

GE T R A M

O p e n T r a f f i c S i m u l a t i o n E n v i r o n m e n t



TEDI
VERSION 4.2

User Manual
February 2004

Table of Contents

1	INTRODUCTION	1
2	TYPES OF OBJECTS IN GETRAM	5
2.1	SECTIONS	5
2.2	DETECTORS	7
2.3	METERING	8
2.4	VARIABLE MESSAGE SIGNS (VMS's)	9
2.5	NODES	10
2.6	CONTROLLERS	13
2.7	BUS STOPS AND PUBLIC TRANSPORT LINES	13
2.8	VEHICLE TYPES	13
2.9	CAMERAS	13
2.10	TEXTS, POLYGONS, POLYLINES, CIRCLES AND GRAPHICAL OBJECTS	13
2.11	CONTROL PLANS	14
2.12	RESULT CONTAINERS	15
2.13	O/D MATRICES	17
2.14	PUBLIC TRANSPORT PLAN	17
3	THE MENU BAR	18
3.1	FILE MENU	18
3.2	EDIT MENU	24
3.3	VIEW MENU	40
3.4	ARRANGE MENU	48
3.5	SECTION MENU	49
3.6	CONTROL MENU	54
3.7	RESULT MENU	60
3.8	O/D MATRIX MENU - BASIC	69
3.9	OD MATRIX MENU – STATEMENTS&PATHS	71
3.10	TRANSYT	80
3.11	PT	81
3.12	WINDOWS	82
3.13	HELP	86
4	THE TOOL BAR AND THE DRAWING AND MESSAGE AREA	88
4.1	THE MESSAGE AREA	88
4.2	THE TOOL BAR	88
4.3	SELECTING OBJECTS	90
4.4	MOVING AROUND THE REAL WORLD AREA	91
4.5	TEXT, POLYLINES, POLYGONS, CIRCLES, GRAPHICS	92
4.6	CREATING AND MANIPULATING SECTIONS & POLYSECTIONS	94
4.7	SECTION PROPERTIES	100
4.8	JUNCTIONS	104
4.9	CENTROIDS	109
4.10	BUS STOPS	111
4.11	PUBLIC LINE	112
4.12	DETECTORS	118
4.13	VMS's	119
4.14	METERING	120
4.15	CONTROLLER	121
5	ACTUATED CONTROL	124
5.1	ACTUATED PARAMETERS	124
5.2	COORDINATED PARAMETERS	126
5.3	MULTI-RING CONTROLLER	127
5.4	ACTUATED CONTROL DEFINITION	128
5.5	BASIS PHASE DEFINITION	128

5.6	ACTUATED PHASE DEFINITION	130
5.7	DETECTOR PHASE DEFINITION	132
5.8	GRAPHICAL PHASE DIAGRAM	132
5.9	BUS PRE-EMPTION	133
6	FUNCTIONS AVAILABLE IN FUNCTION EDITOR	136
6.1	BASIC COST FUNCTIONS	136
6.2	COST FUNCTIONS CONSIDERING VEHICLE TYPES	141
6.3	ROUTE CHOICE FUNCTION	144
7	ASCII FORMATS	147
7.1	RESULT CONTAINER & TRAFFIC STATES	147
7.2	ORIGIN-DESTINATION MATRIX	149
8	QUICK REFERENCE GUIDE TO TEDI COMMANDS	151

1 INTRODUCTION

GETRAM's graphical traffic editor, TEDI, has been designed to make the process of network geometry data entry as user-friendly as any of today's office applications, using the mouse as the main tool. Its main purpose is the construction of traffic models for use with traffic simulators such as AIMSUN. To facilitate this task, the editor accepts a background of the network area, so that sections and nodes can be built subsequently in the foreground using this as a reference.

The editor supports both *urban* and *interurban* roads, which means that the level of detail goes down to elements such as, for example, side lanes, entrance and exit ramps, intersections, traffic lights, and ramp metering. The geometry of the links is specified at the micro level, but the ease of use of the editor makes it even faster than specifying unidimensional links in previous systems, and nodes can be created automatically. The wealth of parameters available for characterising the different types of objects and traffic conditions means that the only limitations on the precision of the model are the quantity and accuracy of the data collected. For example, variables as diverse as the fuel consumption rates of different vehicle types, turning speeds in intersections or the reactions of drivers to a particular message from a Variable Message Sign (VMS), are easily configurable.

TEDI is integrated into the GETRAM Simulation Environment (Generic Environment for Traffic Analysis and Modelling), which consists of the AIMSUN traffic simulator (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks), a network database, a module for storing results, and an Application Programming Interface, through which interfacing with assignment models and other simulation models is perfectly feasible.

TEDI provides an interface to the EMME/2 DATA BANK, which means that a macroscopic analysis can very easily be complemented by a microscopic one using the same traffic data (i.e. O/D matrices) and enjoying a one-to-one match between the elements of the two models.

An interface is provided to TRANSYT and Synchro, the main function of which is to convert a GETRAM network into their respective input data file format and recover the results from GETRAM.

Other interfaces are available with that same purpose of integrating control systems or other popular software tools into the GETRAM environment: interfaces to SCATS, VS-Plus, Utopia and GIS.

Particular attention has been paid to ease of use: the user specifies the network employing realistic concepts such as sections, junctions, ramp metering, bus stops, etc., instead of constructing a wire-frame image with links and unidimensional nodes; the final look of the network is very close to a real-world setting.

The editor accepts a background image that can serve as a reference thus avoiding the need for digitising.

TEDI supports the geometries found in both urban and interurban roads.

TEDI is designed to support the levels of detail required by regional, intermediate and local areas, with increasing modelling detail.

Both static and dynamic assignment models are accepted.

A totally user-defined network state is allowed, thus allowing simulations to run in the absence of assignment results.

TEDI is programmed in C and C++ using X-Windows. It is available for PCs running Windows NT, XP, 95 or 98. Xwin32 5.0 software is required to emulate the X-Windows environment.

TEDI gives the user a realistic view of the traffic network (see

Figure 1-1). In comparison, most other network editors for traffic modelling deal with links (unidimensional segments) and nodes (points). Although traffic engineers may be used to such highly simplified models, we believe that TEDI's approach has some clear advantages. These include the following:

It provides greater geometrical accuracy in sections and nodes. Objects in the network are built following their real-world counterparts very closely. Moreover, a modern visual interface makes this as fast as it would be in a link-node model.

TEDI can accept a set of different formats of graphical files of an area, which can then be loaded as a background, so that digitising maps becomes unnecessary (see Figure 1-2).

In TEDI, all editing capabilities—section geometry, turning movements, junction signal setting, and so on—are available in the same window; there is no need to switch to another module or to change the view of the network. Network states and control plans are also specified within the main window.

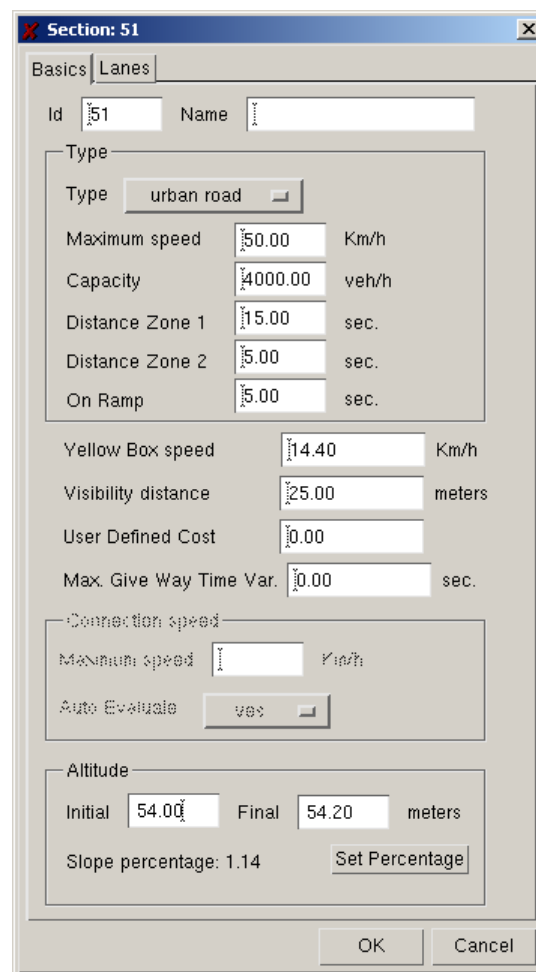
Dialog windows follow a cascade approach, going down to the required level of detail in an object (see Figure 1-3).



Figure 1-1: A TEDI freeway node



Figure 1-2 Example of a TEDI background



Section: 51

Basics | Lanes

Id: 51 Name:

Type: urban road

Maximum speed: 50.00 Km/h

Capacity: 4000.00 veh/h

Distance Zone 1: 15.00 sec.

Distance Zone 2: 5.00 sec.

On Ramp: 5.00 sec.

Yellow Box speed: 14.40 Km/h

Visibility distance: 25.00 meters

User Defined Cost: 0.00

Max. Give Way Time Var.: 0.00 sec.

Connection speed:

Maximum speed: Km/h

Auto Evaluate: yes

Altitude:

Initial: 54.00 Final: 54.20 meters

Slope percentage: 1.14 Set Percentage

OK Cancel

Figure 1-3 A Section Dialog Window: the user cascades through the different levels of information

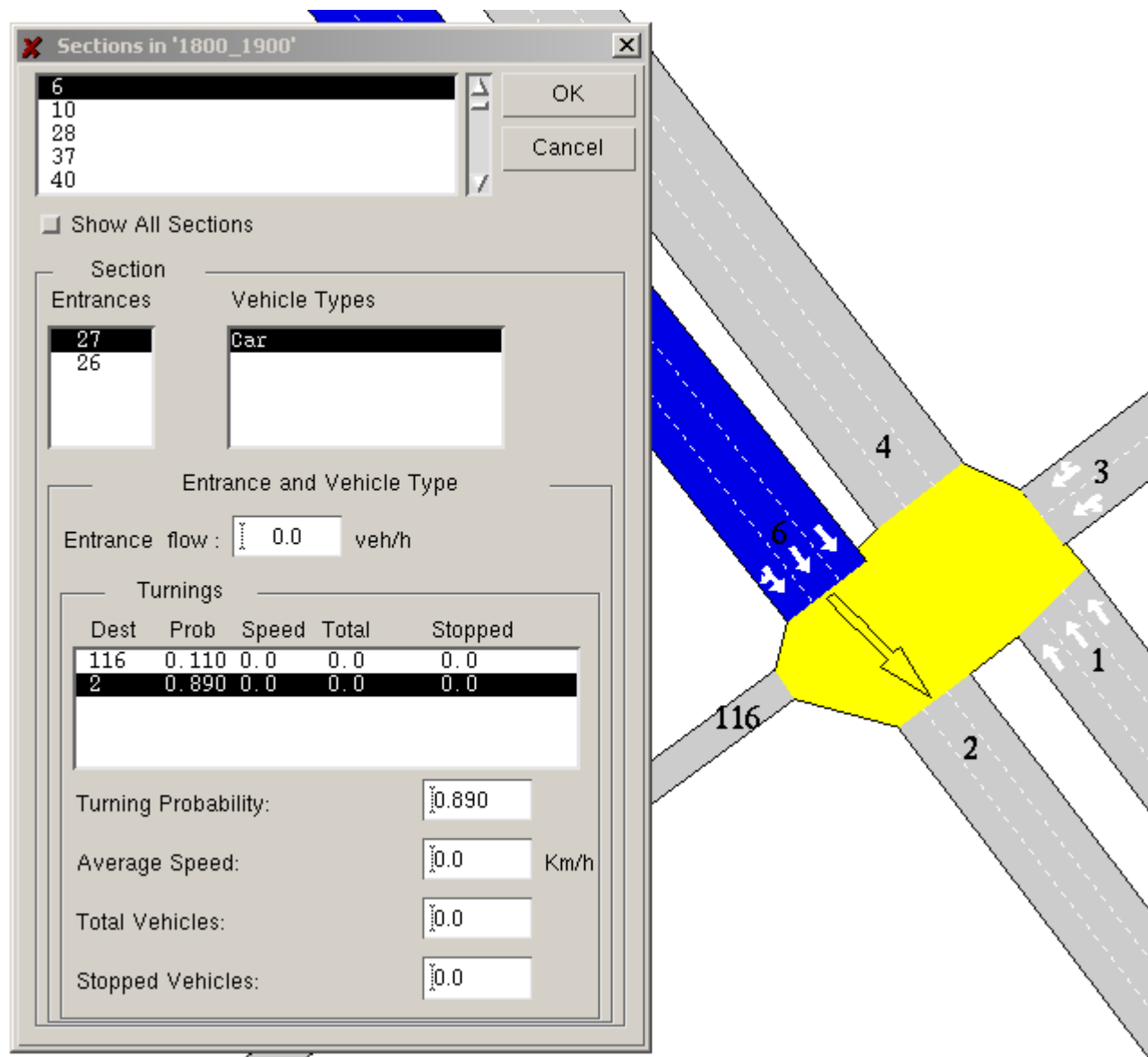


Figure 1-4 Displaying and editing traffic states in TEDI

2 TYPES OF OBJECTS IN GETRAM

For the purpose of defining the network being studied—network geometry, movements and so on—GETRAM uses certain basic objects that provide the foundations upon which more elaborate concepts—control plans, network states, and so on—are built. These basic objects are texts, blocks, lines, sections, nodes, centroids, controllers, detectors, ramp meters, Variable Message Signs (VMS), public transport stops, public transport lines and classes of vehicles. They are all stored in the database.

2.1 Sections

A section is a group of contiguous lanes where vehicles move in the same direction. The partition of the traffic network into sections is usually governed by the physical boundaries of the area and the existence of turning movements. In an urban network, a section corresponds closely to the road from one junction to the next. In a freeway area, a section can be the part of the road between two ramps.

Entrance, exit, and right and left sides of a section are defined according to the vehicles' movement, as shown in Figure 2-1.

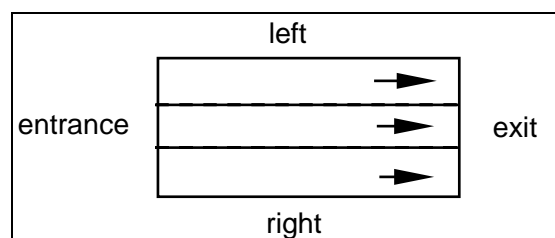


Figure 2-1: Notation used in sections

2.1.1 Main and side lanes

Lanes in a section are classified as main lanes or side lanes. In a main lane, vehicles can enter from other sections and exit to other sections: a main lane is both an entrance and exit lane. A side lane is positioned either on the right or the left side of the section, and is either an entrance or an exit lane (see Figure 2-2). While a section cannot exist without one or more main lanes, side lanes are optional. A section cannot have more than one side lane per side.

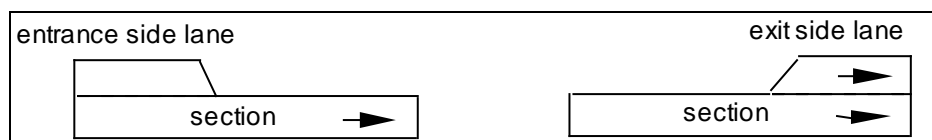


Figure 2-2: Types of side lanes

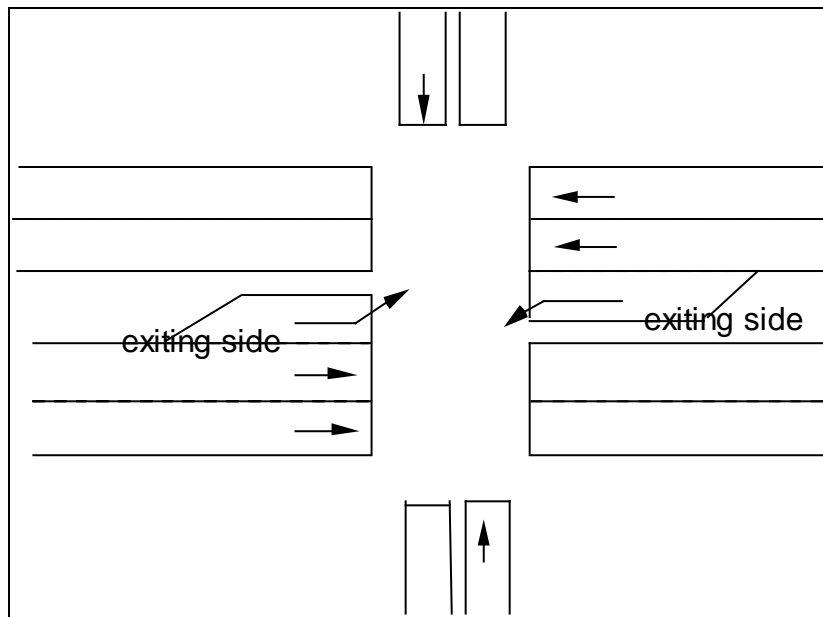


Figure 2-3: Side lanes used for left turnings at a junction

2.1.2 Lane identification

Each lane in the section is identified with a number in the range 1,..., n lanes, from the right to the left of the section, where side lanes are also counted, (see Figure 2-4).

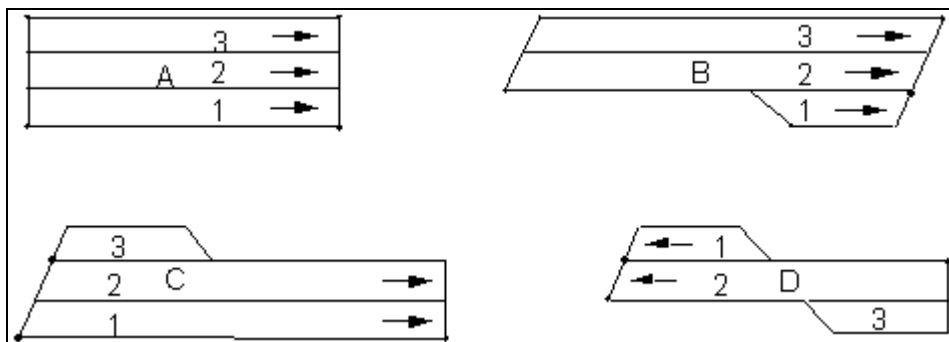


Figure 2-4: Lane identification in a section

2.1.3 Geometry specification

A section is mainly specified by the four coordinates corresponding to the four corners of the set of main lanes, (see Figure 2-5), and side lanes are characterized by their width and length, as shown in Figure 2-6. A library of high-level functions deals with these details and gives the user the coordinates with the required detail.

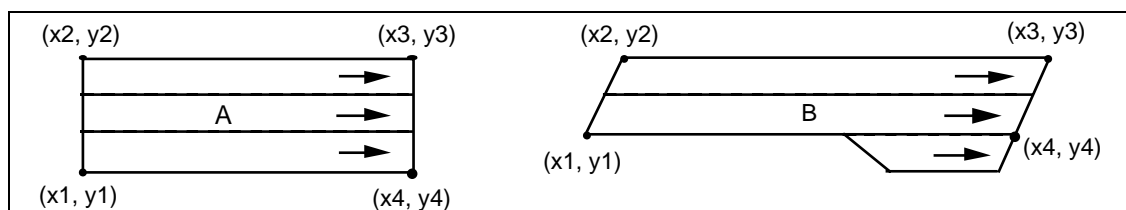


Figure 2-5: Examples of how the section's main lanes are specified by their four corner coordinates

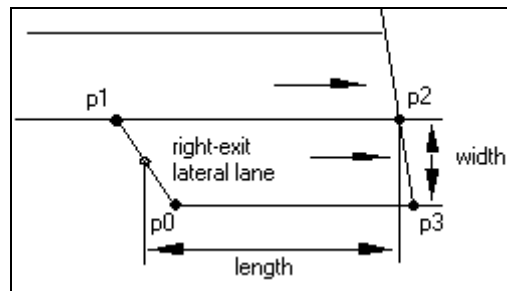


Figure 2-6: The geometry of a side lane is specified by its width and length

2.1.4 Reserved lanes

Reserved lanes in a section are lanes where only one class of vehicles is allowed. A section can have only one set of reserved lanes, and it must be positioned either on the right or on the left side of the section. Reserved lanes can include both main and side lanes but must in all cases be contiguous. The most obvious example is bus lanes. Sometimes taxis can also use these lanes, and this can be specified by defining the ‘public transport’ class of vehicles to which the ‘public bus’ and ‘taxi’ vehicle types may belong. Classes of vehicles and vehicle types are explained in sections 3.2.11, 3.2.12 and 3.7.10. Reserved lanes may be of two types: compulsory or optional. “Compulsory” means that vehicles allowed in that reserved lane are forced to use it, and “optional” means that vehicles allowed in that reserved lane may use it or not depending on the traffic conditions (e.g., bus lanes and high-occupancy vehicle lanes).

2.2 Detectors

Many types of detectors exist: pressure, magnetic, loop, video, and others, but all of them are characterized by their measuring capabilities. Measuring capabilities include vehicle count, presence, speed, occupancy, density, headway and, as a special one, whether the detected vehicle is an equipped one. Also, the detector may have the capacity to distinguish between different vehicle types.

A detector can be positioned at any point in a section, and its width can extend to more than one lane. A section can have no, one, or multiple detectors. The only restriction is that a detector cannot be shared between sections.

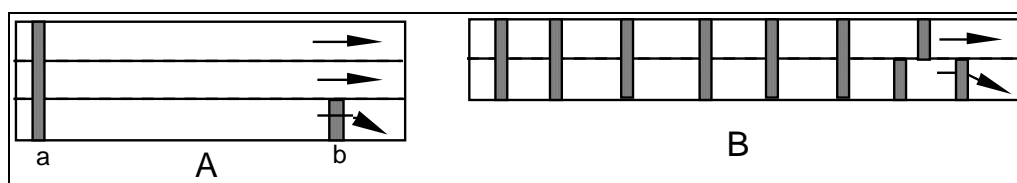


Figure 2-7: Examples of detector positioning in a section. In Section A, detector b is used to detect right-turning vehicles. Multiple detectors as in (B) are possible using video detection

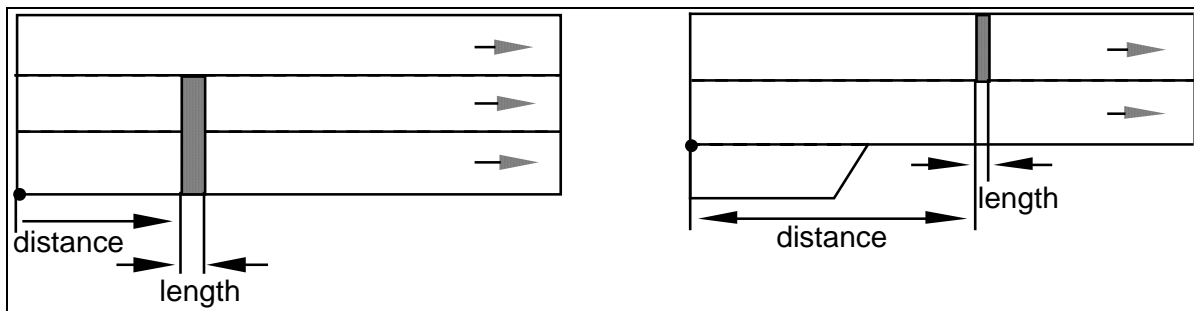


Figure 2-8: Examples of how a detector's position is specified

Figure 2-9 shows an example of detector positioning at a ramp metering point, using standard notation:

Queue detector: to detect a queue before it blocks the upstream section.

Check-in detector: to detect vehicles waiting.

Check-out detector: to adjust the green time to the desired platoon length (can be connected directly to the ramp metering)

Merge detector: to detect vehicles waiting to merge, and thus inhibit the green time in the ramp meter.

Main line detector: to detect the vehicles in the freeway and thus adapt the control policy.

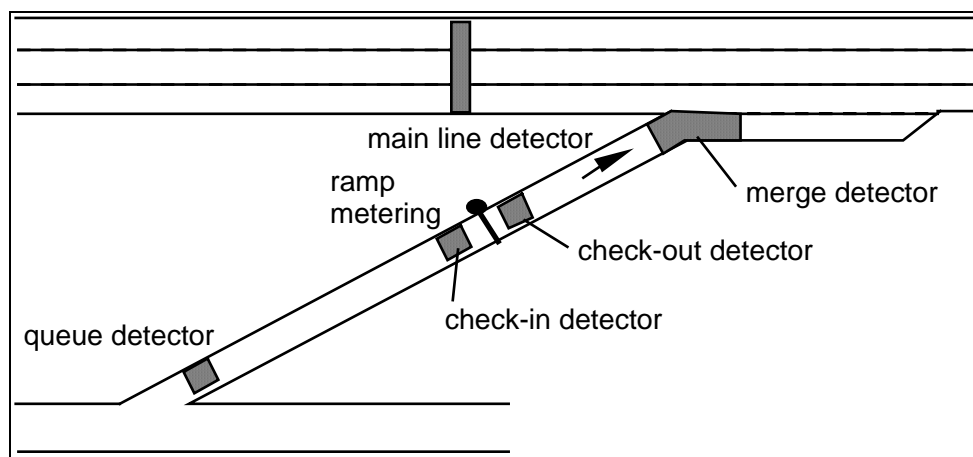


Figure 2-9: Example of detector positioning at a ramp metering point

2.3 Metering

Metering takes place on entrance ramps to a freeway in order to limit the flow of vehicles entering, thus ensuring a non-congested flow on the freeway itself. Figure 2-10 shows an example in a single-lane section. Ramp meters can be implemented in a variety of ways. TEDI uses three types:

Green-time metering. Usually implemented as a semaphore, this is characterized by a cycle time and a green time duration. This is being implemented mainly in Europe.

Flow metering. This controls the maximum flow of vehicles that can pass through the signal. An important parameter in flow metering is the number of vehicles allowed through each time

(normally one, but occasionally two). This is assumed to be fixed throughout the history of the device.

Delay metering. This delays every vehicle crossing it by a fixed number of seconds. This type of metering is useful for modelling certain types of obstacles to free flow, such as tolls or customs barriers

Meters can be positioned anywhere along the section; all exit lanes are affected (unlike detectors, single lane metering is not possible), and only one meter per section is allowed.

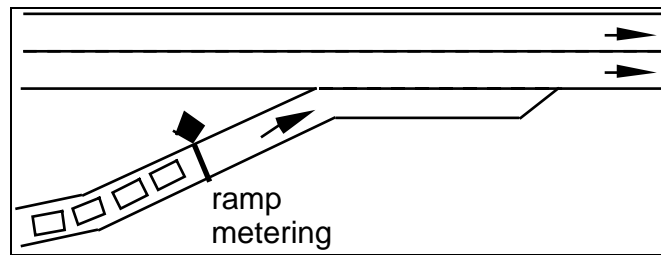


Figure 2-10: Metering in a lane

As will be seen in the case of junctions, there are three types of control policies for metering:

Uncontrolled policy: the metering device is assumed to be disconnected.

Fixed policy: a maximum flow of vehicles is allowed through.

External policy: the flow of vehicles initially permitted can vary between a given maximum and minimum depending on traffic conditions and according to an external user-set algorithm (GETRAM Extensions).

In some situations, selective metering is used, i.e., an adjacent lane with no metering where only certain selected classes of vehicles (for example, public vehicles) are allowed through. This is modelled by considering the unmetered lane as a different section. Figure 2-11 shows an example: vehicles that are not metered turn from section 1 into section 3.

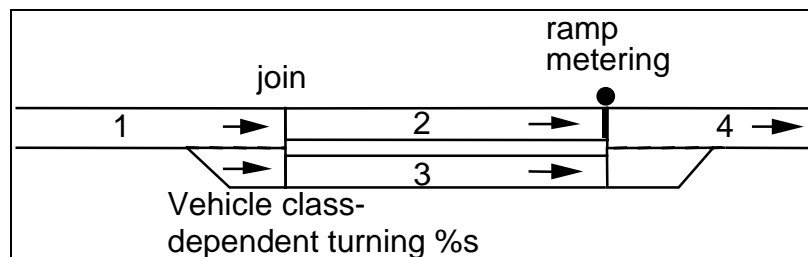


Figure 2-11: Modelling a selective meter

2.4 Variable Message Signs (VMS's)

Variable message signs (VMS's) are used to inform drivers of regulations or instructions that are applicable only during certain periods of the day or under certain traffic conditions (Doughty, 1982). The current implementation allows for the selective display of a number of messages.

The VMS is modelled as being located at a given position in a section (see Figure 2-12) and as having a set of possible messages. At Aimsun each message will be assigned one or more *actions*, which will model the possible effects of the message on the different drivers that read it.

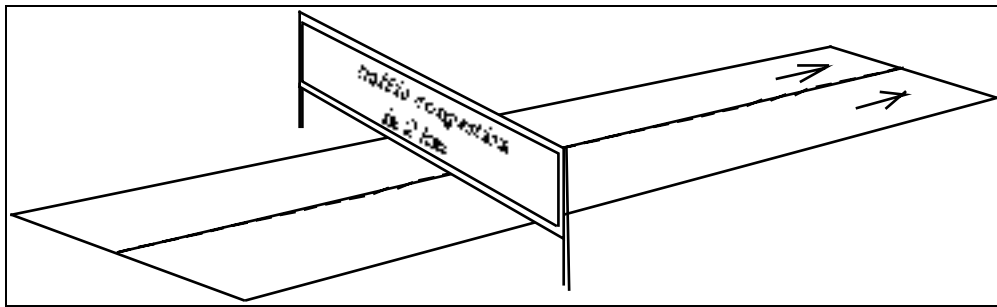


Figure 2-12: A VMS in a section

2.5 Nodes

A node is a point or an area in the network where vehicles change their direction and/or disperse. Hence, a node has one or more origin sections and one or more destination sections. We distinguish between two types of node: join nodes (joins) and junction nodes. The difference between them is that, while in a junction there is a space between the origin and destination sections, in the joint node there is no space between these two sections. In a join node, the number of origin lanes equals the number of destination lanes. Junction nodes are often found on arterials and streets; join nodes, on roads and highways.

The generic concept of node, including both junctions and joins, is useful for assigning a common identifier, and characterising both views and origin/destination zones.

2.5.1 Join nodes

Joins often model bifurcations (A), exit ramps (B), mergings (C), and entrance ramps (D). Figure 2-13 shows how these various situations can be modelled

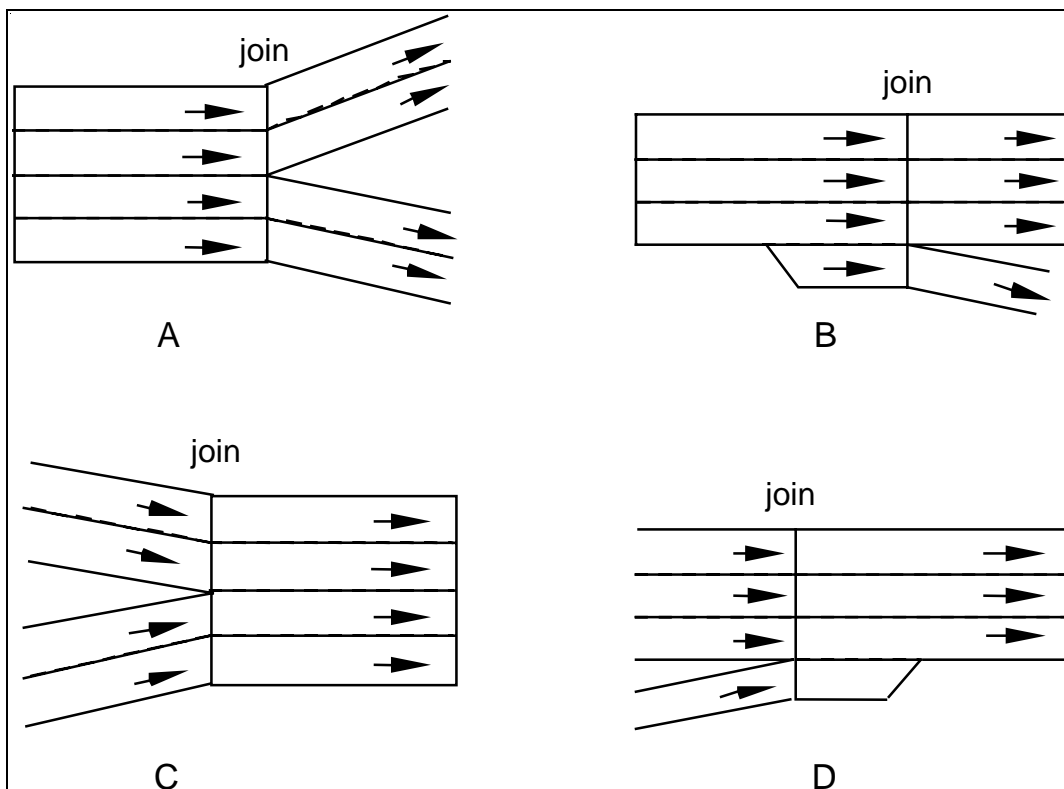


Figure 2-13: Examples of joins

2.5.2 Junctions

Turnings occur within a junction and the user can specify give-ways between them. Imposing give-ways can solve conflicts between turnings arising in junctions without signals. In signalled junctions a sequence of stages is designed to avoid conflicts, but give-ways can also be specified.

Signal lights at a junction can be connected to a controller that sends signal light changes according to a Pre-timed (fixed controlled), Actuated or Adaptive strategy. Controllers are explained later.

2.5.3 Turnings

At a junction turning, origin lanes are not clear to the editor, nor are vehicles restricted to entering the destination section by a lane subset. Therefore, both origin and destination lanes have to be specified.

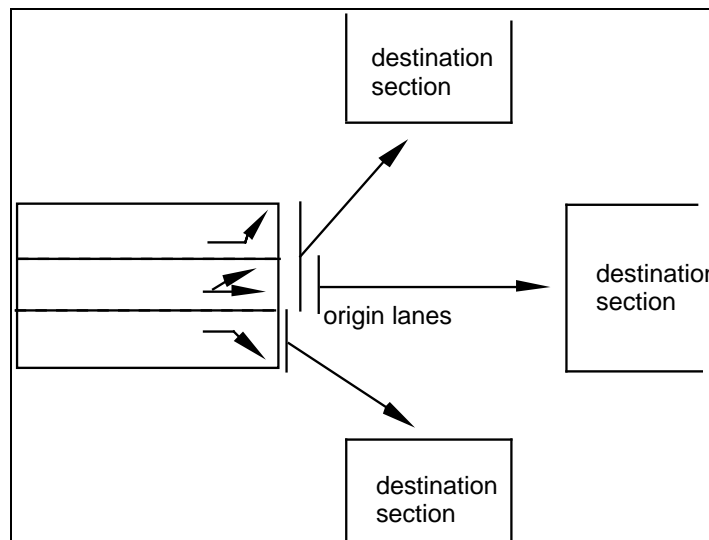


Figure 2-14: Notation used in turnings

2.5.4 Give-way and stop

Vehicles approaching a give-way check the trajectory and speed of crossing vehicles and make the appropriate corrections to avoid collisions. In the case of a stop sign, the vehicle stops at the entrance of the junction and accelerates when it finds a big enough gap between two successive vehicles.

2.5.5 Signal groups

A signal group is the set of turning movements given right of way when a given traffic light turns green (see Figure 2-15). A phase-based approach is applied, in which the cycle of the junction is divided into phases, each one activating a particular set of signal groups (see Figure 2-16).

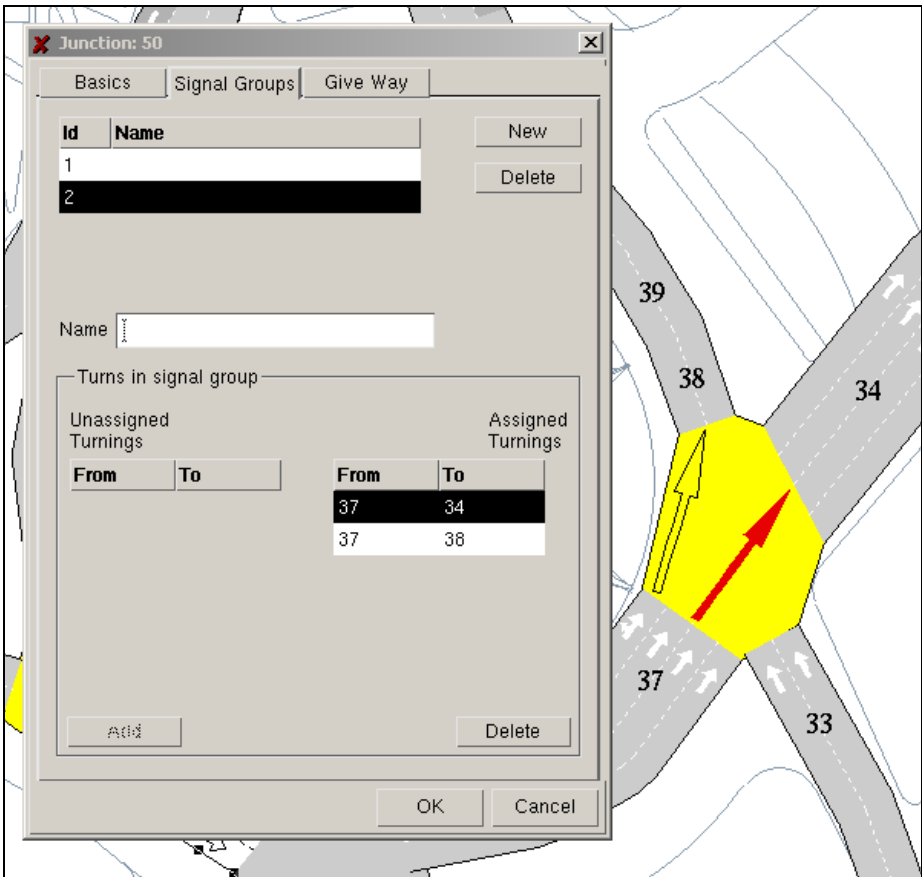


Figure 2-15: Example of the correspondence between a signal group and a set of turnings

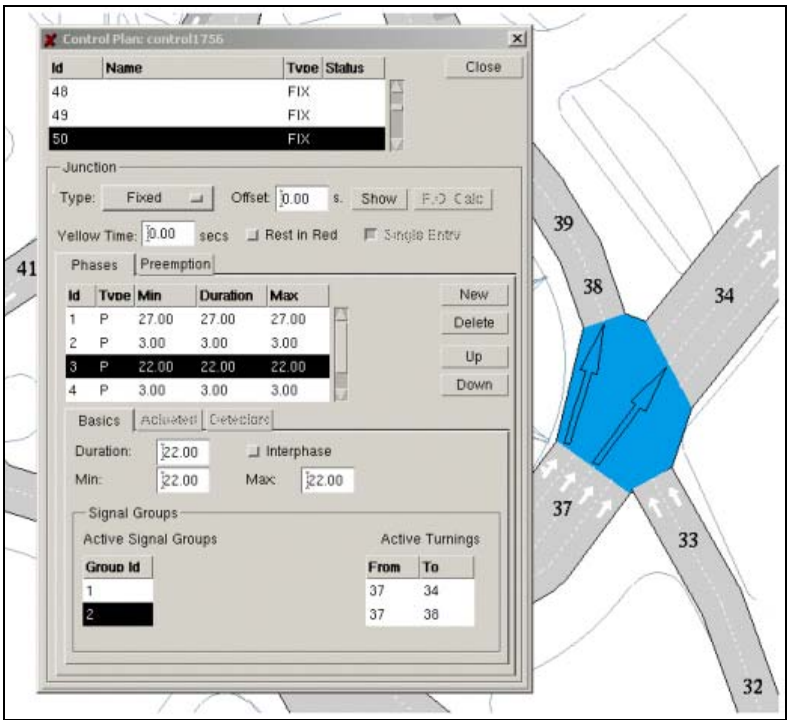


Figure 2-16: Example of a cycle and its phases

2.6 Controllers

Controllers are entities designed to group network equipment: signal groups, VMS's, ramp metering devices and detectors. Their purpose is to represent the equivalent real world devices used by traffic control systems or traffic management software such as Transyt, for example.

2.7 Bus stops and public transport lines

Bus stops are objects that can be positioned on and associated with sections. There are three types of bus stops: normal, bus bays and terminals. Normal bus stops are those which occupy a certain lane length, bus bays are those buses to one side of the section and terminals are the model of bus stations with a certain capacity for a given number of vehicles.

A Bus line is modelled as a set of connected sections (a fixed route) with a set of assigned bus stops subset to the total number of bus stops on those sections. The information associated with the bus line also includes the timetable for that line for the different periods of the day. That information includes departure times and stop times for each bus stop. Different values can be input for different scenarios for different days and time periods; one of those scenarios is then chosen when designing a Public Transport Plan. Public transport lines and public transport stops are connected in the following way: a public transport line has several stops and, reciprocally, several lines pass through a unique public transport stop. Different public transport lines can be connected to each other if they give a coordinated service, and the timetable departure times will then be conditioned to the on-time arrival of its complementary line.

2.8 Vehicle types

A vehicle type is the group into which a set of vehicles and drivers with common physical characteristics has been aggregated for its modelling in the simulation environment. For instance we can work with car, taxi, public bus, private bus, LRT, SUV, truck and long truck.

2.9 Cameras

This object both models real life cameras and/or sets the main perspectives for the later 3D visualization of the simulation animation. Cameras can be translated, rotated in the 3D environment, they can also zoom the image and it is possible to set a route for a given camera so that it automatically guides us through the network during the animation.

2.10 Texts, Polygons, Polylines, Circles and Graphical Objects

Texts, polygons and polylines are used to write text and represent the framework of additional features—walls, squares, blocks of buildings—in which the traffic network is embedded. These can be convenient for clarifying the view of a network, and adding reference points that help to recognise the area to be modelled, (see Figure 2-17 and Figure 2-18). Also, a library of graphical objects is provided with 2D and 3D representations of traffic signals, trees, street lights, vehicles, bus stops and other sort of items usually found in traffic networks.

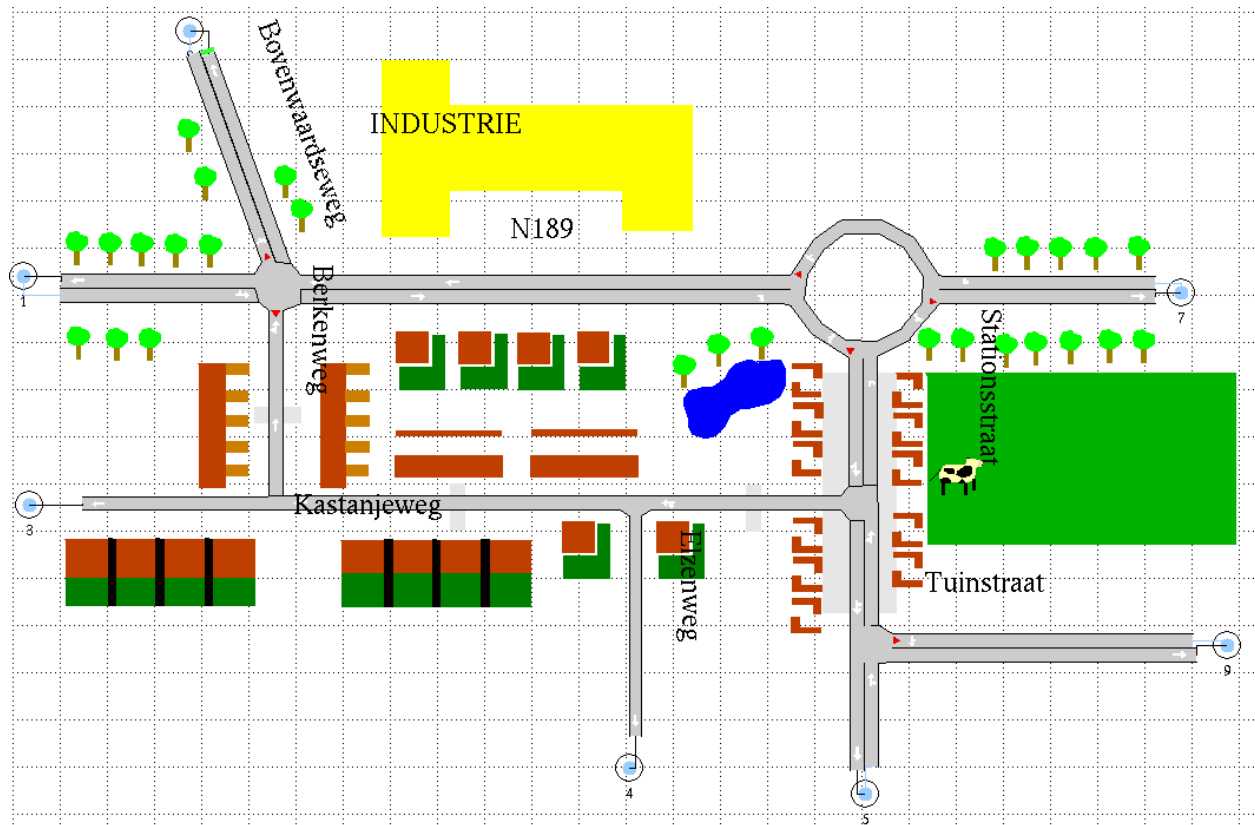


Figure 2-17: 2D decoration

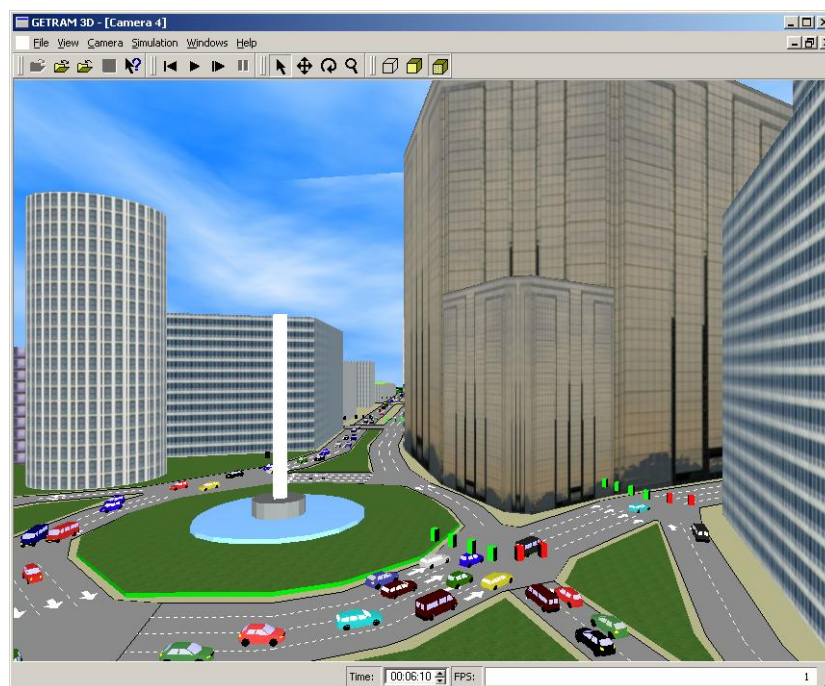


Figure 2-18: 3D decoration

2.11 Control plans

A control plan specifies the control parameters applied to each junction and meter in an area (see Figure 2-15).

The parameters in a **junction** are:

Control type: fixed control, external or Actuated control.

Critical junction flag: a critical junction could be considered differently in an Adaptive or external model.

Offset: used to set the beginning of the sequence of phases, and therefore synchronise adjacent junctions within the same control plan.

Phases folder: sequence of phases with three folders:

Basic: duration of each phase and the active signals during each phase.

Actuated: corresponding parameters in case of an actuated junction.

Detectors: list of the detectors that will provide traffic data to the actuated system, plus the conditions to use that data.

Preemption folder: edit window for the different pre-emption strategies, which lines to attend, the nature of the service and which detector will trigger the service.

Metering parameters depend on they type of meter. The main parameters for each case are:

- **green-time meter**: the cycle time and the green-time duration **flow metering**: maximum allowed flow
- **delay metering** mean delay for each vehicle and the standard deviation from the specified mean.

An important issue is how to deal with changes in the network. When the network configuration changes—for example, when a new junction is added to the network—then the previously stored control plans may be incomplete or inconsistent. TEDI and AIMSUN alert users to such inconsistencies at the time a control plan is opened, requiring the user to supply consistent values for those junctions and meters that have changed.

2.12 Result containers

A result container is characterised as having been produced in a network area, either as the output of a simulation, or as an implementation of the data gathered on traffic conditions.

A result container consists of two parts:

2.12.1 Vehicle types

Vehicle types are specified as types of vehicles with certain physical characteristics.

Vehicle Types (MODEL)

Vehicle Types

- car
- truck

New
Delete
Duplicate
To Lib

Attributes

Name: car

Parameter: Articulated | Classes | Fuel | Pollutants

Name	Mean	Devia	Min.	Max.	Units
Length	4.00	0.20	3.80	5.00	meter
Width	2.00	0.00	2.00	2.00	meter
Max Desired Speed	80.00	20.00	60.00	100.00	Km/h
Max Acceleration	2.80	0.20	2.40	3.20	m/s ²
Normal deceleration	4.00	0.20	3.60	4.40	m/s ²
Max deceleration	8.00	0.20	7.60	8.40	m/s ²
Speed Acceptance	1.10	0.10	0.90	1.25	
Min Distance Veh	1.00	0.20	0.50	1.50	meter
Give Way time	15.00	15.00	10.00	30.00	sec.
Guidance Acceptance	0.00	0.00	0.00	0.00	

Guided: 0.50 Equipped: 0.00 Cruising Tolerance: 0.80 m/s²

OK Cancel

Figure 2-19: Vehicle type

2.12.2 Network states

Each network state contains a time offset, and a description of the traffic state (vehicle inputs, turning proportions and initial occupancy) in each section in the network view (see Figure 1-4). The time offset is measured in relation to the reference time 00:00:00. The traffic state in a section is characterised by the following averaged measurements:

- For those sections with entries not connected to any other section (sections that act as vehicle entries to the network), the entrance flow per vehicle type.
- For each following section, vehicle type, and turning: the average speed, total number of vehicles, number of vehicles stopped, and turning percentage.

As with the control plans, changes in the network are dealt with by highlighting inconsistencies at the time a network state is opened, and requiring the user to supply consistent values for those sections.

2.13 O/D Matrices

An O/D matrix consists of the flow values gathered for each type of vehicle from each vehicle source to each vehicle sink for successive time periods. An O/D matrix is made up of the traffic demand for each vehicle type, each time slice and each OD pair.

Statements in 'calibrado'

Time Period
 From: 08 : 00 To: 10 : 00 (hh:mm) 4 intervals Set ...

Vehicle Type car Time Interval 2

O/D Matrix (Origin in Rows, Destination in Columns)

	202	204	206	215	216	218	224	Total
202								107.7
204								103.8
206				3.5				72.2
215			10.6				8.2	189.3
216				3.4				146.8
218		2.7						66.4
224				3.4				249.8
Total	104.3	122.8	40.6	257.1	82.2	145.5	110.6	25312.7

Multiply By

Export... Import... Close

Figure 2-20: OD Matrix Statements

2.14 Public Transport Plan

When creating a public transport line, a set of timetables is edited with the departure times and stop times for each period of the day. A public transport plan simply consists of assigning one timetable to each public transport line. In this way, we can use a set of PT plans to model the different departure times and stop times of each line at each time period through the simulation.

3 THE MENU BAR

3.1 File Menu

3.1.1 New

Starts a new network-editing session.

3.1.2 Open...

Opens an existing network. This option opens the following dialog window:

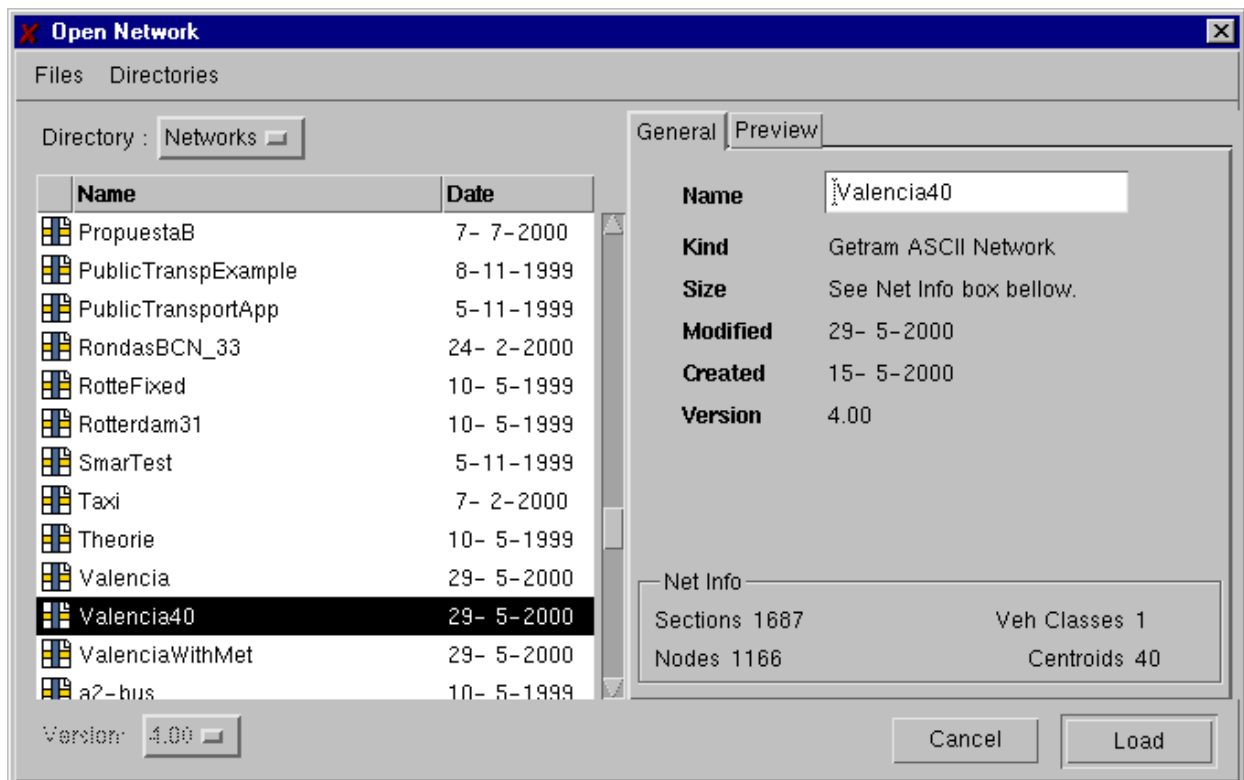


Figure 3-1: Main window

Several fields can be observed in the window:

- Menu bar












Files Directories

A record of the 10 most recently loaded networks and 10 most recently visited directories is kept below the *Files* and *Directories* headers at the top of the window. Choosing one of these networks automatically selects it for loading, and choosing one of the directories causes the browser to jump to it automatically, thus avoiding having to navigate through the directory structure to the intended final location.

- Browser

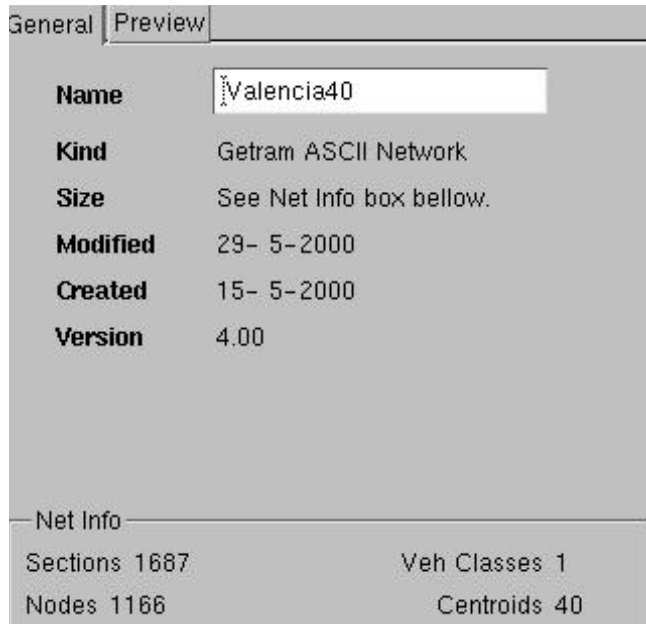
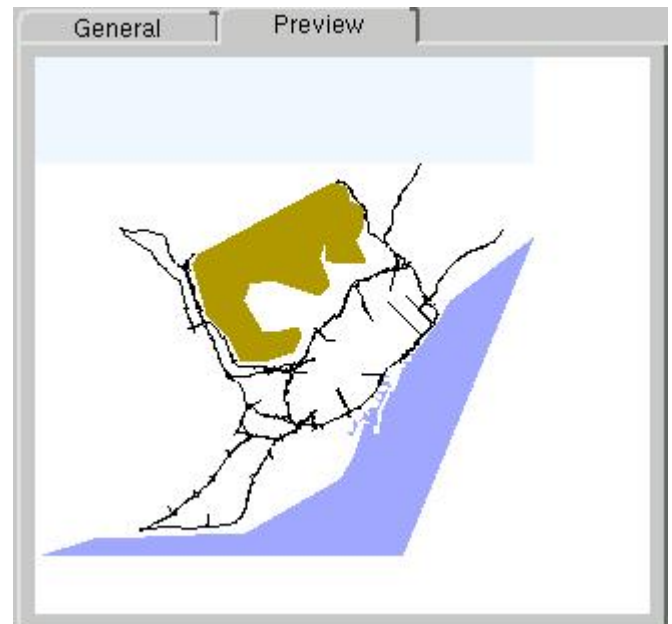
The *Directory* indicator shows the current directory. If the toggle is selected, the path of directories up to the root is shown, and any of them can be selected in order to go there directly. The contents of the present directory are shown under the *Type*, *Name* and *Date* headers. The *Type* header specifies whether the objects displayed are directories (folders), networks (documents), or backgrounds (documents, but filenames end with an extension). A double-click on a folder opens the directory. A double-click on a network loads it.

Figure 3-2: Network folder

Directory : Samples		
Name	Date	
 Aimsun3D demo Nets	12-11-2003	
 Backgrounds	19-11-2003	
 Donosti	12-11-2003	
 BCN 3d	12-11-2003	
 BCNeixample	12-11-2003	
 Dublin	17-11-2003	
 Example	12-11-2003	
 GettingStarted	26-11-2003	
 NTypes	12-11-2003	
 PTNet40	12-11-2003	
 rondas_bcn	12-11-2003	

- *General & Preview folders*

The general folder outlines basic information about the network, both as a network and as a file. The preview folder shows a view of the network so we can check whether it is the one we want before loading it.

**Figure 3-3: General folder****Figure 3-4: Network preview**

3.1.3 Close

Closes the currently open network. The user is always prompted for confirmation on whether to save the latest changes or not.

3.1.4 Save

Saves the currently open network.

3.1.5 Save as...

Saves the currently open network, optionally with a new name and format. This option opens the following window:

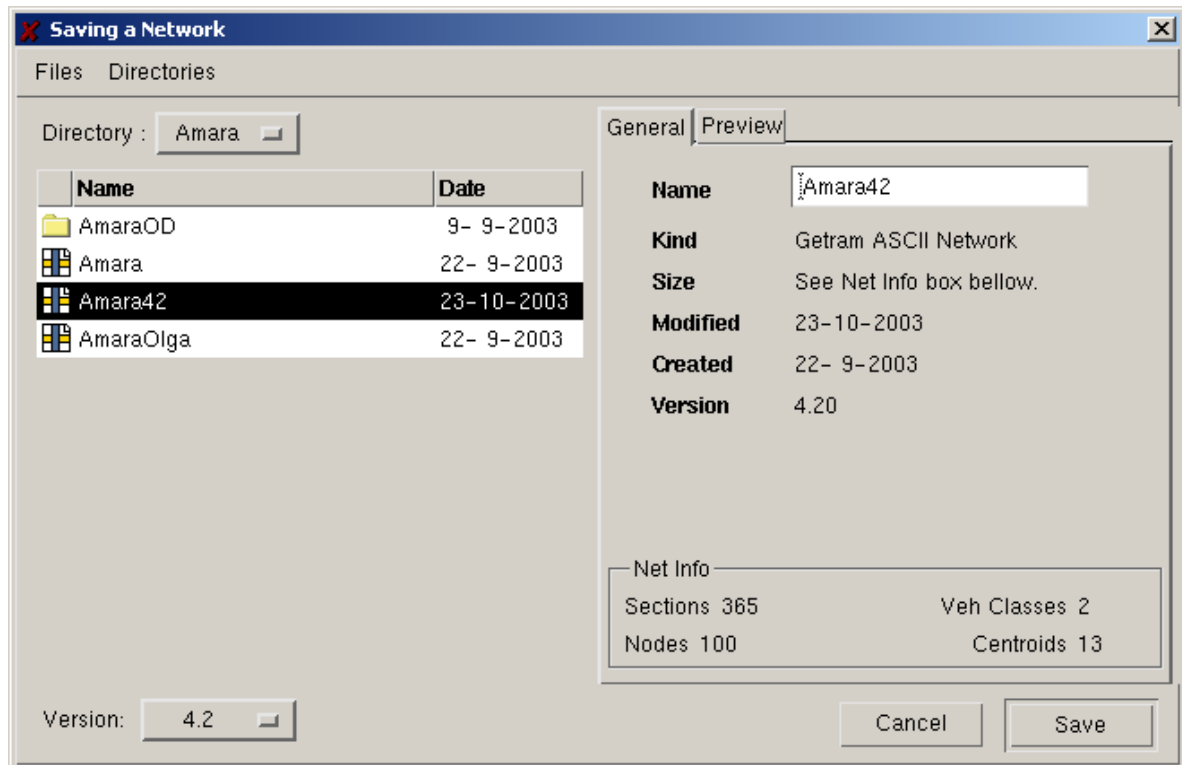
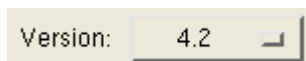


Figure 3-5: Save network window

This window is almost exactly the same as the one used to open a network, but is used for the complementary function. The only differences are that:

1. Instead of the *Cancel-Load* pair of buttons, the *Cancel-Save* pair appears.
2. A new field is present, namely:



This field is included for the sake of completeness, it allows users to choose to save the network in any of the older versions' formats, and it allows old models to be converted to newer formats.

3.1.6 Print

Opens a standard print window to send the print file to the selected peripheral.

3.1.7 Import...

Opens an import window from which an EMME/2 network (see Figure 3-6) can be imported to GETRAM. The data from the EMME/2 model can be used to build the initial skeleton of a GETRAM network, that can then be completed editing it with TEDI. For further information see the EMME/2 Interface Manual.

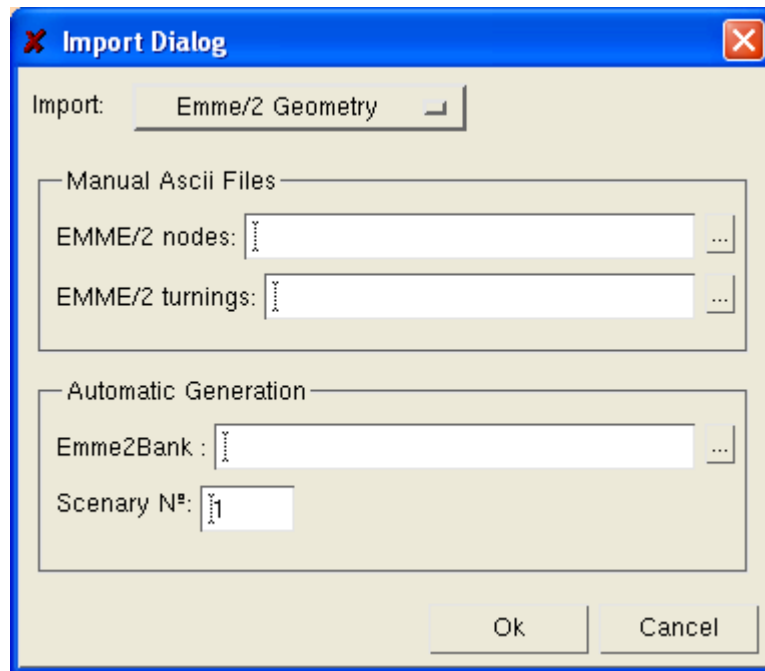


Figure 3-6: Import EMME/2 geometry

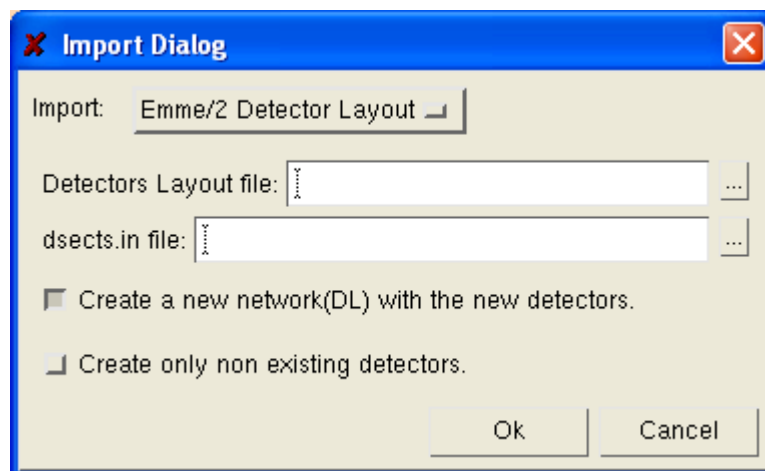


Figure 3-7: Import EMME/2 detector layout

3.1.8 Export...

It is possible to export the network to three different formats:

- To an EMME/2 model: a complete EMME/2 network is built using the information from the GETRAM model. Only calibration remains to be done in order to complete the full modelling. For further information see the EMME/2 Interface Manual.
- To GIS format: the GETRAM information is stored in shape file format.
- As an image: this makes it possible to generate images of the whole network or the viewed area in the TEDI window. You will notice that the first option is not *Output directory*, as in the cases of EMME/2 or GIS, but *Output file*. This is because the user is left to decide the name and format of the image file that has to be added at the end of the default directory already present (or the directory input by the user).

In Figure 3-11, the name added is *TH_im* and the format is *.jpg*. The accepted formats are:

bmp, gif, jpg, png, tif

When generating an image make sure that *text* is deactivated from the *show objects* option.

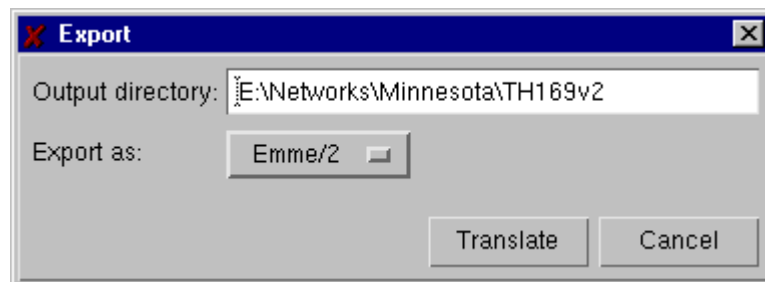


Figure 3-8: Export to EMME/2 window

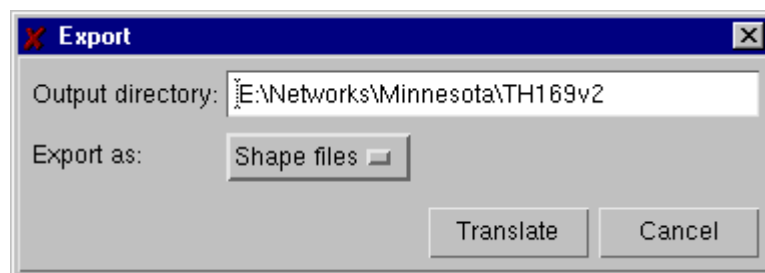


Figure 3-9: Export to GIS window

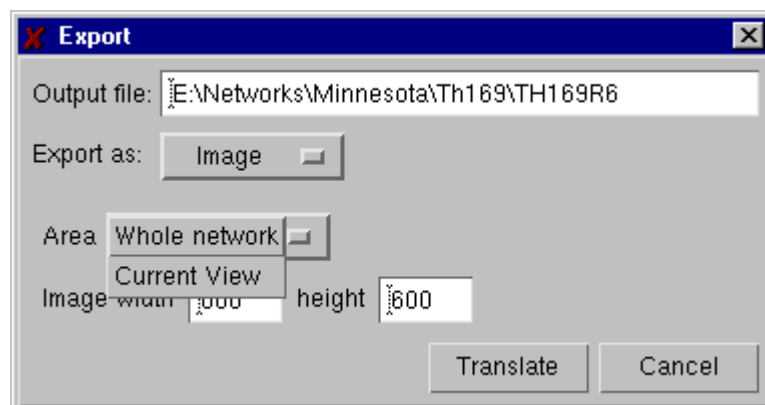


Figure 3-10: Generate an image window

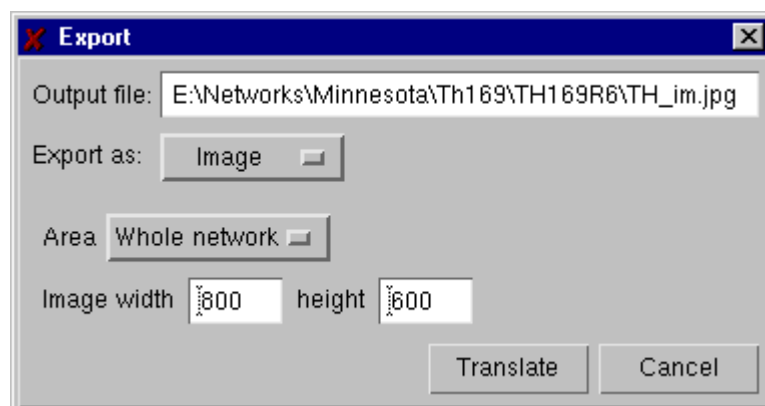


Figure 3-11: Generate an image window

3.1.9 Report Generation

This option allows you to ‘filter’ the model for a wide range of errors before loading it into

AIMSUN for its simulation. The results of the network check are stored in the errors.html file, in the html subfolder of the network folder. The Sections and Nodes toggles allow you to generate a report on the network: a set of html files, one for each section and each node, with all the information contained in each of those objects. If you also choose to generate images, each of these html files will include an image of the object.

Figure 3-13 shows an example of the files produced for a report generated on sections, nodes and errors with images. The file index.html is the one that organises all this information and from which it is easy to access the information associated with a particular object (see Figure 3-14).

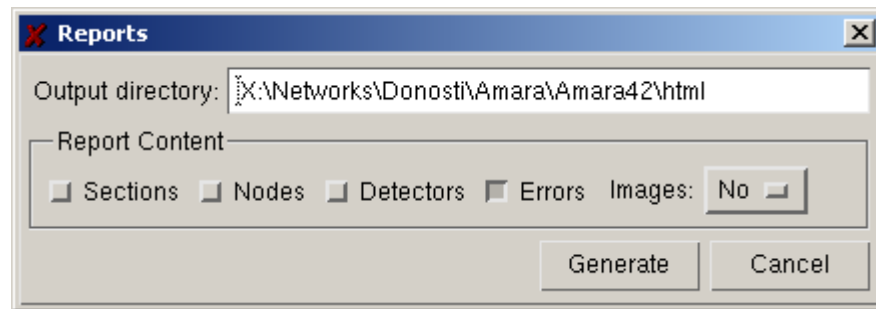


Figure 3-12: Report Generation window

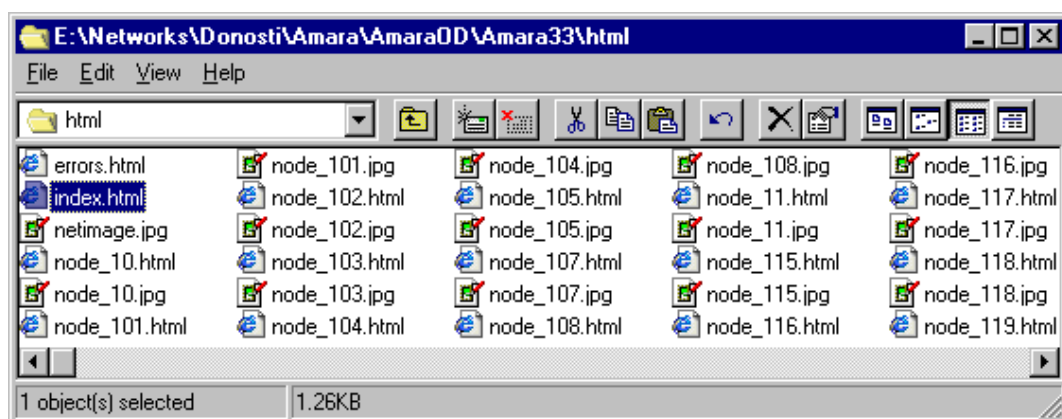


Figure 3-13: Files produced by a report generated on sections, nodes and errors with images

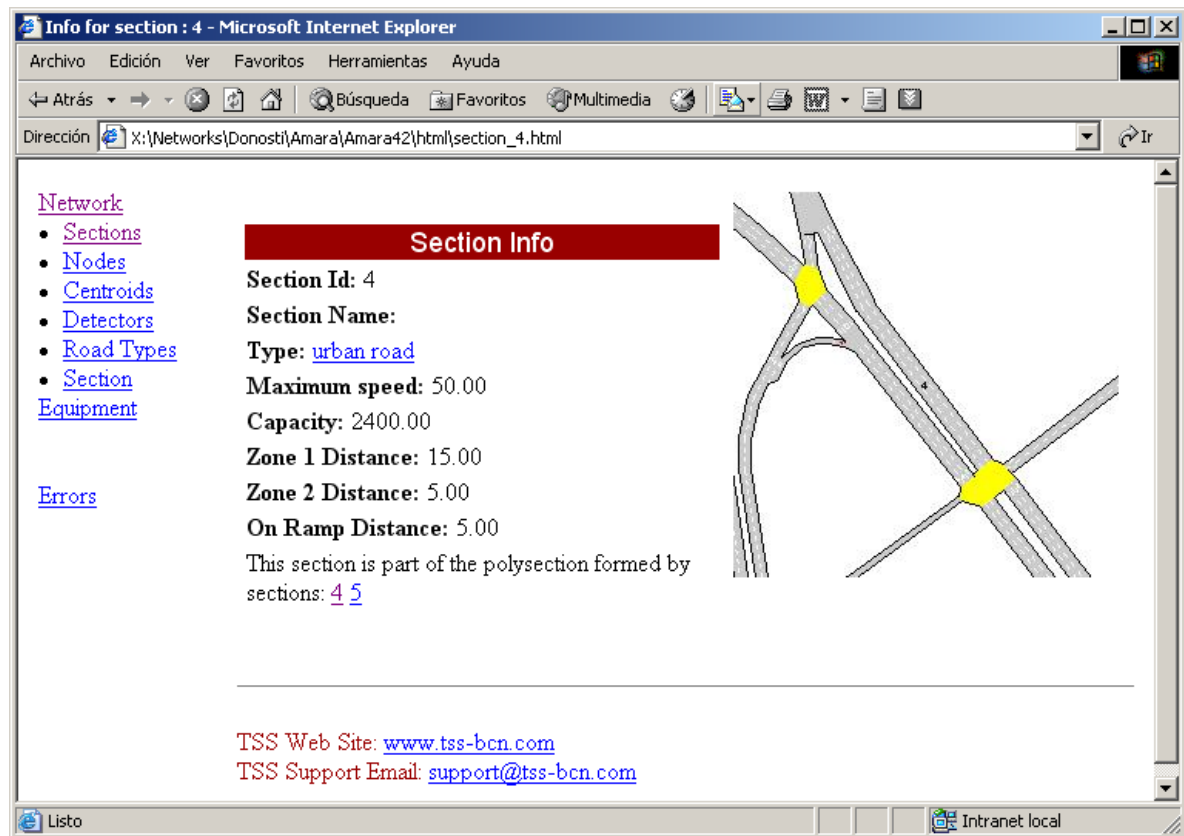


Figure 3-14: Visualising the information associated to section 451

3.1.10 Quit

Exits the application. The user is always prompted for confirmation, and asked whether to save the latest changes or not.

3.2 Edit Menu

3.2.1 Undo

This version of the Undo option, undoes translations, rotations and changes in the vertex of polygons and lines.

3.2.2 Redo

This simply the reverses the Undo option.

The options Cut, Copy and Clear in the edit menu are only activated when one or more objects have been selected. The Paste option is only activated when a Cut or Copy has been carried out.

3.2.3 Cut

Removes the selected objects and puts them on the clipboard.

3.2.4 Copy

Copies the selected objects to the clipboard.

3.2.5 Paste

Copies objects from the clipboard to the drawing area. The paste function places the contents of the clipboard in the middle of the TEDI window view if possible, otherwise it will paste to the middle

of the drawing area. If the object to be pasted does not fit the drawing area, the editor will issue a warning message and will not paste.

We recommend that you check the scale of the present view before pasting, in order to ensure that the objects to be pasted will fit in the window.

3.2.6 Clear

Deletes the selected objects.

3.2.7 Select all

Selects all the objects in the drawing area (also called the “real world”).

3.2.8 Select Inverse

Automatically unselects the selected objects and selects all the other ones.

3.2.9 Unselect All

The inverse operation to that of ‘select all’, all selected objects are unselected. This same effect is accomplished by clicking with the mouse on any area of the real world with no objects.

3.2.10 Find...

Finds objects in the drawing area. This is carried out via the window displayed in Figure 3-15.

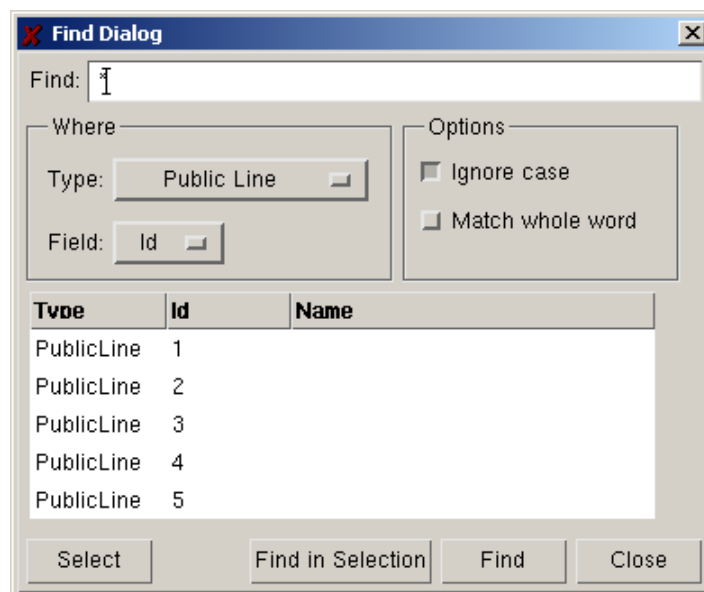


Figure 3-15: Find Object window

The *Object* field (see Figure 3-16) is used to select the type of object to be found. The options are: All types, Section, Node, Centroid, Public Line, Detector, Metering, VMS (Visual Message Sign), Bus Stop, Controller, Text, Polyline, Polygon, Circle, 3D Image, and Shape Group.

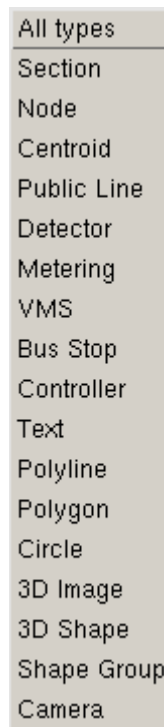


Figure 3-16: Find Object Type

The *Field* field (see Figure 3-17) allows you to search for an object using its identifier or its name.



Figure 3-17: Field options

You can also specify whether the search distinguishes between uppercase and lowercase characters in the names or identifiers. You can also select either a complete match or a partial match.

Example:

We have built a network in which 12 sections form “Baker Street”. The different sections are named “Baker_1-27”, “Baker_28-62”, etc., depending on the numbers in the street included for each section. In this case, if we were searching for the section of Baker street from numbers 28 to 62, we would use *Section-Name-Match Whole Word* and enter “Baker_28-62” in the *Find* field. One further option is to use the wildcard ‘*’ search function. This gives us a full list of all objects of the selected type. Just by clicking on the line of the list containing the required object causes the editor to automatically centre on that object. Double-clicking on that same line will open the dialog window for that object. One last option is that of searching for an object within a group of selected objects or a selected area, instead of searching throughout all the network. In order to achieve this it is necessary to first select the area of interest and then use the ‘find in selection’ option instead of the normal ‘find’.

3.2.11 Vehicle classes

Specifies the classes of vehicles used in the network. This concept is not associated with the physical characteristics of each type of vehicle (e.g., cars, buses, and lorries), but rather with the sort of vehicle it represents as far as the topology of the network is concerned. Examples include private vehicles, public vehicles, and emergency vehicles. This is used later to define for which sort

of vehicles a reserved lane is dedicated. A set of vehicle classes to which each vehicle type belongs is also specified. For example, a 'bus' vehicle type can have selective authorisation in some lanes because it belongs to the 'public transport' class. Again, the 'guided taxi' type belongs to the classes 'public transport' and 'guided'. So the concept of vehicle *type* is that of vehicles with common physical characteristics such as cars, buses, trucks, bicycles, motorbikes, etc., whereas the concept of vehicle *class* is that of the different tags with which we mark the vehicles – the different classes we can form, taking into account one common characteristic of different vehicle types, such as being public transport vehicles.

There is also a library of vehicle tags used in the common editor to facilitate the creation of new vehicle classes in the network, or store them for future use in other networks. The library is independent of the network being edited.

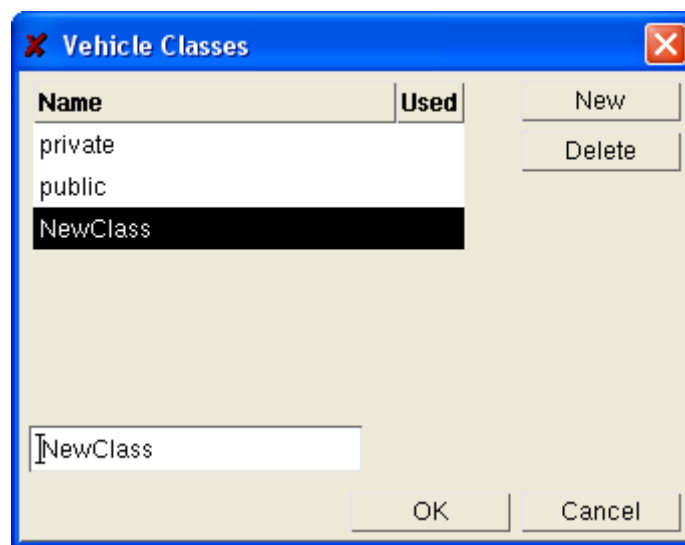


Figure 3-18: Vehicle Classes dialog window

The process of defining a class begins by pressing the *New* button. This makes the *NewClass* appear in both the top and bottom areas of the window. To change *NewClass* to the desired class name, the name has to be written in the bottom window replacing *NewClass*, and then has to be validated by pressing *Enter*. The name then appears in the top area of the window. If the name is not validated, closing the window or pressing *New* again will preserve the *NewClass* name instead of changing it to the desired one.

3.2.12 Vehicle Types Library

This option provides access to a library of different vehicle types. These can be edited, and any new types we needed for the model can be created.

Parameters: The parameters of a vehicle type are changed by selecting the desired parameter in the main window, typing the desired values in the bottom set of slots **and** pressing *Enter* before selecting a new parameter, in order to validate the changes.

Articulated: Trams, buses, lorries or other sort of vehicles can be composed by segments of different lengths. This folder allows us to define the number of segments and the length of each segment. The *Add* button serves to create a segment, whose length we can then specify at the bottom slot. The total length of all the segments has then to approximately match the average length defined at the *length* parameter of the *Parameter* folder. If we set a *Deviation*>0 for the *length* parameter this means that vehicles of different lengths will be generated, so the final length of very

few of them will exactly match the total fixed length of the segments of the *Articulated* folder. 3D takes care of this inconsistencies by adjusting the final length of the last segment, transforming its length into a slightly variable parameter.

Classes: This folder is dedicated to the association of the vehicle type with the different vehicle classes when designing an O/D matrix or a traffic state (pg 63). This folder is deactivated when dealing with the general library.

Fuel: In this folder the parameters of the fuel consumption model are set. The meaning of each parameter is:

Fi: fuel consumption rate for idling vehicles

Fd: fuel consumption rate for decelerating vehicles

F1: fuel consumption rate for vehicles travelling at a constant speed of 90 km/h

F2: fuel consumption rate for vehicles travelling at a constant speed of 120 km/h

Vm: speed at which the fuel consumption rate is at a minimum

C1, C2: constants in the equation for the fuel consumption rate for accelerating vehicles in ml/s, $(c_1 + c_2 a v) \Delta t$

For details on the fuel consumption model please refer to the *AIMSUN User Manual*.

The screenshot shows the 'Attributes' window for a vehicle named 'truck'. The 'Fuel' tab is active, displaying the following parameters:

Parameter	Value	Unit
Fi (idling):	3.00	ml/s
C1 (accelerating):	1.00	ml/s
C2 (accelerating):	0.05	ml/s
F1 (at 90 Km/h):	14.00	l/100Km
F2 (at 120 Km/h):	18.00	l/100Km
Fd (decelerating):	3.50	ml/s
Minimum consumption speed		
Vm:	60.00	Km/h

Figure 3-19: Fuel Consumption folder

Pollutants: The parameters of the pollution emission model are set in this folder. This utility is designed to draw up a balance sheet of the pollution generated during the simulation. The usual procedure for creating a new pollutant is to press the *New Pollutant* button, make the necessary changes in the editing windows **and** to press Enter to validate the changes. IER stands for Idling Emission Rate; AER and DER stand for Acceleration and Deceleration Emission Rates. The user

then presses the *New Emission* button for each time they wish to add a new speed-emission pair value to the set of pairs used by the model. As for pollutant definition, the speed-emission pair is set by writing the values in the editing slots **and** pressing the Enter button to validate the changes. At least one pair of values must be entered here. If there is only one pair, that emission rate will be the one used for all cars travelling at a constant speed. If there is more than one pair, the emission rates for cars travelling at a constant speed will depend on the speed intervals defined by the different pairs. Using the data specified in the figure, emission of pollutants would be calculated as follows. For vehicles idling, accelerating or decelerating, the emission would be the one specified with the name of the pollutant. For cars travelling at 0 to 10 km/h, the emission rate would be the one specified by the value given for a speed of 10 km/h. For cars travelling at 10 to 20 km/h or 60 to 70 km/h the emission rate would be the one associated with the top speed of that interval. Although it is not apparent, the table only goes up to speeds of 70 km/h. Cars travelling at speeds greater than 70 km/h are assumed to have the same emission rate as those travelling at 70 km/h. This is an experimental-design decision. In the case of this example, the simulation is intended for an urban network, so effort was concentrated on defining small speed intervals. Since there were going to be very few vehicles travelling at speeds over 70 km/h, it was deemed better to have greater detail at that interval of speeds than to stretch to 20 km/h intervals. For more information on the model please refer to the AIMSUN *User Manual*.

Attributes

Name

Parameter | Articulated | Classes | Fuel | Pollutants

Name	IER (g/s)	AER (g/s)	DER (g/s)
CO	0.0600	0.3770	0.0720
NOx	0.0010	0.0100	0.0010
HC	0.0070	0.0200	0.0070

Cruising Rates

Speed Km/h	Emission Rate (g/s)
10.0000	0.0600
20.0000	0.0910
30.0000	0.1300

Figure 3-20: Pollution emission folder

Vehicle types of interest can be sent to the particular library of the result container, OD matrix or Public Transport Plan open at that time. In order to transfer vehicle types to/from a result container, OD matrix or Public Transport Plan, both the *Vehicle Types* dialog windows of the library and the result container, OD matrix or Public Transport Plan must be open (see Figure 3-21).

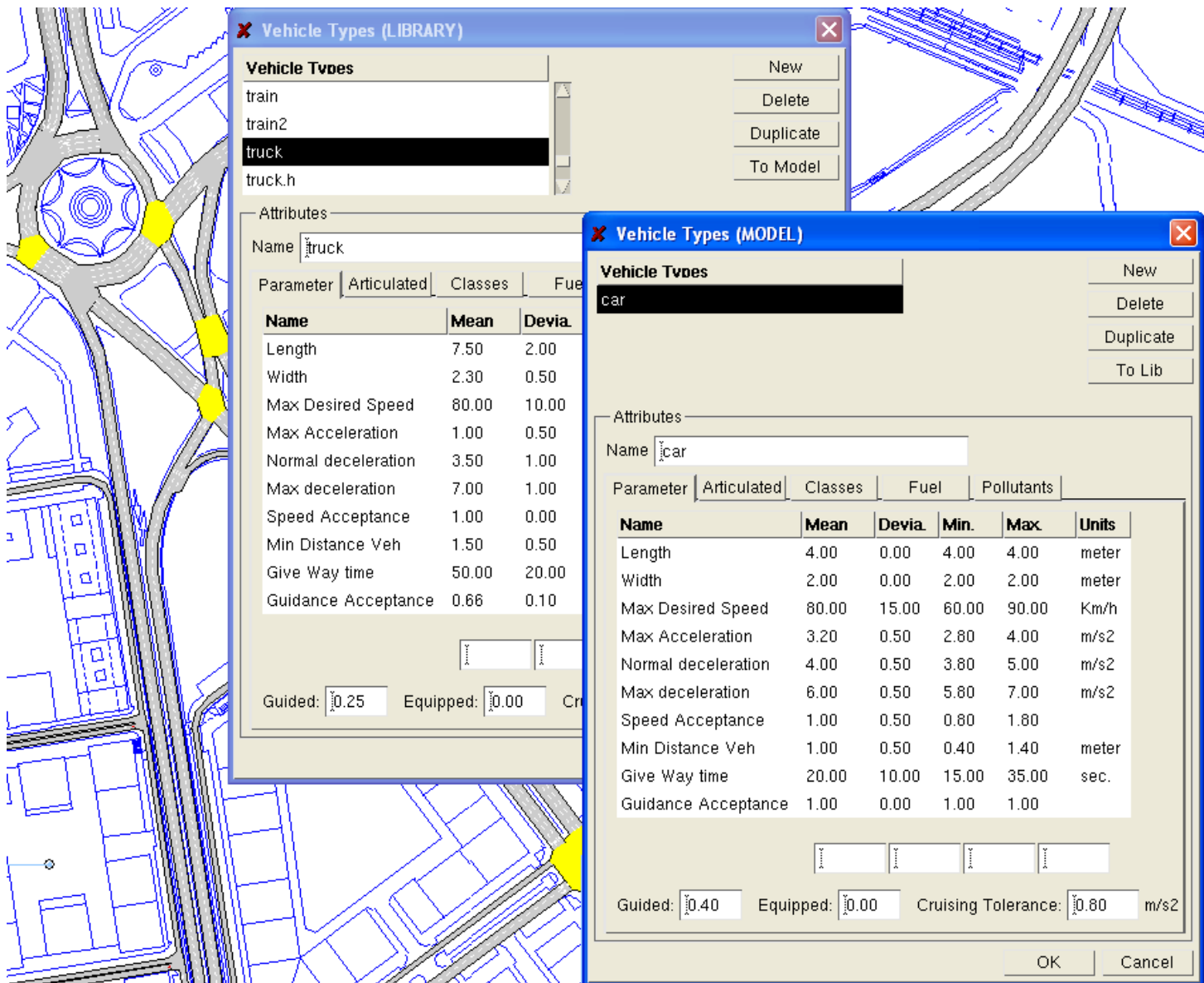


Figure 3-21: Vehicle type transfer to/from Model

3.2.13 Function Editor

When the simulation is based on O/D matrices and routes or paths instead of input flows and turning proportions, it is called a Route-Based simulation model. In this model, vehicles are fed into the network according to the demand data defined as an O/D matrix and they drive along a certain path within the network in order to reach their destination. AIMSUN makes use of two different types of functions to distribute the vehicles among the different paths between each origin and destination: the cost and the route choice functions. The cost functions calculate the cost perceived by the driver from the beginning to the end of a certain link. The route choice function takes the total costs of the routes to a particular destination and assigns a certain percentage of vehicles to each path depending on the cost difference between them. For a more detailed explanation go to AIMSUN user manual.

This option opens the dialog window of Figure 3-22. The dialog window includes six functions: creating a new void function (New), deleting the selected function (Delete), editing the selected function (Edit), cloning an already existing function (Duplicate), updating (Change) a network for

which one of the functions has been edited since it was initially assigned to the network, and copying a function to the Model/System Library from the System/Model Library (To System/Model).

The first step is to decide whether we want to edit a function for the System Library or for the Model Library. The System Library is the library of functions common to all TEDI sessions. It will be possible to access this library when editing all networks. But the functions that we will be able to assign to the links of a model will be those ones included in the Model Library. We can edit the function in the Model Library in case it is a function specific to that network and that we will very unlikely use with other models. Or we can edit the function in the System Library in case we think we might use it with future models, and we can then add it to the functions of our Model just by selecting it and pressing the 'To Model' button.

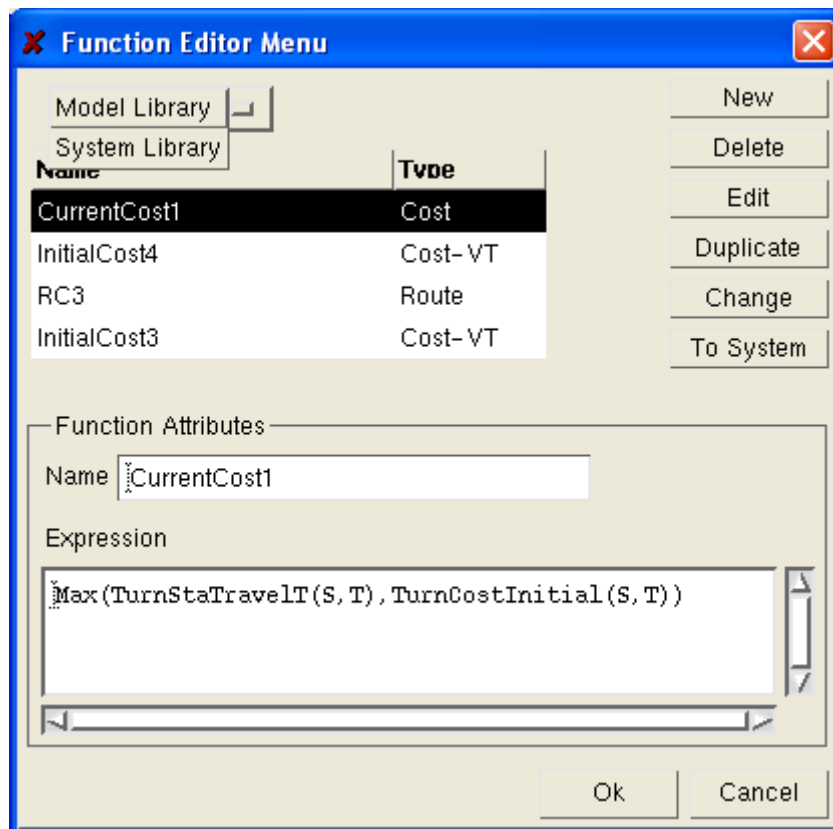


Figure 3-22: Function Editor Menu Dialog Window

Three different types of functions can be edited (see Figure 3-23): Cost functions, Cost functions with Vehicle Type and Route Choice functions. Cost functions with Vehicle Type are those cost functions that may give a different output depending on vehicle type. For instance a ring road could be an attractive route for private cars while taxi drivers could prefer to drive through the city centre.

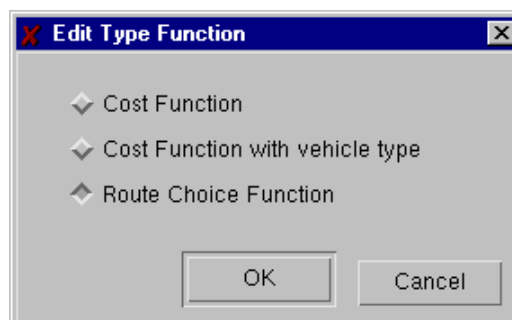


Figure 3-23: Function Type selection

Once a function type has been selected, this opens a window containing the different function types that are active depending on the chosen option, as can be seen in Figure 3-24, 3-25 and 3-26

Function		
Networks Description	NetMaxSectCapacity()	
Section Description	Rand(S)	
Turning Movement Description	TurnMaxSpeed(S,T)	
Statistical Section Data	SectStaFlow(S)	
Statistical Turning Data	TurnStaFlow(S,T)	
Vehicle Type Description	VehTypeMaxSpeed(VT)	
Statistical Section Data VT	SectStaFlowVT(S,VT)	
Statistical Turning Data VT	TurnStaFlowVT(S,T,VT)	
Turning Costs	TurnCostInitial(S,T)	TurnCostInitialVT(S,T,VT)
Shortest Path	SPCost(C,TR,S,K,VT)	
Check Vehicle Type	CheckVehType(VT,NameVehicleType)	
Clear	Ok	Cancel

Figure 3-24: Function Editor Dialog Window for Cost Function

Function	<input type="text"/>	
Networks Description	<input type="text" value="NetMaxSectCapacity()"/>	
Section Description	<input type="text" value="Rand(S)"/>	
Turning Movement Description	<input type="text" value="TurnMaxSpeed(S,T)"/>	
Statistical Section Data	<input type="text" value="SectStaFlow(S)"/>	
Statistical Turning Data	<input type="text" value="TurnStaFlow(S,T)"/>	
Vehicle Type Description	<input type="text" value="VehTypeMaxSpeed(VT)"/>	
Statistical Section Data VT	<input type="text" value="SectStaFlowVT(S,VT)"/>	
Statistical Turning Data VT	<input type="text" value="TurnStaFlowVT(S,T,VT)"/>	
Turning Costs	<input type="text" value="TurnCostInitial(S,T)"/>	<input type="text" value="TurnCostInitialVT(S,T,VT)"/>
Shortest Path	<input type="text" value="SPCumCost(C,TR,S,T,VT)"/>	
Check Vehicle Type	<input type="text" value="CheckVehType(VT,NameVehicleType)"/>	
<input type="button" value="Clear"/> <input type="button" value="Ok"/> <input type="button" value="Cancel"/>		

Figure 3-25: Cost Function with Vehicle Type

Function	
Networks Description	NetMa/SecCapacity()
Section Description	Rand(S)
Turning Movement Description	TurnMa/Speed(S,T)
Statistical Section Data	SecStaFlow(S)
Statistical Turning Data	TurnStaFlow(S,T)
Vehicle Type Description	VehTypeMa/Speed(VT)
Statistical Section Data VT	SecStaFlowVT(S,VT)
Statistical Turning Data VT	TurnStaFlowVT(S,T,VT)
Turning Costs	TurnCostInitial(S,T) TurnCostInitialVT(S,T,VT)
Shortest Path	SPCurrCost(O,TR,S,K,VT)
Check Vehicle Type	CheckVehType(VT,NameVehicleType)

Clear Ok Cancel

Figure 3-26: Route Choice Function

Functions are edited by choosing the required function from each function type library, and typing in any other recognised basic functions.

A list of all functions available and their functionalities is included in chapter 6.

Cost functions can be assigned to turning movements to generate the cost associated to that particular link. The cost is then used by the route choice function. AIMSUN comes with a set of default functions, but this feature allows you to set the functions of your preference for the entire network or for a subset of links. This is described in greater detail in section 4.8.1.

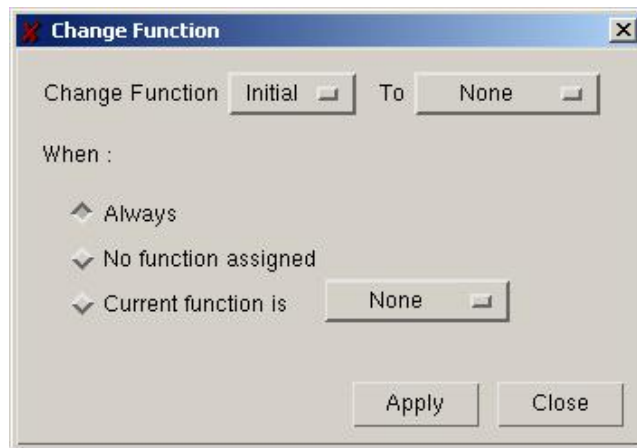


Figure 3-27: Change Function dialog window

The *Change* option opens the dialog window at Figure 3-27. A possible configuration of that window is the one at Figure 3-28.

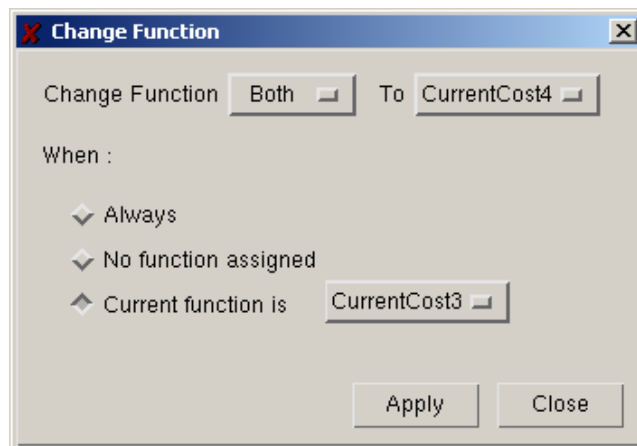


Figure 3-28: Possible utility

The meaning of the dialog window at Figure 3-28 is to change Both the Initial and Current functions to the function CurrentCost4, but only on those cases when the present Current function is CurrentCost3. This is a fast way of updating to CurrentCost4 all those links where we have set CurrentCost3 as the cost function without having to edit each one of them.

3.2.14 Preferences

This allows the configuration of parameters that are global to the traffic network currently being edited, using the dialog window shown in Figure 3-29.

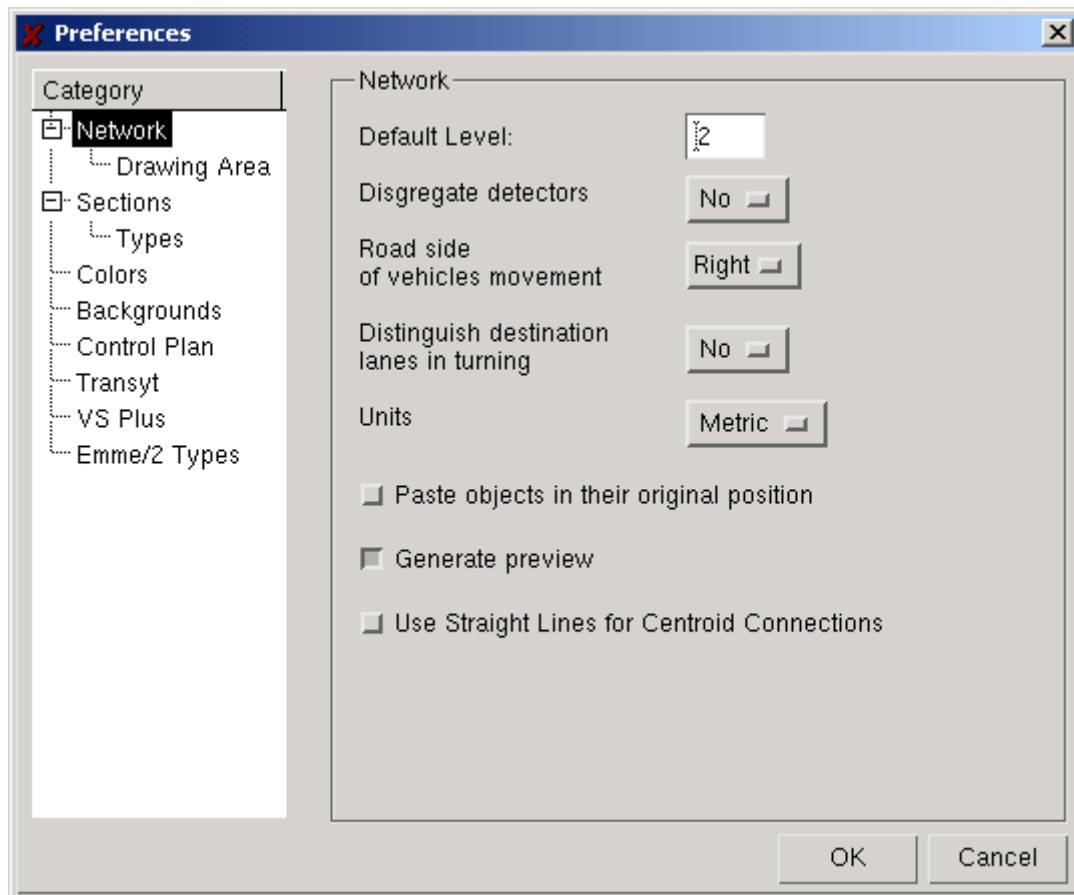


Figure 3-29: Preferences Dialog Window (Network)

There are different sets of global parameters:

– **Network:**

- The *Default Level* option specifies the default graphical level at which any new objects are drawn. The graphical level specifies which objects are drawn on top or below other ones.
- The *Disregate detectors* option specifies, for detectors covering more than one lane, whether the statistics are going to be collected for separate lanes or aggregated into a single statistic.
- The *Road side of vehicle movement* option is used to indicate to the simulator which side will be used by slow traffic. Setting the *Distinguish destination lanes in turning* option to *Yes* allows to decide not only the origin lanes but to decide which subset of the destination section are going to be the destination of the turning movement. This is because the destination lanes in the destination section of a turning don't have to be selected one by one, but are all simultaneously selected just by selecting the destination section. On the other hand, there might be special turnings for which vehicles coming from a certain origin lane might not be allowed to all lanes of the destination section but only to a subset of them. If this is the case, it is not necessary to design the whole network with a *Yes* for this option. It is only necessary to change this option when designing these particular junctions. Changing this option does not affect any junctions previously designed.
- *Units*, allows the metric or the Imperial system to be used.
- *Paste objects in their original position*, pastes the copied objects on top of the original ones,

in their original position. Otherwise objects are pasted in the centre of the TEDI screen (in case they fit there) or in the centre of the drawing area (if they didn't fit in the view).

- *Generate preview* generates the model's preview for the *preview* folder of the *open network* window each time the model is saved.
 - *Use Straight Lines for Centroid Connections*, draws the connectors as a straight line from the centroid to the network, instead of drawing them with horizontal and vertical lines.
- **Drawing Area:** this is where the traffic network is displayed, providing a realistic view of all its features. It sets the size of the real world environment within which elements can be drawn. The real-world area is a rectangular window to the real world, within which drawing is permitted. When the *Drawing Area* option is selected, the information displayed in Figure 3-30 is displayed at the right hand side of the window. This is where you can specify the coordinates of the bottom left and top right corners that define the real-world rectangle. The user specifies the minimum and maximum coordinates for this area. The minimum and maximum coordinates must be in the positive quadrant. The values displayed in the figure are the default values assigned by the editor to a new network. *Grid point* sets the distance between consecutive grid points. If the grid is activated (*View* menu), the different objects will snap to grid points instead of moving or changing dimensions.

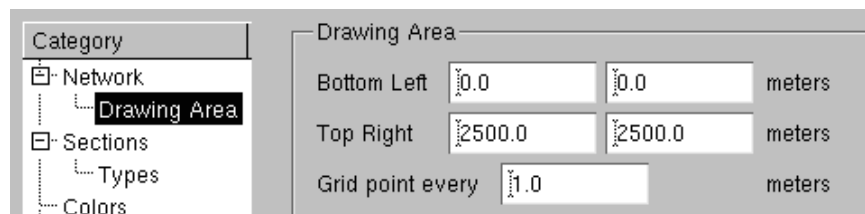


Figure 3-30: Preferences Dialog Window (Network Drawing Area)

- **Default Lane Width:** This specifies the default lane width of new sections. Changing the default value does not affect the sections created before, only the ones created from that point on. The default value can be changed as often as necessary.

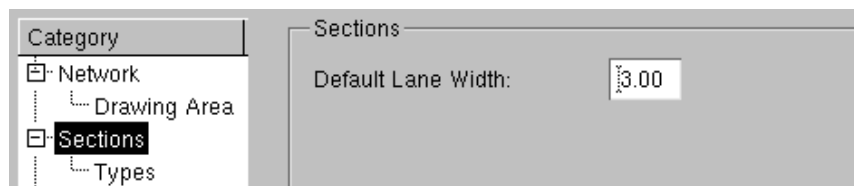


Figure 3-31: Preferences Dialog Window (Sections)

- **Types:** allows the configuration of the existing types of sections and the design of new ones. Eight section types with their corresponding attributes are predefined. The different possible actions are:
- *New:* creates a new section type.
 - *Delete:* removes the selected section type. It does not allow you to remove the eight default section types.
 - *Duplicate:* duplicates an existing section type.
 - *Default Type:* activating this toggle for a given section type makes it the default section type, that is, all new sections will be created as this section type.
 - *Modify Sections:* modifies the attributes of the sections in the network of an existing section

type. Using this button, the attributes of existing sections of this type will be changed to the new values.

To either remove or modify a section type, it must first be selected in the window. There are two types of sections, basic sections and custom sections. Basic sections are the ones initially provided with the editor, which cannot be deleted. Custom sections are developed by the user.

The significance of each parameter and the effect on the simulation when one of these values is changed is explained in the AIMSUN User Manual.

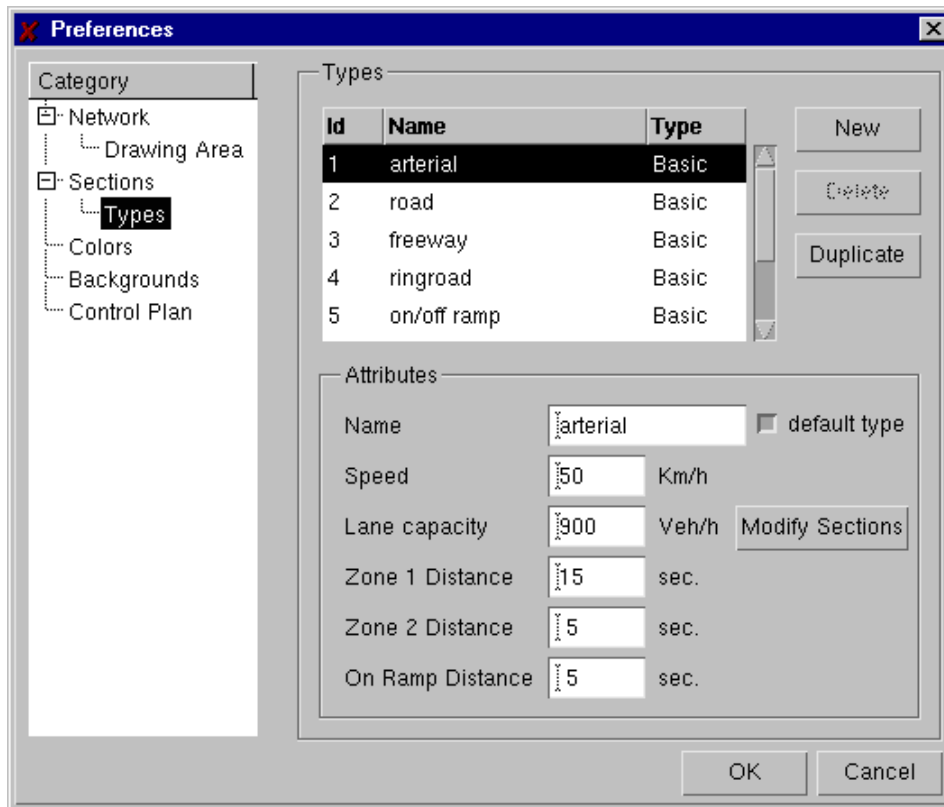


Figure 3-32: Preferences Dialog Window (Section Types)

- **Colours:** clicking on the coloured rectangle positioned to the right of each object opens the window shown in Figure 3-34. Changes are made by altering the proportions of the three primary colours. These changes are reflected by the change in colour of the top square, while the bottom square retains the original colour for purposes of comparison.

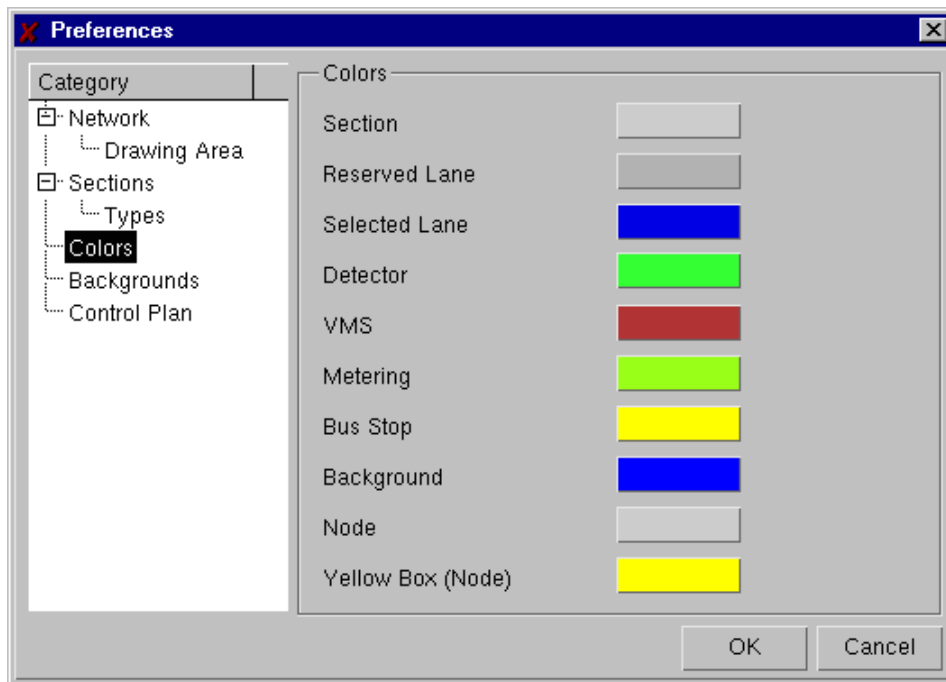


Figure 3-33: Preferences Dialog Window (Colours)

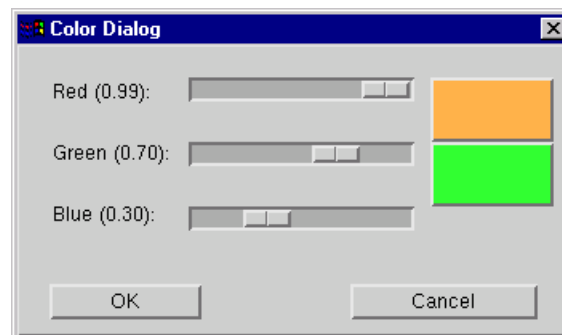


Figure 3-34: Colour Dialog Window

- Background: this is where you can set the toggles to determine, among other options, whether to show the background contents while dragging it to its desired location or to draw each layer of the dxf background with the same colour or with the ones indicated in the file.

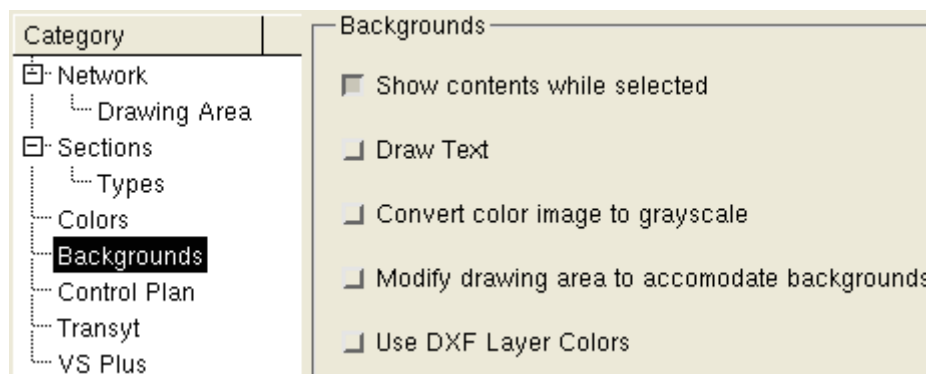


Figure 3-35: Preferences Dialog Window (Backgrounds)

- Control Plan: this is where you set the flag to specify whether or not you wish to edit maximum and minimum times for the traffic signals phases, in addition to the initial timing. These maximum and minimum times can then be used as a limit for possible changes of phase timing

set by an adaptive traffic control system, otherwise the ‘min’, ‘max’ slots for those timings remain deactivated.

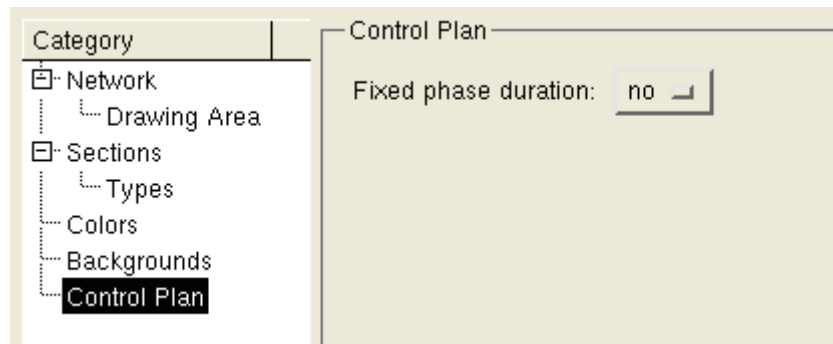


Figure 3-36: Preferences Dialog Window (Control Plan)

- Transyt & VS-Plus & Emme/2: this options are used to set a few of the parameters used by the Transyt, VS-Plus and Emme/2 interfaces.

3.3 View Menu

3.3.1 The Grid

The grid is a common feature in all CAD programs. In the Layout menu, there are two options associated with the grid: Turn autogrid on and Show grid. When the grid is activated, the end points of new sections, texts, blocks, etc, are limited to grid points, its update points snap to the closest grid points when created. Selecting the option *Show grid* from the *Layout* menu makes the grid visible to the user. The option toggles to ‘Hide grid’ so that the next time it is chosen it hides the grid. Hiding the grid does not deactivate it, just as showing it does not activate it. This means that the grid can be used merely as a background reference, with no geometrical consequences, by displaying it without deactivating it.

By default, the distance between grid points is set to 1 meter, but it can be changed in the *Drawing Area* folder of the *Edit - Preferences* option (see Figure 3-30).

3.3.2 Show line pointer

Shows the two lines whose intersection defines the tip of the cursor. Changes to *Hide line pointer* once activated.

3.3.3 Show objects

Displays a dialog window in which the user can specify, for each object, whether to display it, its identifier and its name. In the window there are four columns. The header for the first column is *object* and it describes the type of object to which the rest of options of the row refer. The second column indicates whether the object is visualised at TEDI. The third column indicates whether the Identifier of the object is displayed with the object. The final column indicates whether the name is displayed with the object. These options can be changed with a single mouse click on an empty position (to change it to a *Yes* in order to display the object/id/name) or on a *Yes* (to hide an object/id/name).

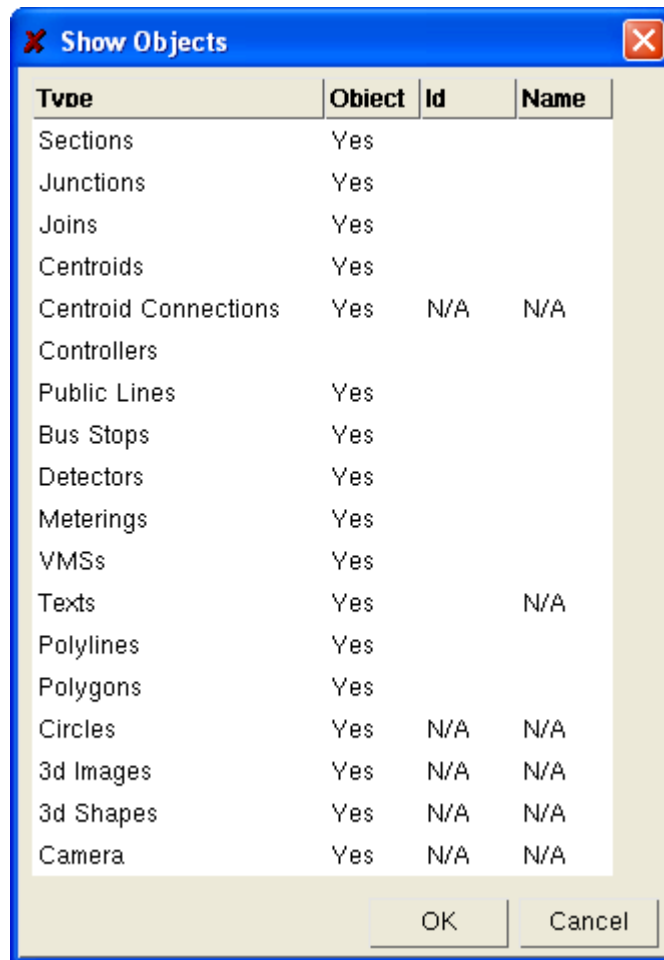


Figure 3-37: Show objects window

3.3.4 Show section speed

Displays a sign with the top speed permitted for each section. An example of this can be seen in Figure 3-38. Once activated, this option changes to *Hide section speed*.

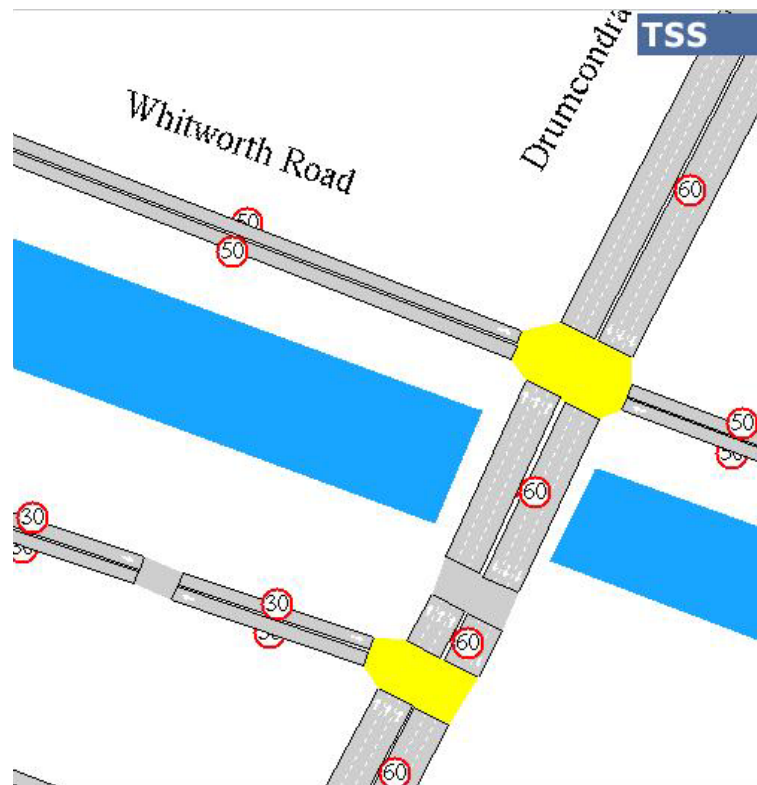


Figure 3-38: Show Speed view

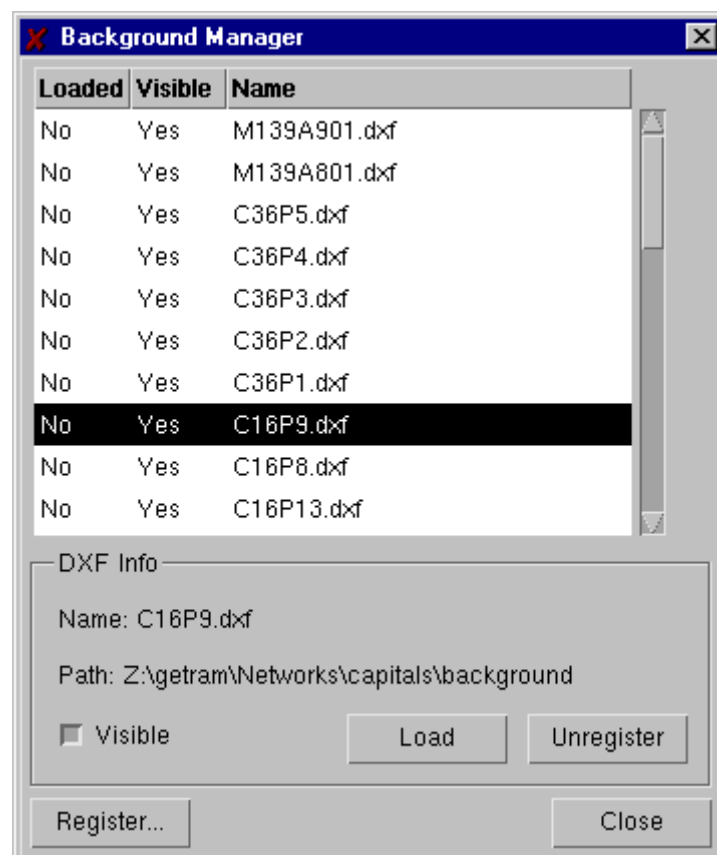
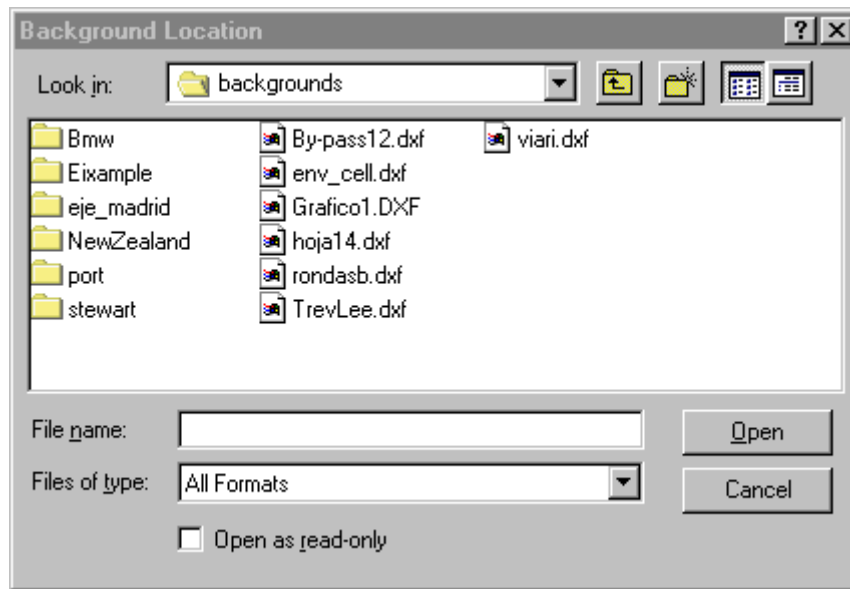


Figure 3-39: Background window

3.3.5 Background

This sets the background pixmap of the editor to the contents of a set of graphical files. The user can load a set of background images as an aid to editing large traffic networks. This is particularly useful for modelling the sections with the correct real-world dimensions. Once loaded into the drawing area, the graphical images can be modified by translations, rotations and changes of scale, so as to align them with other backgrounds and traffic objects.

Figure 3-40: Register Backgrounds window



The “Register...” button calls the dialog window shown in Figure 3-40. An image must be registered before it is used. This is done by pressing the *Register...* button. A dialog window then appears (next image) in which the user simply selects the required image file and clicks on the *Open* button.

Once the image has been registered, it can be loaded as a background. This is done by selecting it in the list box and then choosing the *Load* button (see Figure 3-39). An example of the result can be seen in Figure 3-41. The accepted formats are **dx**f, **bm**p, **jp**g, **ti**f and **gi**f.

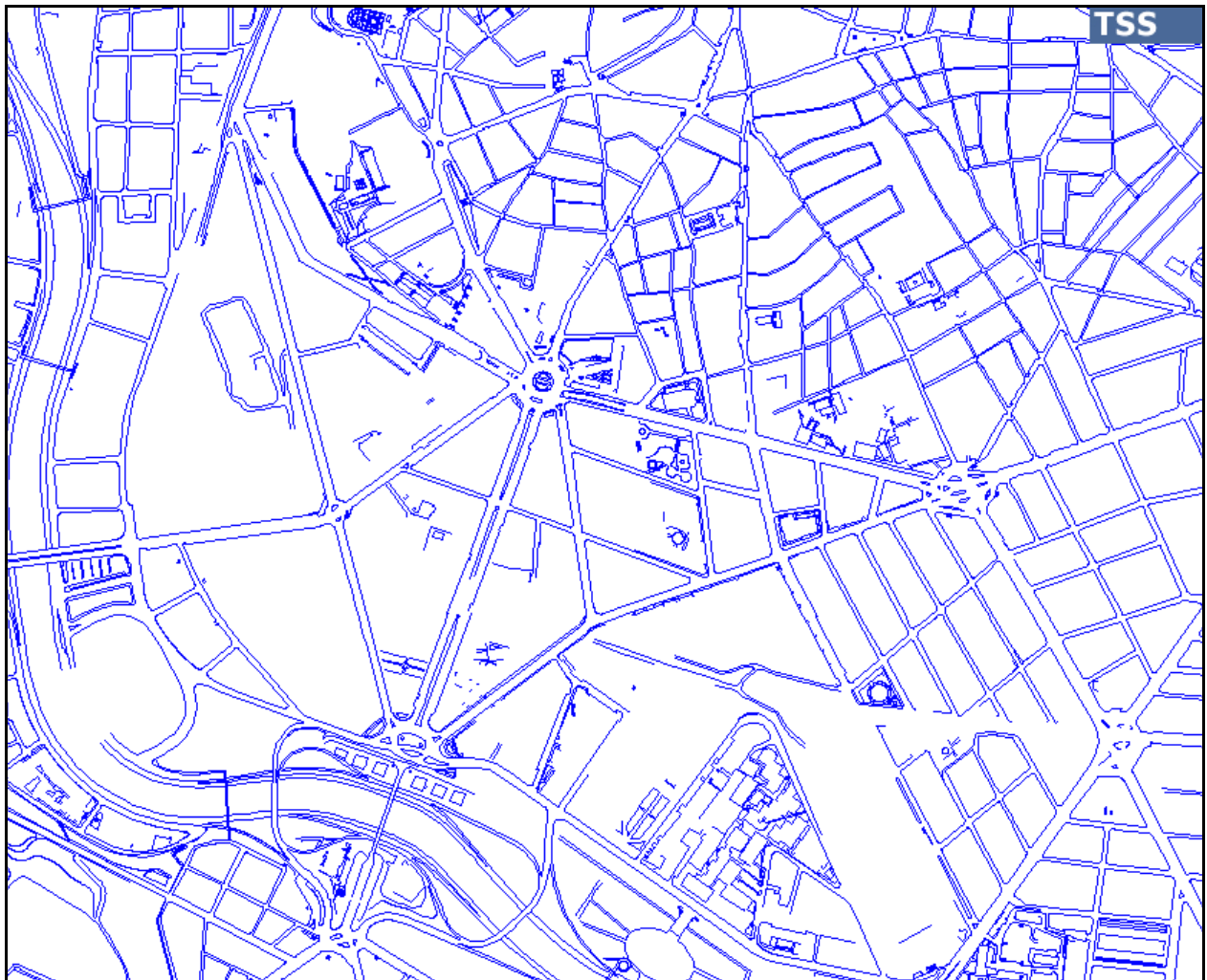


Figure 3-41: Example of a loaded background

To edit a background, you first double-click on its name in the Background Manager window. This opens the window shown in Figure 3-42. This Background dialog window allows us to change the path to the background (in case we have copied the network from somewhere else or we have changed the location of the backgrounds), translate the background by typing the required coordinates for the lower left vertex, to alter its scale in order to make it correspond to its real-world dimensions, and to rotate it to the desired orientation. All these transformations **must** be confirmed by pressing *Enter*. The background can also be moved by dragging it.

The other folder in the dialog window is the one containing the background's layers (see Figure 3-42: Background Dialog Window – Basic folder

). If the background has not been loaded, the layer list will appear empty.

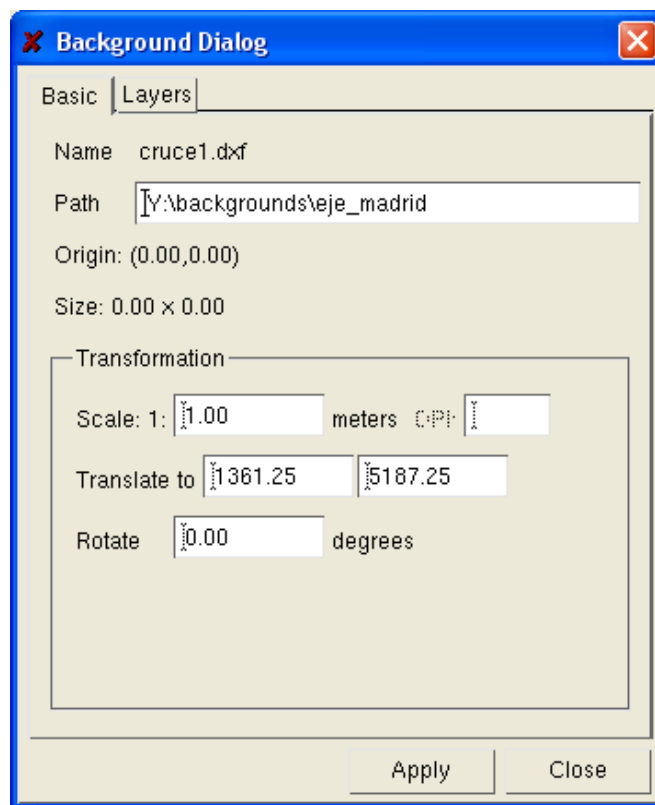


Figure 3-42: Background Dialog Window – Basic folder

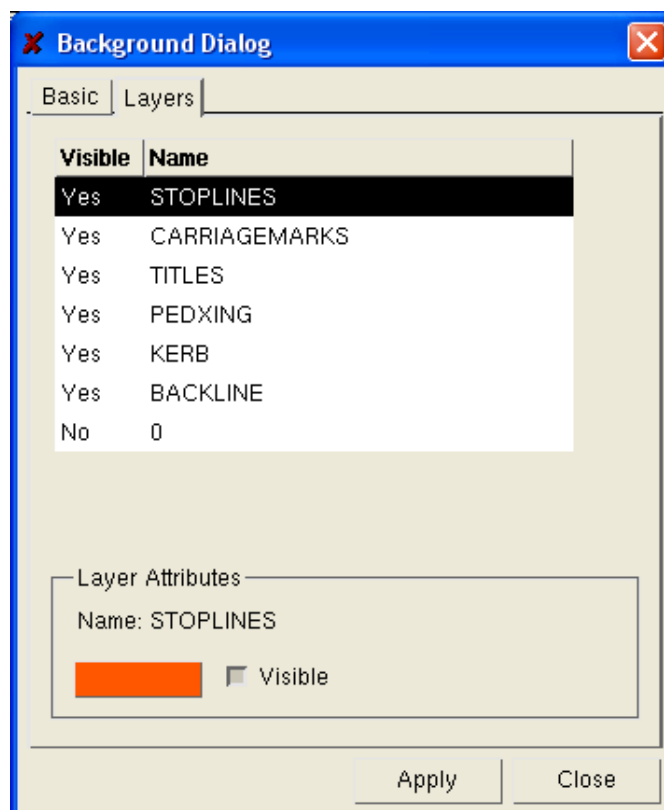


Figure 3-43: Background Dialog Window – Layers

This dialog window allows you to select layer-by-layer and activate/deactivate each layer using the

Visible toggle.

Problems often occur when loading very large backgrounds. The editor tries first to load the background to the co-ordinates defined by the image file. If these co-ordinates do not lie within the real world or if the background does not fit within the area because the box that contains the real background is larger than the real-world environment, then the Background manager cannot guarantee that it will load the background correctly or translate it to its position. The warning that is issued when this occurs is shown in Figure 3-44.



Figure 3-44: Warning window

The recommended action is either to enlarge the real world area to at least the size of the background and then translate the background, or either activate the option *Modify drawing area dimensions to accommodate backgrounds* at the *backgrounds* folder of the *preferences* window of the *edit* menu. Here, however, an additional complication arises. When editing a background, its size can be read as part of the information contained in the *Basic* folder. The warning message may also appear when the background is initially loaded with negative co-ordinates, even if the real world is already big enough for the image box.

3.3.6 Extrude Background Layer

This option allows to automatically generate 3D blocks from a 2D background. This is a function added with the objective of simplifying the edition of 3D networks, and doing so using the DXF file loaded as a background.

It takes a layer from the loaded DXF background/s and extrudes all the polygons contained in the layer. Extrude can be interpreted as creating TEDI polygons. And those polygons are converted into 3D blocks as soon as they are assigned a height $\neq 0$.

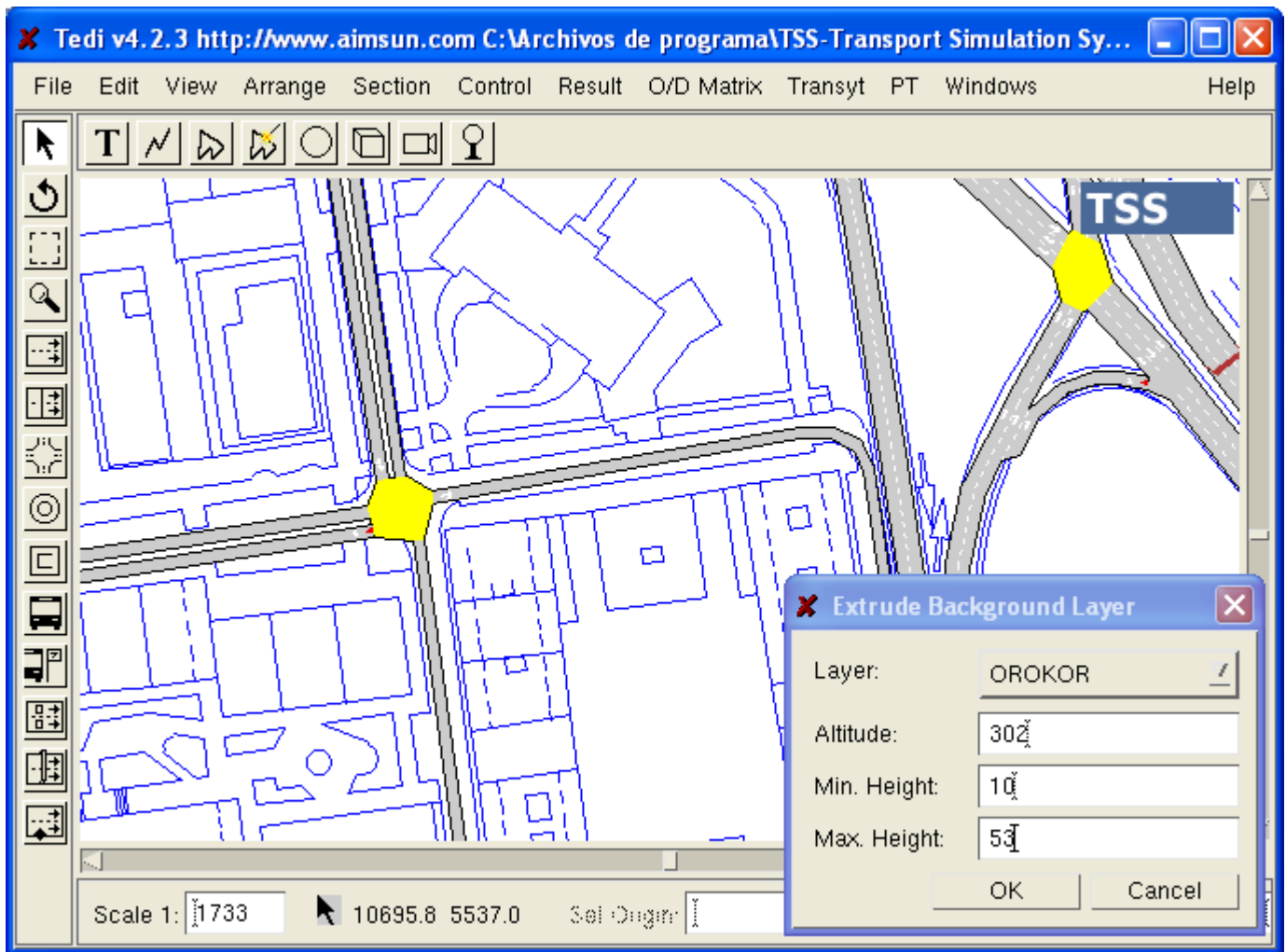


Figure 3-45: Extrude Background Layer Dialog Window

You can specify the following parameters:

- a) Altitude: the altitude (above sea level) of the TEDI polygons (0 by default)
- b) Min Height and Max Height: each TEDI polygon will be assigned a random height between *Min Height* and *Max Height*. Height. If *Min Height* = *Max Height* then all the polygons will have the same height.

Note that not all the DXF closed shapes are always polygons. Quite frequently a shape that looks as closed polygon really is a collection of unconnected segments that looks like a polygon. In such a case TEDI will not be able to create a polygon.

Such a tool is very useful when you have a DXF that contains, in one of its layers, buildings. You can then create a new 3D scene in no time.

3.3.7 Whole drawing

Changes the scale so that the whole real-world rectangle fits into the display window: so that we can see the whole network

3.4 Arrange Menu

3.4.1 Move forward

Moves the selected object(s) forward, adding one level to the original level of the object. This means increasing the graphical priority of these objects, i.e., showing them in front of other objects.

3.4.2 Move backward

Moves the selected object(s) backwards, decreasing in one the level of those objects. This means decreasing the graphical priority of these objects, i.e., showing other objects in front of them.

3.4.3 Align Centres

Moves the selected graphical decorative objects (circles, lines, polygons, text, 'trees' and graphical objects) one on top of the other positioning their centres on the same point.

3.4.4 Group

Groups a set of selected graphical objects into a single object, in the same way as, for instance, Microsoft's Word image editor. This is done to allow an easier manipulation of a graphical object composed by several other ones.

3.4.5 Ungroup

Ungroups the selected graphical object into the set of objects that compose it.

3.4.6 Rotate

This option is the same as the 'rotate' icon below the mouse icon in the toolbar. This is used when manipulating any object or group of selected objects that you want to rotate. When an object is selected, the rotational axis is positioned at the centre of the object, but it is possible to drag that point to any other desired location that simplifies the edition. An example of this feature is illustrated in Figure 3-46.

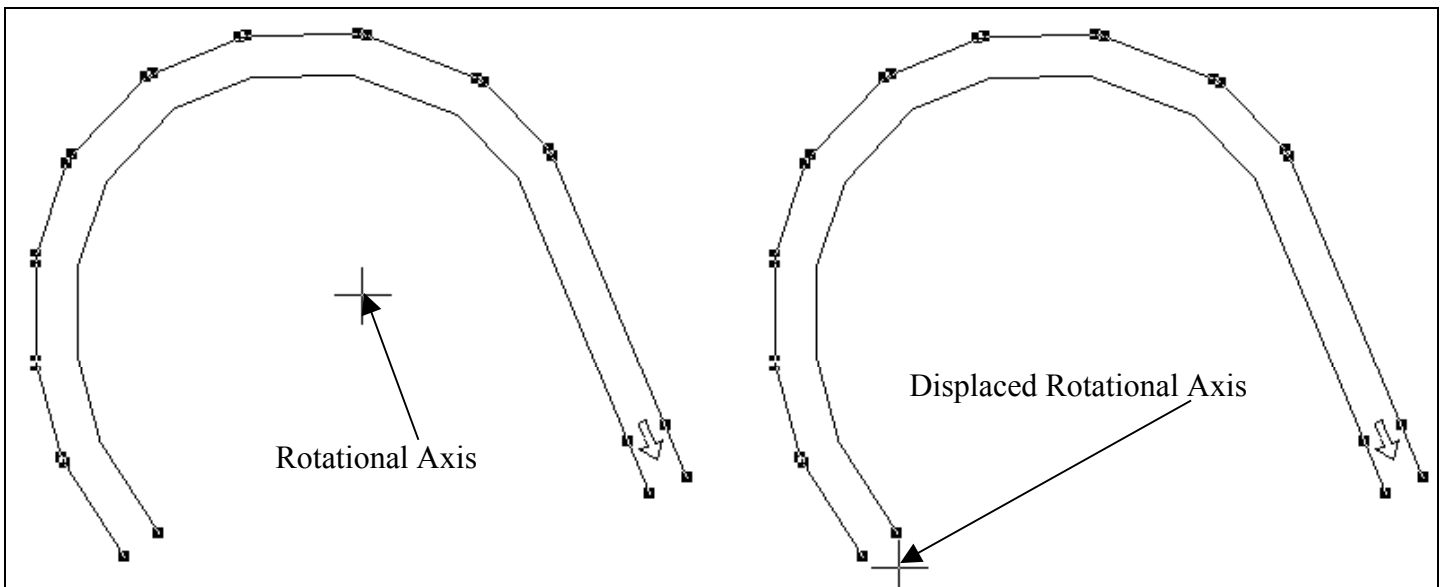


Figure 3-46: Rotate and Rotational Axis

3.5 Section menu

3.5.1 Disable side lanes

Disables the creation/update/deletion of side lanes.

3.5.2 Make join

Joins the selected sections automatically creating a new node. For example, you may often need to link a section arising from a highway exit to a road entry (see section 3 in Figure 3-47). In the figure, we have generated the section from the exit, but the other end of the section is not linked. In order to link the end of section 3 with the entry to the acceleration lane of section 5, the sections must touch one another, in such a way that the visual appearance is approximately the same as the one desired for the final join. Once this is done, the last step is to select both sections and to choose the *Make join* option. However, there is also another option: you can make several joins simultaneously by selecting all the sections involved.

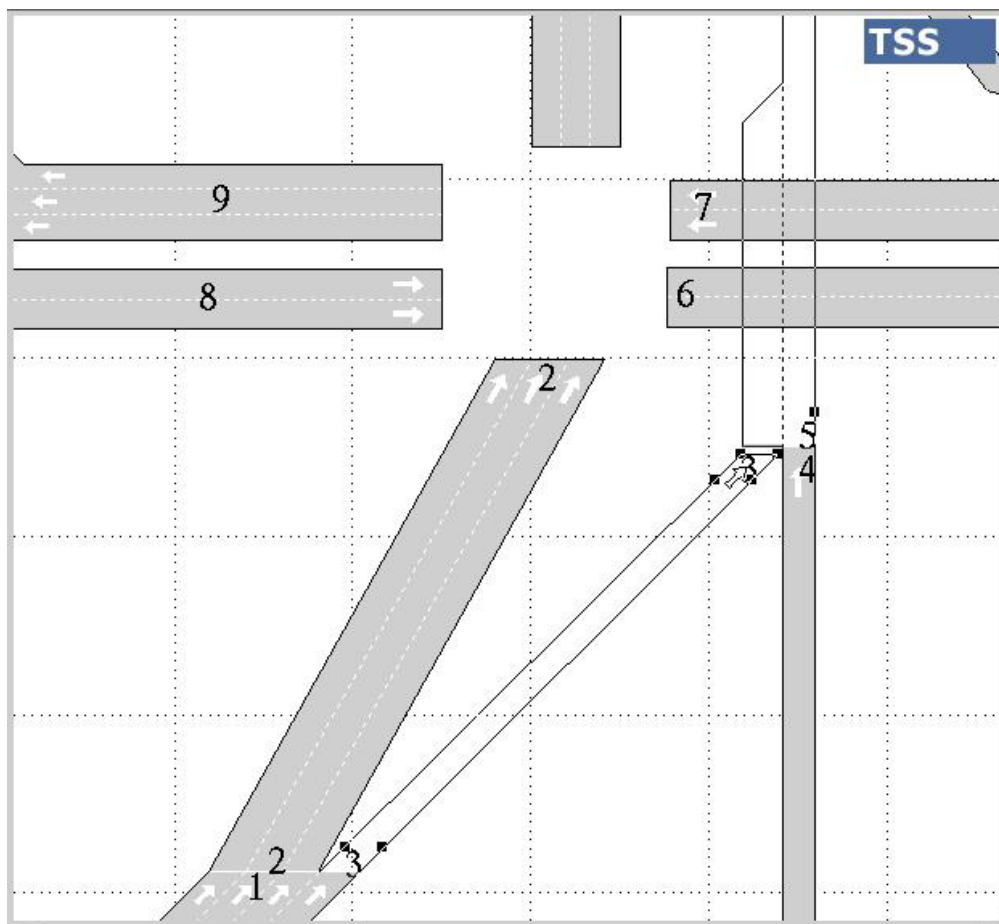
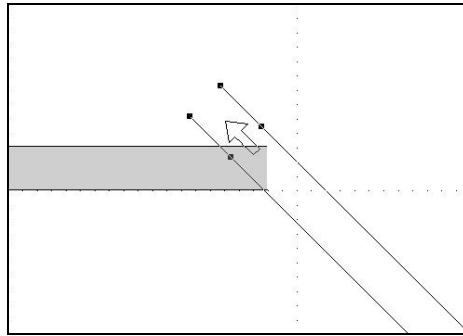
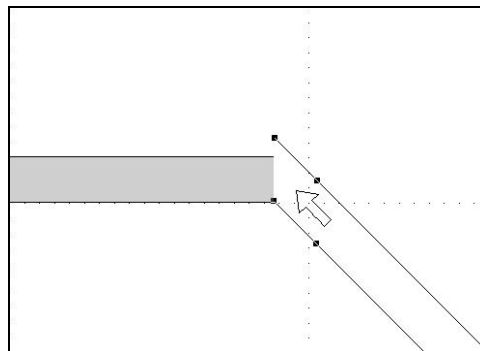
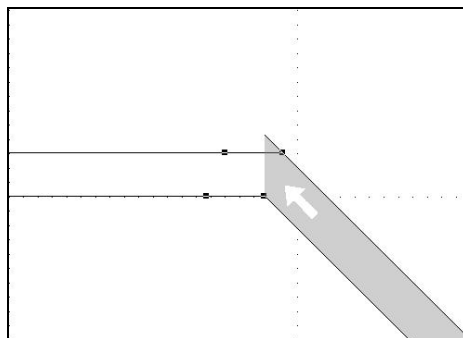
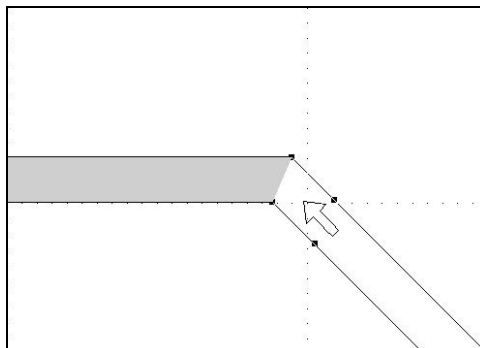


Figure 3-47: Making a join

Quite often the user may find that the two lanes to be joined have different widths. There are several ways of dealing with this problem. In the standard situation displayed in Figure 3-48, the most commonly used procedure to follow is illustrated in Figure 3-48

, Figure 3-49

and Figure 3-50

**Figure 3-48****Figure 3-49****Figure 3-50****Figure 3-51**

As can be seen from the sequence of images, you can adjust the geometry in such a way that the two lanes fit nicely together by altering the orientation of the exit and entry of the sections involved. Double-clicking on the division between two connected sections (see Figure 3-52) can open a join.

The main use of this function is to check and edit the turning speed allowed for vehicles going from one section to the next. If the turning speed assigned by the editor is considered incorrect, the speed can be changed and the *Auto Re-evaluate Speed* toggle deactivated, so that the system does not recalculate and change the new turning speed.

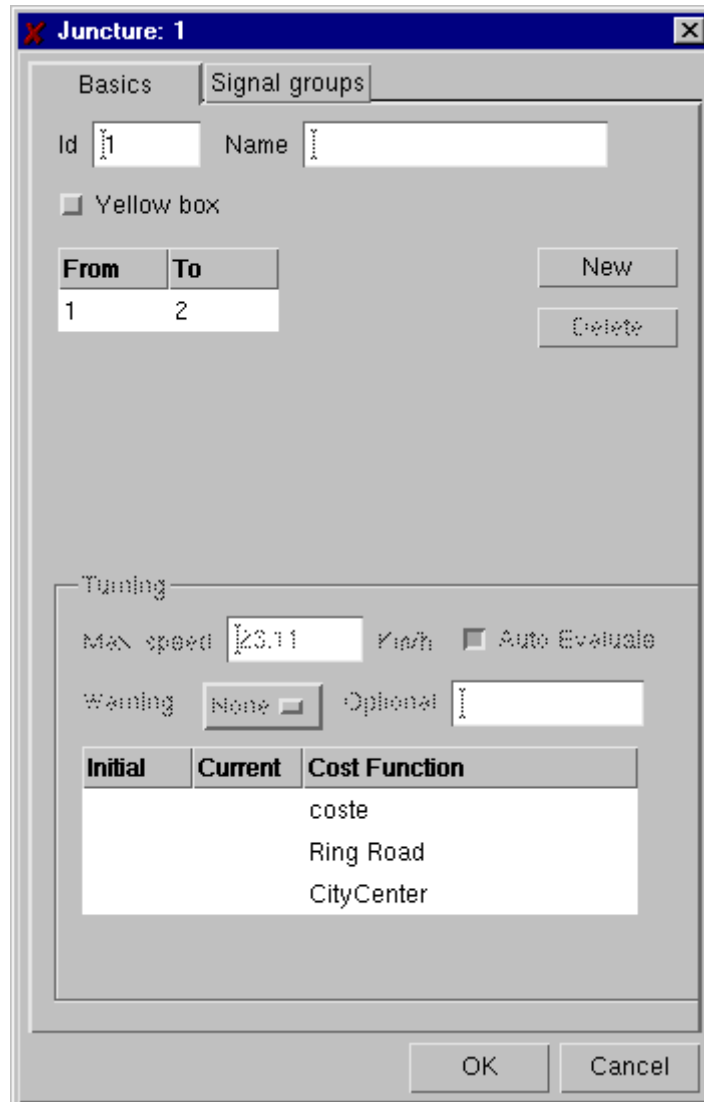


Figure 3-52: Open Join

3.5.3 Break join

This breaks the selected sections apart. The join between two sections is broken similarly to making a join: selecting both of them and choosing this option. In this case we have the same option as previously: we can break several joins simultaneously by selecting all the sections involved.

3.5.4 Make polysections

Groups a set of selected individual sections into a polysection (see Figure 3-53 and **Figure 3-54**). The mechanism is to select the set of sections of interest, and then, after selecting this option, the result is as shown in

Figure 3-54. The sections have to be selected in the order of circulation because they are connected in the order of selection. Once grouped, it is only possible to create side lanes at the entrance and exit of the polysection. It is also possible to manipulate the individual sections forming the

polysection, see Figure 3-55: the top section of this polysection has been rotated to its new orientation, and then its length and position adjusted.

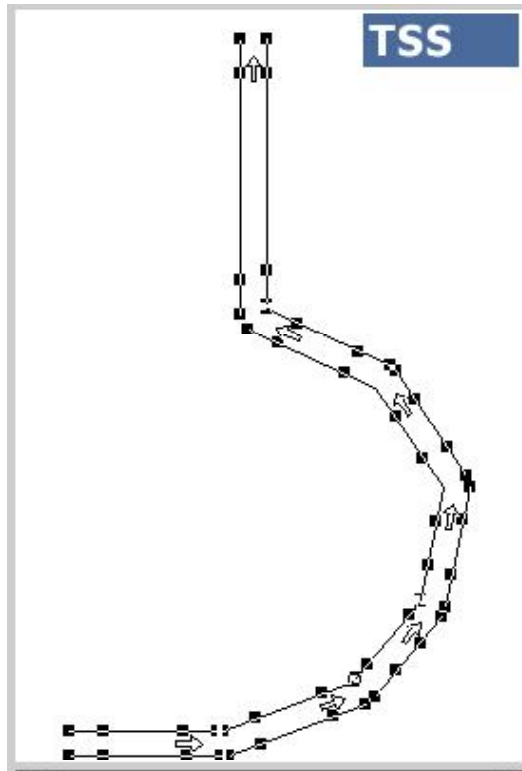


Figure 3-53: Selection prior to making the polysection

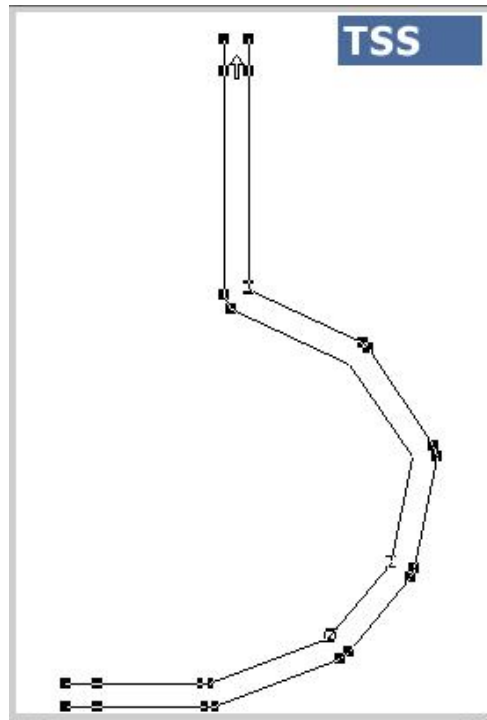


Figure 3-54: Polysection

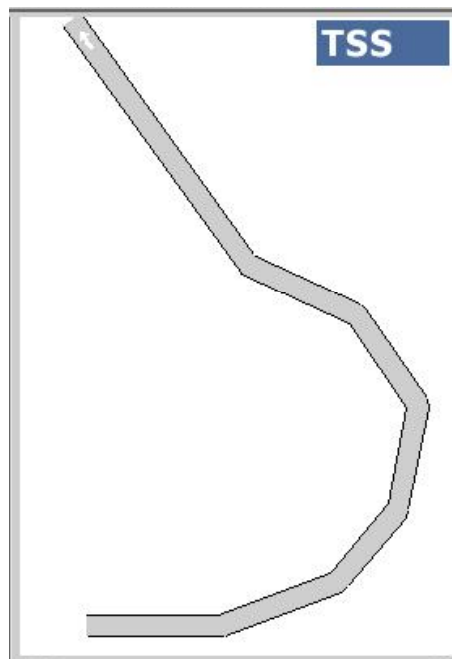


Figure 3-55: Altered polysection

A group of joined individual sections can be grouped into a single polysection as if they were isolated sections: there is no need to break the joins before grouping the sections.

3.5.5 Break polysection

This ungroups the sections of a polysection into individual sections.

Figure 3-54 and Figure 3-53 could serve as an example, but in the reverse order to that of the *Make Polysection* option. You first select the sections of the polysection, and then activate the option.

3.5.6 Change direction

This changes the direction of the selected sections. It only changes the direction of disconnected sections or isolated polysections, and has no effect on any section connected to another section by a junction or a join. If you wish to change the direction of, say, a few kilometres of highway formed by several sections connected through joins, one option, besides that of rebuilding, is to select all the sections involved, select the *Break join* option, select the *Change direction* option, and finally, select the *Make join* option to relink the sections.

3.5.7 Number of lanes (#)

This displays a submenu containing the following options: *1, 2, 3, 4, 5, More..., Default*. The numbers represent the desired number of lanes for the selected sections or for future sections. We can increase/decrease the number of lanes of a section by increasing/decreasing its width or by selecting it and then choosing in the *Number of lanes* option the desired number of lanes for that section. If we choose a number of lanes without having selected a section, then that number will be set as the default number of lanes, that is, it will become both the number in parentheses beside *Number of lanes* and the original number of lanes of any new section at the time of its creation. The *More ...* option opens the image in Figure 3-56, which is used when the required number of lanes is greater than 5. Again, if you have previously selected a set of sections, then the number of lanes in those sections changes to the number that you choose. Otherwise, if you have not selected any sections, then the selected number of lanes will become the default number of lanes for any sections created from that point on. In case of having selected section, then the width of the sections will change to *number of lanes times default lane width* (parameter set in the *Preferences* option of the

Edit menu).

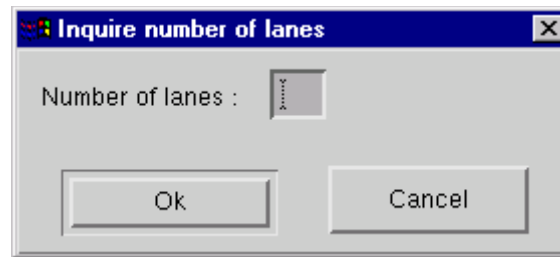


Figure 3-56: Inquire number of lanes

3.6 Control menu

3.6.1 New Control

This creates a new control plan in the current view. Although nothing appears to happen after selecting this option, selecting the *Control* menu again will reveal that several options that were not initially accessible are now activated. It is only possible to create control plans once the network has been given a name, i.e., has been saved from *untitled* to a specific name.

3.6.2 Open Control

This opens an existing control plan. As with the *New control plan* option, nothing appears to happen at first, but you will then notice that several options have become accessible. The window opened for this purpose is displayed in Figure 3-57.

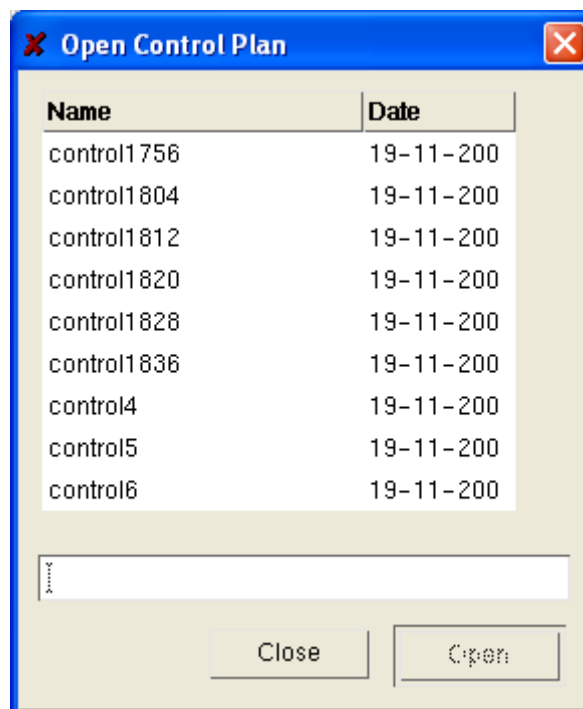


Figure 3-57: Open Control Dialog Window

3.6.3 Close Control

This closes the current control plan. If the latest changes to the control plan have not been saved, a dialog window will ask the user to confirm the action.

3.6.4 Save Control

This saves changes to the currently opened control plan. It is not necessary to close the *Junctions* (Figure 3-61) or *Meterings* (Figure 3-62) window to save changes.

3.6.5 Save Control as

This is the same as *Save Control*, except that it provides you with the option to change the name under which the control plan is saved. The dialog window for this option is displayed in Figure 3-58.

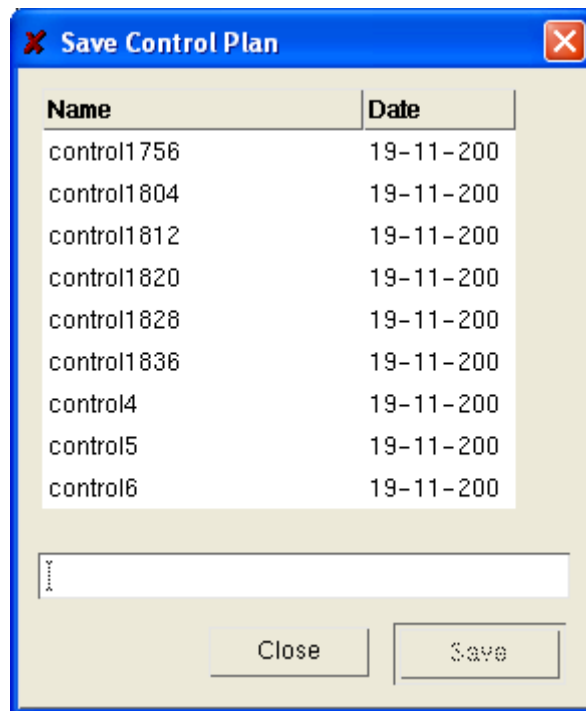


Figure 3-58: Save Control Plan Dialog Window

3.6.6 Import Control

This option allows selecting two Synchro files (.csv format) to import them and automatically create a GETRAM control plan.

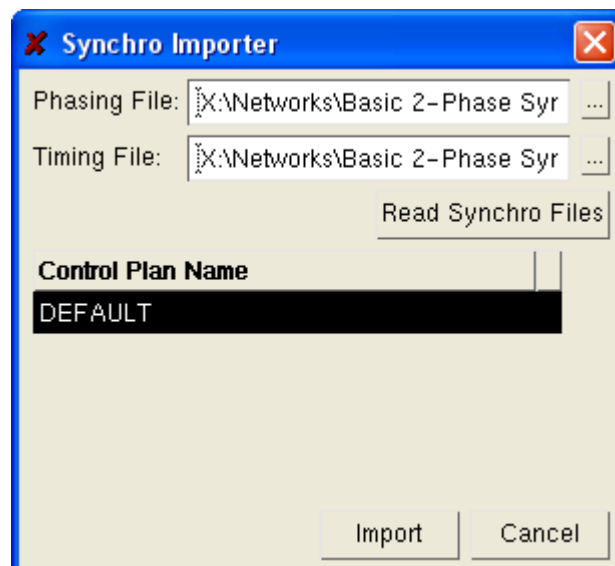
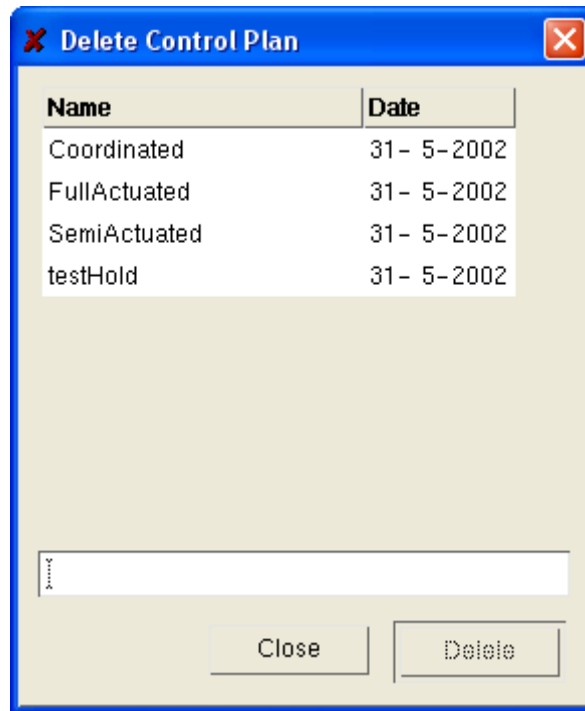


Figure 3-59: Synchro Importer Dialog Window**3.6.7 Delete Control**

This deletes a control plan. A control plan can be selected and deleted using the dialog window displayed in Figure 3-60. A control plan cannot be deleted while it is still open.

**Figure 3-60: Delete Control Plan Dialog Window****3.6.8 Junctions**

This changes a junction's control plan. Choosing this option opens the dialog window displayed in Figure 3-61 (dialog window for traffic signals in a control plan). The different parameters associated with a junction are displayed in the window when the junction is selected by clicking on the *id* in the window.

The first of these parameters is **junction type**. Four different choices are available:

Uncontrolled: junction managed by stops and give-ways. The traffic lights, if any, are disconnected.

Fixed (or pre-timed): junction managed by traffic lights with fixed green-light phases. Different control plans may be assigned to different time periods, to simulate the change of timings depending on the time of day.

Actuated: Traffic control system that follows the NEMA (National Electrical Manufacturers Association) standards (Section 5). For further details on those standards please refer to *Traffic Control Handbook* by the Federal Highway Administration.

External: It follows pre-timed control but can be combined with an external control policy implemented using the GETRAM Extensions.

Beside the *type* indicator a *critical* toggle can be seen. This is used to mark those junctions that may have a preferential treatment in a given adaptive system.

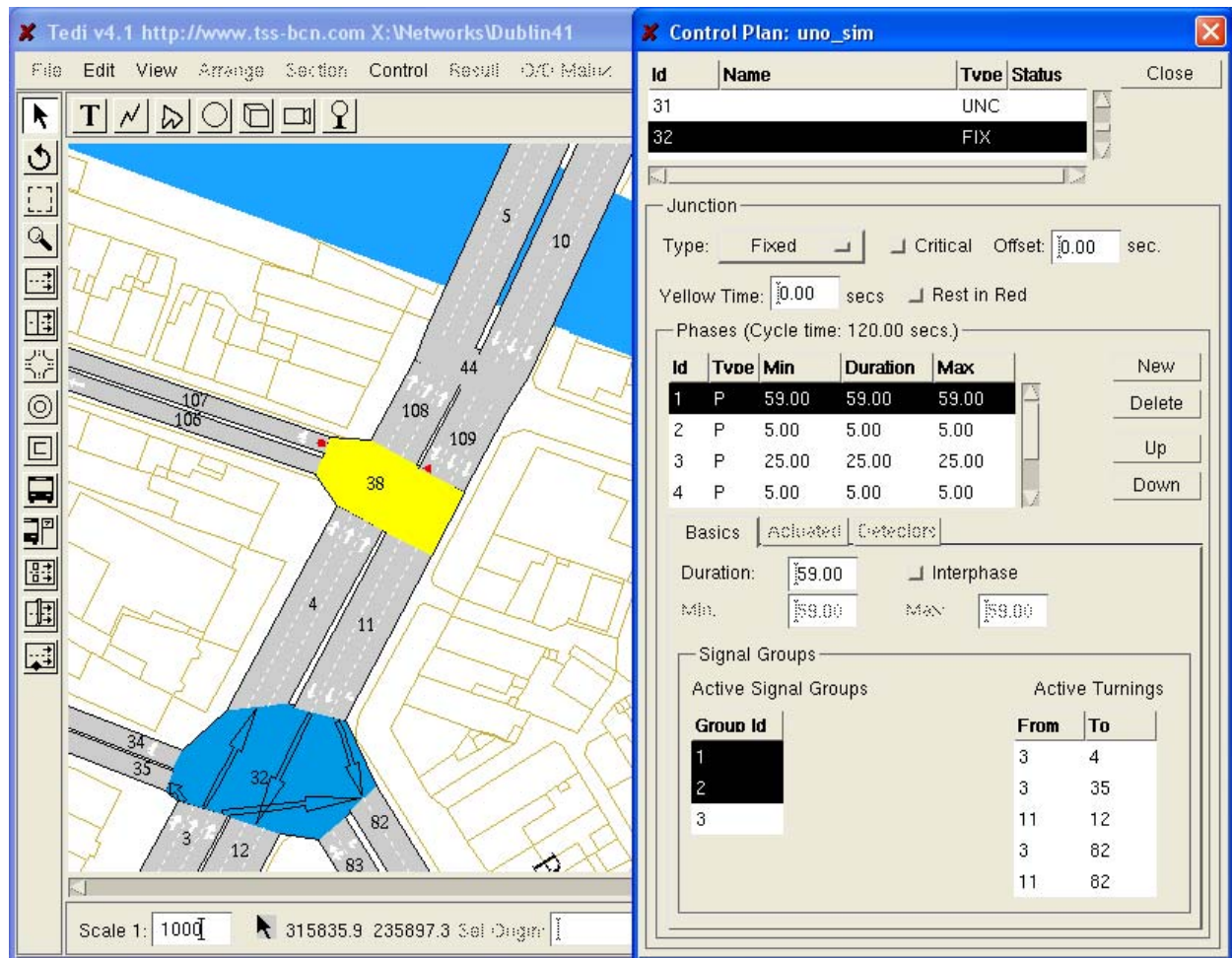


Figure 3-61: Dialog Window for traffic signals in a control plan

Offset: This parameter is used to synchronise adjacent junctions (green wave). It formally specifies at which point of the cycle the junction is situated at the beginning of the simulation. In other words, the moment at which we locate the beginning of the cycle when the control plan is initialised (e.g., an offset of -3 means that 0 seconds after the beginning of the control plan, the state of the traffic lights will be that defined for second 3 in the control plan).

Yellow Time: The yellow time given is not inserted between one phase and the next but it replaces the initial seconds of the red phase. Consequently, if for a certain turning movement its green phase is assigned 34 seconds and the rest of the cycle lasts 66 seconds, then setting a 4 second yellow time will model that traffic light as 34 seconds of green light plus 4 seconds of yellow plus 62 seconds of red light. The behaviour of vehicles under yellow light has two phases. The first phase is the first half of the yellow time (i.e. 2 seconds in case of a 4 second yellow time) and during this time the drivers consider that yellow time is basically the same as green time. The rest of the yellow time the vehicles apply a normal braking rate, not the maximum one they could reach under red light.

Rest in Red: only necessary for actuated control (section 5.1).

Phases: This is where the different phases of the junction cycle are defined. A phase is defined as a time period for which there are no changes to the state of the traffic lights at that junction. Phases are built in the following way: first the phase is created by pressing the *New* button; then it is characterised by selecting it in the top window and then selecting the signal groups (each active signal group enables its associated turnings) as being active (green) during that phase. The signal groups are selected by clicking on the signal group in the *Group Id* window. Lastly, a time is

assigned to each phase defined in the junction by selecting the phase and then assigning the desired timing in the *Duration* area.

Interphase: This is an optional type of phase to model the fixed clearance time in between phases. This type of phase is then set between phases. For instance, a junction with a 60 second cycle and two equally important movements (symmetric distribution of time), could be modelled with:

- 2 phases
- 2 interphases (each interphase in between the two phases)
- 24 second phases and 6 second interphases
- 3 second yellow time

This would produce 24 seconds of green light, 3 seconds of yellow light and 3 seconds of clearance time (a total of 33 seconds of red light as perceived by the complementary movement).

If you are not interested in using the 'min' and 'max' values, it is only necessary to set the *Bound phase duration* flag at the *Control Plan* folder of the *Edit – Preferences* option to *yes*.

Duration: time length of each phase (the Min.-Max. specifications are only meaningful in actuated frames or in combination with an external package). (See also, AIMSUN- Object menu - junctions - Phase/Control).

Signal Groups: The left hand window lists the signal groups associated with the selected junction and, if a phase has been selected, it shows the signal groups active during that phase and allows them to be selected/deselected. The right-hand window only shows the turnings enabled for that phase, that is, the turnings associated with the active signal groups.

Actuated & Detectors folders: For further details on those standards please refer to section 5.0 or to *Traffic Control Handbook* by the Federal Highway Administration.

3.6.9 Metering

This option changes ramp-metering control plans. As for junctions, control plans for metering are implemented through a set of parameters. The first parameter determined is control type:

Uncontrolled: the metering device is assumed to be disconnected.

Fixed: a non-dependent policy on traffic conditions is applied.

External: as in the case of junctions, this is a user label for those meterings you have the intention of controlling from an external system through the GETRAM Extensions.

During green light metering, the dialog window issued is merely a specialised version of the one used for junctions, in which there is only one turning, one traffic light and one phase. The parameters to be specified are the cycle duration, the duration of the green-light phase of the cycle, and the maximum and minimum acceptable values for this green-light time if it is changed by an external metering policy (see Figure 3-62). It is always necessary to set the timing with non-zero values. The procedure is to set all the timings and press the ENTER button only after setting the last parameter. ENTER must be pressed to validate the changes, otherwise the changes will be undone as soon as another meter is selected.

Control Plan: c1

Id	Name	Type
1	Metering 1	Flow
2	Metering 2	Delay
3	Metering 3	Green

Metering

Control Type: Fixed

Green-time Metering (sec)

Cycle: 60 Offset: 0.00 Yellow Time: 4

min init max

0.00 0.00 0.00

Close

Figure 3-62: Green light metering

During delay metering, the parameters are the mean time that the vehicles spend stopped, and the standard deviation from that mean (see Figure 3-63). (Press the ENTER button only after setting both values.)

Metering

Control Type: Fixed

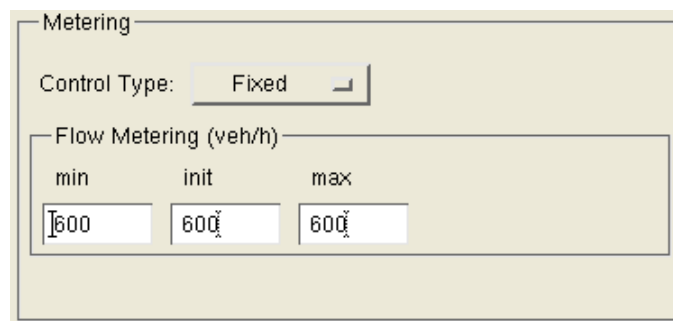
Delay (sec)

Mean: 15.00

Standard deviation: 8.00

Figure 3-63: Delay metering

For flow metering, the desired flow of vehicles is specified along with it the maximum and minimum flows allowed during adaptive metering (see Figure 3-64). The desired flow is achieved by allowing the vehicles through the metering point in packets. The size of these packets is defined along with the type of metering when configuring metering in its respective section.



Metering

Control Type: Fixed

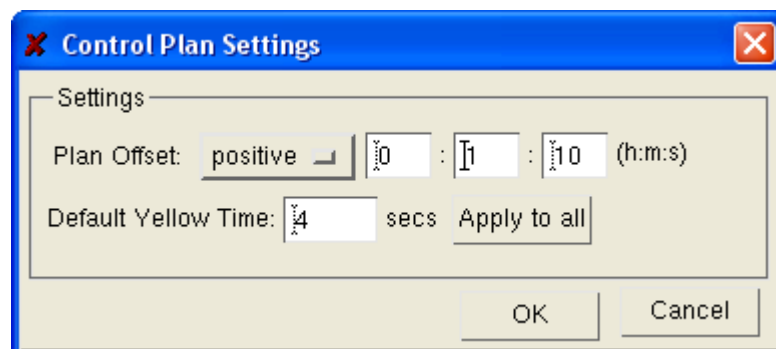
Flow Metering (veh/h)

min	init	max
600	600	600

Figure 3-64: Flow metering

3.6.10 Settings

This window allows you to edit a set of global parameters for the control plan. The first of these is the plan offset. This allows us to specify the plan's offset with respect to the moment of application of the control plan during the simulation, in the same way that the offset of a junction is carried out within a control plan (section 3.6.8). For example, the start point of the control plan in Figure 3-65 will be as if it had been started one minute and ten seconds before. The positive/negative toggle is used to set the direction of the time translation of the control plan, as with a junction's offset (section 3.6.8). The yellow time parameter is the default yellow time that will be set at each junction when editing its control plan. At the control plan's window for that junction the user can then leave that value or input a new one. Some more details concerning the yellow time are given at section 3.6.8. *Apply to all* inserts this default yellow time into all the junctions characterised previous to that moment.



Control Plan Settings

Settings

Plan Offset: positive 0 : 1 : 10 (h:m:s)

Default Yellow Time: 4 secs Apply to all

OK Cancel

Figure 3-65: Control Plan Settings window

3.7 Result menu

3.7.1 New Result Container

Creates a new result container. A result container is where the results of a simulation are stored, and where you can store the traffic data for a traffic simulation. The simulator has to know all turnings, the proportion of vehicles that take each turning in a junction, the quantity of vehicles that occupy the network at the beginning of the simulation, etc. As with the control plans, it is only possible to create new result containers once the network has been given a name, i.e., after it has been saved from *untitled* to a specific name. Also, as with the control plans, when a result container is created nothing appears to happen at first, but opening the result menu again will show that several new options have been activated.

3.7.2 Open Result Container

Opens an existing result container.

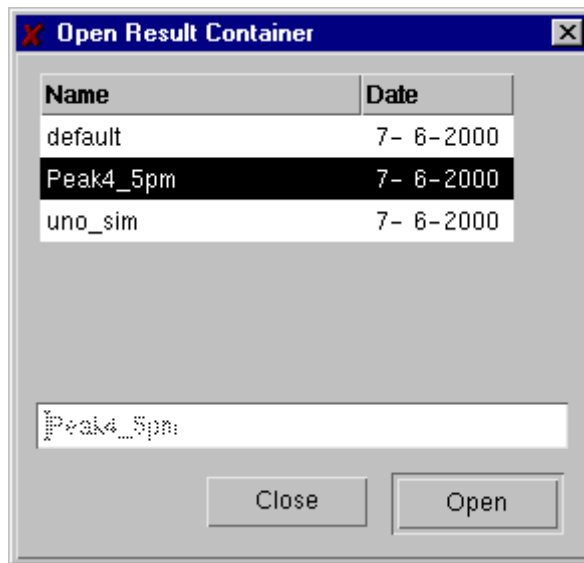


Figure 3-66: Open Result Container Dialog Window

3.7.3 Close Result Container

This closes the current result container. If any new states have been created and not saved, the editor will prompt for confirmation.

3.7.4 Save Result Container

Updates the currently open result container, saving any new states or effectively deleting any undesired states deleted via the *Delete State* option.

3.7.5 Save Result Container as...

This is the same as “Save Result container”, except that it provides you with the option to change the name under which the result container is saved. The dialog window for this option is displayed in Figure 3-67.

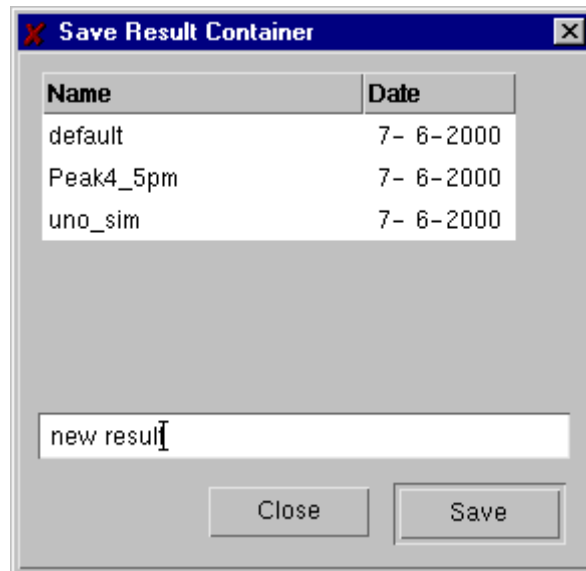


Figure 3-67: Save Result Container Dialog Window

3.7.6 Import Result Container from Ascii File

In case we have access to information on traffic flows and turning proportions in a tabulated Ascii file, this feature automatically creates a new result container with the imported data. Please refer to section 7.1.1.

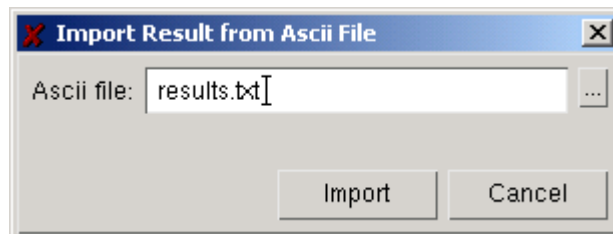


Figure 3-68: Import dialog window

3.7.7 Export Result Container from ASCII file

You can choose the traffic states of your interest and the vehicle types for which you want the information and automatically export that information into a tabular Ascii file of the same format as the one required for importing traffic results into TEDI (section 7.1.1).

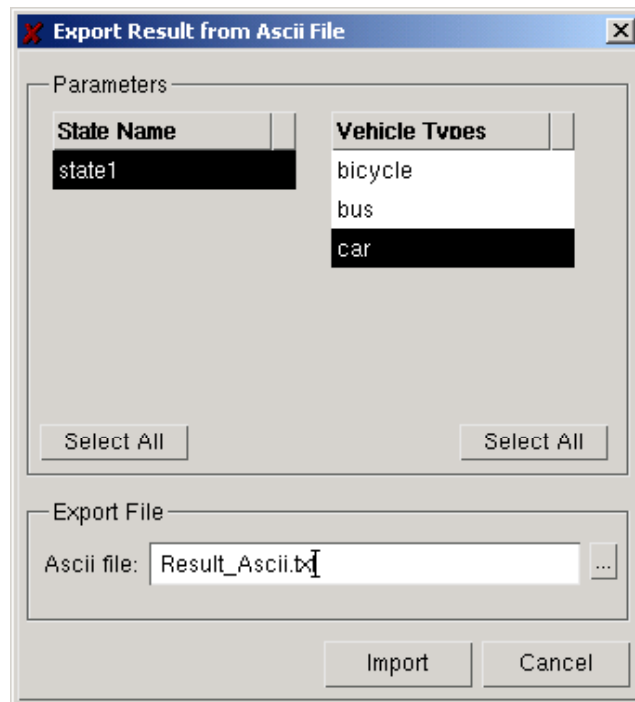


Figure 3-69: Export dialog window

3.7.8 Import Result Container from EMME/2

Creates a new result container using the EMME/2 flow volumes. For further information see EMME/2 Interface Manual.

3.7.9 Delete Result container

Deletes a result container. The dialog window for this option is displayed in Figure 3-70.

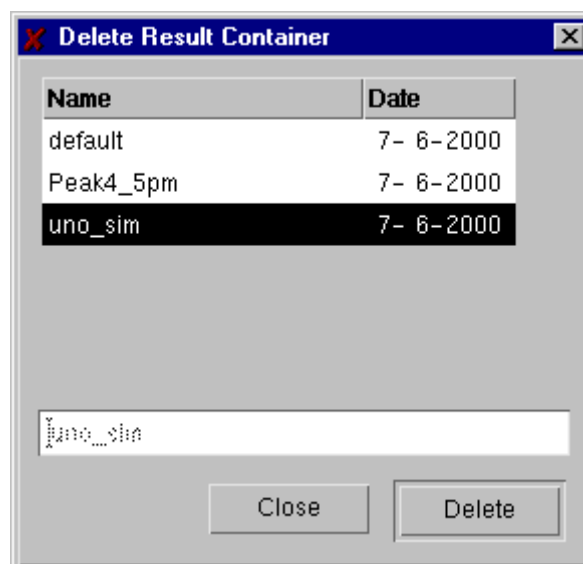


Figure 3-70: Delete Result Container Dialog Window

3.7.10 Vehicle types...

This is where you can edit the vehicle types that are going to circulate through the network when simulating with the current result container. Choosing this option opens a window that allows the configuration or creation of the type of vehicle required, if it is not found among the libraries. The

window opened when selecting the vehicle types is the one displayed in Figure 3-71, almost exactly the same window as when opening the vehicle types library (section 3.2.12). The main difference is that this window only contains the vehicle types that we took from the general library and the ones created by ourselves; and the only other difference is that now instead of having 'to model' on the last button, we have 'To Lib' in order to add a vehicle type, initially created only for the result container of a particular model, to the general library. Vehicle types can be opened for alteration besides adding them to the libraries. New vehicle types can also be created. When creating a vehicle class, a default set of values is assigned to the parameters, or the parameters of a selected type are used if the 'duplicate' option is selected.

Vehicle Types (MODEL)

Vehicle Types

- Car
- bus
- truck

New
Delete
Duplicate
To Lib

Attributes

Name: Car

Parameter | Articulated | Classes | Fuel | Pollutants

Name	Mean	Devia	Min.	Max.	Units
Length	4.00	0.50	3.50	5.00	meter
Width	2.00	0.00	2.00	2.00	meter
Max Desired Speed	80.00	0.00	80.00	80.00	Km/h
Max Acceleration	2.80	0.50	2.00	3.00	m/s2
Normal deceleration	4.00	0.50	3.50	4.50	m/s2
Max deceleration	8.00	1.00	7.00	8.50	m/s2
Speed Acceptance	1.00	0.20	0.80	1.30	
Min Distance Veh	1.50	1.00	1.00	2.00	meter
Give Way time	25.00	15.00	10.00	40.00	sec.
Guidance Acceptance	0.00	0.00	0.00	0.00	

Guided: 0.00 Equiped: 0.00 Cruising Tolerance: 0.80 m/s2

OK Cancel

Figure 3-71: Vehicle Types Dialog Window - Parameters

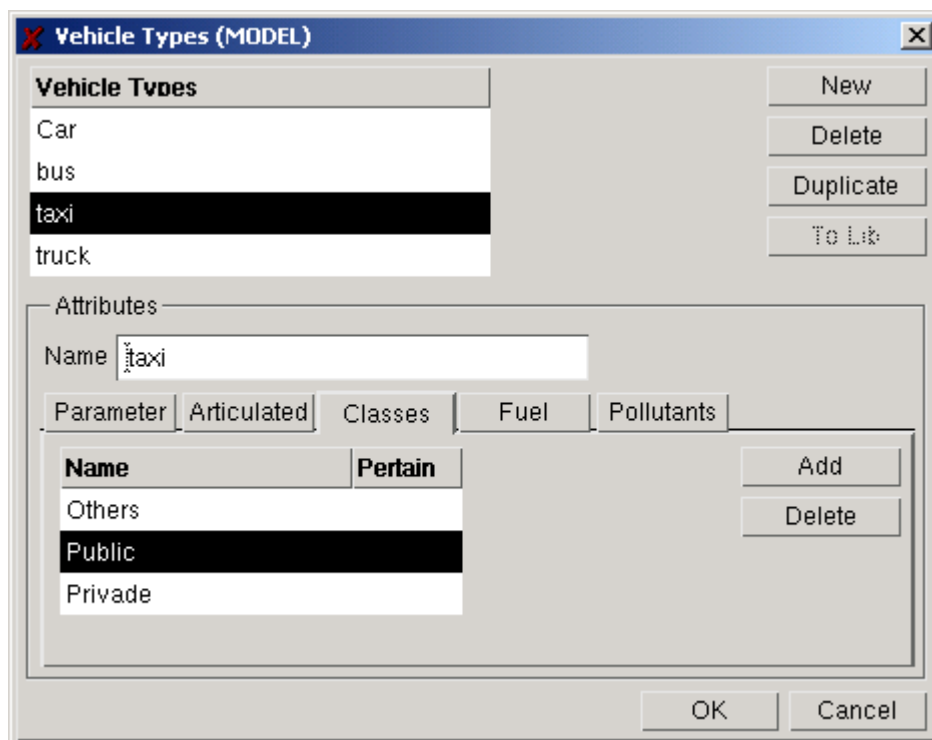


Figure 3-72: Vehicle Types Dialog Window - Classes

You can opt to associate a vehicle with one of the network vehicle classes (vehicle classes folder, Figure 3-72) as defined in the *Edit* menu, vehicle class option. This will allow the vehicle type to circulate through the lanes reserved for that particular class.

If calculations such as estimates of general fuel consumption or pollution emissions are to be made, then you must feed in the required data (section 3.2.12).

Following the general policy of the TEDI editor, Enter **must** be pressed after each input in order to validate it; otherwise any changes made will be ignored and lost.

3.7.11 New State

This creates a new network state in the current result container. A state is a set of data that defines the input flows to the network, and the turning proportions at every node for every vehicle modality, taking into account their origin sections (for more information on states see section 3.7.13). This process here is similar to that for control plans and result containers: it is only possible to create new states once the result container has been given a name, i.e., once it has been saved from *untitled* to a specific name.

You have two different options when creating a new state. You can either create an empty state, or a default. In the first case, the network will not have any vehicles at the beginning of the simulation. In the second case, the network will be loaded with vehicles from the very beginning. The first case, besides its intrinsic interest, has two frequent applications. The simulator allows for a 'warm-up' period (during this period, no time is taken from the simulation and no data files are produced) to load the network with vehicles in the right proportions and with the right distribution. Another application is to carry out a first test simulation, stopping it at the point considered to reproduce the desired initial state, and saving that state for later use as the initial state for other simulations. The default state allows the network to be tested quickly to detect any 'logical' errors, such as missing turnings that do not imply a lack of network connectivity or missing stages in the control cycles. It also allows for another form of testing: since vehicles start off uniformly distributed across the

network, an analysis of the conditions for traffic circulation over the different parts of the network can be carried out.

3.7.12 Open State

This opens an existing network state. The windows for this option are displayed in Figure 3-73.

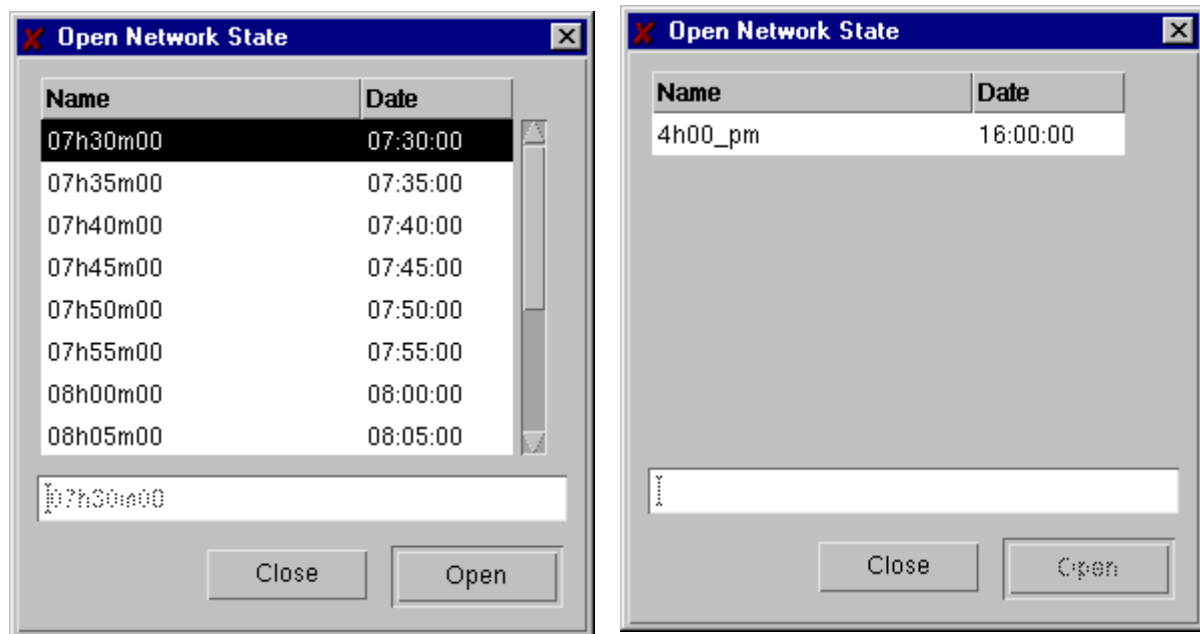


Figure 3-73: Open Network State windows

The second of these windows displays a typical view of a result container containing the first state that has been constructed for the model. The first window displays a result container containing the states generated by a simulation every five minutes, each state containing the information on the traffic conditions at that time.

3.7.13 Sections

This changes the state of the sections. The real creation of the state comes from configuring the initial conditions of each section (i.e., turning proportions' input flows). This is carried out with the *Sections* option. The window for this option is displayed in Figure 3-74. The figure shows that in section 3 (named 'Dorset (N)'), cars entering from section 2 have a 12% chance of going to section 82, 67% of going to section 4 and 21% of going to section 35. 0.2 cars on the section are going to section 82 in the first moments of the simulation, 0 cars are stopped, and the cars are trying to flow in this direction with an average speed of 50 km/h. Of course, the simulator will not position 0.2 cars on the section, but will round this figure up to the nearest integer. Another required piece of data is the average entry flow of vehicles whenever we characterise an entry in the network. In the example provided in Figure 3-74, the entry flow of cars is shown to be 328 cars/hour, but in this case this number (created as default data) is ignored, as it has no meaning for a section without an unbounded entry. This characterisation process must be carried out for each turning movement for each vehicle class coming from each entry section.

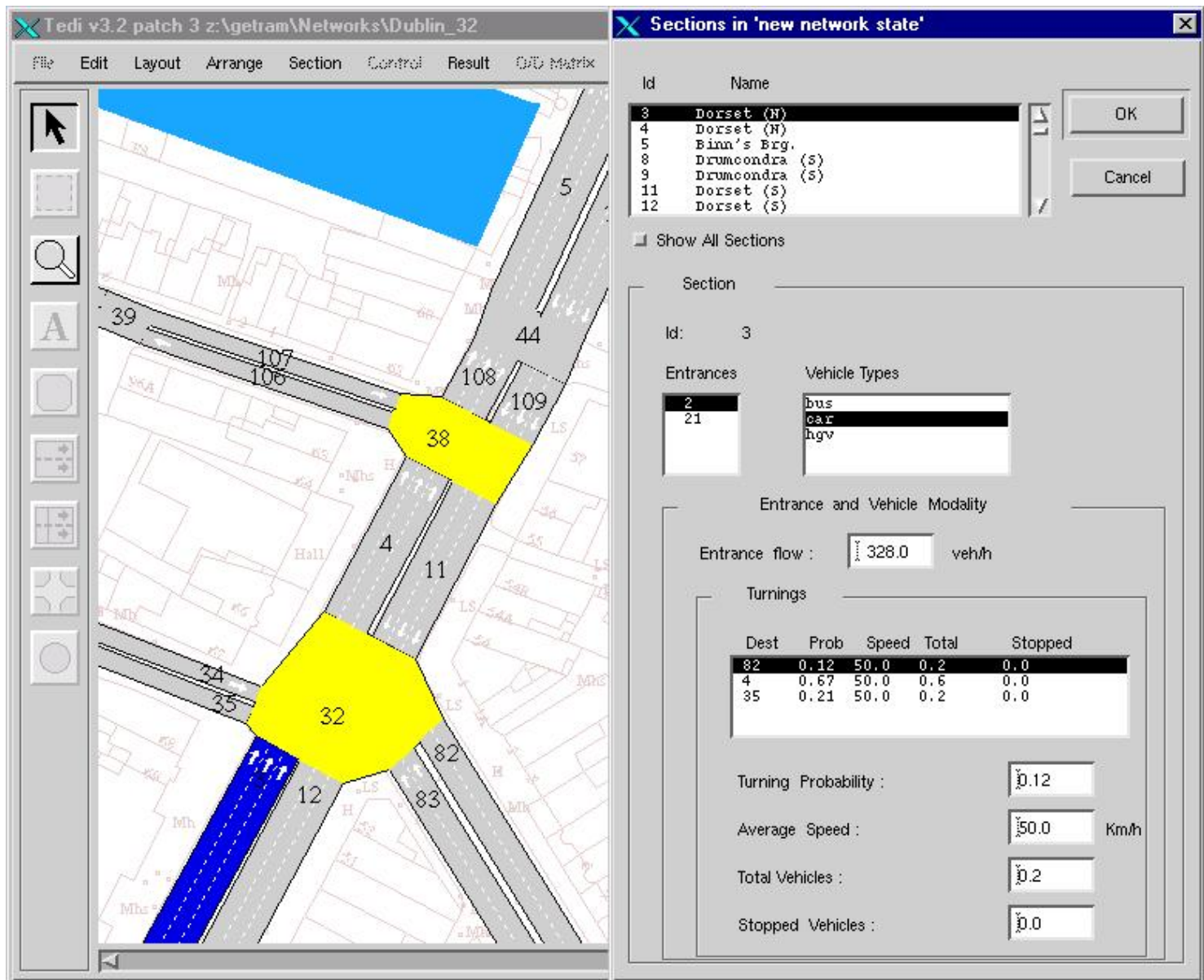


Figure 3-74: Sections Dialog Window

Information concerning a particular section is displayed by clicking on the graphical representation of the section in the drawing area or on its Id in the dialog window. In the dialog window, the names of the sections are also shown; this is done to make it easier to find sections.

3.7.14 Close State

This closes the current state. If the latest changes to the state have not been saved, a dialog window will ask the user for confirmation before closing.

3.7.15 Save State

This saves changes to the current network state. It is not necessary to close the state to save it, so it is recommended that you save the state every once in a while during a long data input session.

3.7.16 Save State as...

This is the same as *Save State*, except that it provides you with the option to change the name under which the state is saved. The dialog window for this option is displayed in Figure 3-75.

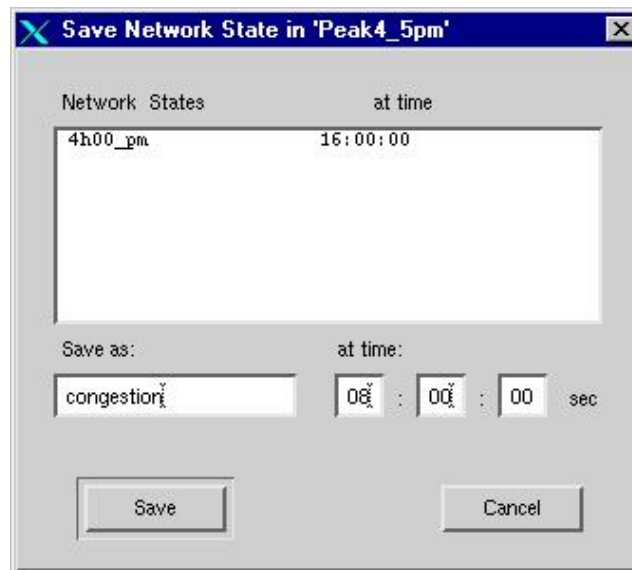


Figure 3-75: Save as Dialog Window for Network States

3.7.17 Delete State

This deletes a network state. The dialog window for this option is displayed in Figure 3-76. The user selects the unwanted state and presses the *Delete* button. An opened state cannot be deleted. Trying to do so will open the window displayed in Figure 3-77.

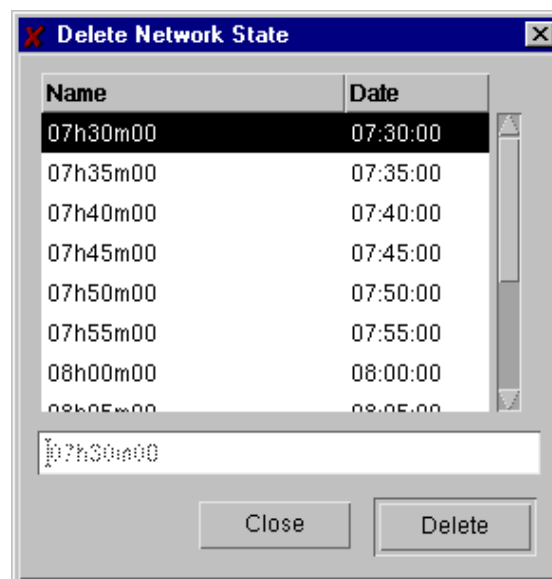


Figure 3-76: Delete Network State Dialog Window

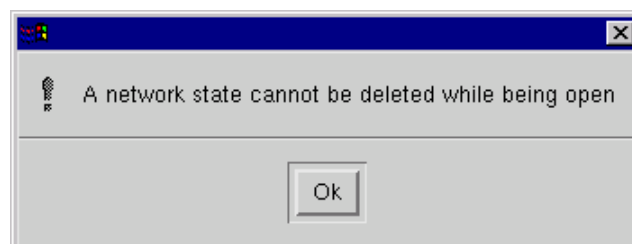


Figure 3-77: Network State Deletion warning window

3.8 O/D matrix menu - basic

3.8.1 New O/D matrix

This creates a new O/D matrix. The new matrix bears the default name of *Untitled*.

3.8.2 Open O/D matrix

This opens an existing O/D matrix. The dialog window for this option is displayed in Figure 3-78.

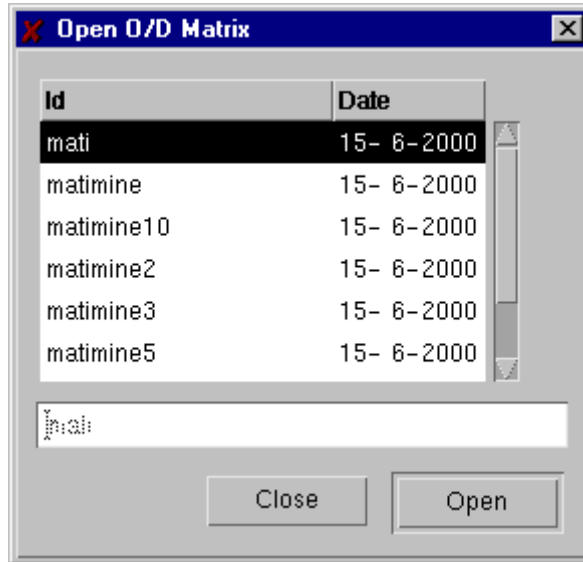


Figure 3-78: Open O/D Matrix Dialog Window

3.8.3 Save O/D matrix

This saves changes to the currently open O/D matrix. It is not necessary to close the statement dialog window (dialog window of Figure 3-82) in order to save the changes.

3.8.4 Save O/D matrix as...

This saves changes, except that it provides you with the option to change the name under which the matrix is saved. The dialog window for this option is displayed in Figure 3-79.

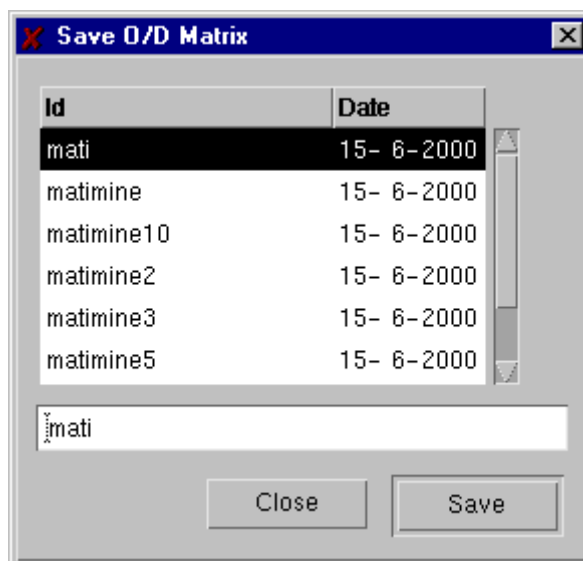


Figure 3-79: Save as O/D Matrix Dialog Window

3.8.5 Delete O/D matrix

This deletes an O/D matrix (not possible if the matrix is open). The dialog window for this option is displayed in Figure 3-80.

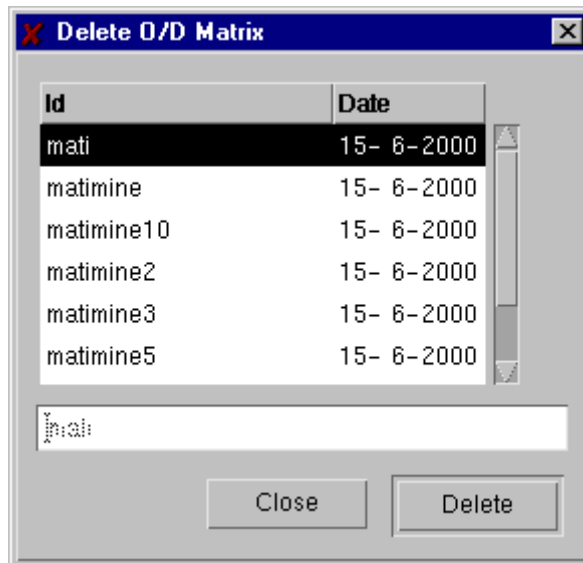


Figure 3-80: Delete O/D Matrix window

3.8.6 Close O/D matrix

This closes the current O/D matrix. A prompt for confirmation appears if the most recent changes have not been saved.

3.8.7 Vehicle types

This is used to edit or change the vehicle types in the current O/D matrix. This is exactly the same function as described for result containers (see section 3.7.10).

3.8.8 Import from Emme/2

This imports O/D matrices from the EMME/2 database. This option is activated once the matrix that will contain the EMME/2 matrix/matrices has been designed. The import procedure consists of entering the necessary information in the dialog window displayed in Figure 3-81. First enter the path and the name of the EMME/2 matrix, or search for it using the *Browse* utility. When the system locates the matrix, double-clicking on it will cause the browser to automatically close and then write the path and name in the *File* line. Select the vehicle types and time intervals represented by the imported matrix. Enter the time dimension of the original EMME/2 matrix in order for the system to convert the matrix from its original duration to that of the selected time intervals.

For further information see the EMME/2 Interface Manual.

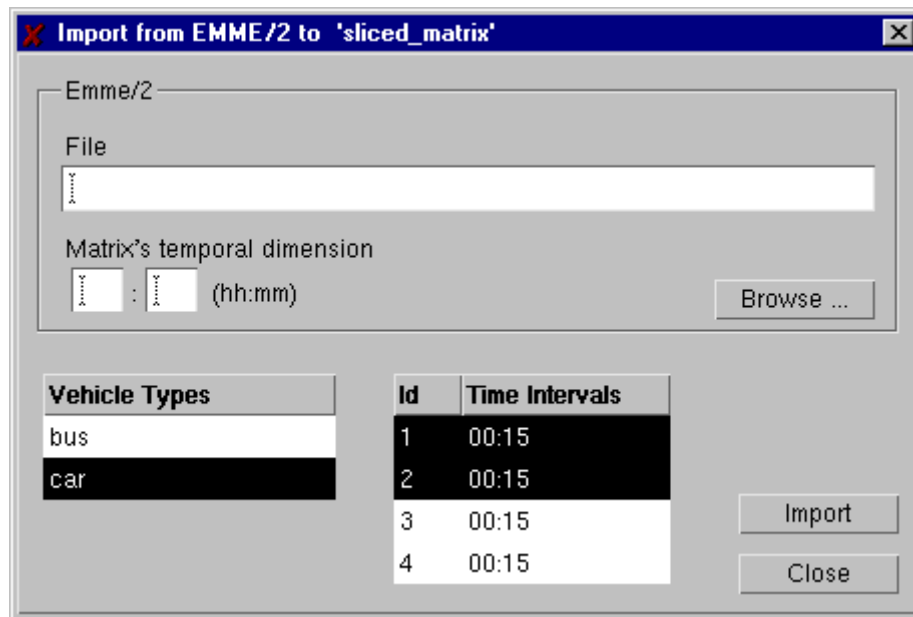


Figure 3-81: Importing an O/D matrix from EMME/2

3.8.9 Export to Emme/2

This exports O/D matrices to the EMME/2 data bank. The dialog window for this option is displayed in Figure 5-86; the process is the reverse of that for the import process. For further information see the EMME/2 Interface Manual.

These two last options only work in conjunction with the GETRAM – EMME/2 interface.

3.9 OD Matrix menu – Statements&Paths

3.9.1 Main window

This sets the different time periods for a time-sliced matrix, as well as setting the vehicle flows from each origin to each destination for each vehicle type and time interval. The dialog window for the *Statements* option is displayed in Figure 3-82. The method to build an OD matrix from this point on is explained in section 3.9.2 and 3.9.3.

Statements in 'untitled'

Time Period

From: 00 : 00 To: 00 : 00 (hh:mm) 0 intervals Set ...

Vehicle Type all Time Interval all

O/D Matrix (Origin in Rows, Destination in Columns)

	1	3	4	5	6	7	8	Total
1								
2								
3								
4								
5								
6								
7								
8								
Total								0.0

Multiply By 1 All Row Col Cell

Export... Import... Close

Figure 3-82: Statement Dialog Window

3.9.2 Time period

The first thing to do is to set the time period that is going to characterise the matrix. This is done by clicking on the *Set* button to open the window displayed in Figure 3-83. First, you define the time period in the *From, To* boxes.

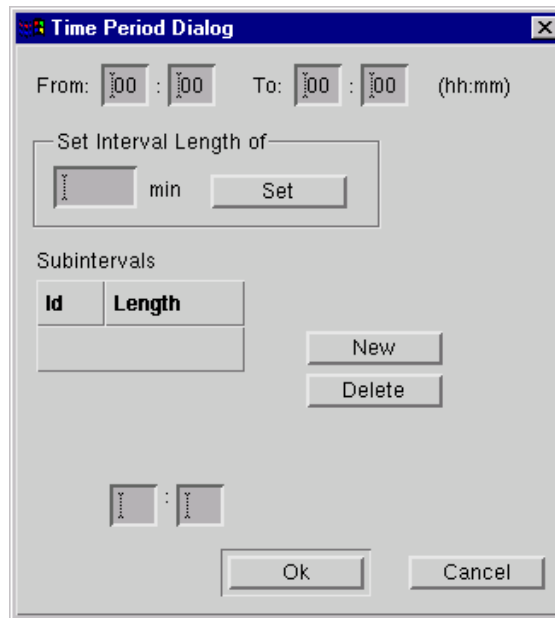


Figure 3-83: Setting the time characteristics of the matrix

Now there are two options if a time-sliced-matrix is required. When the different intervals have the same time length, you enter it in the *Set Interval Length of* box (Figure 3-84) and a list of the different intervals will automatically appear in the *Subintervals* list when the *Set* button is pressed (Figure 3-85).

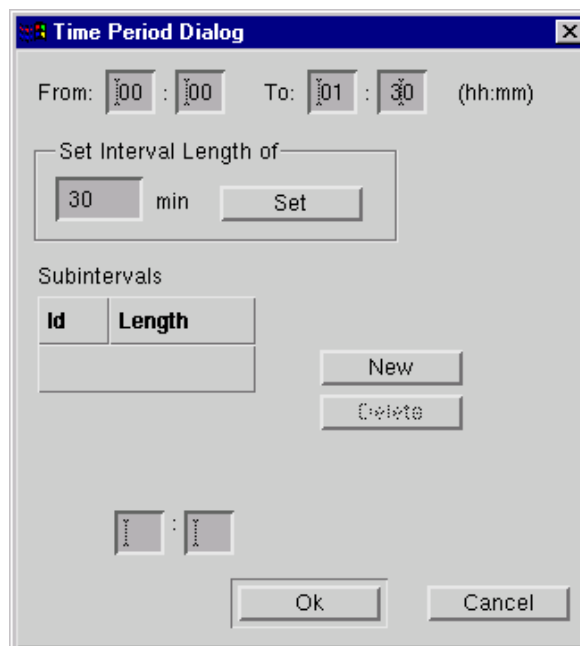


Figure 3-84: Setting the interval time length -1

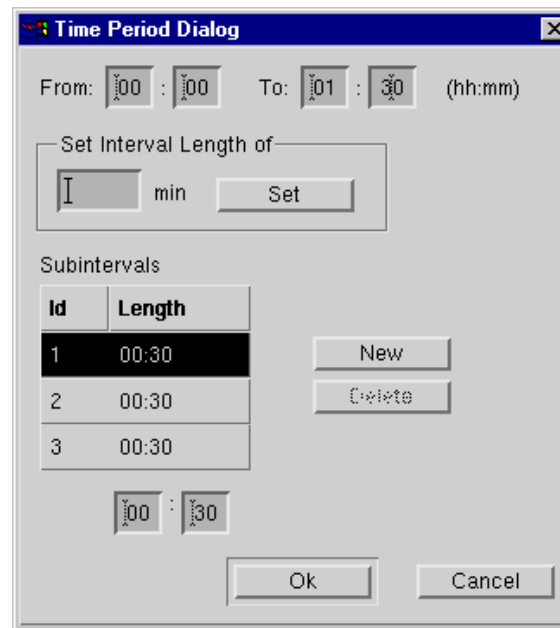


Figure 3-85: Setting the interval time length -2

When the intervals have different time lengths, the list is generated by creating the intervals one by one. This involves clicking on the *New* button, setting the time length in the boxes at the bottom, and pressing the *Enter* key to validate the information (see Figure 3-86).

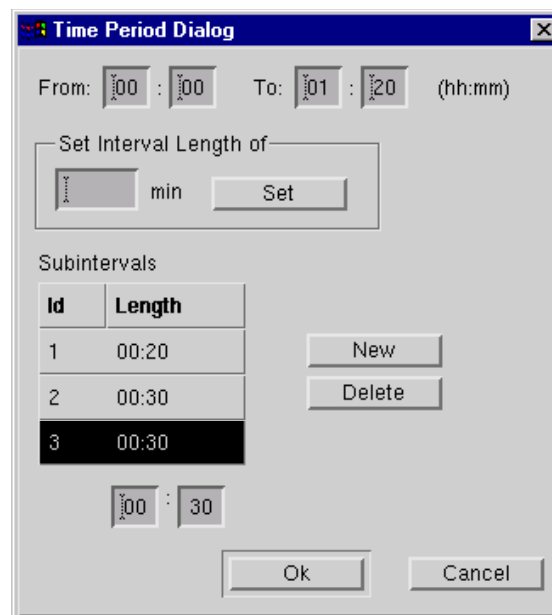


Figure 3-86: Creating different time intervals

3.9.3 Traffic demand data for each OD pair

Upon completing the time characterisation of the matrix you return to the parent window (the *Statements* window). This is where you feed your traffic data into the matrix. There are several ways of doing this depending on the *Vehicle Modality* and *Time Interval* options chosen. If these tags are left as they appear ('all' and 'all'), the traffic information will involve all time intervals and all vehicle types. This means that if we enter a '54', for instance, at position (4,5), a total of 54 vehicles, considering all vehicle types, will circulate from centroid 4 to centroid 5 across the total length of the simulation (Figure 3-87).

Consequently, in a matrix with two vehicle classes and three time intervals, 27 cars and 27 trucks will circulate from centroid 4 to centroid 5 (see Figure 3-88), or 18 vehicles will flow within each one of the three intervals (see

Figure 3-89). If we select the types 'car' and '1' from the tags mentioned above, we will see a '9' in position (4,5) (see Figure 3-90). The different combinations of these two tags give us all the partial views of the O/D matrix that may be of interest: the data for the different time intervals and/or the data for the different vehicle types.

Vehicle Type

Time Interval

Origin (in Rows) / Destination (in Columns) Centroids

	2	3	4	5	6	Total
1						
2						
4				54		54
7						
Total				54		54

Figure 3-87: Data

Vehicle Type

Time Interval

Origin (in Rows) / Destination (in Columns) Centroids

	2	3	4	5	6	Total
1						
2						
4				27		27
7						
Total				27		27

Figure 3-88: More data

Vehicle Type

all

Time Interval

1

Origin (in Rows) / Destination (in Columns) Centroids

	2	3	4	5	6	Total
1						
2						
4				18		18
7						
Total				18		18

Figure 3-89: Yet more data

Vehicle Type

Time Interval

Origin (in Rows) / Destination (in Columns) Centroids

	2	3	4	5	6	Total
1						
2						
4				9		9
7						
Total				9		9

Figure 3-90: Further data

If you have different traffic densities at different time intervals and you do not have the same proportion of vehicles in each class, data has to be entered in another way. The data is fed in independently for each time interval and for each vehicle class. This is done by setting the class tag to first vehicle class, and the interval tag to the first time interval. You then enter data and change the class tag to second vehicle class, etc. Repeat for the remaining time intervals.

If the demand for each vehicle type during each time interval is defined as a percentage of the total aggregated demand, then we can:

- set the *vehicle type* and *time interval* tags to 'all'
- input the global demand
- calculate the corresponding percentages to correct the uniformly distributed demand
- multiply each submatrix (vehicle type and time interval) by its corresponding correction factor

For instance, if we have a matrix with a single OD pair, with:

- a global demand of 1.000 vehicles
- 4 time slices (4 time intervals)
- 2 vehicle types, for instance car and truck

This means that we have 8 submatrices and that $1.000/8 = 125$ vehicles will be assigned to each one of them. But we can have that:

- an 80% of the demand corresponds to car
- the remaining 20% of the demand corresponds to truck
- a 10% of the demand corresponds to the first time slice
- a 25% of the demand corresponds to the second time slice
- a 45% of the demand corresponds to the third time slice
- a 20% of the demand corresponds to the fourth time slice

Consequently, although we have a 12.5% of the demand at each submatrix, for car we should have:

- an 8% of the demand at the first time slice
- a 20% of the demand at the second time slice
- a 36% of the demand at the third time slice
- a 16% of the demand at the fourth time slice

and for truck we should have:

- a 2% of the demand at the first time slice
- a 5% of the demand at the second time slice

- a 9% of the demand at the third time slice
- a 4% of the demand at the fourth time slice

So the final step is to multiply each submatrix by the factor that corrects the demand from a 12.5% to its corresponding proper percentage.

For car:

- we should have an 8% of the demand at the first time slice, but we have a 12.5%, so we have to multiply this submatrix by $8/12.5=0.64$
- corrective factor for the second time slice: 1.6
- corrective factor for the third time slice: 2.88
- corrective factor for the fourth time slice: 1.28

For truck:

- we should have a 2% of the demand at the first time slice, but we have a 12.5%, so we have to multiply this submatrix by $2/12.5=0.16$
- corrective factor for the second time slice: 0.4
- corrective factor for the third time slice: 0.72
- corrective factor for the fourth time slice: 0.32

In case of a matrix of 400 OD pairs, it means that instead of having to input 3.200 data elements we would only have to input 400 and carry out 8 mathematical operations.

NB: 1) Always press Enter or change matrix position after feeding data into a position before changing the interval tag or the class tag. Be careful when entering specific data for each vehicle type and time interval. Always check the tags before entering data: entering data with one of the tags set to 'all' will update all the information for that origin-destination pair; that is, it will be equally distributed among the different vehicle types and/or time intervals, unless data had already been entered for that OD pair. In that case the demand will be distributed according to the proportions set by the original data.

3.9.4 Import

This utility allows to import OD matrices in text tabular format.

The two buttons in the 'Statements' window open the following window.

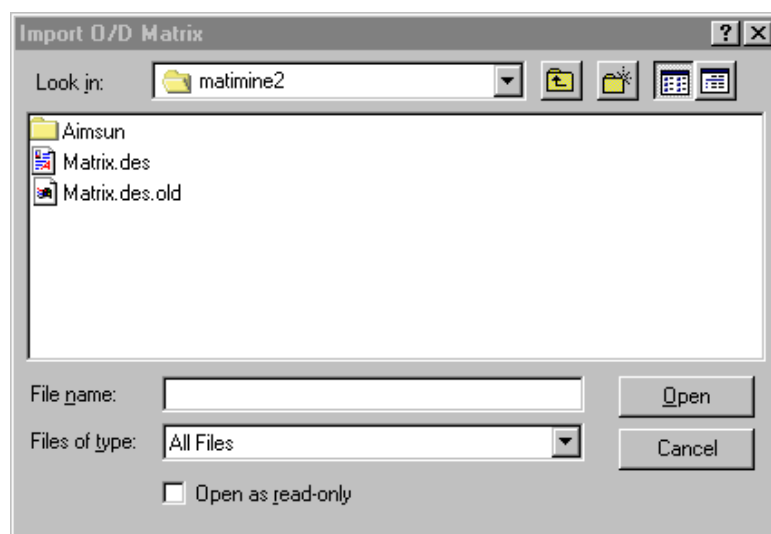


Figure 3-91: Import O/D Matrix Dialog Window

The procedure is to select the vehicle modality and time period that you want to import in the 'statements' window. Then open the previous window and select the file containing the required OD information.

3.9.5 Export

The procedure for exporting OD matrices to tabular format files is the same as that used for importing them. Select vehicle modality and time period and then select (or create) the required destination file in the dialog window.

3.9.6 Path Definition

To edit a route, proceed as follows:

- Click on the *New* button
- Click on the sections that make up the route, one after the other and in the order in which vehicles will flow
- Input the OD pair for which you may decide later on to assign a part of its demand to this route
- Press the OK button

The OD pair has to be set for the system to check the consistency of the route, that is: that the route is connected (consecutive connected sections) and that it is a valid path from the chosen origin to the destination. An example of this is displayed in Figure 5-84.

If you are editing an old network and only changing part of it, you do not need to re-edit a whole route again. When you open the path, you will see the whole route with the exception of the old sections that were deleted. From that point on you cannot close the dialog window, as it will issue a warning message informing you that the route is unconnected. This is because deleting sections within the route has 'broken' the route into two groups of connected sections.

If the network is complete then you can re-edit the route from the last connected section and add the sections from that point until the first section of the next connected group for the original route.

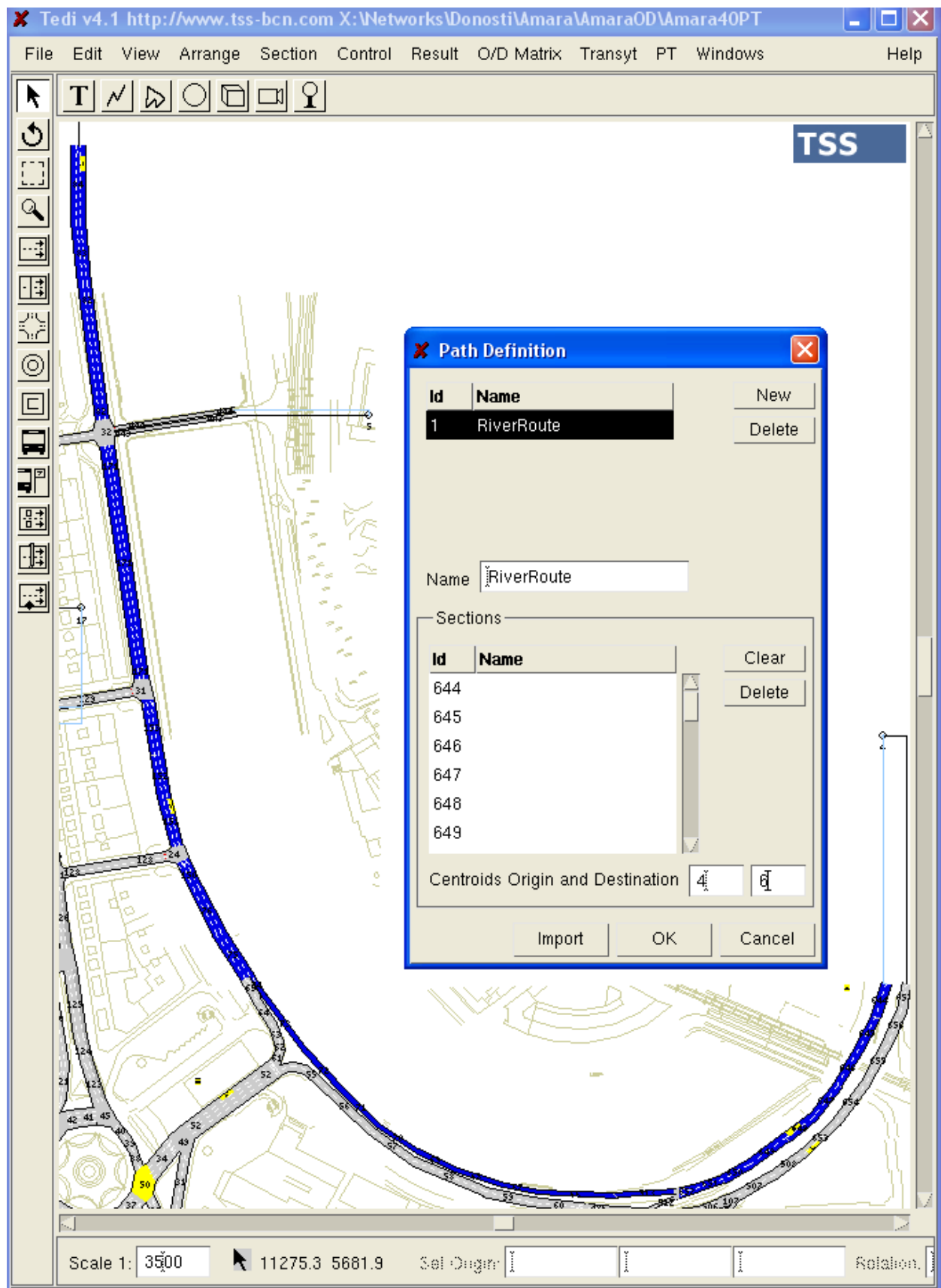


Figure 3-92: Fixed Path

3.9.7 Import

Imports into TEDI the path routes file created in EMME/2. For further information see the GETRAM - EMME/2 Interface Manual.

3.9.8 Path Assignment

Once an O/D matrix has been opened, you can assign the percentage of use of the paths previously defined as well as the percentage for the ICSP (Initial Current Short Path) between a pair of O/D centroids. To do so you must indicate the origin and destination centroids. Then, all the paths defined for that O/D pair will be listed. If it does not have any path defined then its corresponding ICSP will appear. Afterwards you will be able to modify the percentage of use of each path in each vehicle type of the matrix. If there is no flow defined between a pair of centroids its paths percentages won't be editable. If you want to list all the centroids an '*' can be written. The total sum of the percentages for each O/D pair and vehicle type cannot be higher than 1.00.

If a percentage is written in the textbox located at the bottom part of the dialog and the 'All Paths' button is pressed, this percentage will be assigned to the first path of each O/D pair

All the changes made by the user will be saved when saving the matrix.

Figure 3-93 shows how an 80% of cars and a 70% of trucks travelling from centroid 1 to centroid 22 will be distributed among 4 paths in the specified percentages. The remaining 20% of cars and 30% of trucks will choose among one of those paths or other alternative paths according to the active route choice function.

Stochastic Path Assignment of 'matrix_18-20'

Origin Centroid: 1 Destination Centroid: 22

Oria.	Dest.	Path Name	car	truck
1	15	ICSP		
1	15	TOTAL	0.00	0.00
1	16	ICSP		
1	16	TOTAL	0.00	0.00
1	22	inside	0.1	0.05
1	22	mix	0.3	0.25
1	22	outside	0.3	0.4
1	22	ICSP	0.10	0.0
1	22	TOTAL	0.80	0.70

Percentage input field: All Paths Ok Cancel

Figure 3-93: Stochastic Path Assignment

3.10 Transyt

This is the menu associated with the special TRANSYT/10 interface. The TRANSYT/10 functions are explained in the corresponding manual for this complementary software.

3.11 PT

3.11.1 PT Lines

This option opens a browser window with a list of the public transport lines designed up to that moment. Clicking on a line's number will center the Tedi image on the line's icon. Double clicking on a line's number will not only center the image but also open the dialog window of the line.

3.11.2 PT Plans

This option allows you to design a public transport plan, understood as a combination of timetables for public transport lines. The dialog window for this option is displayed in the following figure.

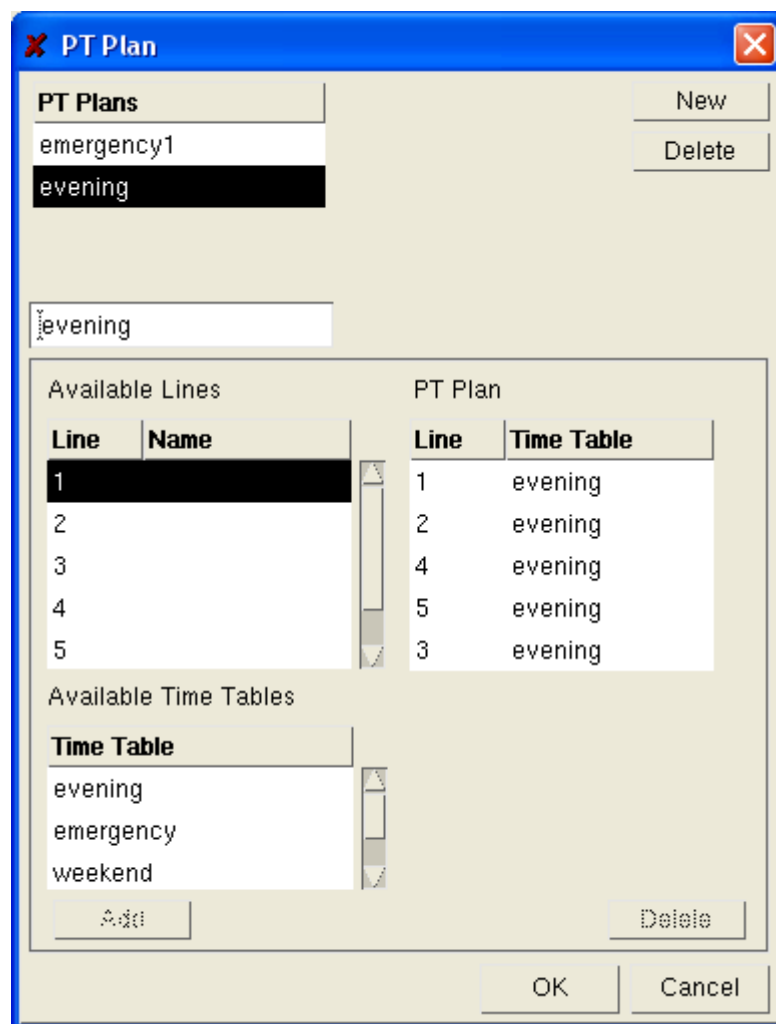


Figure 3-94: PT Plans Dialog Window

The process is as follows:

- select or create a PT plan with the *New* button
- select the first line number
- select one of the timetables available for that line containing the *Schedule Slices* for the simulation interval in which this plan is to be used.
- click on the *Add* button
- repeat the process with the remaining line numbers

You have now created a public transport plan. A package of information grouped under the name of this PT plan assigns a timetable to each line in the model. If you plan to use this plan in simulations from 16:00 to 18:00, then the selected timetables must include information from 16:00 to 18:00.

The general method for creating a public network is as follows:

- create the bus stops (section 6.8)
- assign the public transport vehicle types (section 5.10.3)
- create the public lines and their time tables (section 6.9)
- create the PT plan

See the section on Public Lines to get a full picture of the nature of timetables and schedule slices.

3.11.3 Vehicle Types

This option opens the same window as the option of the same name in the *Result* and *O/D Matrix* menus. This is also used exactly as in those options (section 3.7.10).

The different types of buses that are going to circulate in the public transport lines must be taken from the *Vehicle Types Library* to this library, in the same way that they are taken from the *Vehicle types* windows in the *Result* and *O/D Matrix* menus (section 3.7.10).

3.11.4 Import Transit Line from EMME/2

This option opens a browser to the EMME/2 transit line to be imported. For further information see the EMME/2 Interface Manual.

3.12 Windows

3.12.1 Show Map

This option opens the small window with the Navigator header as displayed at the top left of Figure 3-95.

The smaller window displays an image of the whole network, and, within a red rectangle, the part of the network currently being viewed in the TEDI main window. Clicking on any part of the network in the small window will also move the main window to that location. You can also zoom or navigate (right button held pressed) in the same way as in the main window (see section 4.4). The main purpose of this feature is to provide a fast and easy method for moving around large networks. The image at the bottom right is just an augmented image overprint of the part of the image in the dashed rectangle in the *Navigator* window, allowing the user to view the area in greater detail.

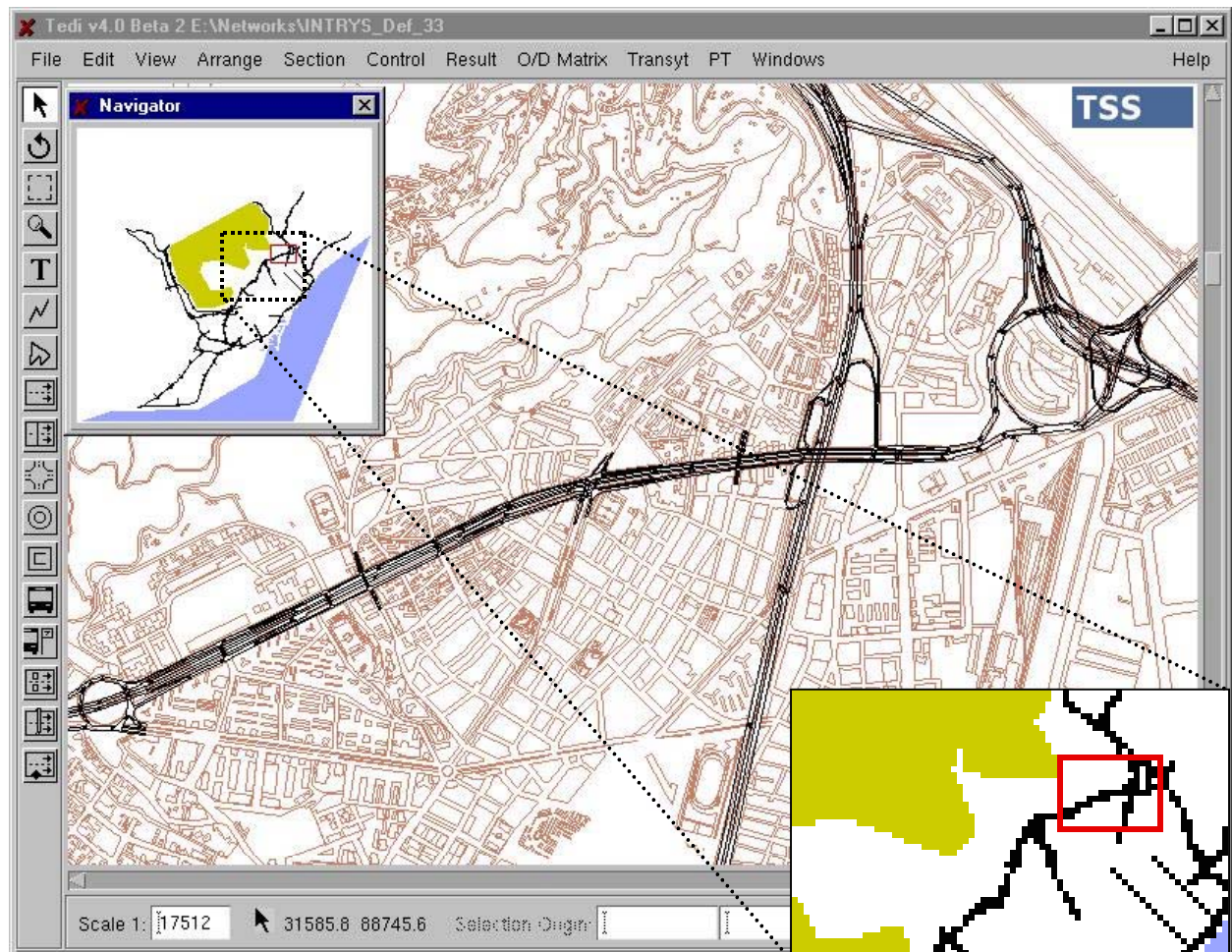


Figure 3-95: Small Map

3.12.2 Show Object Info

This option opens the window displayed at the top right of Figure 3-96. This window displays information associated with the selected objects, which in some cases is not included in the message window or in the dialog window for the open object. Figure 3-96 displays a section. In this case, we can see the section's id in the window header, along with the name, level and physical dimensions of the section. The id and name can also be accessed by opening the section. The level can also be found in the message window. The physical dimensions, however, can only be read in the information window. Information on centroids, detectors, VMS's and other objects is viewed in a similar manner.

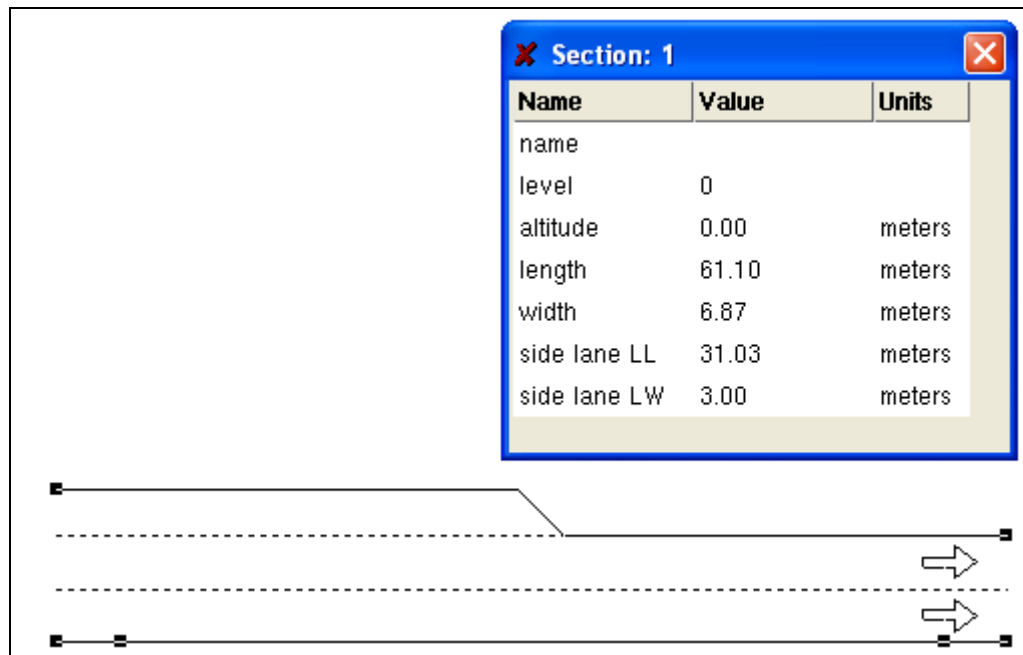


Figure 3-96: Info Window

3.12.3 Show Log Window

This option opens the window displayed in Figure 3-97. In this dialog window, you can view a list of the different messages, concerning actions like loading data sets, that the editor may have generated.

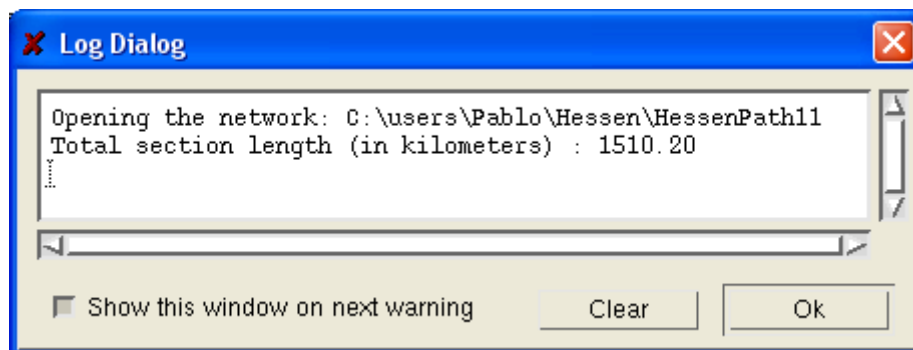


Figure 3-97: Log Window

3.12.4 Hide Distance & Hide Tools

Hide Distance eliminates the message window and Hide Tools eliminates the tool bar (see Figure 3-98, allowing a larger network area to be edited).

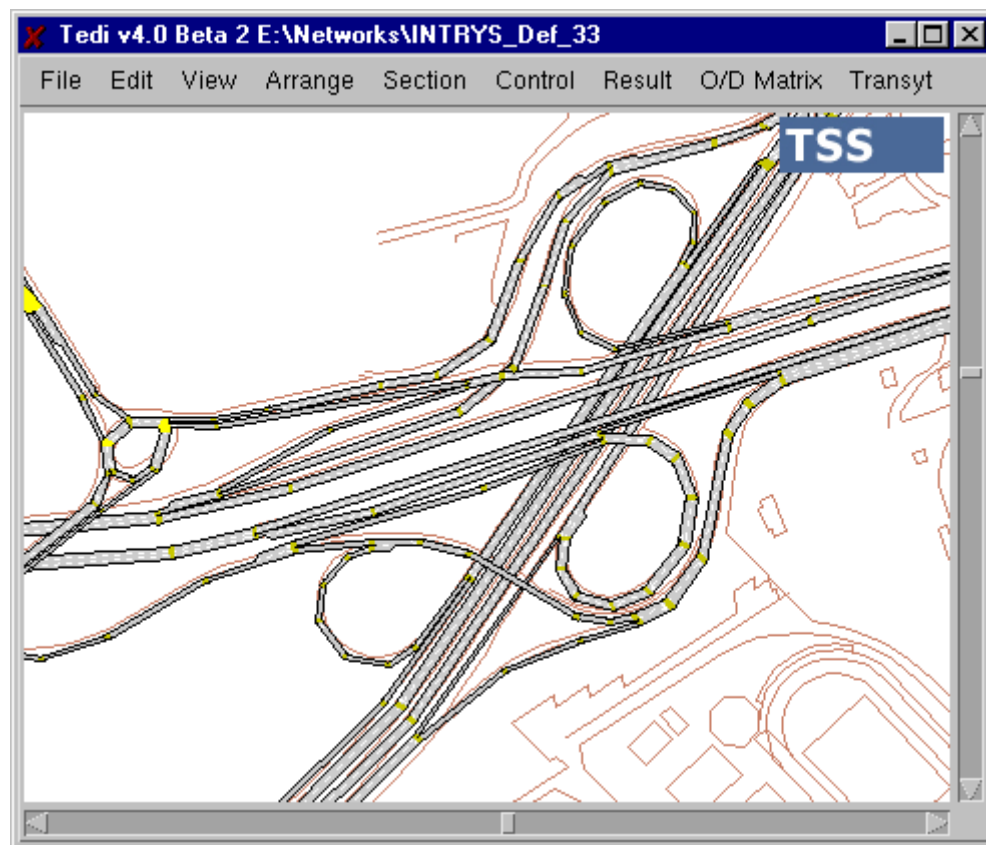


Figure 3-98: Hidden message window and tool bar

These options change to *Show Distance* and *Show Tools* respectively after being selected.

3.13 Help

3.13.1 Licenses

This option opens the window displayed in Figure 3-99, which lists the information contained in the license.dat file, defining the active software components and the number of possible users.

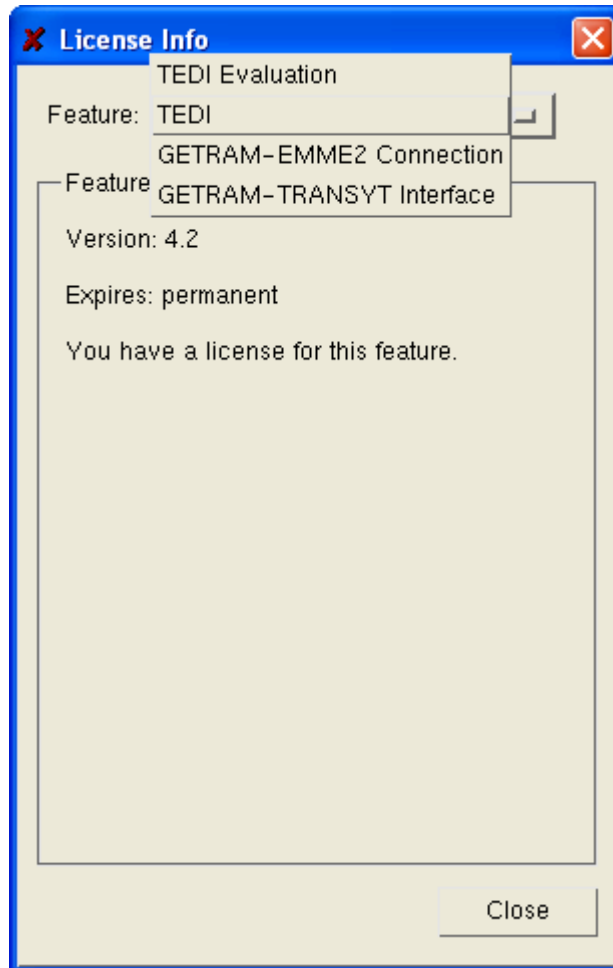


Figure 3-99: License Info window

The *Update Info* button allows you to refresh this information if it has changed since the beginning of the editing session.

3.13.2 About

This option just reopens the TEDI presentation window.

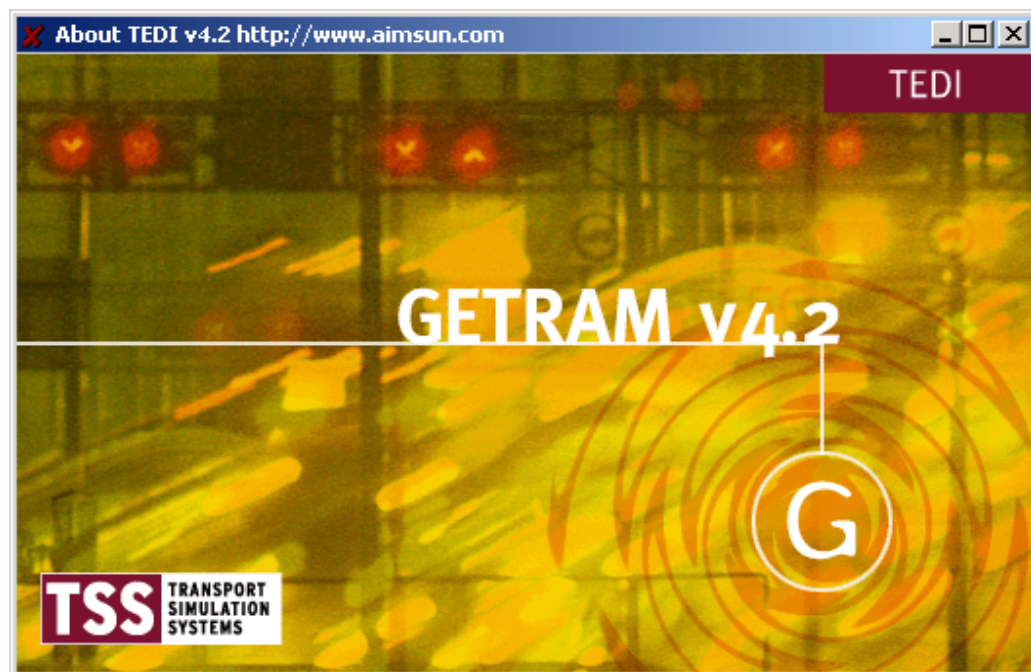


Figure 3-100: Presentation window

4 THE TOOL BAR AND THE DRAWING AND MESSAGE AREA

4.1 The Message Area

The message area is located at the bottom of the window and provides data such as the position of the mouse pointer in the real world, the scale, and the dimensions of the selected object (see Figure 4-1). For example:

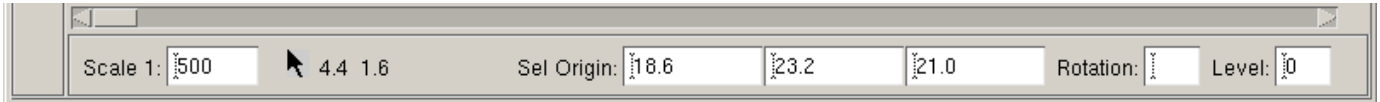


Figure 4-1: Message area—displayed when a section is selected

- Always displayed:

Scale: 1: ... map scale, the relationship between the dimensions of the drawing area window and those of the real world. The desired scale can be directly input in this window.



Real-world co-ordinates of the mouse pointer.

- When selecting an object or a set of objects:

Selected Origin: location of a selected object or set of objects (x,y,z coordinates of the origin of the selection).

Rotation: window in which to input the number of degrees when rotating the selected object or set of objects.

Level: window in which to input the desired level (graphical display order) for a selected object or set of objects.

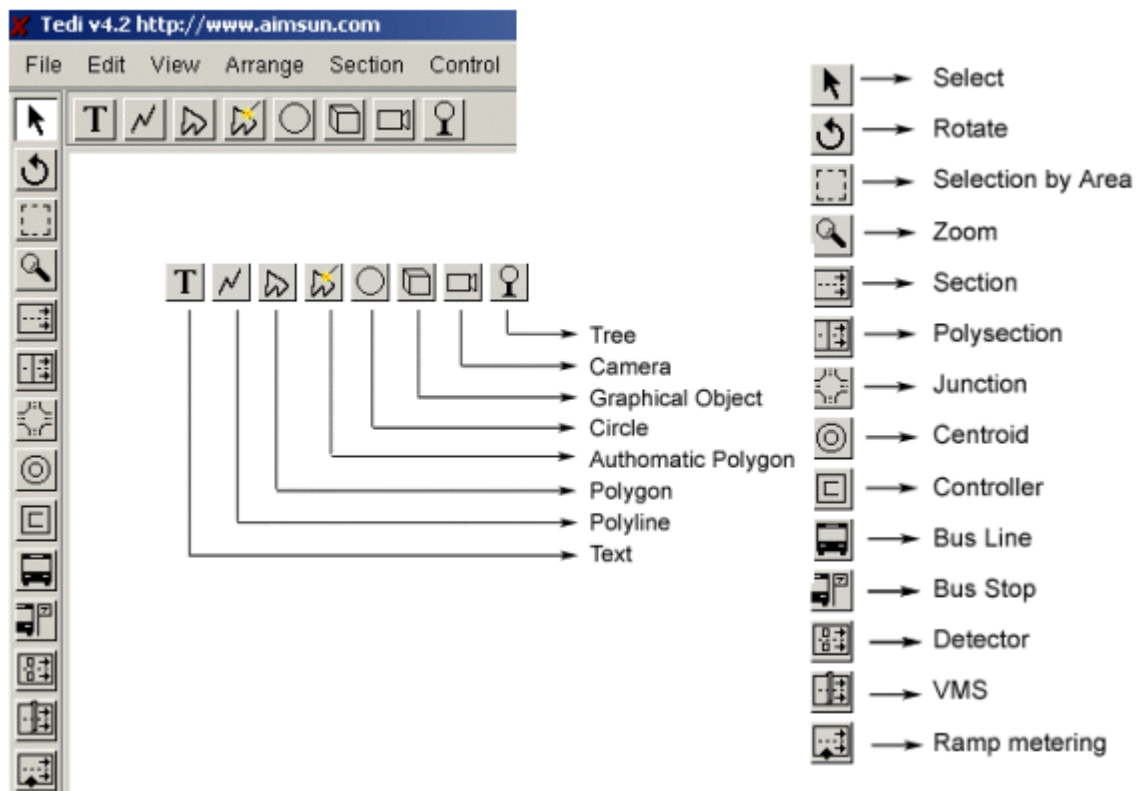
4.2 The Tool Bar










This is a column of buttons located on the left-hand side of the window and below the menu bar, and which enables the creation of objects such as text, blocks, sections, and junctions. When the user clicks on a button, it is highlighted showing that an operation has been activated: the tool remains selected until the user has finished (or aborted) that operation. The user can cancel an operation either by choosing the *Cancel* button in the object's associated dialog window, or by escaping to a select mode (the two first buttons). The drawing tools can be classified into three types according to function:

Select buttons. The first and third button of the left-hand column. The first one (arrow) provides the normal select mode, while the third provides the select-by-area mode.

Zoom button. The fourth button. Provides a 'zoom in' function. If pressed while holding the 'Shift' key, it allows you to 'zoom out'.

Object creation & manipulation buttons. The second and the remaining ten buttons, and the horizontally ordered buttons. These allow the direct creation and positioning of the objects, which are graphically represented in the drawing area, namely text, blocks, sections, polysections, junctions, and centroids.



-  **Select:** This is the default mouse selection tool; with it the user can click, shift-click, control-click or double-click on different objects in the drawing area in order to select or open them.
-  **Rotate:** When pressed, a cross appears at the centre of the object or group of objects. Then it is possible to position the cross at the desired centre of rotation and drag the object around it.
-  **Selection by area:** Same as *select*, but enables all the objects contained in a rectangle to be selected. The rectangle is specified by clicking on the position of one of the desired vertices and, without releasing the button, dragging to the position of the opposite vertex and then releasing the button.
-  **Zoom:** When pressed, clicking on the drawing area zooms in, taking that point as the centre of the area to be displayed. Shift-clicking—that is, clicking while keeping the Shift key pressed—zooms out. The middle mouse button also provides the zoom function. To zoom into a particular area you must select it by clicking on the position of one of the desired vertices and, without releasing the button, dragging to the position of the opposite vertex and then releasing the button.
-  **Section:** When pressed, allows the creation of new sections in the drawing area by clicking on the origin and ending of the section. A section is a set of contiguous lanes.
-  **Polysection:** When pressed, allows the creation of new polysections in the drawing area by clicking on the vertex of the desired polysection, and double clicking at the final point.
-  **Junction:** When pressed, allows the creation of new junctions. A junction—or intersection in the USA—is an area where sections with different directions intersect.
-  **Centroid:** When pressed, allows the creation of a new centroid, by clicking on its desired location. A centroid is a source-sink of vehicles.
-  **Controller:** When pressed, allows the creation of a new controller. A controller represents an element to group traffic control elements: signal groups, detectors, ramp metering and VMS.



Public Transport Line: When pressed, allows the creation of new bus routes.



Public Transport Stop: When pressed, allows the creation and characterisation of bus stops.



Detector: When pressed, allows the creation of a detector by clicking on its desired location.



VMS: When pressed, allows the creation of a VMS by clicking on its desired location.



Ramp Metering: When pressed, allows the creation of a ramp-metering device by clicking on its desired location.



Text: When pressed, allows the insertion of text.



Polyline: When pressed, allows the insertion of a continuous line by clicking on the vertex of the desired line, and double clicking at the final spot.



Polygon: When pressed, allows the insertion of blocks by clicking on the vertex of the desired polygon, double clicking on the last vertex.



Automatic Polygon generation: When pressed, allows the selection of a background polygon in order to automatically generate an AIMSUN polygon using the background data.



Circle: When pressed, allows the insertion of circles (cylinders when viewed as 3D objects).



Graphical object: When pressed, allows the insertion of one of the graphical objects present in the dialog window of the graphical objects' library that opens.



Camera: When pressed, allows the insertion of a camera from which to view the network when working in 3D mode.



Tree: When pressed, allows the insertion of a 3D 'tree' object. It is then possible to associate to it one of the different tree types present at the graphical objects' library or any other of the graphical images.

When the user clicks on a button, it is highlighted, showing that an operation has been activated. The tool remains selected until the user has completed (or aborted) that operation. The functions for each tool and its most frequent uses are described below.

4.3 Selecting objects

As explained in section 4.2, the first two buttons provide selection functions, the mouse (see Figure 4-2) for single-object selection and the rectangle for selection by area (see

Figure 4-3).



Figure 4-2: Single-object selection tool



Figure 4-3: Area selection tool

4.3.1 Selection of an object

The user can select objects by clicking on them. Objects are then displayed as selected. For a section, this means that the section shows its ‘sizing handlers’ or ‘update points’ and becomes transparent (for instance Figure 3-96). When selecting an object by clicking on it, any previously selected objects are deselected. Clicking on a section in a polysection will select that individual section; control-clicking on it will select the whole polysection.

4.3.2 Selection of more than one object

To avoid deselecting the previously selected objects, the user must Shift-click on the next object (i.e., click on that object while pressing the Shift key). A Control-click on a polysection will select all sections composing the polysection, and a Control-click on a junction will select the junction and all sections connected to it. Selecting a whole polysection without deselecting any previously selected objects requires a Shift-control-click

Another method for selecting multiple objects is ‘by area’. This is done by first pressing the ‘select by area’ button on the Tool Bar, then clicking in the drawing area and moving the mouse while keeping the mouse button pressed down. A selection area appears as a dashed rectangle. All the objects that are completely inside the dashed rectangle are selected when the mouse button is released. The selected objects can be of different types. Previously selected objects are deselected, unless the user presses the Shift key throughout the operation.

Selected objects can be moved around by dragging them or, when high precision is required, using the cursor keys.

4.3.3 Deselecting objects

As stated earlier, unless the user holds the key while selecting objects, all previously selected objects are deselected. The same result is achieved by clicking anywhere on an empty space. When more than one object has been selected, and only one of them is to be deselected, you can deselect by Shift-clicking on it.

Double-clicking on an object opens it, that is, its associated dialog window is displayed (for instance, Figure 4-7). Any previously selected object is deselected. The following sections of this document describe what happens when you open each of these objects.

The ‘Delete’ key, if activated, deletes all selected objects.

4.4 Moving around the real world area

The horizontal and vertical scroll bars can be used to view other parts of the real world area. The user can also zoom in and out with the zoom tool (see Figure 4-4) on the Tool Bar—zooming out is done by pressing the ‘Shift’ key at the same time. You can also use the middle button to zoom in and zoom out. If your mouse only has two buttons, the same effect can be obtained by pressing both buttons simultaneously. And if the middle button of your mouse is a scrolling wheel, then you can zoom in/out just by turning the wheel on one direction or the other. One way of using the zoom to move is to zoom out until the desired final location appears in the view, and then zoom in on it. An initial tactic after loading the background could be to display the whole real-world area (*Whole drawing* option of the *Arrange* menu) and then zoom in on it. Another way to move around the real world is by ‘panning’. This is done by moving the mouse pointer and, at the same time, pressing the right mouse button. The result is, sometimes, more convenient than clicking on the horizontal and/or vertical scroll bars. The last way of moving around is by changing the scale factor. This is

done by typing the new scale value directly into the text box located in the message area. This has the same effect as zooming in/out, but alters the scale by a factor that you determine.



Figure 4-4: Zoom tool

4.5 Text, Polylines, Polygons, Circles, Graphics

The tools illustrated in Figure 4-5 are the editor's cosmetic tools. The text tool, for instance, allows the street names to be included in the graphical view of the model. The text, when selected, appears enclosed in a dashed box with four update points at the vertexes. The update points can be used to rotate it in the same way as if it were a section (see section 4.6.1) once the *rotate object* option has been selected (from the *Arrange* menu).



Figure 4-5: Text, Polyline, Polygon, Automatic Polygon, Circle, Graphical Object, Graphical image

The polylines or the polygons are defined by clicking on the required position for the vertices of the lines that compose the final line or the vertices of the final polygon, and then double-clicking on the last vertex to close the polygon or on the last position that the line reaches. Polygons can be opened in order to assign them with a colour and a name. Figure 4-6 shows an example of this. To change the colour of a polygon, the coloured rectangle must be pressed. This action opens the dialog window displayed in Figure 4-7, which is a general window for editing the colours of various object types. In the dialog window in Figure 4-6 we can also see how it is possible to associate different textures to each face of the polygon.

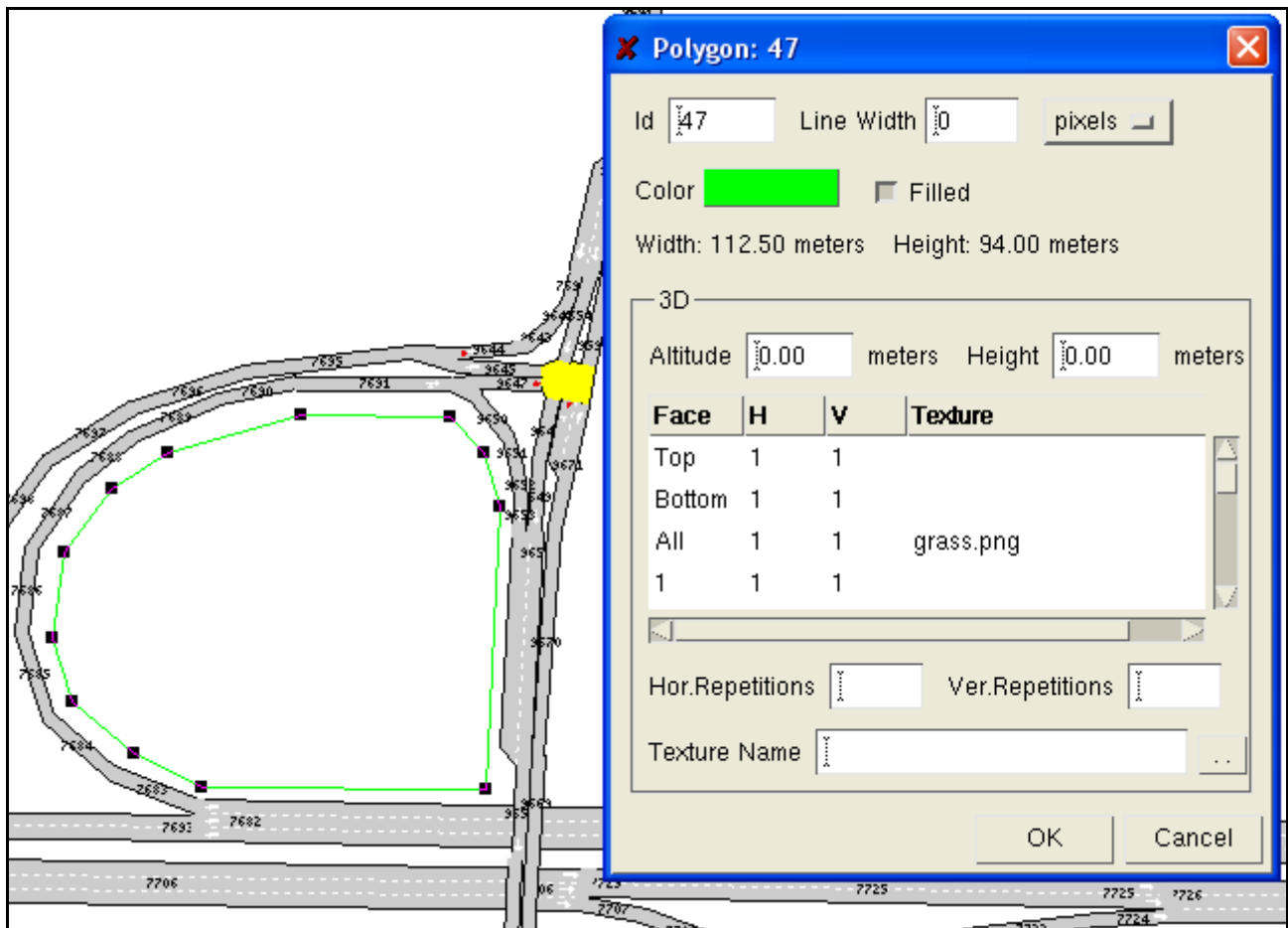


Figure 4-6: Open Polygon

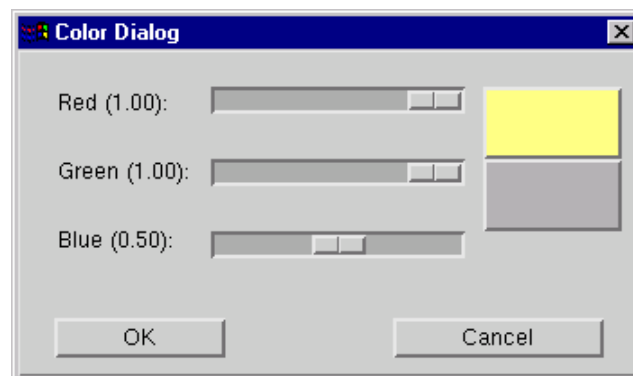


Figure 4-7: Edit colour

The markers indicate the proportions for each of the primary colours. The top rectangle displays the result of the current combination, while the bottom rectangle displays the original colour.

Clicking on a point in the network after having selected the graphical object icon opens a dialog window (figure 6.7) in which to select the object that we want to copy into the model.

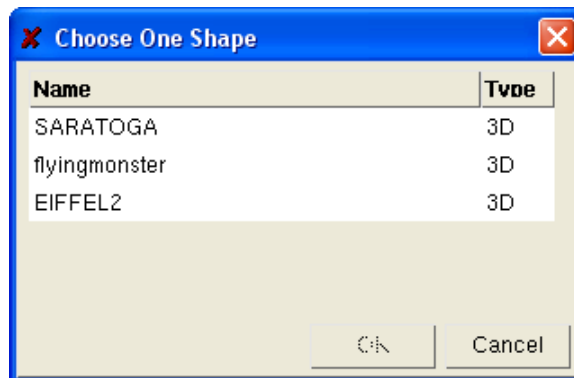


Figure 4-8: Adding a shape

Once the object has been selected it is copied into the model, and from that point on it can be selected, dragged around, rotated, deleted or opened (figure 6-8).

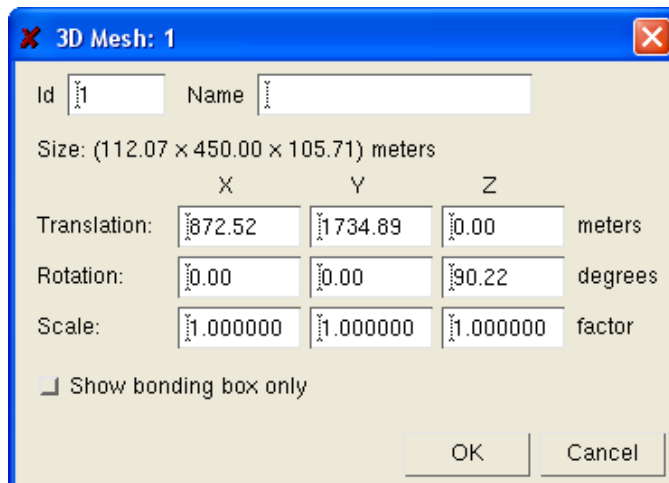


Figure 4-9: Shape Properties

A more complete explanation on the 3D qualities of these objects and their edition can be found in the 3D Viewer Users Manual.

4.6 Creating and Manipulating Sections & Polysections

Sections and polysections are road segments with common characteristics between junctions or mergings. Sections are straight segments created using the tool displayed by Figure 4-10. Polysections are irregular segments formed by a set of sections and are created using the tool displayed in Figure 4-11.



Figure 4-10: Section tool



Figure 4-11: Polysection tool

To draw a section, press the Section button, click on the point of origin, release the button after the

click, move the cursor to the section end and then point and click again. This creates something similar to the section displayed in Figure 4-12.

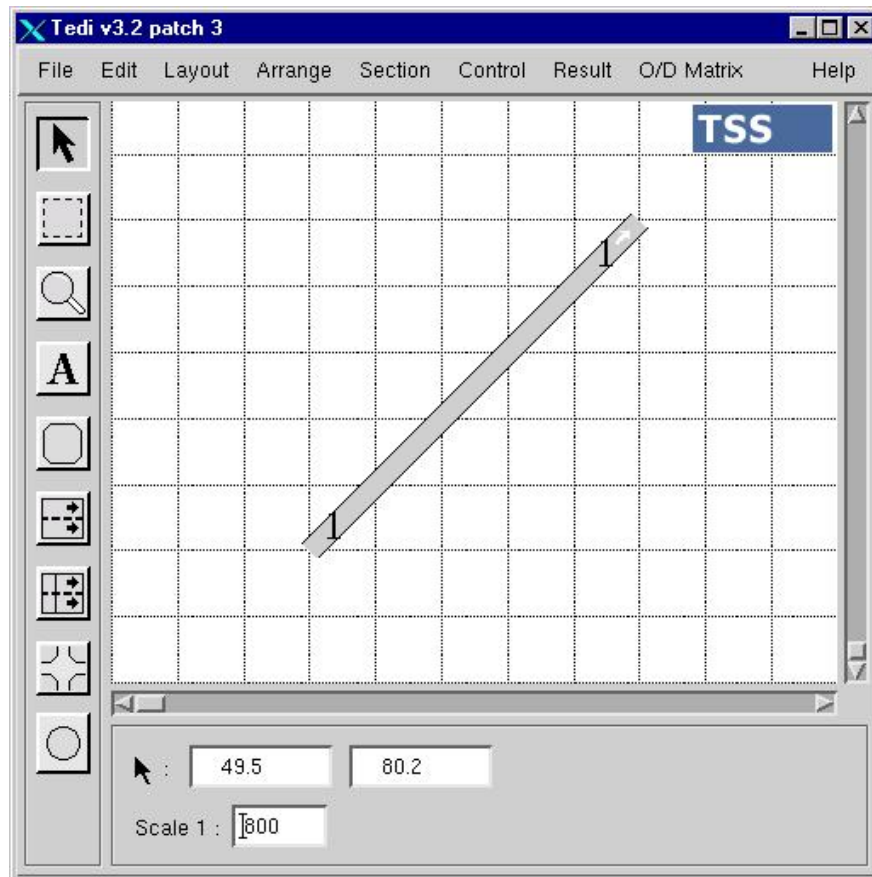


Figure 4-12: Initially drawn section

4.6.1 Manipulating sections and polysections

To change the original section into, for instance, the one of Figure 4-13, carry out the following:

- move the selected section by clicking on it without releasing the button and then drag it to the correct location (be sure to click well inside the section in which the mouse retains its usual arrow shape)
- give the section the desired width by moving the cursor to one side of the section, to the point where the arrow cursor changes to a circle with a dot at its centre, and then drag only that side of the section, widening it as much as needed (as the section widens, the number of lanes increases each time there is enough space for a new one)
- repeat this process with the leading edge of the section in order to adjust its length
- orientation of the entry and exit is carried out by clicking on the dots on the vertices of the rectangle and moving the cursor to the desired position of the chosen vertex while holding down the mouse button
- the side lane is added by clicking on the dots situated behind those on the vertices and pulling outwards, keeping the mouse button pressed
- the length and width of this extra side lane is adjusted in the same way as a section, by dragging its side with the mouse.

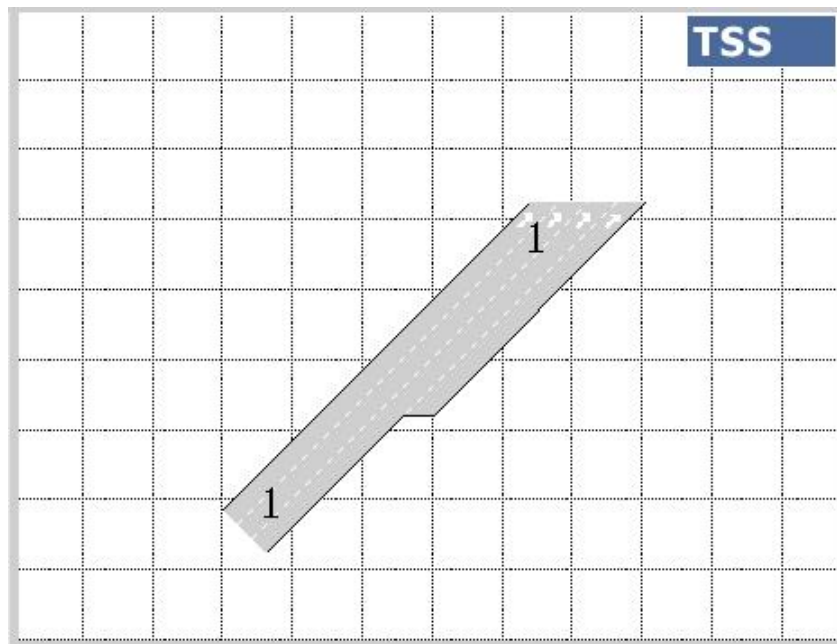


Figure 4-13: Transformed section

Polysections are generated as follows. Firstly, activate the tool by clicking on it. Then define the origin with a single click of the mouse as for a section. Then move the cursor to the position of the exit for the first section of the polysection and click to simultaneously define the exit for this section and the origin for the subsequent section. Of course, these exits and entrances are linked upon creation. You can manipulate a polysection at two levels: either the polysection as a whole, or the individual sections composing the polysection. This concept is illustrated in Figure 4-14.

- section 2 has been selected by control-clicking on it
- the number of lanes or the lane width can be changed as if it were a single section using the *Section-Number of lanes* option (not with the mouse)
- please note that the entry and exit sections of a polysection are the only side lanes allowed; this arises from the concept of a polysection as an irregular section
- the other update points are visible between the sections of the polysections in polysection 3 (they are active to allow for minor cosmetic adjustments)
- the entry and exit sections of polysection 1 have also been selected to show the difference with the same sections of polysection 2; this difference consists of the vertex update points, which are present in polysection 1 but not in polysection 2: this is because polysection 2 is connected to other sections

A single section of polysection 3 has been selected to illustrate earlier explanations: the selected section can be deleted, moved or rotated to a new position, up to a certain point, to adjust its position within the polysection. Entry and exit sections offer all this editing capabilities and can also be given side lanes. Sections 4 and 5 have been selected to facilitate comparison between sections and polysections.

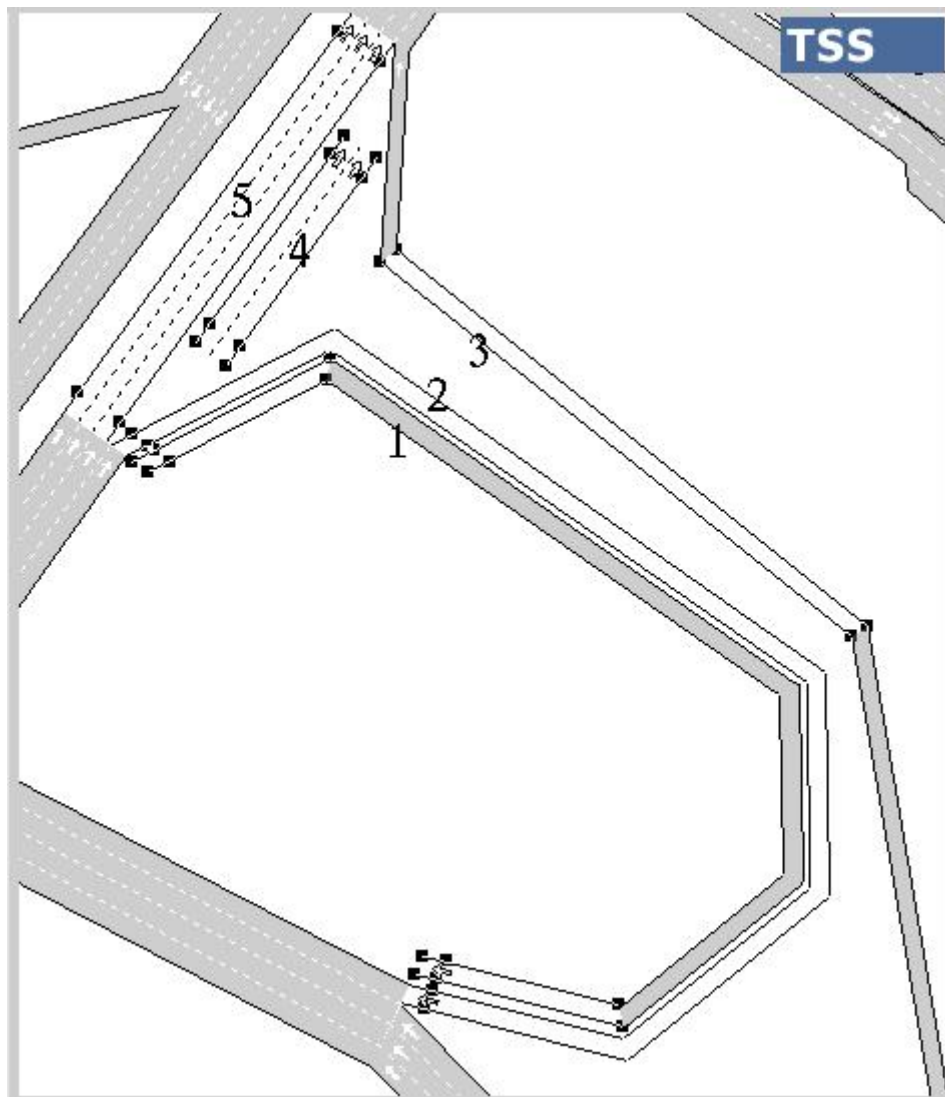


Figure 4-14: Polysections

4.6.2 The build-up process

Usually, all you need to do is to simply add a section to the exit of a previous section due to an increase or decrease in the number of lanes, an exit or entry ramp or merely an extension of the original lanes in a different direction. To do this, select the section from which the new section has to follow on, and press the *section* button on the Tool Bar. If you were to do this to the section in Figure 4-13, the result would look like the example displayed in Figure 4-15.

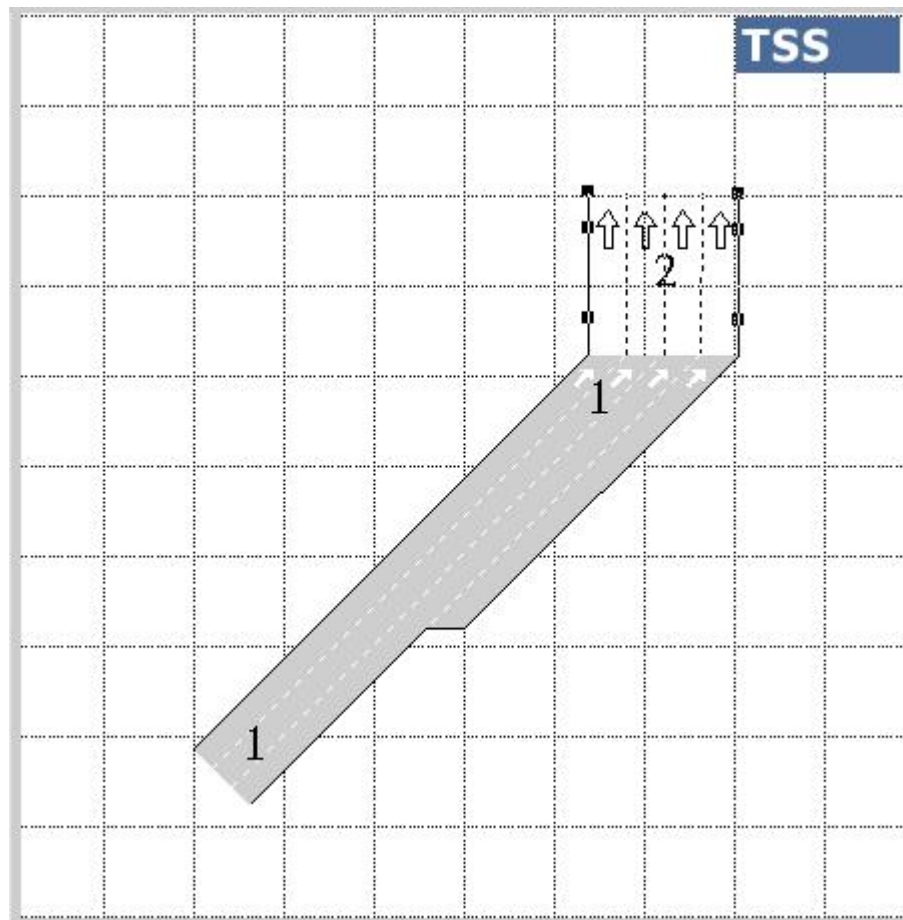


Figure 4-15: New section

If you had wanted the new section to emerge from the entry rather than the exit, you would have followed the same process but held down the shift key while pressing the *section* button, that is, you would have Shift-clicked the *section* icon. The new section can be manipulated according to your requirements. Supposing you want to achieve the final result displayed in Figure 4-16.

- the change of orientation and length can be accomplished by clicking and holding the mouse well inside the section and dragging the section to the desired final orientation and length: it will then grow and rotate around its link with the mother section
- the number of lanes is reduced from four to three as if it were an isolated section, that is, dragging one side with the cursor reduces the section by one lane
- the mother section is selected again and the *section* icon pressed again, to generate the lane that follows from the side lane of the mother section
- this one-lane section can be manipulated in the same way as the previous one

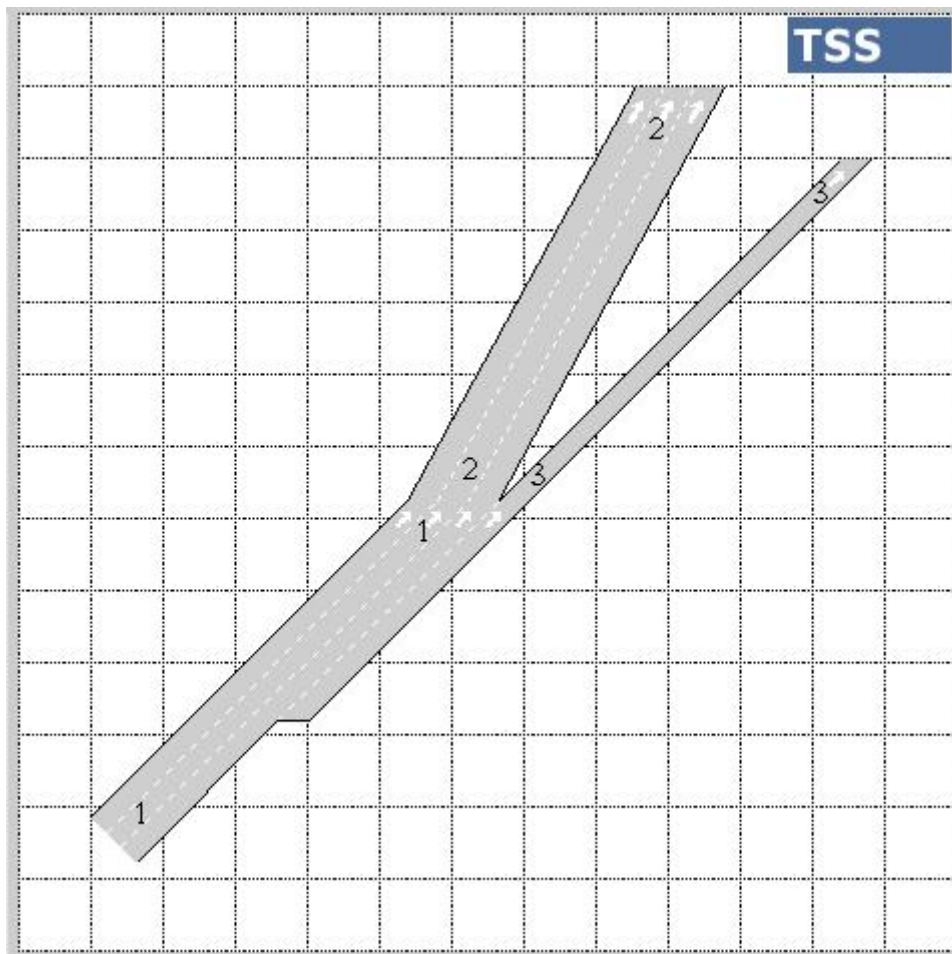


Figure 4-16: New section transformed

Another important task involves forming a join between two sections, that is, connecting two sections that were created separately. We may have a section originating from an exit ramp that needs to be connected to an acceleration lane. An example can be found in Section 3.5.2, which covers the topic of making and breaking joins.

4.7 Section properties

4.7.1 Basics

Section: 10

Basics | Lanes

Id: 10 Name:

Type

Type: arterial

Maximum speed: 50.00 Km/h

Capacity: 1800.00 veh/h

Distance Zone 1: 15.00 sec.

Distance Zone 2: 5.00 sec.

On Ramp: 5.00 sec.

Yellow Box speed: 14.40 Km/h

Visibility distance: 25.00 meters

User Defined Cost: 0.00

Max. Give Way Time Var.: 0.00 sec.

Connection speed

Maximum speed: 50.00 Km/h

Auto Evaluate: yes

Altitude

Initial: 0.00 Final: 0.00 meters

Slope percentage: 0.00 Set Percentage

OK Cancel

Figure 4-17: Section folder

Section type: There are eight different section descriptions: arterial, road, freeway, ring road, on/off ramp, urban road, street and signalised street. Each of these types implies different values for the capacity and maximum speed. The user can create new section types and edit the existing ones. Great care must be exercised when editing the eight basic types.

Maximum speed: Legal speed limit of the section.

Capacity: Maximum flow the section can take. AIMSUN makes use of capacity only with the route choice models; there are no capacity restrictions on the simulation of the section. Capacity is a result of microscopic simulation, not a constrain.

Distance Zone 1: Distance from the next turning point at which the drivers begin to look for a gap to change lane, if necessary.

Distance Zone 2: Distance from the next turning point at which the drivers begin trying to force a gap to change lane, if necessary.

On Ramp: This can represent the distance from the end of the acceleration lane at which the drivers begin to look for a gap. For further information see AIMSUN user manual.

Yellow Box speed: Vehicles travelling at this speed or slower, in the junction that could be found at the exit of this section, are considered to be queuing. This means that in case of a yellow junction, a vehicle travelling towards it through this section that detects other vehicles in the junction at this or lower speeds will brake, stop and wait at the end of the section until the junction clears, before entering it.

Visibility distance: This is a parameter used with the give-way algorithms. It represents the distance to the give-way from which the driver can begin to look for other vehicles driving towards the intersection from other directions. It only needs to be set when a give-way is positioned at the end of the opened section.

User defined cost: This is a field accessible with the function editor's functions. It has been designed as the area where the user can introduce any relevant data for the cost functions that might not be already found among GETRAM's standard data set or any other local cost, like, for instance, toll costs.

Max. Give Way Time Var.: Value added or subtracted from each vehicle's Maximum Give Way Time parameter (Section 3.7.10) when driving through that section. Maximum Give Way Time is an attribute of the vehicle, while Maximum Give Way Time Variation is a parameter of the section. This parameter is a tool to modify the vehicle's Maximum Give Way Time at certain spots. Maximum Give Way Time affects:

- 1) The behaviour of vehicles at a Give Way. So, with the Maximum Give Way Time Variation you can increase/decrease the level of aggressiveness (increase/decrease the flow of vehicles through the give way).
- 2) The behaviour of vehicles when lane-changing at a section's zone 3 (increase/decrease the level of aggressiveness).

You should take into account that Maximum Give Way Time Variation is a value in seconds, that can be positive or negative, and that is constant for all vehicles because each vehicle type already counts with its own standard variation to its own Maximum Give Way Time.

Connection speed: Turning speed from one segment to the next one of a polysection. Connectors are the links that connect one segment of a polysection to another segment of that same polysection. It is not necessary to edit it in case of sections.

Altitude: This is used to specify the slope of the section (it can also be left blank). This is done by specifying the altitude at the origin and at the end of the section (initial and final altitude), or it can also be done by specifying the initial altitude and the degree of slope in the section (in %). When dealing with polysections it is also possible to select the whole polysection and set the initial and final altitude of the origin and the end of the polysection. In this way it is not necessary to edit every single segment of the polysection but to accomplish the same result in a single editing move.

A more complete explanation of these parameters and their effects on a simulation run is given in the AIMSUN User Manual.

Figure 4-18 and Figure 4-19 show how there is a set of parameters that should have the same value

for all the segments of a polysection, and that is the reason why they can only be commonly edited through the last section of the polysection. Figure 4-18 and Figure 4-19 show these detail.

Figure 4-18: Folder from the last segment

Figure 4-19: Folder from any other segment

But a section is connected to another section through a join, not a connector. A join is a separate object. To set the speed for vehicles going from one section to another you only need to open the join and enter the desired speed. This is displayed as in Figure 4-20.

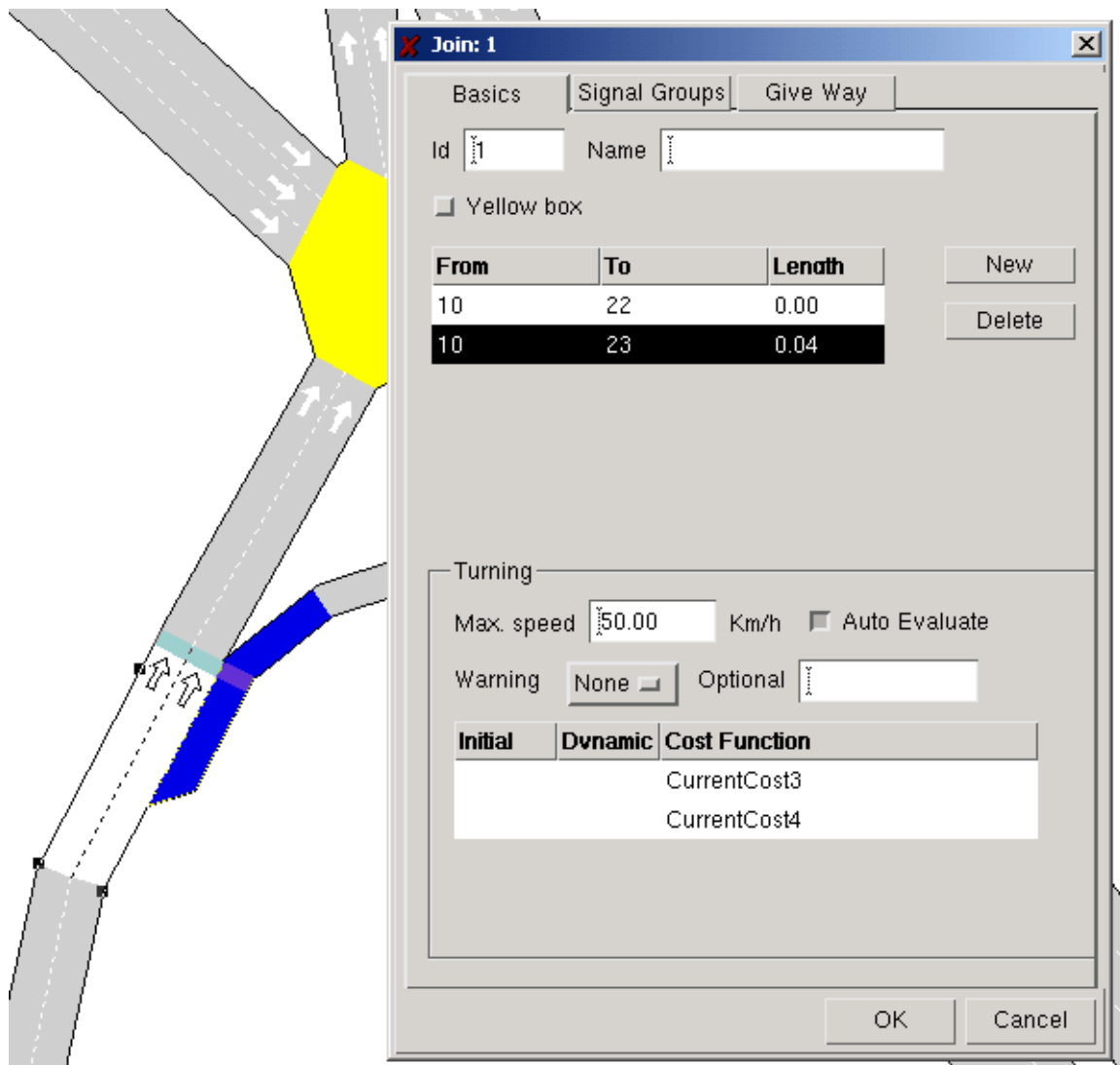


Figure 4-20: Open Join Dialog Window

4.7.2 Lanes

This is the field in which you select the lanes of a section to be reserved for a subgroup of vehicle classes, and in which you can “draw” any of the continuous lines that might limit the lane changing manoeuvres in the section. You can select the lanes by specifying the number of reserved lanes, the side of the section on which to place them, the classes of vehicles that are going to be allowed to circulate on those lanes and whether these vehicles will be forced to take those lanes (compulsory) or whether they will take them only if the traffic conditions on them are better than in others (optional). Before reserving lanes, the vehicle classes must be defined (section 3.2.11).

For instance, the characteristics of the section displayed in Figure 4-21 are 1 compulsory reserved lane on the right for vehicle types classified as ‘Public’, and a continuous line that does not allow vehicles on the second lane to change to the first one for the last 50 meters of the total 130 meters of the section.

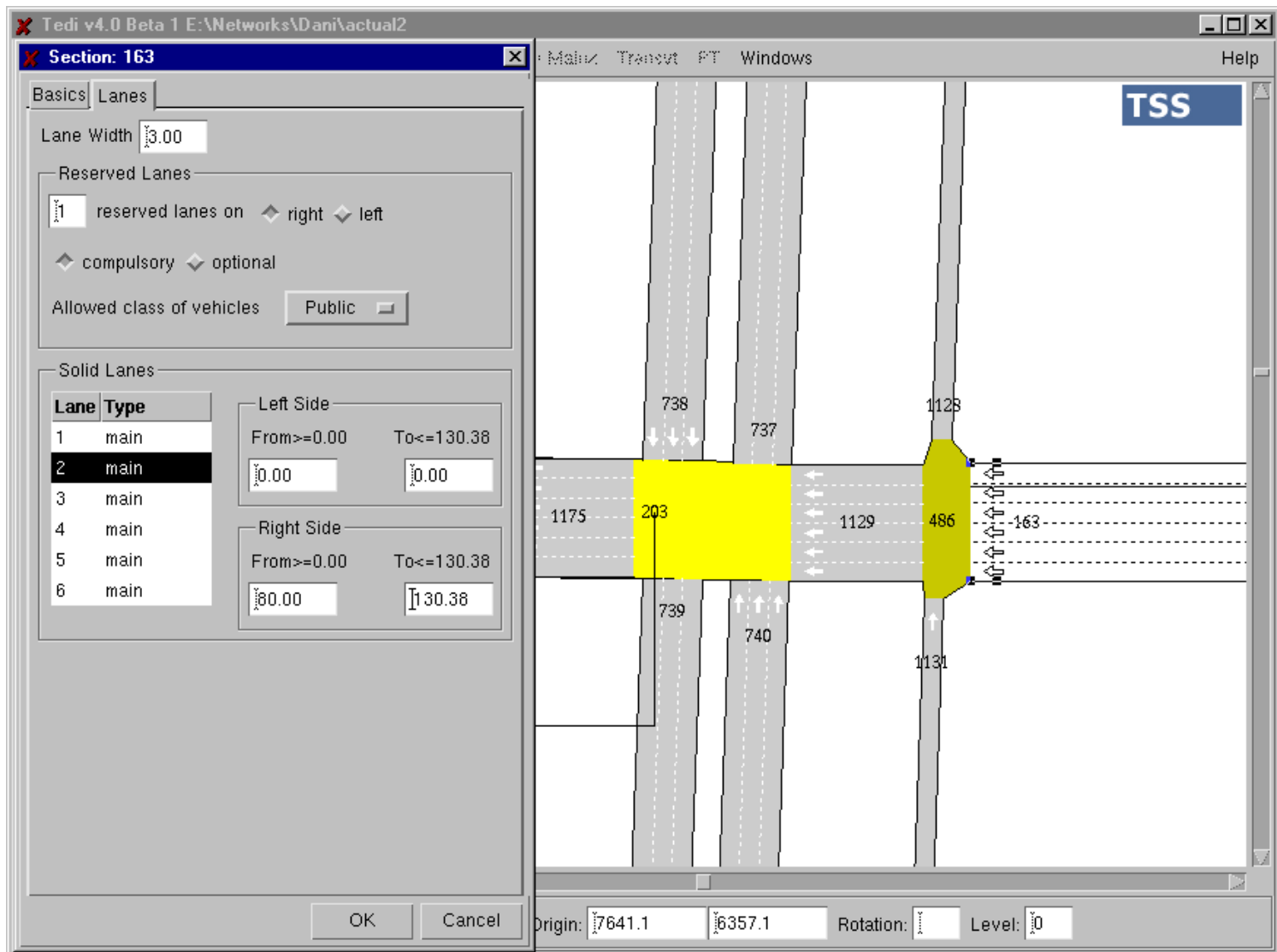


Figure 4-21: Reserved folder

4.8 Junctions

Once all the necessary sections of a network have been drawn, we end up with a geometrical replica of the real-world network. You now it is necessary to add the junctions and define the allowed directions a vehicle can take wherever more than two sections meet. Let us assume that we have built the network displayed in Figure 4-22.

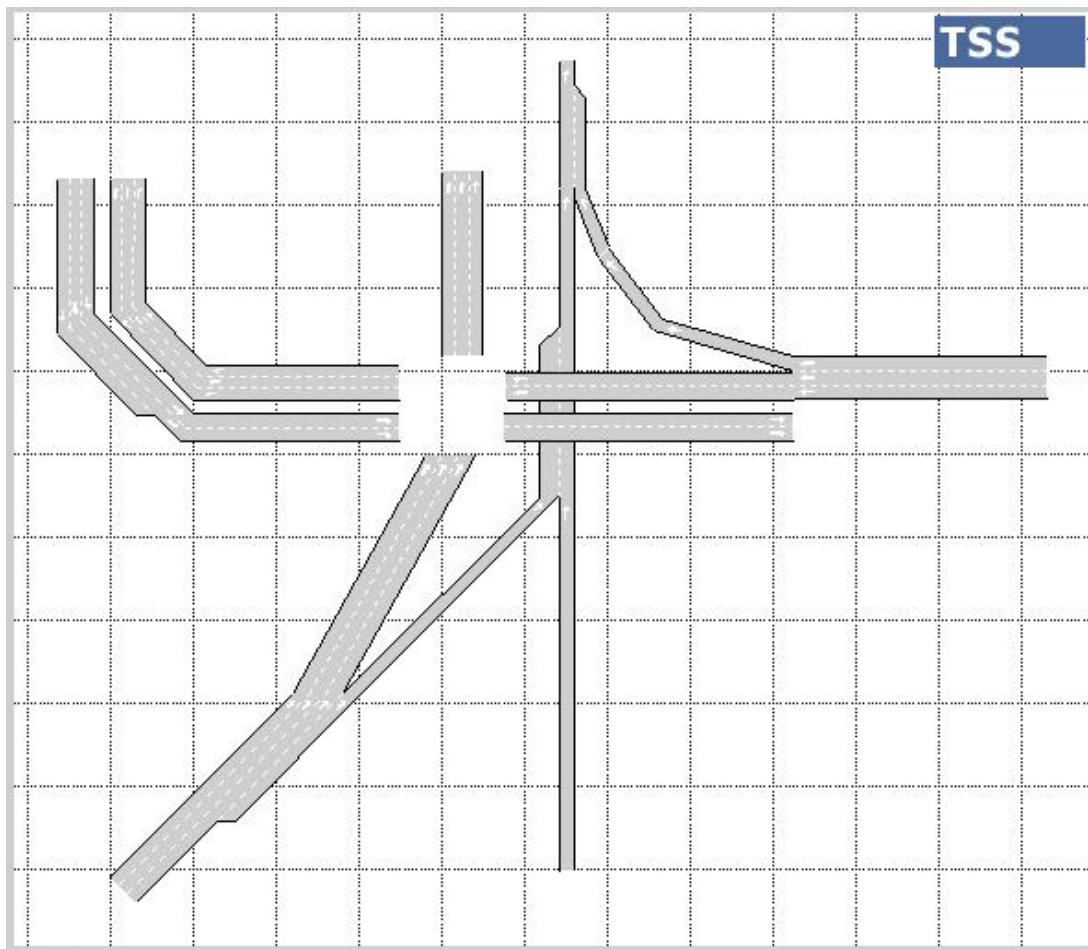


Figure 4-22: Traffic network (sections only)

To characterise a junction, you begin by setting up a detailed view of the required junction, and then press the *Junction* icon on the Tool Bar. A dialog window will appear (see Figure 4-23). The window has three different fields: *Basics*, *Signal Groups* and *Give Way*.

4.8.1 Basics

If the *Turnings* icon is pressed, an additional window appears. This is where you can define the turnings allowed in the junction. To do so, you first press the *New* button, then click on the origin lane and click on the destination lane. The result is displayed in Figure 4-23.

If there are two or more origin lanes, you need to select all of them by Shift-clicking on them in the same way as when selecting more than one section. You repeat this procedure with the destination lanes if you want to distinguish between destination lanes (option in *Global parameters* in the *Edit* menu). Otherwise, when clicking on one lane of the destination section, all the lanes will be selected as possible destinations for the vehicles in the origin lane. Each turning can be characterised using different parameters:

Max. speed: This represents the maximum turning speed. The editor calculates a value from the geometrical configuration of the junction and the maximum speed allowed through the sections. This value can be changed, however, by typing the desired value directly into the 'Max. speed' slot. The 'Auto Re-evaluate Speed' option re-evaluates the turning speed automatically whenever there is a change in the configuration of the junction.

Warnings: The 'Warnings' parameter has three options: none, give-way, stop. None means there is no sign, and give-way and stop mean that the corresponding sign is present in the origin lane. These warning signs then simulate the corresponding traffic conditioning.

Optional: This is left blank, providing an area in which to introduce any additional information not yet included.

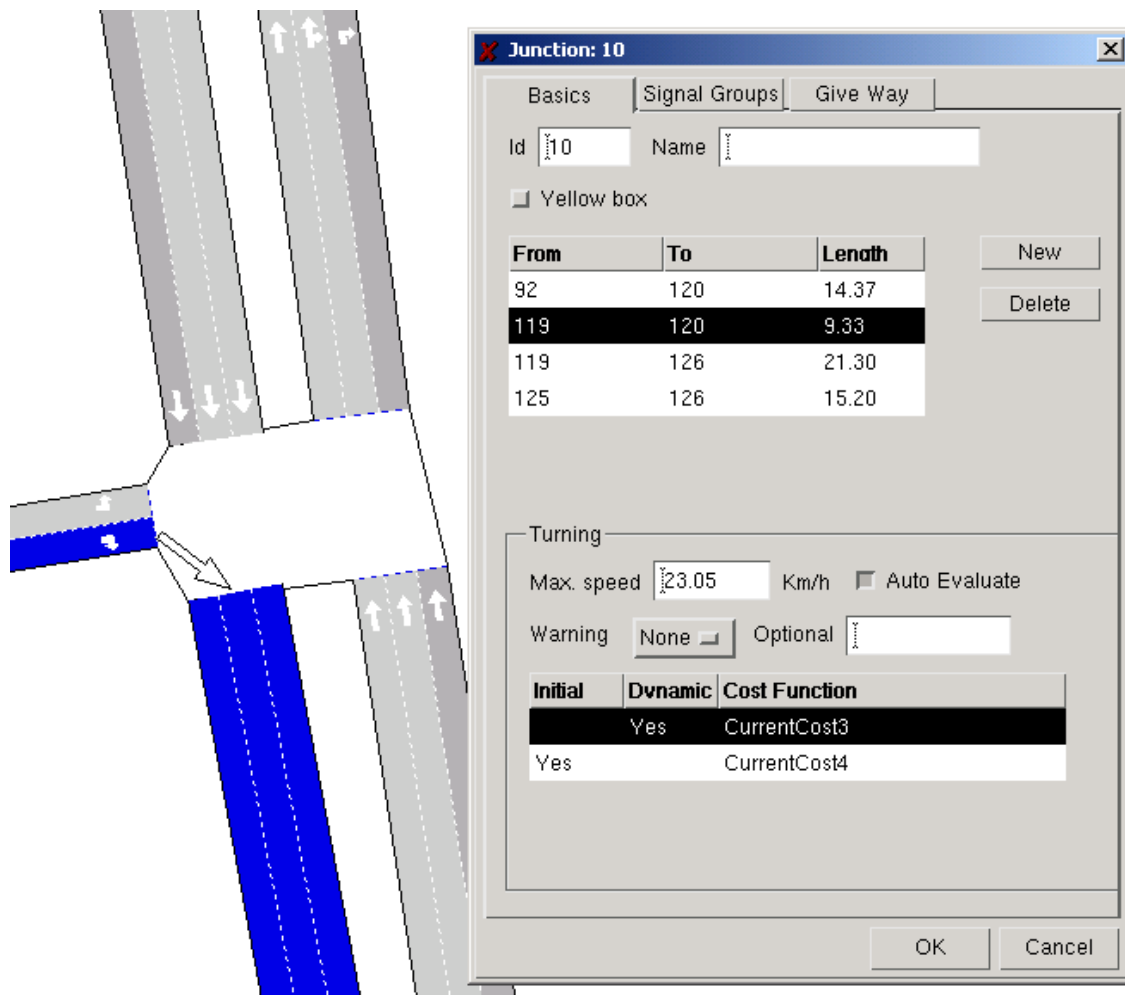


Figure 4-23: Creating a turning

Cost Function: When running the simulation with an OD Matrix, a route choice systems distributes de vehicles over the different paths from each origin to each destination. The distribution per path can depend on the cost of each link that composes the path. An link is the concept illustrated by Figure 4-24 and Figure 4-25: it is the section/polysection length dedicated to a turning movement from one node to the next, plus that same turning movement. The cost of each link depends on the cost function associated to it. A different cost function can be used for the very first calculation of routes at the beginning of the simulation (*Initial* at Figure 4-23) than for the successive calculations from that point on (*Dynamic* at Figure 4-23). The default cost function is travel time, but you can use other cost functions for specific situations like tolls or, for instance, cost functions that take into account other factors like the psychological weight of low speeds. In order to do this it is only necessary to edit the desired cost functions (section 3.2.13) and associate them to the specific links. In order to associate a cost function to an link it is only necessary

to click at the 'Initial' and/or 'Current' column beside the cost function at the bottom window of the Junction dialog window. The row of that cost function appears then selected in negative style as it can be seen at Figure 4-23.

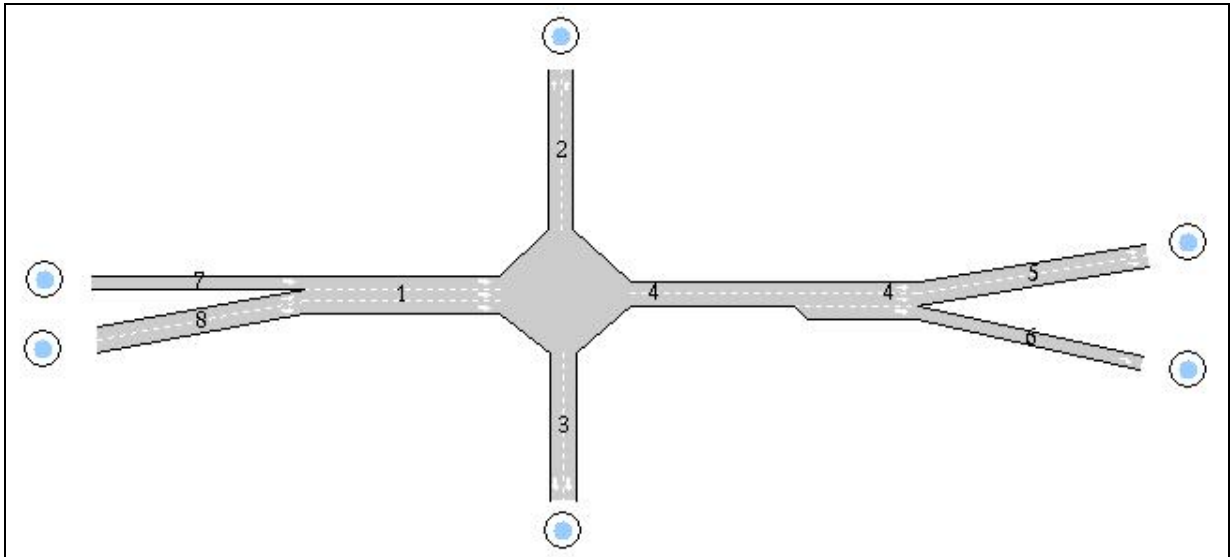


Figure 4-24: example network

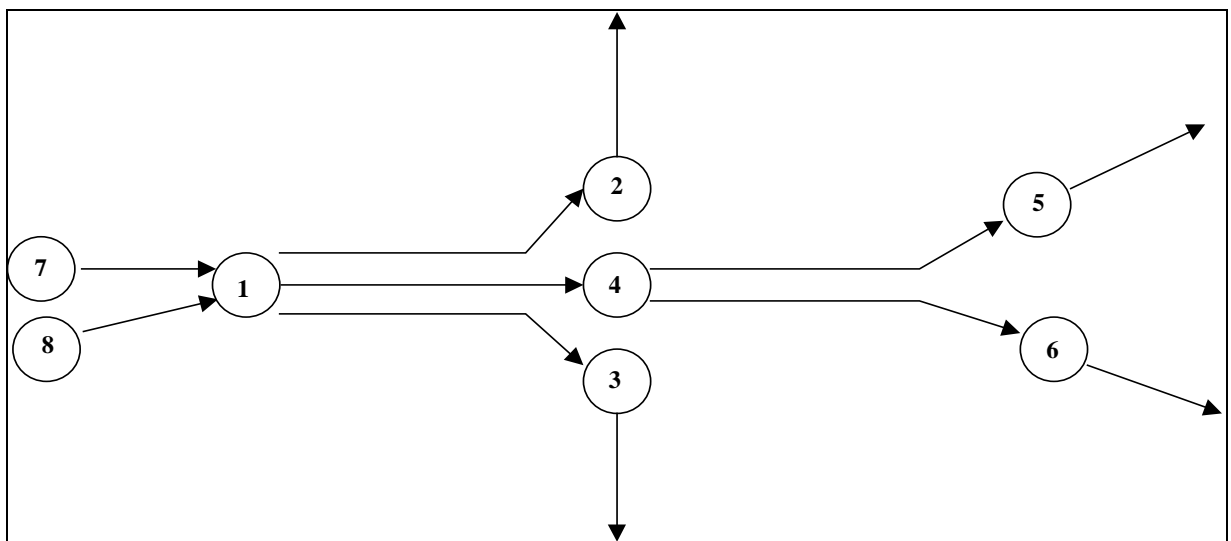


Figure 4-25: Representation of Network as Links for Shortest Routes calculation

4.8.2 Signal Groups

The *Signal Groups* area is where you can store the traffic lights present in the intersection, and the turning movements associated to each traffic light. When the *Signal Groups* folder is selected, the display changes to the dialog window in Figure 4-26. What we can see in Figure 4-26 is that:

- signal group '2' has been selected
- we can see it controls 2 turning movements (from section '3' to sections '2' and '116'); we can visualize this information both at the dialog window and in a graphical format as arrows in the junction
- one of its assigned turnings has been selected (from section '3' to section '2'); the arrow that

represents this turning movement is black-coloured

- a turning that has not been assigned to any signal group has also been selected to visualise it (from section '3' to section '4'); the arrow that represents this turning movement is red-coloured

To enter the traffic lights:

- click on the *New* button
- click on the turnings active when the traffic light just created is green
- click on the *New* button again and choose the turnings associated with another traffic light, repeating this process until all traffic lights have been included in the junction model.

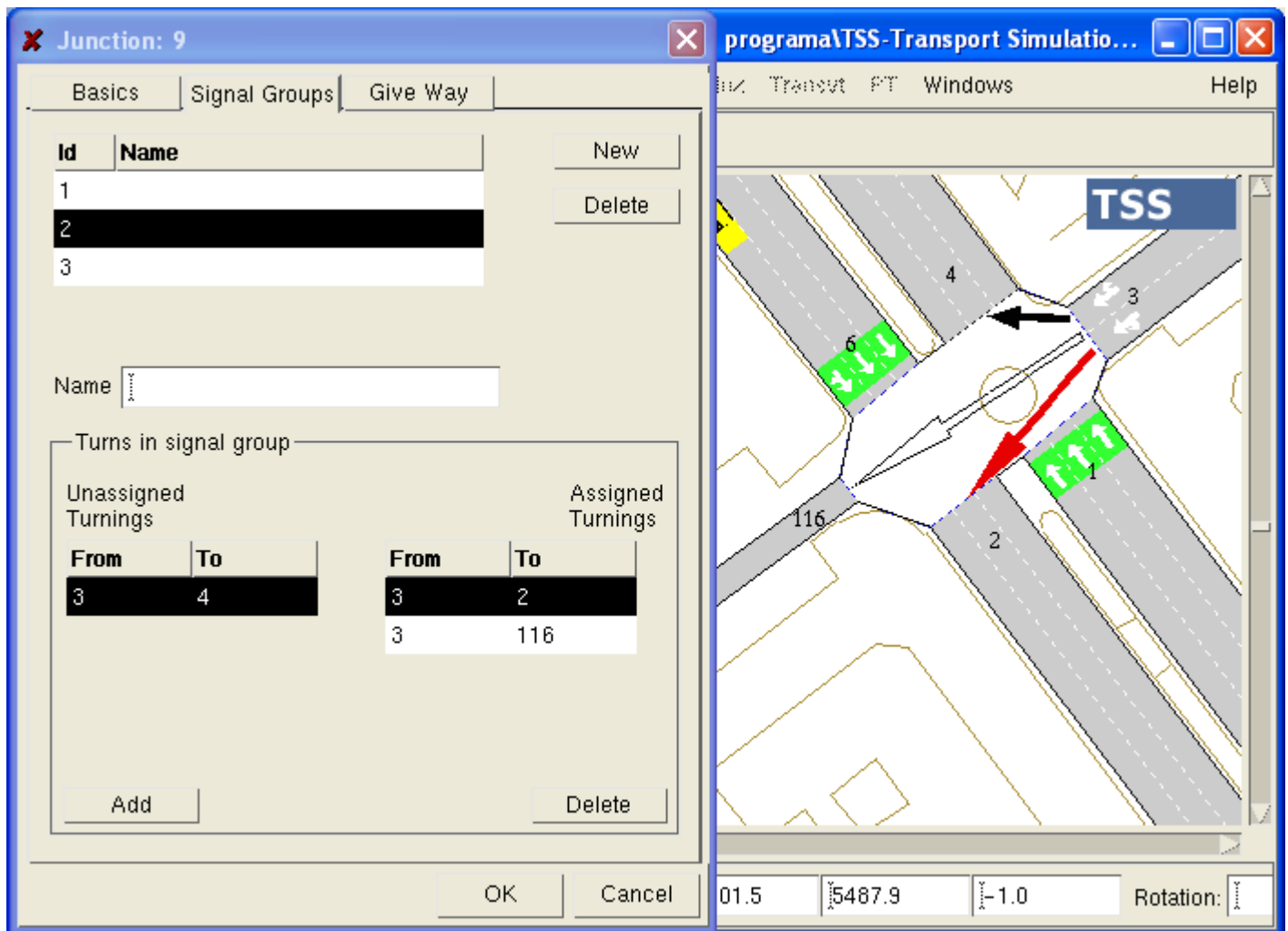


Figure 4-26: Signal Groups

To make any changes to a previously designed junction you must first reopen it. As explained previously, you simply double-click on an object to open it. If a join is opened, you can see that it is no more than a simplified junction.

4.8.3 Give Way

For the turning movements with yield sign defined (Give Way or Stop sign) the user can define what are the relative turning priorities through the Give Way folder which is in the Junction dialog.

A matrix with all turning movements of the junction appears in the Give Way folder. Only the cells

corresponding to pair of movements both having yield sign, are editable. It is assumed that any turn with yield sign will give way to all turns without yield sign. Priority definition only affects to the relations between pairs of movements with yield sign.

To set a priority, put a 1 in the corresponding cell, taking into account that the turns in the rows of the matrix give way to the turns in the columns where there is a 1 in the cell.

In Figure 4-27, turn 32-5 will give way to turn 23-5. Of course that turn 23-5 will also give way to turn 2-33, as this turn has not yield sign, so no need to define this priority.

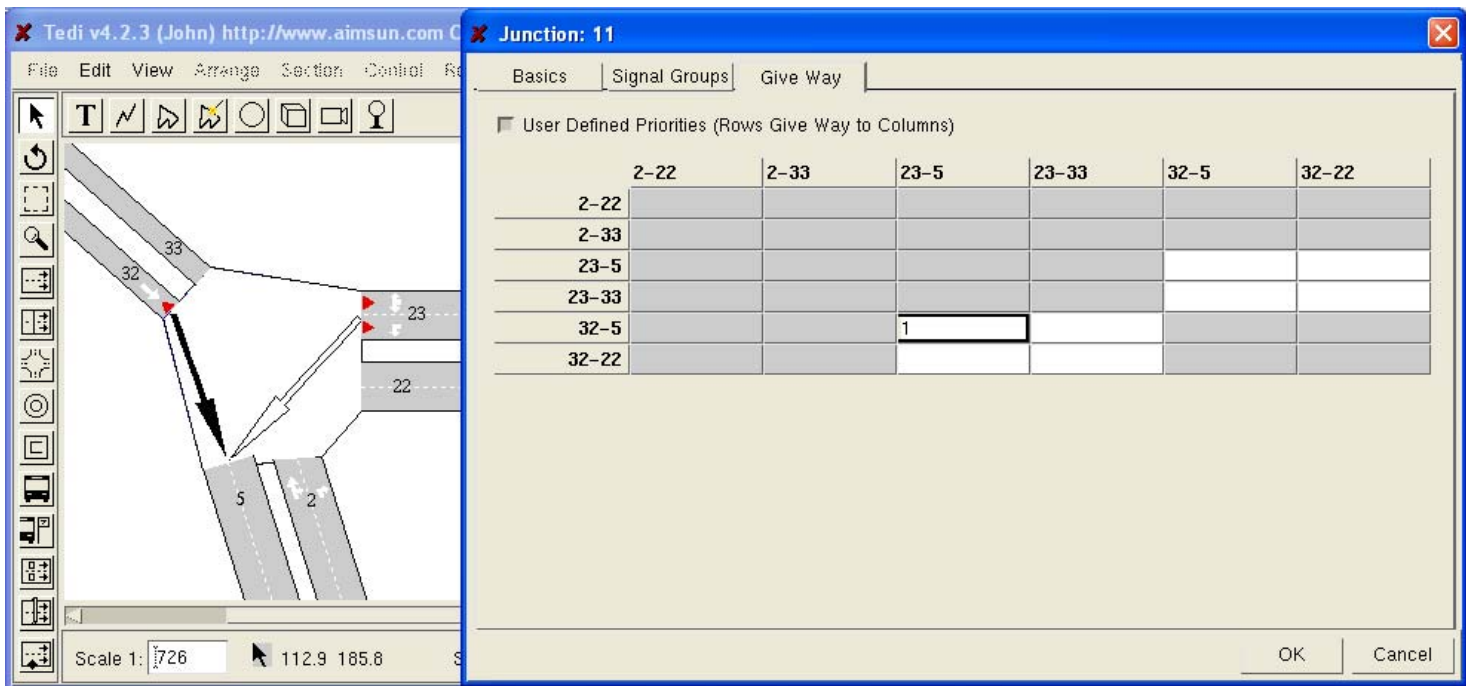


Figure 4-27: Signal Groups

4.9 Centroids

When working with O/D matrices, the centroids located in a network represent the origins and destinations. Each centroid is linked to the network through one or more connections. A connection is a virtual entity that the vehicles use to jump to/from the centroid to/from the entry/exit sections, (see Figure 4-28).

4.9.1 To create a new centroid

- click on the Centroids Icon, which is the last icon (a circle shape) at the bottom of the Tool Bar
- click on the drawing area at the position at which you want the centroid to be located; a centroid is then drawn at that spot, and the 'Centroids' dialog window opens

In this dialog window, you can enter a name for the centroid and then proceed to define the connections.

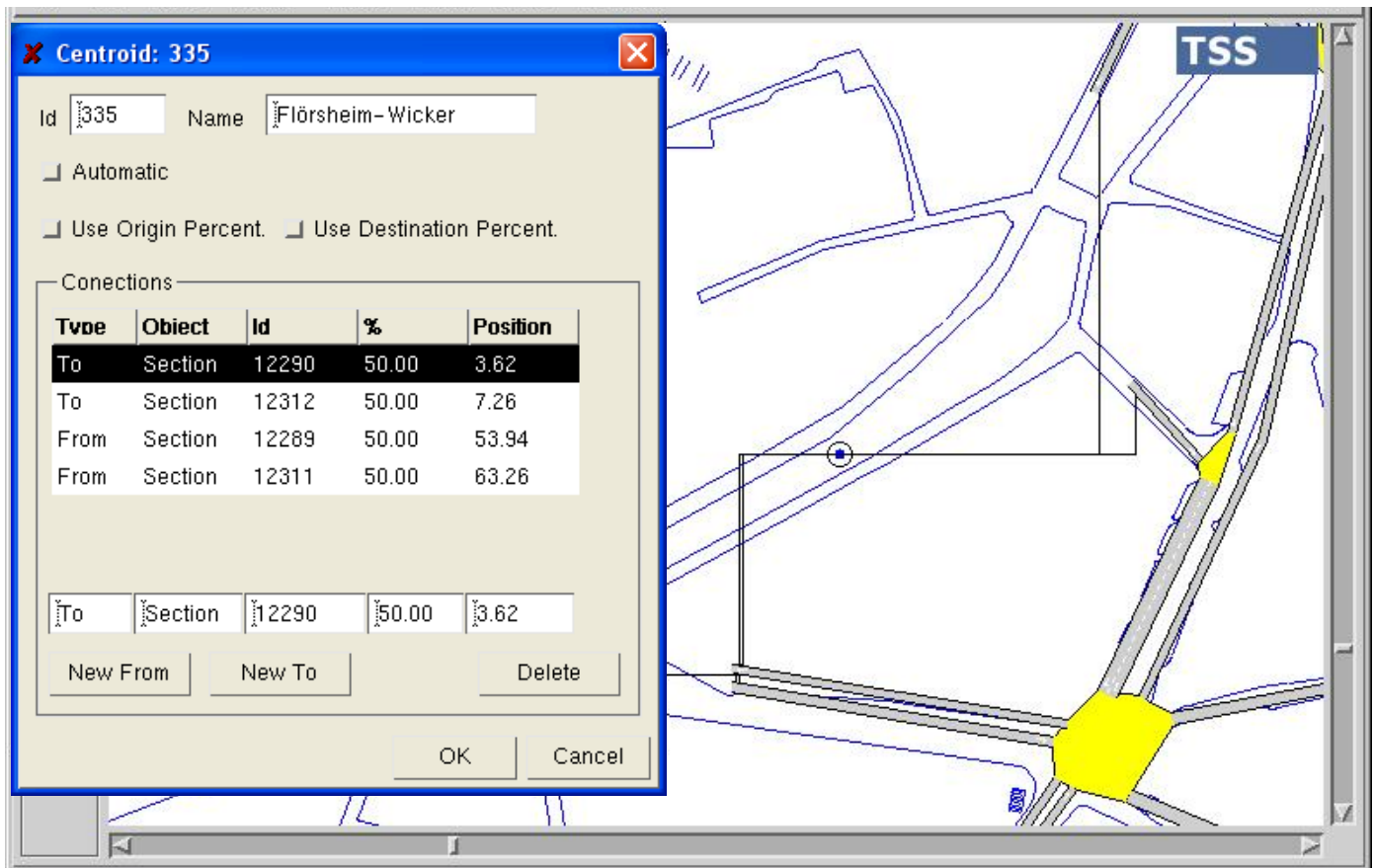


Figure 4-28: Opened centroid and connections

4.9.2 To define connections

If you want to define a type "To" connection (means 'To Object', therefore a Source or Origin of traffic), just click first on the *New To* button (Centroid dialog window) and then click on the section (or node) to which you want to establish the connection.

If you want to define a type "From" connection (means 'From Object', i.e., a Sink or Destination of traffic), first click on the *New From* button and then click on the section (or node) to which you want to connect the centroid.

To change an existing type 'To' connection into a type 'From' connection, or vice-versa, you must delete the connection (using Delete button) and create it again.

In general (but not always), a "To" connection is used with entry sections and a "From" connection is used with exit sections.

The *Automatic*, the *Use Origin Percentages* and *Use Destination percentages* toggles are used to configure the way in which the vehicles are introduced or leave the network through the connections. It works in the following way:

- If *Use Origin Percentages* = deselected: the vehicles entering the network from that centroid are distributed among the different connections taking into consideration their destinations. That is, they are introduced through those connections representing the shortest paths.
- If *Use Origin Percentages* = selected: the vehicles are introduced using as only criteria the proportions established by the fixed percentage associated with each connection.
- If *Use Destination Percentages* = deselected: the vehicles leaving the network to that centroid

use the connection at the end of the path calculated by the route choice systems, taking into consideration the routes' costs.

- If *Use Destination Percentages* = selected: the flow on each route is adjusted in such a way that the relative output flow through each of the output connections is the one set by the percentages.
- If *Auto...* = selected: when creating the connections, each connection will be assigned the same percentage, that is, vehicles will be distributed equally among all connections.
- If *Auto...* = deselected: when creating the connections, each connection will be assigned a 0% of the vehicles generated by the centroid and when updating it will assign the same percentage to all connections, except when adding a connection to a set of connections that already exists. Then, as the existing percentages add up to 100% and the new connection is assigned a 0%, the result will still add up to 100% and the situation will be saved as created. So be careful with percentages for centroids that have been edited more than once when changing their condition from not considering the percentages to considering them.

A centroid may also be connected to/from an internal section (embedded section) or junction. This means that any section or junction in the network can be a 'source' and 'sink' of modelled traffic, for instance, a parking lot.

4.9.3 To edit an existing centroid

Double-clicking on a centroid opens the centroid dialog window. It is then possible to edit the name of the centroid or one of its connections by selecting it. You are only permitted to change the %, which represents the percentage of traffic generated in this centroid that will be fed through each connection (if this modality is active). After any changes, check that all the percentages of all the connections add up to 100%. After making a change, you must press the return button to validate the change. Neither the Object identifier nor the Position can be edited; they are automatically set when the connection is first defined.

4.10 Bus Stops

Bus stops are created and manipulated in the same way as detectors. You just click on the icon, click on the section in which you want to position the bus stop, and then drag it to its final position or change its length by dragging the update points. The length of a bus stop determines its capacity. As many buses as fit within the bus stop will stop simultaneously, those that find the bus stop fully occupied will wait until enough space is generated. The space needed for more than one bus is the sum of the length of the vehicles plus the minimum distance between them. If the bus stop is too small for an arriving bus, then the bus will not enter the bus stop and will remain blocked for the rest of the simulation. You can position more than one bus stop in the same section, as for any other sort of object.

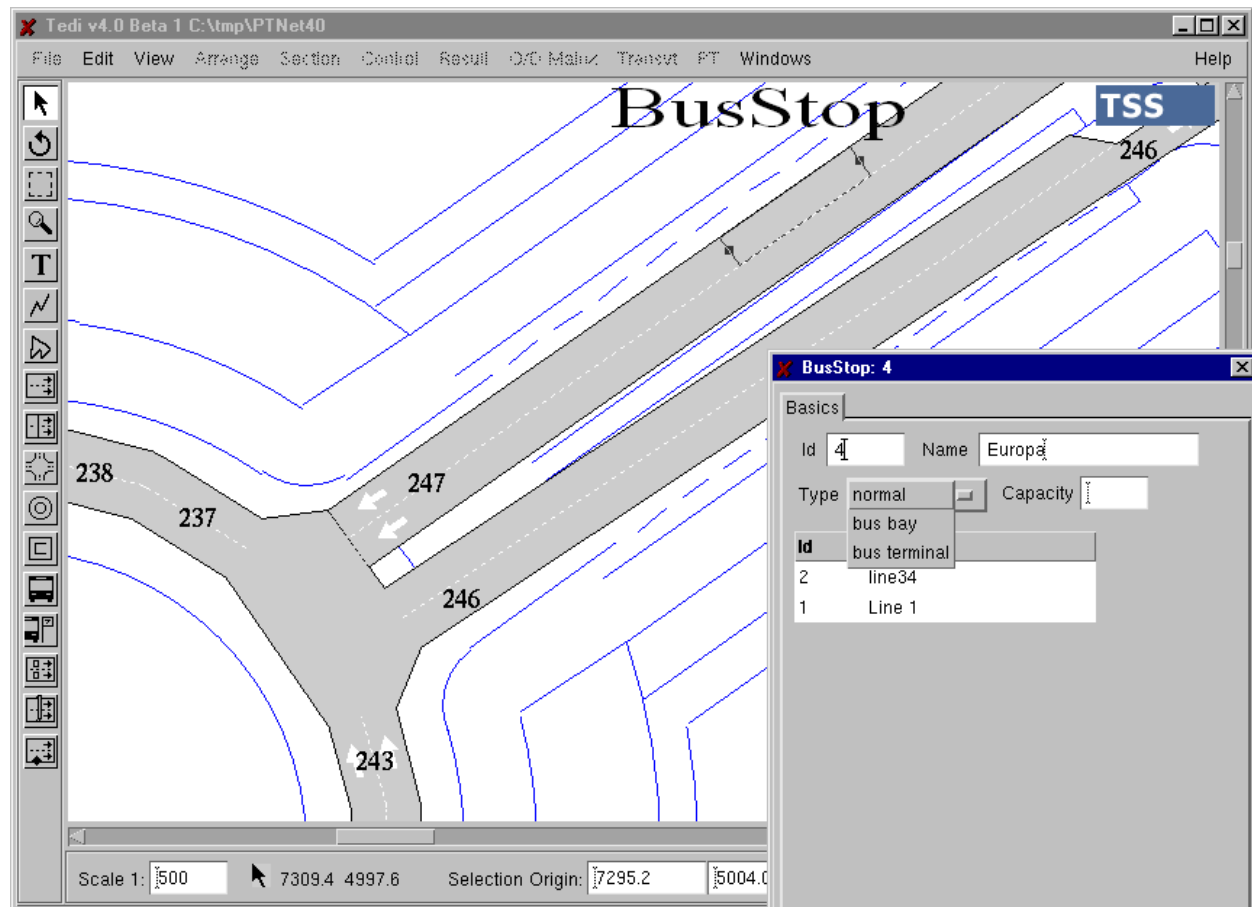


Figure 4-29: Bus Stop Dialog Window

Figure 4-29 displays an open bus stop.

Id: This is where you can name the bus stop or give it an appropriate identifier.

Name: additional field for assigning a name to the detector.

Type: this option characterises the bus stop as:

- Normal: on-street bus stop, that is, bus blocks the lane when it stops
- Bus Bay: bus leaves the lane to a special side lane where the bus stop is located. Only buses have access to a bus bay and buses enjoy priority when leaving the bus bay to re-enter the main traffic lanes.
- Bus terminal: this is the bus station, the buses ‘disappear’ when they enter the terminal and re-enter the model at that point when starting their route again.

Capacity: capacity of the stop. For a bus terminal this is the number of buses that it can park.

Information window: this window displays the bus lines that use this bus stop, both their id and name. This window exists purely for information purposes. A line is assigned to a bus stop in the *Public Line* dialog window.

4.11 Public Line

To create a public line (bus, tram, LRT) you first click on the icon and then on any free area of the model. The specific spot is not especially important because the created object (see the zoomed area

of Figure 4-30, the yellow square with a front view of a bus in it) is just a graphical entity on which you can double-click when you wish to access or consult the data of the public line represented by this image. It is very different from other objects such as detectors or VMS's, which are linked to the section in which they are created.

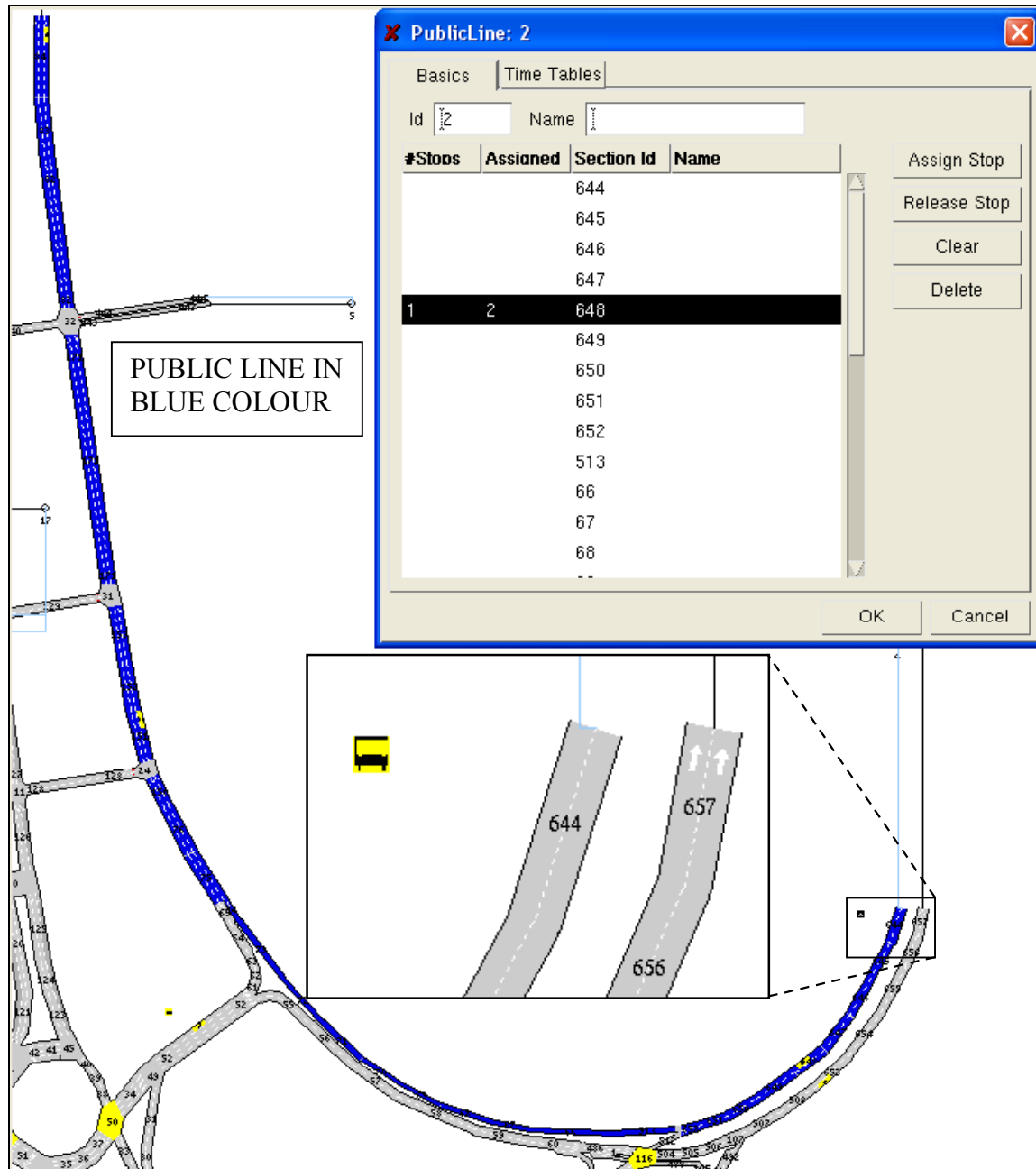


Figure 4-30: Public Line

4.11.1 Routes

To edit a route, proceed as follows:

- Activate the editing capability by clicking on the *Edit Route* toggle
- Click on the sections that make up the route, one after the other and in the order in which the bus will follow them.
- Select a section in which the line has a stop and then select a stop within that section (in case there is more than one) at which you would like the bus to stop.

- Finally, click on the *Assign Stop* button.

If you are editing an old network and only changing part of it, you do not need to re-edit a whole route again. When you open the line, you will see the whole route with the exception of those parts of the network that have been modified. Once you've opened the Public Transport Line (double click on its icon) you can close the dialog window only with the *Cancel* button, as the *OK* button will issue a warning message informing you that the route is unconnected. This is because deleting sections within the route has 'broken' the route into two groups of connected sections. An example of this is displayed in Figure 4-31. In this figure, you can see how three sections of *Line 2* have been deleted. It is now possible to open the public line, but you receive the "Unconnected Route" error message when you try to close it with the *OK* button.

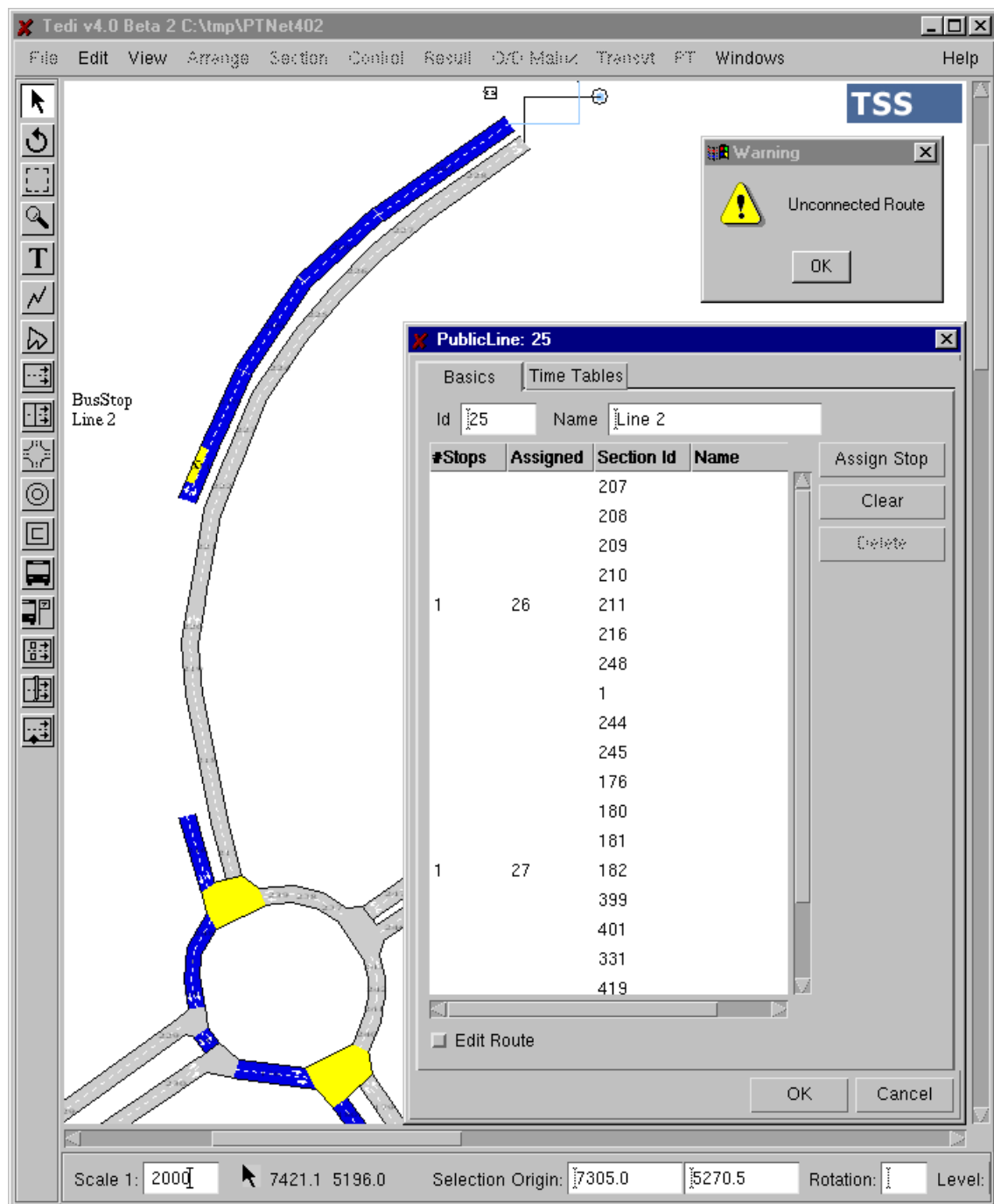


Figure 4-31: Unconnected Route

If the network is complete then you can re-edit the route from the last connected section and add the sections from that point until the first section of the next connected group for the original route. This is displayed in Figure 4-32, where:

- section 211 was selected
- sections 2 and 3 have been added (clicking on them with the mouse and in the flow order)
- now there is only left to click on the only missing section (section 4)

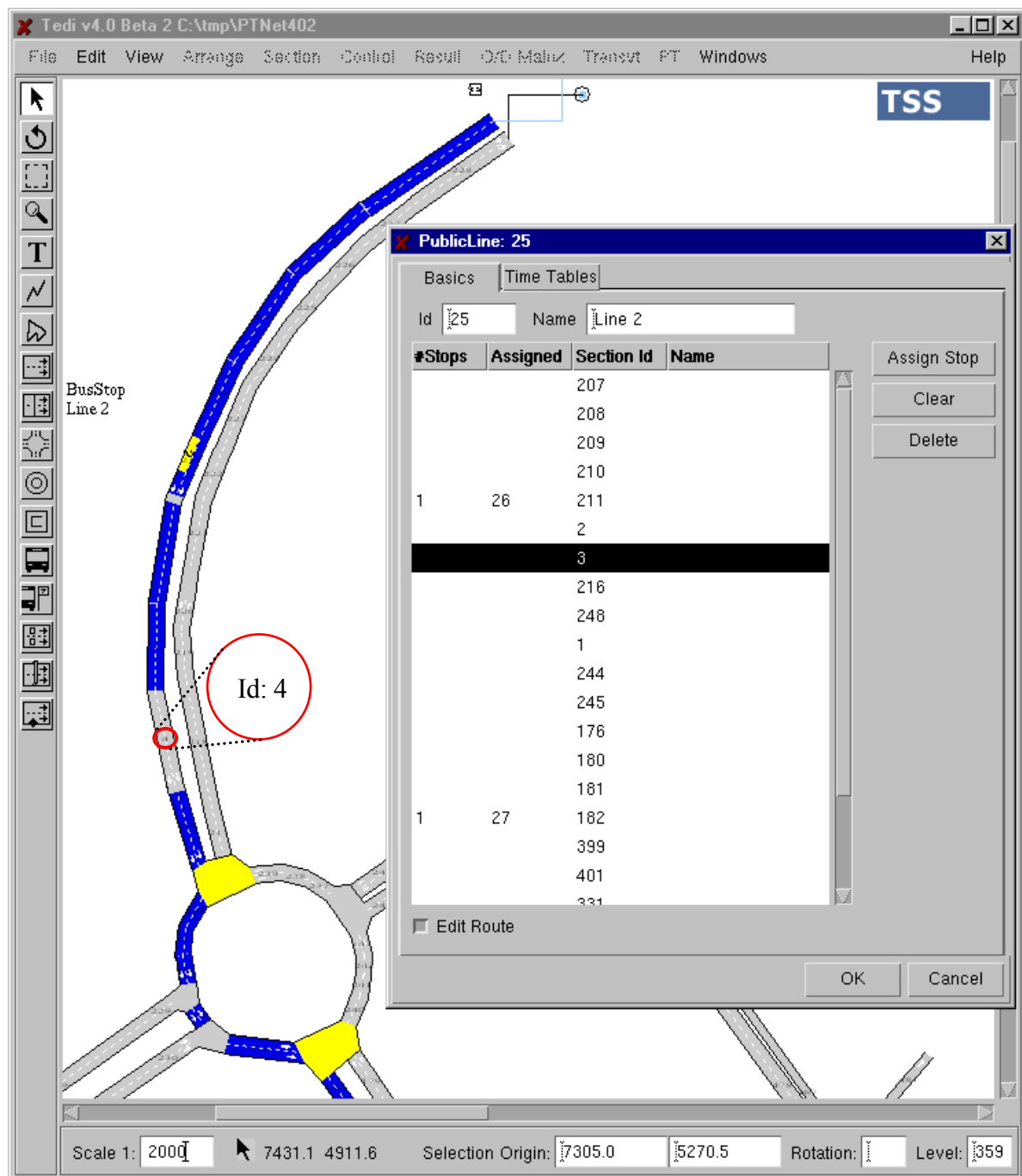


Figure 4-32: Re-editing a Route

4.11.2 Timetables

In the previous figure, you can see that there is a second *Timetable* data folder next to the *Route*

folder. This is displayed in Figure 4-33.

The top left of the window contains a list of the different timetables created so far. The *New* button allows you to create new timetables. The *Delete* button erases the selected timetable. The *Duplicate* button creates a new timetable exactly like the one currently selected. You can then edit the cloned timetable instead of having to create a whole timetable from scratch.

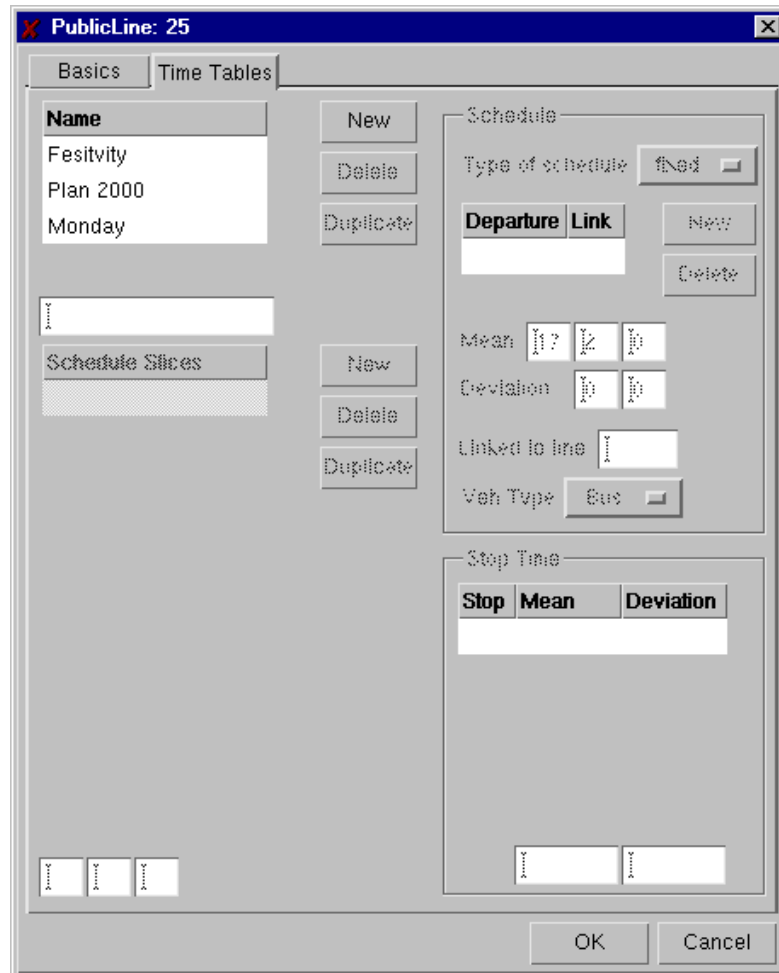


Figure 4-33: Timetables

When you select a timetable, the buttons listed above become activated, and the different timetable periods are listed in the *Schedule Slices* window.

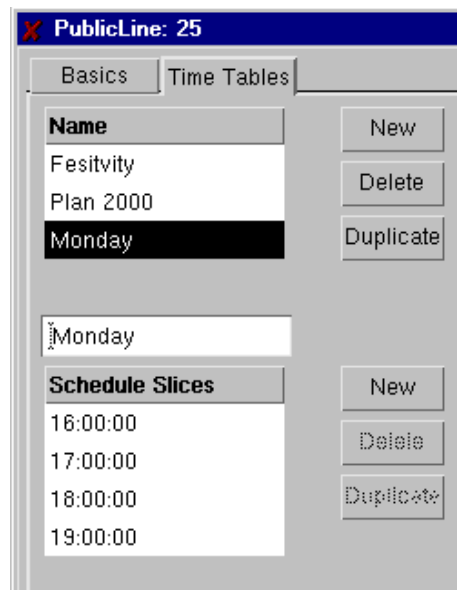


Figure 4-34: Schedule Slices

Here, the *New*, *Delete* and *Duplicate* have the same functionality as for the timetables. Selecting one of the slices not only activates the *Delete* and *Duplicate* buttons, it also displays the information associated with that slice in the right half of the dialog window (see Figure 4-35).

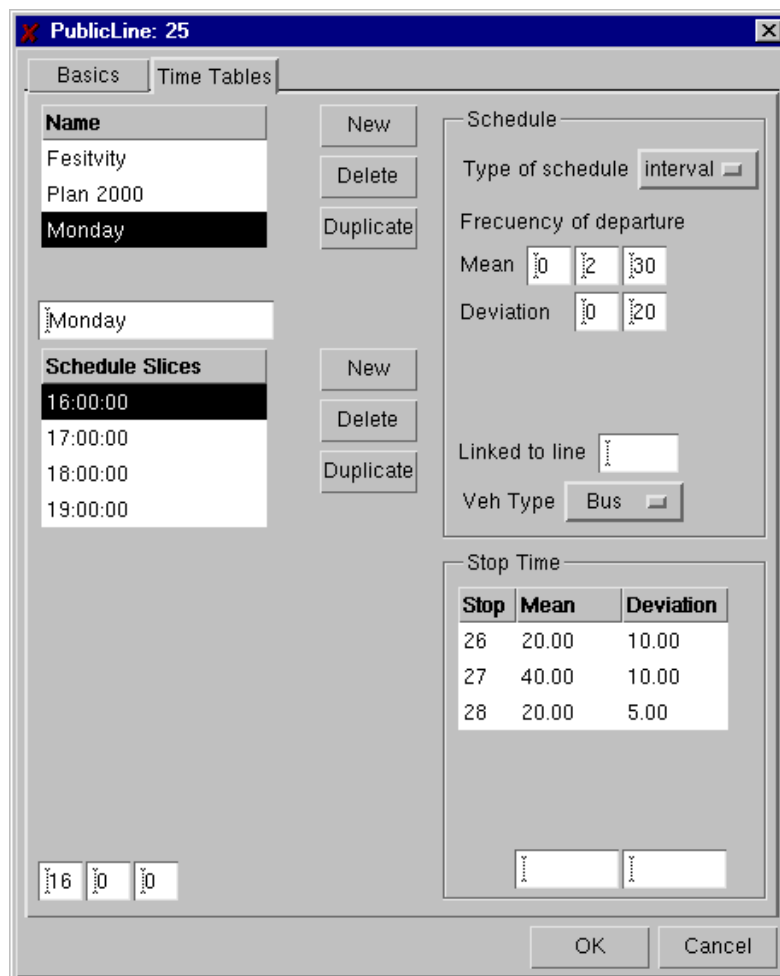


Figure 4-35: Interval Schedule

The screenshot shows a 'Schedule' dialog box with the following fields and controls:

- Type of schedule:** A dropdown menu set to 'fixed'.
- Departure/Link table:**

Departure	Link
18:00:00	
18:02:00	
- Mean:** A text box containing '18:00:00'.
- Deviation:** A text box containing '1:20'.
- Linked to line:** An empty text box.
- Veh Type:** A dropdown menu set to 'Bus'.
- Buttons:** 'New' and 'Delete' buttons are located to the right of the table.

Figure 4-36: Fixed Schedule

A schedule slice is a time period during which the public vehicle generation follows a given pattern, and the time spent by the public vehicles on the stops stays more or less constant.

Type of schedule:

- Interval (see Figure 4-35): this type of schedule is selected when buses are generated periodically at regular intervals for the time interval of the selected slice. This regular interval is defined as the frequency of bus departures. In order to define this, you must specify both that mean time between departures and the standard deviation from that mean.
- Fixed (see Figure 4-36): this second option allows you to create a list or time table with the time of departure of each bus, and also the standard deviation from that departure time, as in the Interval Schedule.

Linked to line: This allows you to provide a precondition for the departure on a line, linked to the arrival of another bus (either from the same line or a different line) during the simulation. Before departing, a linked bus will check whether a bus from the line to which it is linked has arrived during the previous period. If the Interval Schedule is being used, all buses on the edited line would be linked to the other line. This means that the “previous period” checked will be the period since the last departure. That previous period would be 2 minutes 30 seconds for the line displayed in Figure 4-35 if it had been linked to another line (and if the previous bus had departed on time). If a Fixed Schedule is being used, not all departures have to be linked, optionally only a selected subset can be linked. This means that the “previous period” checked will be the period since the last linked departure. This previous period would be 4 minutes for the line displayed in Figure 4-36 if a bus departed every two minutes (this is an arbitrary time difference derived up from the departure times displayed in the figure) and one of every two buses were linked to another line (and the previous bus had departed on time).

Vehicle Type: Section 3.11.3 describes how to associate different public transport vehicle types to a model, in case, for instance, you have buses with three different lengths and physical characteristics. The *Veh Type* toggle is where you define which of these vehicle types is to depart at the selected departure for the selected schedule slice. If you are using the Schedule Interval, all buses will be of the same type for the entire slice. If you are using the Fixed Schedule, you can associate a different type to each departure of the same slice.

4.12 Detectors

In order to create a new detector it is necessary to click on the detector icon of the toolbar and then

click on the point in the section at which you want to position the detector. The detector will appear, but will only occupy one lane at first. You can adjust its width and length by clicking and dragging the corner dots of the detector. Clicking and dragging from the middle of the detector changes its position.

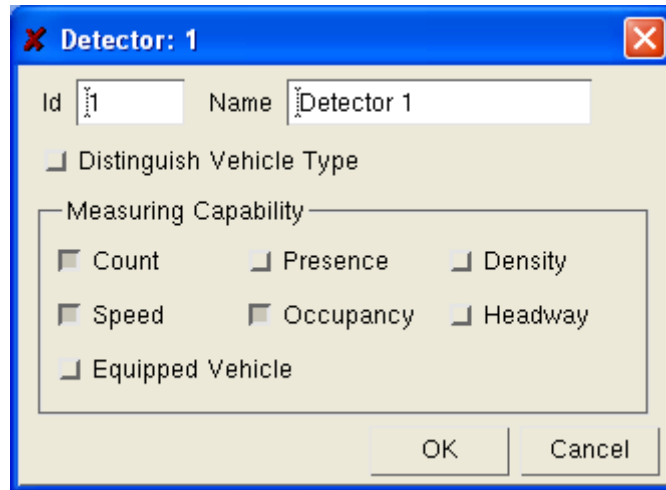


Figure 4-37: Detector Dialog Window

Id: this is where you can name the detector or give it an appropriate identifier.

Name: additional field for assigning a name to the detector.

Distinguish Vehicle Type: this option, when selected, causes the detector to distinguish between different vehicles types, giving the statistics both aggregated for all traffic and differentiated by vehicle type.

Measuring capabilities: any detector can measure up to six different variables, namely: Count, Speed, Presence, Occupancy, Density and Headway.

It has an additional measuring capability, that of detecting an equipped vehicle. A vehicle has the possibility of being somehow equipped, and the detector can have the possibility of emitting a special signal when a vehicle of that sort is detected. This feature is mainly used with PT pre-emption. The buses or LRTs of the line that enjoys that priority are equipped with an electronic device that is detected by a detector with that capability and that has been set at a certain distance from the following intersection. Then that detector generates a signal that starts the process that gives priority to the vehicle of that line.

4.13 VMS's

The process for creating and manipulating a VMS is much the same as that described for detectors, except for the fact that VMS's have fixed dimensions.

Id: this is where you can name the VMS or give it an appropriate identifier.

Name: additional field for assigning a name to the detector.

Type: there are two different types of VMS's, namely Gantry signs and Boom poles. This differentiation is included for the sake of completeness; it has no effect on simulations with AIMSUN.

Message: in this window, a list of all the possible messages of the VMS is shown. If you select a particular message (by clicking on it), its full contents are displayed in the secondary window at

the bottom of the section dialog window. If a message number with no message is selected, you can add a new message just by typing it in the secondary window.

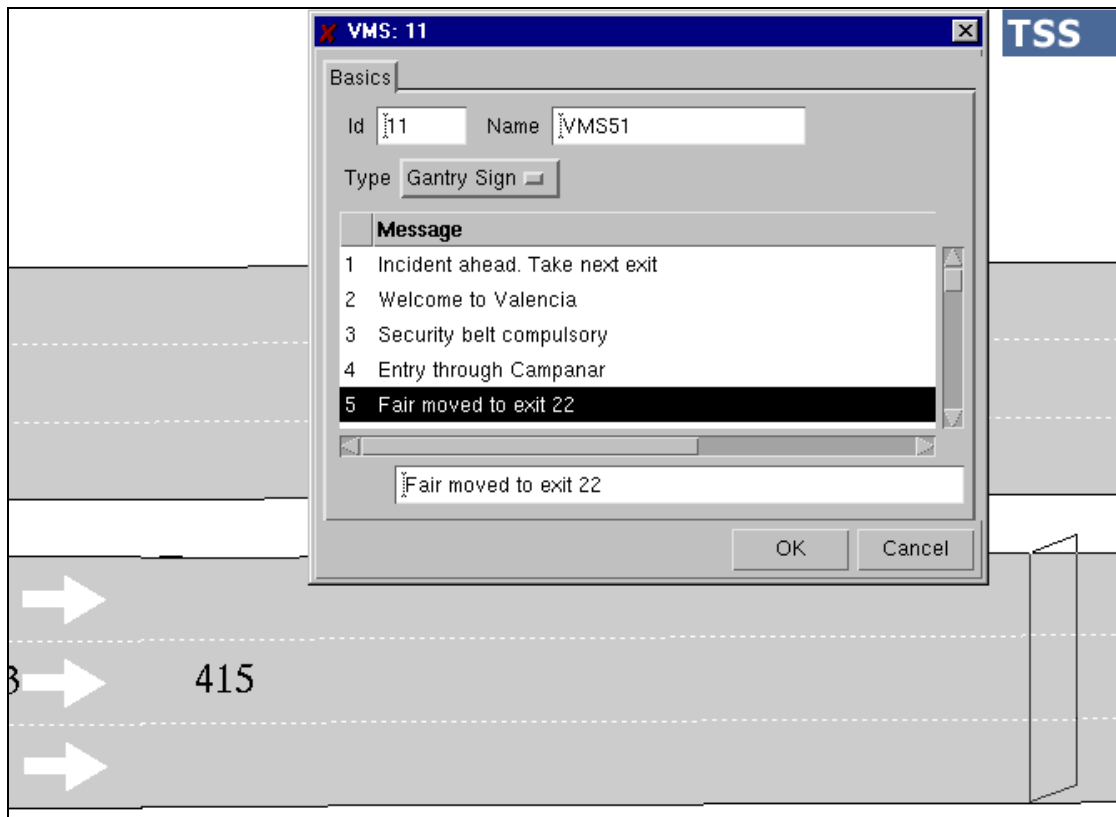


Figure 4-38: VMS folder

4.14 Metering

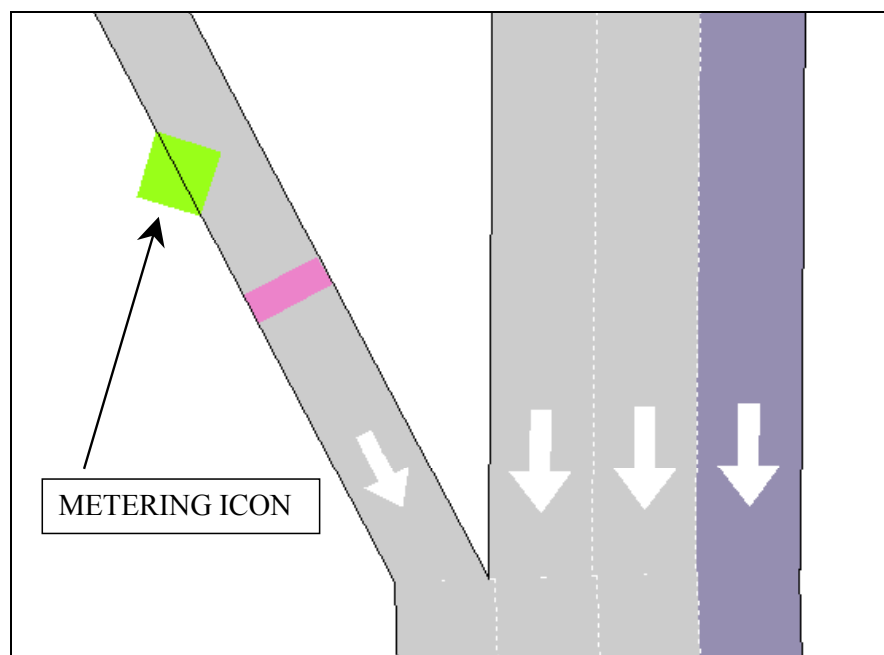


Figure 4-39: Metering Dialog Window

This space is dedicated to the creation of ramp meters in the opened section. To create a ramp

meter, click on the *Create* button. A rhomboid will appear in the middle of one side of the section, representing the controlling traffic light. Clicking and dragging from the middle of the rhomboid will change its position.

Id: this is where you can name the ramp meter or give it an appropriate identifier.

Name: additional field for assigning a name to the detector.

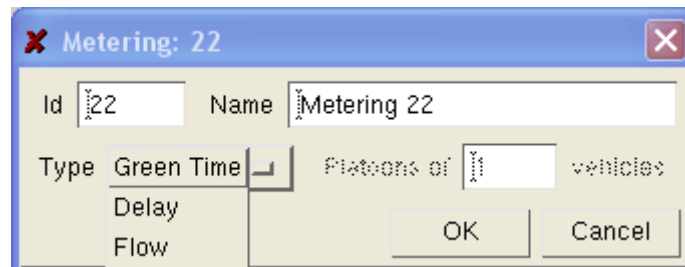


Figure 4-40: Metering Dialog Window

Type: zone with toggle buttons to select whether to meter the green light time, to allow flows of predetermined platoons of vehicles, or to impose a delay time as a means for modelling the stoppage of vehicles due to a control facility, such as a toll or customs barrier.

4.15 Controller

As when creating most objects, you only need to click on the controller icon and then click on the drawing window on the spot in which you want to position the controller. It will then appear, semitransparent because it is selected, allowing you to easily identify it and then drag it to its final position.

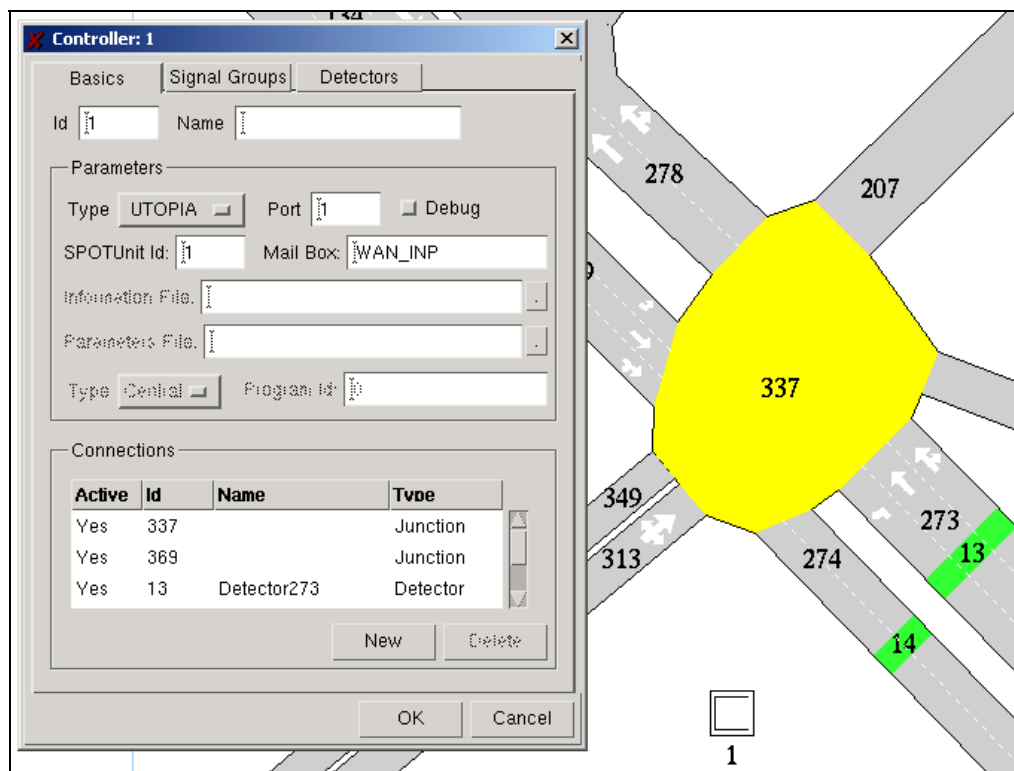


Figure 4-41: Controller Dialog Window (Basics)

A double-click will open the dialog window associated with the controller (see Figure 4-41). In this

window, you can establish the connections between the controller and those devices under its control by pressing the new button and then clicking on the required object (detector, VMS, ramp meter, junction).

When dealing with junctions, the second step is to establish how the controller ‘sees’ the different signal groups. A set of signal groups of different junctions are usually grouped into a single group which is then controlled by the controller, that is, if a controller activates one of its signal groups then all signal groups of all junctions grouped into that controller signal group are also activated. This concept is reflected in Figure 4-42. Here, you can see how signal group 1 has just been created and assigned signal group 2 from junction 369 and signal group 2 from junction 337, so that when controller 1 receives the order to activate its signal group 1, then signal group 2 at junctions 337 and 369 are activated.

Finally, we can define which is the information that the controller has to gather, which is defined in the *Detectors* folder.

SCATS/UTOPIA/VS-PLUS: If you label a controller as SCATS/UTOPIA/VS-PLUS then Aimsun tries to establish connection with SCATS/UTOPIA/VS-PLUS as soon as the model is loaded into the simulator. If it doesn’t succeed then the junction is controlled as pre-timed.

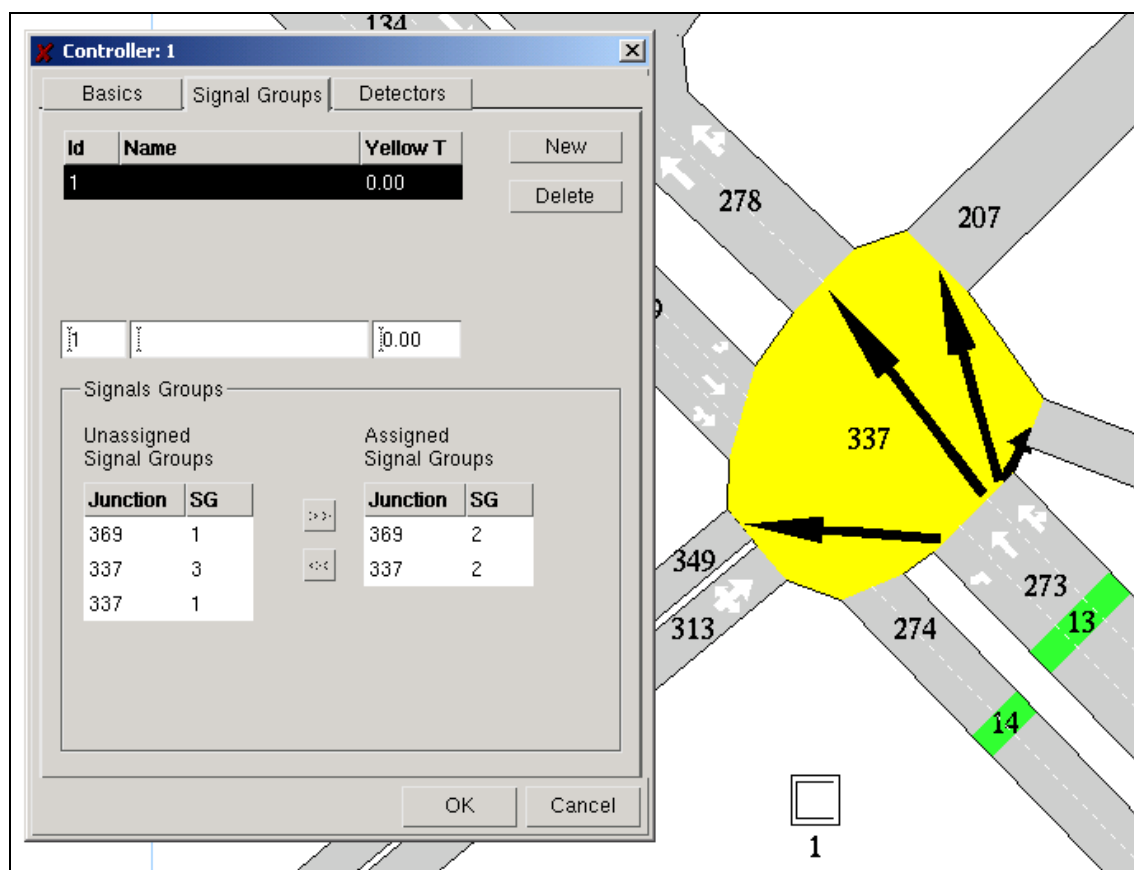


Figure 4-42: Controller Dialog Window (Signal Groups)

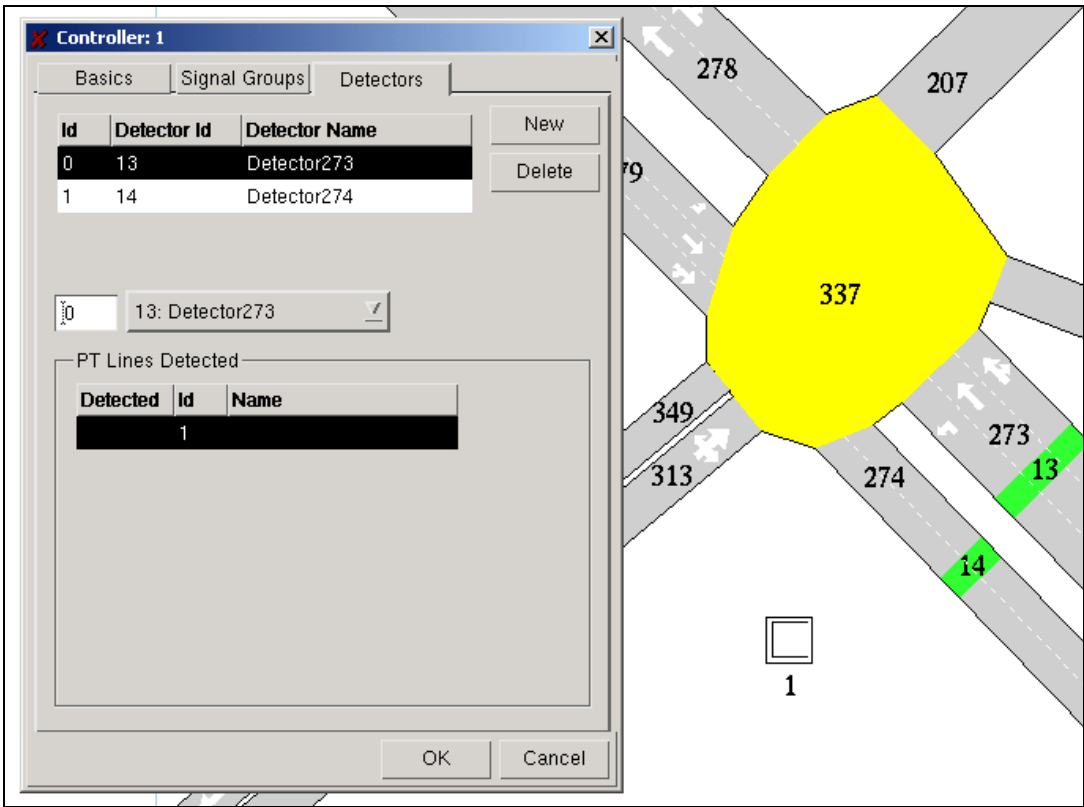


Figure 4-43: Controller Dialog Window - Detectors

5 ACTUATED CONTROL

The actuated control capability in AIMSUN implements the policy used by traffic signal controllers in accordance to NEMA (National Electrical Manufacturers Association) standards. The purpose of traffic-actuated control is to adjust traffic signal timing continuously for an efficient service of traffic demand at the branches of an intersection. These adjustments respond to real-time measures of traffic demand obtained from vehicle detectors installed in some or all of the approaches to the intersection.

Traffic actuated signal control can be split into the following categories:

1. **Semi-Actuated:** In Semi-actuated control, the major traffic movements are the ones active unless there is a conflicting call (=demand) from a minor movement. Detectors are required at the branches of all minor movements.
 - a. **Uncoordinated.** The phase servicing the major movements is placed in Minimum or Maximum recall and rests in green until a call from a conflicting phase is recorded. The next conflicting phase is served as soon as a gap-out or max-out occurs on the major phase if the major phase is actuated. If the major phase is not actuated (no detectors available), the conflicting phase is serviced after the minimum green time out of the major phase. Following the yellow change interval servicing the major phase, the controller places a call for the major phase, unless the rest-in-red feature is specified.
 - b. **Coordinated:** The major movement is the synchronized phase. Minor movement phases are served (if called) only after the synchronized phase's yield point. These minor-movement actuated phases either gap-out or terminate at their respective force-off points. Force-off occurs at a specified point in time within the background signal cycle to ensure that the major phase will be coordinated with those of the adjacent signal controllers.
2. **Full Actuated:** Volume-density operation is an option of full-actuated control. It can calculate the minimum green based on actual demand (demand understood as the number of detector actuations during yellow and red, if the detector is set back from the stop-bar), and it can implement a specified gap reduction function.

5.1 Actuated Parameters

- **Minimum Green:** It is the shortest green time period allowed for that phase. This magnitude may also be specified not as a fixed quantity but as a variable magnitude. It then depends on traffic demand and consists on a time parameter specified as *Seconds per actuation* multiplied by the number of vehicle actuations recorded during the yellow and red signal periods for this phase, up to a *Maximum Initial* time. The *minimum green* is then either that calculated time magnitude or the *minimum initial* (defined in TEDI as Minimum Duration), whichever is greater. The *minimum initial* is a parameter set in order to guarantee a minimum value for the *minimum green* independent from the number of vehicle actuations.
- **Maximum Initial:** The *Extensible Initial* is a dynamism that adds time to the *minimum initial* for each vehicle actuation received by a phase during the yellow and red signal. The *maximum initial* is the maximum value up to which the initial time (*minimum green*) can be extended. This is the volume-density feature. In case we want to deactivate this mechanism, the procedure in TEDI is just to set the same value both for *maximum initial* and for the

phase's minimum duration.. Finally, in order to enable this feature it is necessary to set locking detectors (the detectors must have count as one of their capabilities).

- **Passage Time (or vehicle Extension)** is the maximum gap between detector actuations that will extend the active green indication. An alternative definition is that of the time that the phase will be extended for each actuation. When the time between actuations exceeds the allowable gap, the active phase will Gap-Out. *Passage Time* is typically set as the travel time from the set-back passage detector to the stop-bar, or the specified acceptable headway between vehicles for presence detectors located close to the stop-bar. The allowable gap can change over time if a Gap Reduction function is active.
- **Seconds per actuation:** This is the number of seconds the *Minimum Initial* is extended for each actuation during the yellow and red of that phase.
- **Gap Reduction** is a process that reduces the allowable gap from the specified *Passage Time* down to the specified *Minimum Gap*. This reduction process starts after a specified time (*Time Before Reduce*) that extends from the beginning of the current green phase or from a call to service from a conflicting movement (whichever occurs later), to the time that the gap reduction process begins. The gap is reduced from the passage time to the minimum gap, over a specified amount of time (*Time to Reduce*). Figure 5-1 depicts the allowable gap during the green time.

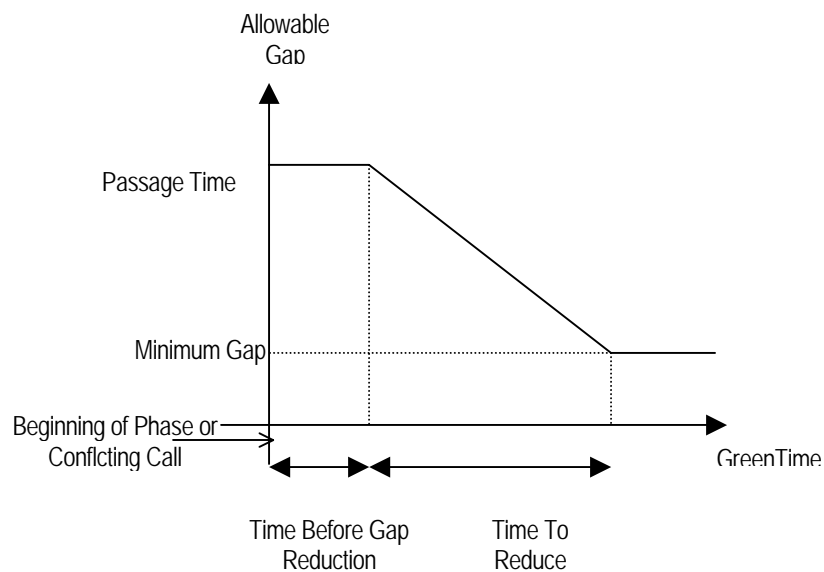


Figure 5-1: Allowable Gaps

- **Maximum Green Time** is the maximum limit to which the green time can be extended on a phase. This parameter is defined as *Maximum Duration* in AIMSUN. If the green phase extends to this maximum limit, then the phase terminates by Max-Out. The Maximum Green Time starts timing at the beginning of the green phase, or when a conflicting call is received; whichever is later.
- **Recall** is a feature whose purpose is to activate a specified phase. The available options are:
 - **No Recall:** An actuated phase will only be activated by a call; if not called (if no demand has been detected), it will be skipped. Once the phase is active, it can be

extended based on vehicle demand. The phase will end by Gap-out or Max-out

- **Minimum Recall:** The phase will be activated at least to its *Minimum Green* value even in the absence of an actuation (i.e. pedestrians are not detected but nevertheless are guaranteed enough crossing time). It can be extended based on vehicle demand and ends by Gap-out or Max-out.
- **Maximum Recall:** A constant vehicle call is applied to the phase; it extends to the *Maximum Green* (Max-out).
- **Coordinated Recall:** The coordinated or “synch” phase (see next point) will be activated every cycle. Its duration to the yield point will include any unused time from preceding phases. In presence of a coordinated phase, the offset is set with respect to this phase instead to the first phase.
- **Rest in Red and Hold:** The *Rest in Red* feature sets all signal indications to red in the absence of a call, then directly activates the phase servicing the next call. The *Hold* feature retains the existing green interval until it yields to a conflicting call. It is the opposite command to *rest in red*.
- **Detectors:** One of the advantages of actuated control is its ability to adjust signal timing parameters based on detector actuations that record actual vehicle demand. The detector parameters that control signal timing are:
 - *Locking/Non-Locking:* The Locking mode retains vehicle actuations until the appropriate phase receives the green interval. This detector counts vehicle actuations during the yellow and red time of the phase (to accomplish this, the detector must have been configured with the *count* capability), for example for extending the minimum initial green time. The *Non-Locking* mode, retains calls only while vehicles are in the zone of detection (The detector must be configured with *presence* capability). This is particularly useful when the detectors are located remotely. For volume-density operation, the locked mode must be selected.
 - *Delay Time:* the detector delays its call for this amount of time following a vehicle actuation. This is used in non-locking mode when we want to produce a signal as if the detector was in a position closer to the stop line.
 - *Extension Time:* This time is used to extend the call of the detector actuation. This is particularly useful for very short-call detectors. This is used in non-locking mode with those narrow detectors for which we want to produce a presence signal as if they were longer.

5.2 Coordinated Parameters

Without signal coordination, the cycle duration varies constantly depending on fluctuations in traffic demand. Coordinated traffic signal operation requires a fixed cycle length for each timing plan so that it is possible to synchronize a system of interconnected signal controllers. The parameters or elements that define a coordinated system are:

- **Coordinated phase:** The coordinated phase (recall set as *Coord*) is activated each cycle without reliance on detector actuations. This phase duration includes any unused time from preceding phases.
- **The yield point** (or “local time zero”) represents the point in the signal cycle that represents the end of the coordinated phase. The yield points of all controllers in the system are used to

coordinate them. All times in a controller are referenced to the yield point.

- **Force Offs:** For each of the non-coordinated phases there is a point in the cycle at which the phase must terminate in order to maintain a fixed cycle. Each force-off time must be computed so that the change period (yellow + all red), plus the minimum durations of all subsequent phases can be completed before the yield point.
- **Permissive Period** for an actuated non-coordinated phase is the time interval, referenced to the yield point, during which the controller can activate that actuated phase.
- **Offset** is the time distance from the beginning of the control plan to the yield point.

5.3 Multi-Ring controller

A Ring consists on 2+ sequentially timed conflicting phases that propagate in an established order. A controller may have a single ring (see an example in Figure 5-2), dual rings (see an example in Figure 5-3) or multi-rings. Another element is that of the *f* barrier. A barrier (compatibility line) is a reference point in the preferred sequence of a multi-ring controller at which all rings are interlocked. Barriers assure there will be no concurrent selection and timing of conflicting phases for traffic movements in different rings. All rings cross the barrier simultaneously to select and start at the same time the selected set of phases on the other side of the barrier. An option on Multi-Ring controllers is the *Dual Entry* or *Single Entry* operation:

- **Dual Entry** is a mode of operation in which one phase in each ring must be in service. If a call does not exist in one of the rings when the barrier is crossed, then a default phase is selected in that ring.
- **Single Entry** is a mode of operation in which a phase in one ring can be selected and timed alone if there is no demand for service in a non-conflicting phase on parallel rings.



Figure 5-2

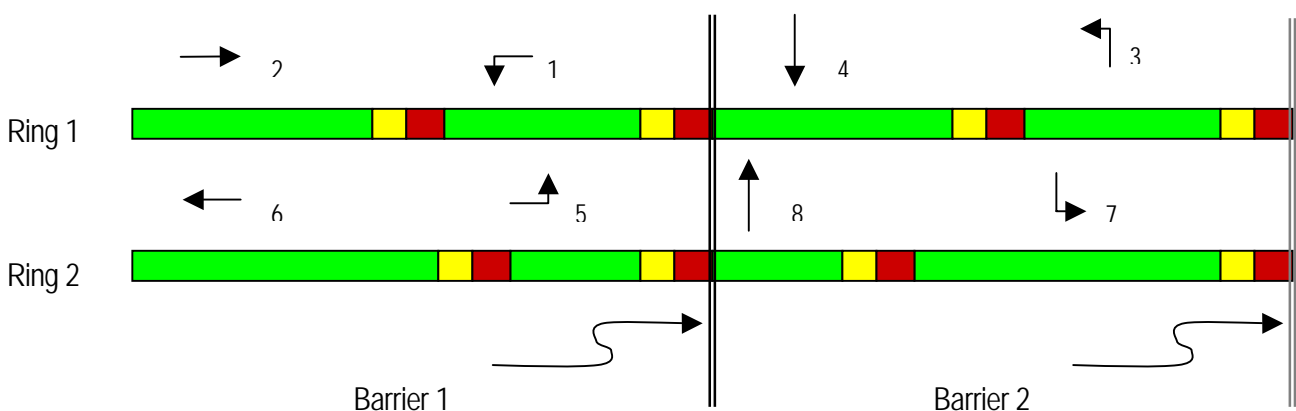


Figure 5-3

5.4 Actuated control definition

Each phase in an actuated control definition normally includes the green interval plus the yellow time and the all red time interval. This phase definition, in AIMSUN generates two phases, one related to the green time and the second one related to the yellow time and the all red interval defined as an interphase. The Table 5.1 depicts the AIMSUN phase's definition, corresponding to the example of Figure 5-2.

Phases Identifiers	AIMSUN phases Identifiers
1	1 + (5)
2	2 + (6)
3	3 + (7)
4	4 + (8)

Table 5.1

The table 5.2 shows the AIMSUN phases, corresponding to the multiring controller example of figure 3.2 (all phases in brackets represents interphases):

Phases Identifiers	AIMSUN phases Identifiers
1	1 + (9)
2	2 + (10)
3	3 + (11)
4	4 + (12)
5	5 + (13)
6	6 + (14)
7	7 + (15)
8	8 + (16)

Table 5.2

The definition of the actuated control in AIMSUN is phase oriented, that is defining the control as a set of different phases with their attributes. The phase attributes could be grouped in:

- Basics Phase definition
- Actuated Phase definition
- Detector Phase definition

5.5 Basics Phase Definition

The Basics Phase definition allows the user to define:

- **Interphase or not:** An interphase will be served always at the end of the previous phase and with a fixed duration. A turning maneuver can be active during the interphase. For instance, on the interphase of the first option of Figure 5-4 we can see how the west-east turning maneuver is kept active. On the other hand that same interphase should be an all-red clearance period in case the actuated system chooses the second option.

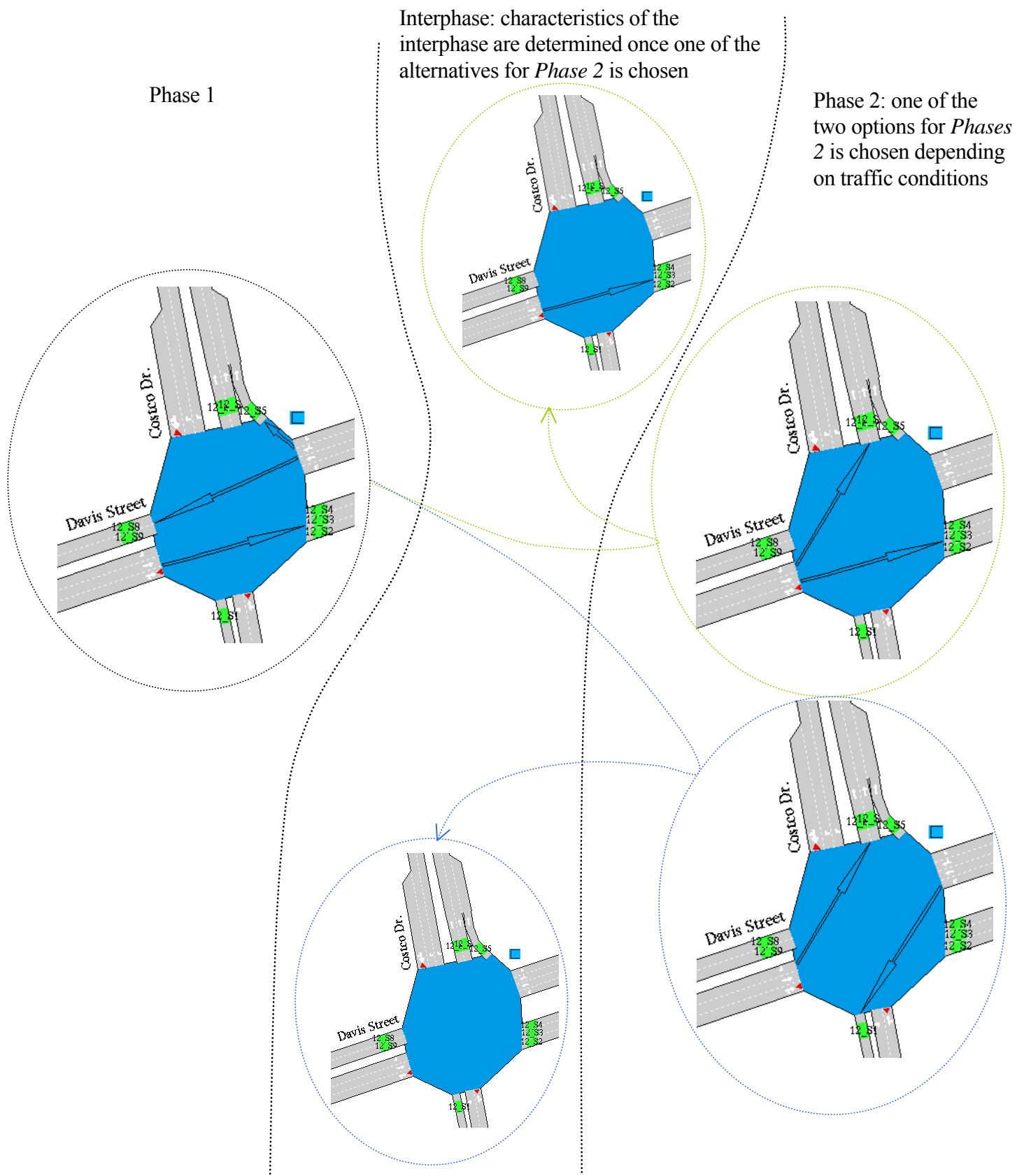


Figure 5-4: Actuated control interphase's dynamic profile

As explained in section 5.4, the user should activate during a phase/interphase only those turning movements that will always be allowed during each phase/interphase. Otherwise it will be the

software the one to calculate whether a turning movement takes place or not. For instance, in the example of Figure 5-4 it will be the system the one to decide whether the west-east turning maneuver is allowed or not during the interphase, and it will do so depending on whether phase 1 takes option 1 or option 2. So, when designing the control plan, you have to define that interphase without any turning maneuvers, and have no fear for missing maneuvers.

- **Minimum and Maximum Duration:** The minimum and maximum green time of the phase. For an interphase these parameters are disabled because an interphase has a fixed duration, not a variable one.
- **Duration:** This attribute is used to define the duration of an interphase.
- **Signal Groups assigned:** the active signal groups when the phase is served.

Control Plan: DUALRING

Id	Name	Tvov	Status
1			ACT

Junction

Type: ☒ Actuated ☐ Offset: 0.00 s.

Yellow Time: 0.00 secs ☐ Rest in Red ☐ Single Entry

Phases ☒ Preemption

Id	Tvov	Min	Duration	Max
1	P	4.00	6.00	6.00
2	P	16.00	46.00	46.00
3	P	4.00	11.00	11.00
4	P	16.00	21.00	21.00

Basics ☒ Actuated ☐ Detectors

Duration: 46.00 ☐ Interphase

Min: 16.00 Max: 46.00

Signal Groups

Active Signal Groups

Group Id
1
2
3
4
5
6

Active Turnings

From	To
1	3

Figure 5-5: dialogue window for the Basics Phase Definition

5.6 Actuated Phase Definition

The Actuated Phase definition allows the user to define the actuated parameters of each phase:

- **Recall:** The phase recall. It can be “No Recall”, “Minimum Recall”, “Maximum Recall” or “Coordinated” (this last is to indicate the coordinated phase)
- **Passage Time:** is the initial maximum time between actuations (gap or allowable headway) without losing the green.
- **Minimum Gap:** minimum acceptable vehicle gap specified by the user. It is the minimum value to which the allowable gap is reduced at the end of the Time to Reduce. All this just in case the *reduce by* feature is selected. Figure 5-1.

- **Time Before Reduce:** a preset time before the allowable gap begins to reduce. During this time period the gap's value is has a constant value equal to *passage time*. Figure 5-1.
- **Time to Reduce:** the amount of time that it takes to reduce the allowable gap from passage time to minimum gap. Figure 5-1.
- **Minimum gap, Time before reduce and Time to reduce** determine the gap reduction process. Figure 5-1.
- **Hold:** The Hold activated retains the existing green interval when there is no other detector actuation.
- **Force-Off an Permissive Period** determine extra parameters when the control is coordinated.
- **MultiRing:** In a multiring controller each phase has to be placed in one ring and barrier.
 - IdRing: Identifier of the ring to which the phase belongs (from 1 to NumRings)
 - IdBarrier: Id. of the barrier to which the phase belongs (from 1 to NumBarriers)
 - Position: Relative position of that phase in its ring (from 0 to NumPhasesInRing-1)
 - Default: Marks that phase as the default one for its ring and barrier.

Control Plan: DUALRING

Id	Name	Tvce	Status
1			ACT

Junction

Type: ☒ Actuated ☐ Offset: 0.00 s.

Yellow Time: 0.00 secs ☐ Rest in Red ☐ Single Entry

Phases ☒ Preemption

Id	Tvce	Min	Duration	Max
1	P	4.00	6.00	6.00
2	P	16.00	46.00	46.00
3	P	4.00	11.00	11.00
4	P	16.00	21.00	21.00

Basics ☒ Actuated ☐ Detectors

Parameters

Recall:

Passage Time: 3.00 Minimum Gap: 3.00

Time Before Reduce: 0.00 Time To Reduce: 0.00

Seconds Actuation: 0.00 Maximum Initial: 5.00

☐ Hold Force Off: 0.00

Permissive Period: From: 0.00 To: 0.00

MultiRings

Id Ring: 1 Id Barrier: 1 Position: 0 ☒ Default

Figure 5-6: actuated control parameters

5.7 Detector Phase Definition

The Detector Phase definition allows the user to define the associated detectors of each phase:

- **Locking:** Apart from detecting presence, this detector will count vehicles for the volume-density feature during the yellow and red time of a phase. In order to accomplish this, the detector must have been configured with the *count* capability.
- **Delay:** If detectors not located at the location of interest, this time is added to the moment of detection. This means that the system uses what would have been the moment at which a detector in the desired position would have emitted the signal
- **Extension:** This time is added to extend the call of the detector. This is particularly useful for very short-call detectors.

The Detector Phase Definition is defined in the AIMSUN GUI by means of the dialogue of Figure 5-7.

Control Plan: DUALRING

Id	Name	Type	Status
1		ACT	

Close

Junction

Type: **Actuated** Offset: 0.00 s. Show F.O. Calc

Yellow Time: 0.00 secs ☐ Rest in Red ☐ Single Entry

Phases Preemption

Id	Type	Min	Duration	Max
1	P	4.00	6.00	6.00
2	P	16.00	46.00	46.00
3	P	4.00	11.00	11.00
4	P	16.00	21.00	21.00

New Delete Up Down

Basics Actuated Detectors

Id	Locking	Delay	Extension
1	No	0.00	0.00

Remove

Locking: ☐ No Delay: Extension:

Figure 5-7: actuated control's detectors

5.8 Graphical Phase Diagram

During the control definition, TEDI allows the user to display the phase diagram of the junctions. This phase diagram contains the structure of phases (with their durations) and the state of each

signal group in the control cycle. Figure 5-8 depicts one phase diagram of a Dual Ring controller.

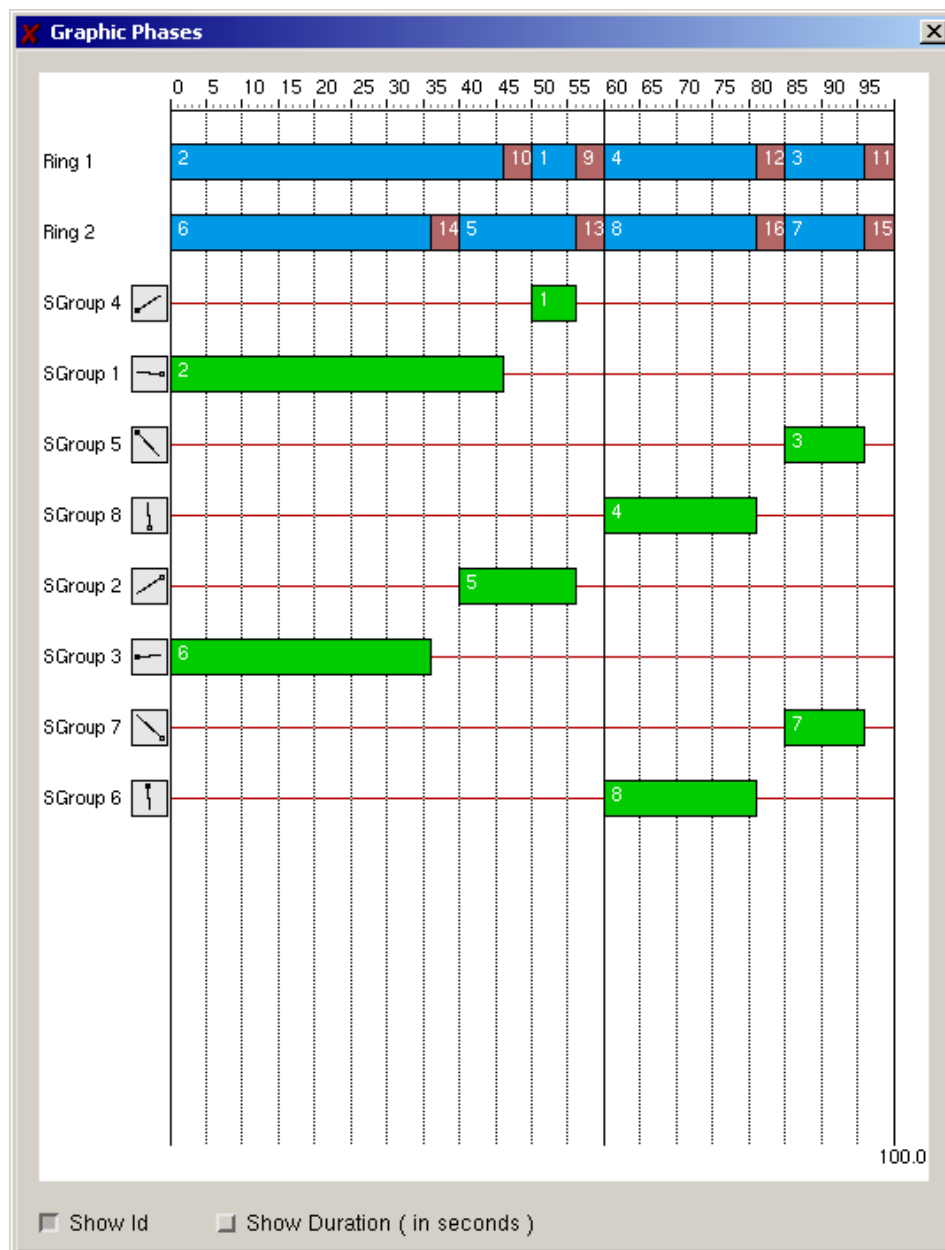


Figure 5-8: graphic phases

5.9 Bus Pre-emption

The bus pre-emption in AIMSUN adds a new element in the control (either actuated or pre-timed) that has to consider the arrival of Public Transport vehicles in order to anticipate the phase that give right of way. It takes into account the following elements:

- Public Transport Lines that activate a priority request, and
- Detectors that determines the position where the priority request is activated or finished.

5.9.1 Bus Pre-emption Parameters

The bus pre-emption definition has the following parameters:

PT Lines: Defines the Public Transport Lines that activate a priority request. The request is accepted if a public transport vehicle belongs to one of the PT Lines and is equipped. (The equipped vehicle is an attribute of the Vehicle Type)

Dwell Phase: The phase that gives right of way to the priority request.

Detector Request: Defines which detectors activate a priority request. All detectors have to have defined the “Equipped Vehicle” capability. When a Equipped Public Transport vehicle, which belongs to one of the PTLines defined, crosses the detector there is a priority request

Detector End: Defines which detectors finish priority request. All detectors have to have defined the “Equipped Vehicle” capability. When an Equipped Public Transport vehicle, which belongs to one of the PTLines defined, crosses the detector, the priority request is finished.

Delay Time is defined to delay the start of the pre-emption timing (this could be zero)

Inhibit Time defines the interval is to prevent any phase that is not the requested priority phase. This interval starts from the beginning of the request. The Inhibit Time has to be less or equal than the Delay Time.

Min Dwell defines the minimum duration of the Dwell phase. In a pre-timed control plan is not considered

Max Dwell defines the maximum duration of the Dwell phase. In a pre-timed control plan determines the duration of the Dwell phase.

Reserve Time defines the time to prevent other priority request when one has been completed. This is used to prevent frequent priority services

Type of algorithm could be “Serve All” or “Alternative”. “Serve All” serves all phases, with its minimum green time in the cycle before the Dwell phase. The “Alternative” serves Dwell phase as soon as possible.

Figure 5-9 depicts the corresponding TEDI dialogue window.

Control Plan: DUALRINGPreOnePha

Id	Name	Type	Status
1		ACT	

Close

Junction

Type: Offset: s.

Yellow Time: secs ☐ Rest in Red ☐ Single Entry

Phases

Preemption Set:

PT Line	Del
1	

Phase	Add	Type	Detector	Del
2	<input type="button" value="Del"/>	Req. 6		
		End 10		

Phase: Type:

Parameters

Delay: <input type="text" value="5.00"/>	Inhibit: <input type="text" value="5.00"/>
Min Dwell: <input type="text" value="5.00"/>	Max Dwell: <input type="text" value="20.00"/>
Reserve: <input type="text" value="10.00"/>	Type: <input type="button" value="Alternative"/>

Figure 5-9: bus pre-emption

6 FUNCTIONS AVAILABLE IN THE FUNCTION EDITOR

The function editor allows to the user define three different types of functions:

- Basic Cost Functions
- Cost Functions considering Vehicle Types
- Route Choice Functions

N.B. In the following descriptions, all measurements have been expressed using the metric system. If the network has been built using English system units, the user should read miles instead of kilometres and feet instead of meters.

6.1 Basic Cost Functions

Basic Cost Functions do not distinguish between vehicle types and therefore can not make use of variables that have any vehicle type reference, so all vehicle types will have a common cost. The parameters for this type of function are S (section reference) and T (turning reference: turnings that have S as origin section).

Gives the cost of a turning movement common to all vehicles (aggregating costs for each vehicle type).

Signature: **FunctionName** (S, T) → *double*

where: **S** is a section reference.

T is a turning reference.

Function Body: Defined as a numerical expression using the following items:

- Basic items and arithmetic operators:
 - **numerical constants**.
 - **+**: Addition.
 - **-**: Subtraction.
 - *****: Multiplication
 - **/**: Division
 - **ABS**: absolute value
 - **^**: power
- Logic items: Give a logical expression and can be used in the **IF(a;b;c)** conditional statement.

- **a==b**: Equal to (where a and b are numerical expressions).
 - **a<b**: Less than (where a and b are numerical expressions).
 - **a<=b**: Less than or equal to (where a and b are numerical expressions).
 - **a>b**: Greater than (where a and b are numerical expressions).
 - **a>=b**: Greater than or equal to (where a and b are numerical expressions).
 - **a&b**: Logical And (where a and b are logical expressions).
 - **a|b**: Logical Or (where a and b are logical expressions).
 - **!a**: Logical negation (where a is a logical expression).
- Mathematic functions:
- **SIN(a)**: Sine of numerical expression a.
 - **COS(a)**: Cosine of numerical expression a.
 - **TAG(a)**: Tangent of numerical expression a.
 - **ASIN(a)**: Inverse Sine of numerical expression a.
 - **ACOS(a)**: Inverse Cosine of numerical expression a.
 - **ATAN(a)**: Inverse Tangent.
 - **SINH(a)**: Hyperbolic Sine of numerical expression a.
 - **COSH(a)**: Hyperbolic Cosine of numerical expression a.
 - **TAGH(a)**: Hyperbolic Tangent of numerical expression a.
 - **SQRT(a)**: Square root of numerical expression a.
 - **EXP(a)**: Exponential of numerical expression a.
 - **LOG(a)**: Base 10 logarithm of numerical expression a.
 - **LN(a)**: Natural logarithm of numerical expression a.
 - **MIN(a,b)**: Minimum between two numerical expressions a and b.
 - **MAX(a,b)**: Maximum between two numerical expressions a and b.
- Conditional:
- **IF(a;b;c)**: Conditional statement that returns the numerical expression b if the logical expression a is true, otherwise it returns the numerical expression c.
- Variables:
- When one of the components of a function's name is 'Turn' the function may return information on the *T* turning movement or information on the turn and a certain road length previous to the turning movement. In this second case we will refer to 'Link (*S,T*)' in the function's description (see the AIMSUN2 User Manual for a detailed description of link capacity).

- Random:
 - **Rand(S)**: Random number between 0 and 1 generated using the seed of Section *S*.
- Network Description:
 - **NetMaxSectCapacity()**: Returns the Maximum Section Capacity of all sections of the Network.
 - **NetMaxTurnCapacity()**: Maximum Link Capacity of the Network.
- Section Description
 - **SectMaxSpeed(S)**: Maximum Speed of Section *S* expressed in Km/h. If section *S* belongs to a polysection then it calculates the average of the maximum speeds of all sections that belong to the same polysection.
 - **SectCapacity(S)**: Total Capacity of Section *S*. If section *S* belongs to a polysection then it returns the capacity of the last section of the polysection.
 - **SectLength(S)**: Length of Section *S* expressed in meters. If section *S* belongs to a polysection then it returns sum of the length of all the sections that belong to the polysection.
 - **SectSlope(S)**: Slope percentage of Section *S*. If section *S* belongs to a polysection then it returns the average slope percentage of all sections that belong to the polysection, adjusting each section's weight according to its length.
 - **SectUserDefinedCost(S)**: User defined cost of Section *S*. If section *S* belongs to a polysection then it returns the sum of the user defined costs of all sections that belong to the polysection.
- Turning Movement Description
 - **TurnMaxSpeed(S, T)**: Maximum Speed of Turning *T* expressed in Km/h.
 - **TurnCapacity(S, T)**: Calculates the Total Capacity of Link (*S, T*).
 - **TurnLength(S, T)**: Length of Turning *T* expressed in meters.
- Statistical Section Data (aggregating all vehicle types). When talking about means it refers to a temporal mean during a time period defined by the route choice cycle and the number of intervals that it considers (*cycle times number of intervals*). For more details AIMSUN user manual.
 - **SectStaFlow(S)**: Mean Statistical Flow of Section *S* expressed in veh/h.
 - **SectStaDensity(S)**: Mean Statistical Density of Section *S* expressed in veh/Km. If section *S* belongs to a polysection it returns the average density of all sections that belong to that same polysection, adjusting each section's weight according to its length.
 - **SectStaSpeed(S)**: Mean Statistical Speed of Section *S* expressed in Km/h. If section *S* belongs to a polysection it returns the average speed of all sections that belong to that same polysection, adjusting each section's weight according to its length.
 - **SectStaTravelT(S)**: Mean Statistical Travel Time of Section *S* expressed in seconds. If section *S* belongs to a polysection then it returns the sum of the travel times of all the sections that belong to the same polysection. The function returns the

statistical travel time as long as at least one vehicle has driven along the whole section. If the section has remained empty for the whole period it returns the travel time in free flow conditions. If no vehicles have driven along the section because all vehicles on the section have remained stopped it returns the delay time of the first vehicles of the section.

- **SectStaDelayT(S)**: Mean Statistical Delay Time of Section *S* expressed in seconds. If section *S* belongs to a polysection then it returns the sum of the delay times of all the sections that belong to that same polysection.
 - **SectStaStopT(S)**: Mean Statistical Stop Time of Section *S* expressed in seconds. If section *S* belongs to a polysection then it returns the sum of the stop times of all the sections that belong to that same polysection.
 - **SectStaNbStopsVeh(S)**: Mean Statistical Number of Stops in Section *S*. If section *S* belongs to a polysection then it returns the sum of the number of stops of all sections that belong to that same polysection.
 - **SectStaQueueLMean(S)**: Mean Queue Length per Lane in Section *S*. If section *S* belongs to a polysection then it returns the sum of the mean queue lengths for all sections that belong to that same polysection.
 - **SectStaQueueLMax(S)**: Maximum Queue Length per Lane in Section *S*. If section *S* belongs to a polysection then it returns the sum of maximum queue length of all sections that belong to the same polysection.
- Statistical Turning Data (aggregating all vehicle types):
- **TurnStaFlow(S, T)**: Mean Statistical Flow of Turning *T* expressed in veh/h.
 - **TurnStaSpeed(S, T)**: Mean Statistical Speed of Link (*S,T*) expressed in Km/h. If section *S* belongs to a polysection then it considers the average speed of all sections that belong to the same polysection.
 - **TurnStaTravelT(S, T)**: Mean Statistical Travel Time of Link (*S,T*) expressed in seconds. If section *S* belongs to a polysection then adds the sum of the travel times for all sections that belong to that same polysection. The function returns the statistical travel time as long as at least one vehicle has driven along the whole section. If the section has remained empty for the whole period it returns the travel time in free flow conditions. If no vehicles have driven along the section because all vehicles on the section have remained stopped it returns the delay time of the first vehicles of the section.
 - **TurnStaDelayT(S, T)**: Mean Statistical Delay Time of Link (*S,T*) expressed in seconds. If section *S* belongs to a polysection then it adds the sum of the delay times for all sections that belong to that same polysection.
 - **TurnStaStopT(S, T)**: Mean Statistical Stop Time of Link (*S,T*) expressed in seconds. If section *S* belongs to a polysection then it adds the sum of stop times for all sections that belong to the same polysection.
 - **TurnStaNbStopsVeh(S, T)**: Mean Statistical Number of Stops in Link (*S,T*). If section *S* belongs to a polysection then adds the sum of number of stops for all sections that belong to the same polysection.
 - **TurnStaQueueLMean(S, T)**: Mean Queue Length per Lane in Link (*S,T*). If section *S* belongs to a polysection then adds the sum of mean queue lengths for all

sections that belong to the same polysection.

- **TurnStaQueueLMax(S, T):** Maximum Queue Length per Lane in Link (S,T). If section S belongs to a polysection then adds the sum of maximum queue lengths for all sections that belong to the same polysection.
- Previous Turning Cost:
 - **TurnCostInitial(S, T):** Initial Cost in Link (S,T), when vehicle types are not considered. In case the turning has an initial cost function per vehicle type, then it returns a -1.
 - **TurnPastCost(S, T, offset):** Past Cost in link (S,T). Past cost are link cost stored in previous simulations. If parameter offset is 0 then takes the past costs stored at the same time. Therefore for a positive offset takes the next past costs stored and for a negative values takes the previous past costs respect the current time.

Examples: Example of default initial cost function used in AIMSUN2, when no distinction is made between different vehicle types:

$$\text{InitialCost1 (S, T)} = (\text{SectLength(S)} / (\text{SectMaxSpeed(S)} / 3.6)) \\ + (\text{TurnLength(S, T)} / (\text{TurnMaxSpeed(S, T)} / 3.6))$$

Example of default initial cost function used in AIMSUN2, when no distinction is made between different vehicle types and with capacity weight 2:

$$\text{InitialCost2 (S, T)} = \text{TurnStaTravelT(S,T)} + \text{TurnStaTravelT(S,T)} * \\ 2.0 * (1 - (\text{TurnCapacity(S,T)} / \text{NetMaxSectCapacity()}))$$

Example of default cost function used in AIMSUN2, when no distinction is made between different vehicle types:

$$\text{CurrentlCost1 (S, T)} = \text{Max}(\text{TurnStaTravelT(S,T)}, \text{TurnCostInitial(S,T)})$$

Example of default cost function used in AIMSUN2, when no distinction is made between different vehicle types and with capacity weight 2:

$$\text{CurrentlCost2 (S, T)} = \text{Max}(\text{TurnStaTravelT(S,T)} + \text{TurnStaTravelT(S,T)} * \\ 2.0 * (1 - (\text{TurnCapacity(S,T)} / \text{NetMaxSectCapacity()})), \text{TurnCostInitial(S,T)})$$

6.2 Cost Functions considering Vehicle Types

Cost Functions considering Vehicle Type can distinguish between vehicle types and consequently can make use of variables that have vehicle type reference. The parameters for this type of function are S (section reference), T (turning reference, this turning has as origin section S) and VT (vehicle type reference).

Gives the cost of a turning movement, taking into account the vehicle type.

Signature: **FunctionName** (**S**, **T**, **VT**) → *double*

where: **S** is a section reference.

T is a turning reference.

VT is a vehicle type reference.

Body: Defined as a numerical expression using the following items:

- ❑ Basic items (see Basic Cost Functions)
- ❑ Logic items (see Basic Cost Functions)
- ❑ Mathematics functions (see Basic Cost Functions)
- ❑ Conditional (see Basic Cost Functions)
- ❑ Variables:
 - Random (see Basic Cost Functions)
 - Network Description (see Basic Cost Functions)
 - Section Description (see Basic Cost Functions)
 - Turning Movement Description (see Basic Cost Functions)
 - Vehicle Type Description:
 - **VehTypeLength(VT)**: Mean Length of Vehicle Type *VT* expressed in meters
 - **VehTypeWidth(VT)**: Mean Width of Vehicle Type *VT* expressed in meters
 - **VehTypeMaxSpeed(VT)**: Mean Maximum Desired Speed of Vehicle Type *VT* expressed in Km/h
 - **VehTypeMaxAccel(VT)**: Mean Maximum Acceleration of Vehicle Type *VT* expressed in m/s^2 .
 - **VehTypeDecel(VT)**: Mean Normal Deceleration of Vehicle Type *VT* expressed in m/s^2 .
 - **VehTypeMaxDecel(VT)**: Mean Maximum Deceleration of Vehicle Type *VT* expressed in m/s^2 .
 - **VehTypeSpeedAc(VT)**: Mean Speed Acceptance of Vehicle Type *VT*.
 - **VehTypeGiveWayTime(VT)**: Mean Give Way Time of Vehicle Type *VT*.
 - **VehTypeGuidanceAc (VT)**: Mean Guidance Acceptance of Vehicle Type *VT*.
 - Check Vehicle Type:

- **CheckVehType(VT, name):** Return 1 if VT belongs to vehicle type with name string.
- Statistical Section Data (is possible to use the statistical section data aggregating all vehicle types, see Basic Cost Functions, and the following distinguishing vehicle types)
 - **SectStaFlow(S, VT):** Mean Statistical Flow of Section *S* of Vehicle Type *VT* expressed in veh/h.
 - **SectStaDensity(S, VT):** Mean Statistical Density of Section *S* of Vehicle Type *VT* expressed in veh/Km. If the section *S* belongs to a polysection then gives the average of density of all sections that belong to the same polysection.
 - **SectStaSpeed(S, VT):** Mean Statistical Speed of Section *S* of Vehicle Type *VT* expressed in Km/h. If the section *S* belongs to a polysection then gives the average of speed of all sections that belong to the same polysection.
 - **SectStaTravelT(S, VT):** Mean Statistical Travel Time of Section *S* of Vehicle Type *VT* expressed in seconds. If the section *S* belongs to a polysection then gives the sum of travel time of all sections that belong to the same polysection. It gives the statistical travel time in the case of some vehicle has cross the section, otherwise it gives the travel time in free flow conditions.
 - **SectStaDelayT(S, VT):** Mean Statistical Delay Time of Section *S* of Vehicle Type *VT* expressed in seconds. If the section *S* belongs to a polysection then gives the sum of delay time of all sections that belong to the same polysection.
 - **SectStaStopT(S, VT):** Mean Statistical Stop Time of Section *S* of Vehicle Type *VT* expressed in seconds. If the section *S* belongs to a polysection then gives the sum of stop time of all sections that belong to the same polysection.
 - **SectStaNbStopsVeh(S, VT):** Mean Statistical Number of Stops in Section *S* of Vehicle Type *VT*. If the section *S* belongs to a polysection then gives the sum of number of stops of all sections that belong to the same polysection.
 - **SectStaQueueLMean(S, VT):** Mean Queue Length per Lane in Section *S* of Vehicle Type *VT*. If the section *S* belongs to a polysection then gives the sum of mean queue length of all sections that belong to the same polysection.
 - **SectStaQueueLMax(S, VT):** Maximum Queue Length per Lane in Section *S* of Vehicle Type *VT*. If the section *S* belongs to a polysection then gives the sum of maximum queue length of all sections that belong to the same polysection.
- Statistical Turning Data (is possible to use the statistical turning data aggregating all vehicle types, see Basic Cost Functions, and the following distinguishing vehicle types)
 - **TurnStaFlow(S,T, VT):** Mean Statistical Flow of Turning *T* expressed in veh/h.
 - **TurnStaSpeed(S, T, VT):** Mean Statistical Speed of Turning *T* of Vehicle Type *VT* expressed in Km/h. If the section *S* belongs to a polysection then considers the average of speed of all sections that belong to the same polysection.
 - **TurnStaTravelT(S, T, VT):** Mean Statistical Travel Time of Turning *T* of Vehicle Type *VT* expressed in seconds. If the section *S* belongs to a polysection then adds the sum of travel time of all sections that belong to the same polysection. It gives the statistical travel time in the case of some vehicle has cross the section, otherwise it gives the travel time in free flow conditions.

- **TurnStaDelayT(S, T, VT):** Mean Statistical Delay Time of Turning *T* of Vehicle Type *VT* expressed in seconds. If the section *S* belongs to a polysection then adds the sum of delay time of all sections that belong to the same polysection.
 - **TurnStaStopT(S, T, VT):** Mean Statistical Stop Time of Turning *T* of Vehicle Type *VT* expressed in seconds. If the section *S* belongs to a polysection then adds the sum of stop time of all sections that belong to the same polysection.
 - **TurnStaNbStopsVeh(S, T, VT):** Mean Statistical Number of Stops in Turning *T* of Vehicle Type *VT*. If the section *S* belongs to a polysection then adds the sum of number of stops of all sections that belong to the same polysection.
 - **TurnStaQueueLMean(S, T, VT):** Mean Queue Length per Lane in Turning *T* of Vehicle Type *VT*. If the section *S* belongs to a polysection then adds the sum of mean queue length of all sections that belong to the same polysection.
 - **TurnStaQueueLMax(S, T, VT):** Maximum Queue Length per Lane in Turning *T* of Vehicle Type *VT*. If the section *S* belongs to a polysection then adds the sum of maximum queue length of all sections that belong to the same polysection.
- Previous Turning Cost:
- **TurnCostInitial(S, T, VT):** Initial Cost in Turning *T* of Vehicle Type *VT*. In the case the turning has a initial cost function not considering vehicle types, then gives the common cost.
 - **TurnPastCost(S, T, VT, offset):** Past Cost in link (*S,T*) of Vehicle Type *VT*. Past cost are link cost stored in previous simulations. If parameter offset is 0 then takes the past costs stored at the same time. Therefore for a positive offset takes the next past costs stored and for a negative values takes the previous past costs respect the current time.

Examples:

Example of default initial cost function used in AIMSUN2 distinguishing vehicle types:

$$\begin{aligned}
 \text{InitialCost3} (S, T, VT) = & (\text{SectLength}(S)/(\min(\text{SectMaxSpeed}(S)*\text{VehTypeSpeedAc}(VT), \\
 & \text{VehTypeMaxSpeed}(VT))/3.6)) \\
 & + (\text{TurnLength}(S,T)/(\min(\text{TurnMaxSpeed}(S,T)*\text{VehTypeSpeedAc}(VT), \\
 & \text{VehTypeMaxSpeed}(VT))/3.6))
 \end{aligned}$$

Example of default initial cost function used in AIMSUN2, distinguishing vehicle types and taking the capacity weight as 2:

$$\begin{aligned}
 \text{InitialCost4}(S, T, VT) = & \text{TurnStaTravelT}(S,T,VT) + \text{TurnStaTravelT}(S,T,VT)* \\
 & 2.0*(1-(\text{TurnCapacity}(S,T)/\text{NetMaxSectCapacity}()))
 \end{aligned}$$

Example of default cost function used in AIMSUN2, when no vehicle types are considered:

$$\text{CurrentlCost3} (S, T, VT) = \text{Max}(\text{TurnStaTravelT}(S,T,VT), \text{TurnCostInitial}(S,T,VT))$$

Example of default cost function used in AIMSUN2, when no vehicle types are considered and taking the capacity weight as 2:

$$\text{CurrentlCost4} (S, T, VT) = \text{Max}(\text{TurnStaTravelT}(S,T,VT) + \text{TurnStaTravelT}(S,T,VT) * 2.0 * (1 - (\text{TurnCapacity}(S,T) / \text{NetMaxSectCapacity}())), \text{TurnCostInitial}(S,T,VT))$$

Example of cost function using a conditional statement to distinguish a different user defined cost :

$$\begin{aligned} \text{CurrentlCost5} (S, T, VT) = & \text{IF}(\text{CheckVehType}(VT, \text{car}) > 0; \\ & \text{TurnStaTravelTVT}(S,T,VT) + \text{SectUserDefinedCost}(S); \\ & \text{TurnStaTravelTVT}(S,T,VT) + \text{SectUserDefinedCost}(S) + 10) \end{aligned}$$

6.3 Route Choice Function

Gives the probability for each one of the n shortest paths to be chosen by vehicles, taking into account the vehicle type.

Signature: *FunctionName* (**O**, **D**, **TR**, **S**, **N**, **F**, **I**, **VT**) → *double*

where:

- O** is an origin centroid reference
- D** is a destination centroid reference
- TR** is a set of SP (shortest path) reference
- S** is a section reference.
- N** is the number of SP to consider.
- F** is the index of the first SP to consider.
- I** is the index of the SP to evaluate (from 0 to N-1).
- VT** is a vehicle type reference.

Function Body: Defined as a numerical expression using the following items:

- ❑ Basic items (see Basic Cost Functions)
- ❑ Logic items (see Basic Cost Functions)
- ❑ Mathematics functions (see Basic Cost Functions)
- ❑ Conditional (see Basic Cost Functions)
- ❑ Aggregation:

- **SUM(J, ini, end, exp)** : Gives the total sum of numerical expression *exp*, changing the parameter *J* from the integer *ini* to the integer *end*. An equivalent algebraic formulae

$$\text{is: } \sum_{J=ini}^{end} expr$$

- **PROD(J, ini, end, exp)** : Gives the total product of numerical expression *exp*, changing the parameter *J* from the integer *ini* to the integer *end*. An equivalent algebraic formulae

$$\text{is: } \prod_{J=ini}^{end} expr$$

□ Variables:

- Check Vehicle Type (see Cost Functions considering Vehicle Types)
- Characteristics of Shortest Path:
 - **SPCurrCost(O, TR, S, K, VT)**: Current Cost of Shortest Path *K* of Vehicle Type *VT* from Section *S*. The value of *K* has to be defined between *F* and *F+N-1*, where *F* and *N* are parameters of Route Choice Functions.
 - **SPCurrTravelTime(O, TR, S, K, VT)**: Current travel time of Shortest Path *K* of Vehicle Type *VT* from Section *S*. The value of *K* has to be defined between *F* and *F+N-1*, where *F* and *N* are parameters of Route Choice Functions.
 - **SPDist(O, TR, S, K, VT)**: Physical Distance of Shortest Path *K* of Vehicle Type *VT* from Section *S*. The value of *K* has to be defined between *F* and *F+N-1*, where *F* and *N* are parameters of Route Choice Functions.
 - **SPExpTravelTime(O, TR, S, K, VT)**: Experimented travel time of Shortest Path *K* of Vehicle Type *VT* from Entrance Section *S*. The value of *K* has to be defined between *F* and *F+N-1*, where *F* and *N* are parameters of Route Choice Functions.
 - **SPCommonDist(O, TR, S, K, L, VT)**: Common Physical Distance between Shortest Path *K* and Shortest Path *L* considering Vehicle Type *VT* from Section *S*. The value of *K* and *L* have to be defined between *F* and *F+N-1*, where *F* and *N* are parameters of Route Choice Functions.
 - **SPCommonCost(O, TR, S, K, L, VT)**: Common Cost between Shortest Path *K* and Shortest Path *L* considering Vehicle Type *VT* from Section *S*. The value of *K* and *L* have to be defined between *F* and *F+N-1*, where *F* and *N* are parameters of Route Choice Functions.
 - **SPPastCost (O, TR, S, K, VT, T)**: Past Cost of Shortest Path *K* of Vehicle Type *VT* from Section *S* at interval *T* (if equal to 0, is the past cost considering the current interval, equal to 1 is the past cost of the next interval, equal to -1 is the past cost of the previous interval and so on). The value of *K* has to be defined between *F* and *F+N-1*, where *F* and *N* are parameters of Route Choice Functions.

Example of Logit model with an scale factor of 60:

$$RC1 (O, D, TR, S, N, F, I, VT) = 1 / \text{SUM}(J, 0, N-1, \text{EXP}(-60 * (\text{SPCurrCost}(TR, S, F+J, VT) / 3600.0 - \text{SPCurrCost}(TR, S, F+I, VT) / 3600.0)))$$

Example of Route Choice Model where the cars follow a Logit model with an scale factor of 60 and the other vehicle follow a uniform distribution:

$$RC2 (O, D, TR, S, N, F, I, VT) = \text{IF}(\text{CheckVehType}(VT, \text{car}) > 0; 1 / \text{SUM}(J, 0, N-1, \text{EXP}(-60 * (\text{SPCurrCost}(O, TR, S, F+J, VT) / 3600.0 - \text{SPCurrCost}(O, TR, S, F+I, VT) / 3600.0))) ; 1/N)$$

Example of Logit model with an scale factor of 60 and using the past cost of the current interval:

$$RC3 (O, D, TR, S, N, F, I, VT) = 1 / \text{SUM}(J, 0, N-1, \text{EXP}(-60 * (\text{SPPastCost}(O, TR, S, F+J, VT, 0) / 3600.0 - \text{SPPastCost}(O, TR, S, F+I, VT, 0) / 3600.0)))$$

Example of Logit model with an scale factor of 60 and using a combination of current cost, past cost of the current and next interval:

$$RC4 (O, D, TR, S, N, F, I, VT) = 1 / \text{SUM}(J, 0, N-1, \text{EXP}(-60 * ((0.5 * \text{SPCurrCost}(O, TR, S, F+J, VT) + 0.25 * \text{SPPastCost}(O, TR, S, F+J, VT, 0) + 0.25 * \text{SPPastCost}(O, TR, S, F+J, VT, 1)) / 3600.0 - (0.5 * \text{SPCurrCost}(O, TR, S, F+I, VT) + 0.25 * \text{SPPastCost}(O, TR, S, F+I, VT, 0) + 0.25 * \text{SPPastCost}(O, TR, S, F+I, VT, 1)) / 3600.0)))$$

7 ASCII FORMATS

7.1 Result Container & Traffic States

7.1.1 Format of the file *ResultAscii*

Result_name

Number_of_vehicle_types

Vehicle_Type_1 (*name of vehicle type*)

Vehicle_Type_2

...

Vehicle_Type_N

Number_of_states

State_name_1 Initial_Time_State_1 (*name and initial time of the traffic state*)

State_name_2 Initial_Time_State_2

...

State_name_N Initial_Time_State_N

Entrance_Section1

Flow_Veh1_State_1 Flow_Veh2_State_1 ...

Flow_Veh1_State_2 Flow_Veh2_State_2 ...

...

Entrance_Section2

Flow_Veh1_State_1 Flow_Veh2_State_1 ...

Flow_Veh1_State_2 Flow_Veh2_State_2 ...

...

0 → *The '0' is the signal that indicates the end of the information on entrance sections.*

Next: percentage of vehicles from SectionAt to SectionTo for each vehicle type and during each state.

SectionAt1 SectionTo1

Percentatge_Veh1_State1 Percentatge_Veh2_State1...

Percentatge_Veh1_State2 Percentatge_Veh2_State2...

...

SectionAt2 SectionTo2

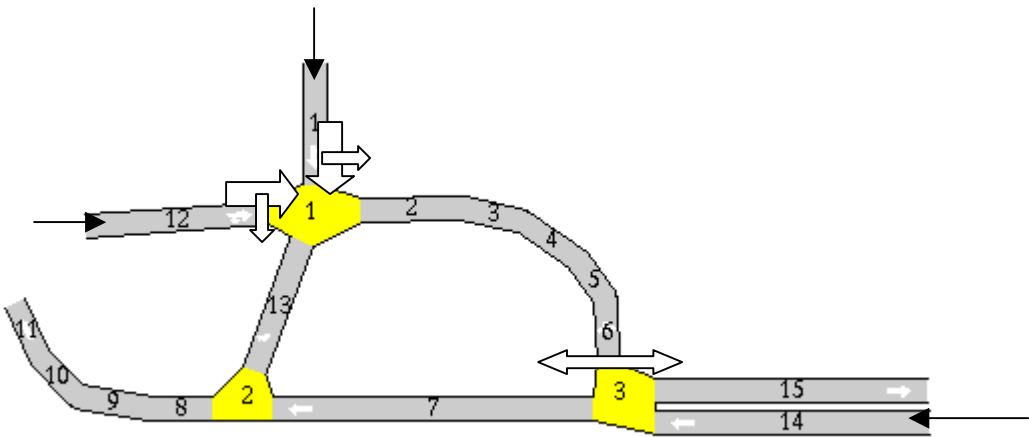
Percentatge_Veh1_State1 Percentatge_Veh2_State1...

Percentatge_Veh1_State2 Percentatge_Veh2_State2...

...

0 → *The '0' is the signal that indicates the end of the information on turning percentages.*

7.1.2 Example for the TestResult network



ResultAscii
2
car
bus
2
state1 080000
state2 090000
1
200 50
230 80
12
300 70
330 100
14
400 80
430 110
0
1 2
0.2 0.3
0.2 0.3
1

1	13
0.8	0.7
0.8	0.7
6	7
0.4	0.3
0.4	0.3
6	15
0.6	0.7
0.6	0.7
12	2
0.8	0.6
0.8	0.6
12	13
0.2	0.4
0.2	0.4
2	

7.2 Origin-Destination Matrix

7.2.1 Format of the file *MatrixAscii*

Ascii Matrix Format (Tabular file) for a matrix with M vehicle types and N time slices:

Id_Time_Interval_1 Duration_Time_Interval_1 Vehicle_Type_1

MATRIX (this matrix is for the defined interval and vehicle type, and the format is the standard one for matrices: rows and columns for origins and destination)

Id_Time_Interval_2 Duration_Time_Interval_2 Vehicle_Type_1

MATRIX (following matrix is the one for the same vehicle type but for the next time interval)

...

Id_Time_Interval_N Duration_Time_Interval_N Vehicle_Type_1

MATRIX (last matrix for Vehicle_Type_1, so we have N time slices)

Id_Time_Interval_1 Duration_Time_Interval_1 Vehicle_Type_2

MATRIX (the same list of matrices, that is N matrices for the N time slices, begins again, but now for Vehicle_Type_2)

Id_Time_Interval_2 Duration_Time_Interval_2 Vehicle_Type_2

MATRIX

...

Id_Time_Interval_N Duration_Time_Interval_N Vehicle_Type_2

MATRIX

...

Id_Time_Interval_1 Duration_Time_Interval_1 Vehicle_Type_M

MATRIX (the list of N matrices for each vehicle type is repeated until the last vehicle type)

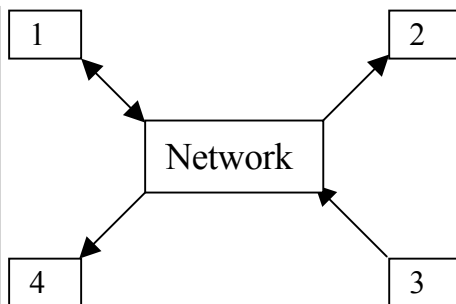
...

Id_Time_Interval_N Duration_Time_Interval_N Vehicle_Type_M

MATRIX

Example:

1	15	bus	
2	4	1	
1	74	1.8	
3	12.7		22.2
2	15	bus	
2	4	1	
1	81.6	1.8	
3	16.8		18.9
1	15	car	
2	4	1	
1	1294	32.2	
3	221.2		398.3
2	15	car	
2	4	1	
1	1429.2	30.8	
3	294.7		445.2
1	15	truck	
2	4	1	
1	147.9	3.7	
3	25.3		43.1
2	15	truck	
2	4	1	
1	163.3	3.5	
3	33.7		51.0



8 QUICK REFERENCE GUIDE TO TEDI COMMANDS

8.1.1 File menu

New: Opens a new network.

Open... Opens an existing network.

Close: Closes the currently open network.

Save: Saves the currently open network.

Save as... Saves the currently open network, with the option to provide a new name and format.

Print: Sends print file to the peripheral.

Import: Opens the dialog window to set the parameters for importing an EMME/2 network.

Export... Exports the network as an EMME2 model. This option is available only in those versions equipped with the EMME/2 interface (GETRAM kit 2).

Report Generation... Generates a report of possible errors and on the geometry of the network.

Quit: Exits the application.

8.1.2 Edit menu

Undo: Undoes last translation/rotation.

Redo: Redoes last translation/rotation.

Cut: Removes the selected objects and places them on the clipboard.

Copy: Copies the selected objects to the clipboard.

Paste: Copies from the clipboard to the drawing area.

Clear: Deletes the selected objects.

Select all: Selects all the objects in the drawing area.

Select Inverse: Unselects all selected objects and selects all those that were not selected.

Unselect all: Unselects all selected objects.

Find...: Finds objects in the drawing area.

Vehicle classes... Specifies the classes of vehicle types used in the network.

Vehicle Types library....: Library with predefined vehicle types.

Function Editor.....: Opens the dialog window of the editor of route choice and cost functions.

Preferences....: Specifies the overall parameters of the traffic network currently being edited.

8.1.3 View menu

Turn autogrid on: This allows the user to toggle between an active and inactive drawing grid. Activating the grid means that, when creating new objects, end points will be restricted to the grid.

Show grid: Toggles between hiding and displaying grid points.

Show line pointer: Shows the two lines whose intersection defines the tip of the cursor. Changes to "Hide line pointer".

Hide distance: Hides the 'Message area'. Changes to "Show distance".

Show objects: Displays a modeless dialog window where the user can specify, for each object, whether to display it and/or its identifiers.

Show section speed: Displays a sign with the top speed permitted on each section.

Background... Sets the background pixmap to the contents of a set of DXF files.

Whole drawing: Zooms out to the scale in which the whole drawing area can be seen.

8.1.4 Arrange menu

Move forward: Moves the selected objects forward. This means increasing the priority of these objects, i.e., showing them in front of other objects.

Move backward: Moves the selected objects backward. This means decreasing the priority of these objects, i.e., showing other objects in front of them.

Align Centres: Moves the selected graphical decorative objects one on top of the other positioning their centres on the same point.

Group: Groups a set of selected graphical objects (text, polylines, polygons) into a single object for easier manipulation as a whole.

Ungroup: Inverse operation to that of Grouping.

Rotate: Turns the mouse pointer to a cross, indicating that selecting a set of objects and dragging the end point of one of them will rotate them. Choosing this option again returns the mouse pointer to normal mode.

8.1.5 Section menu

Disable laterals: Disables the creation/update/deletion of side lanes in the sections. This is useful when modifying a network at a high scale.

Make join: Joins the selected sections, automatically creating a new node.

Break join: Breaks the join between the selected sections. This may delete the node between them.

Make Polysection: Groups a set of consecutively selected sections into a single polysection.

Break Polysection: Separates a polysection into the different sections that compose it.

Change direction: Changes the direction of the selected sections.

Number of lanes: Displays a submenu containing the following options: "1", "2", "3", "4", "5", "More...", "Default". All them serve to change the number of lanes in the selected sections. The "Default" option determines the number of lanes according to section width and default lane width.

8.1.6 Control menu

New Control: Creates a new control plan in the current view.

Open Control... Opens an existing control plan.

Close Control: Closes the current control plan.

Save Control: Saves changes in the currently open control plan.

Save Control as... Same as "Save Control", with the option to change the name under which it is

saved.

Import Control: Select the files of a Synchro control plan and import them to a GETRAM control plan

Delete Control... Deletes control plans.

Junctions... Changes the control plan junctions.

Meterings... Changes ramp-metering control plans

Settings... Sets some general parameters of the control plan such as the offset or the yellow time.

8.1.7 Result menu

New Result container: Creates a new result container in the current view.

Open Result container... Opens an existing result container.

Close Result container: Closes the current result container.

Save Result container: Saves changes in the currently open result container.

Save Result container as... Same as “Save Result container”, with the option to change the name under which it is saved.

Import Result container from EMME/2...: Automatically adds all the input flows and turning proportions from an EMME/2 assignment. Only works with EMME/2 interface.

Delete Result container: Deletes result containers.

Vehicle types... Changes the vehicle types in the current result container.

New State: Creates a new network state in the current result container.

Open State... Opens an existing network state.

Sections... Changes the state in the sections.

Close State: Changes the state in the sections.

Save State: Saves changes in the current network state.

Save State as... Same as “Save State”, with the option to change name under which it is saved.

Delete State... Deletes network states.

8.1.8 O/D matrix menu

New O/D matrix: creates a new O/D matrix.

Open O/D matrix: opens an existing O/D matrix.

Save O/D matrix: saves changes in the current open O/D matrix.

Save O/D matrix as: saves changes, with the option to change the name under which it is saved (no space in filename). This option also makes it is also possible to define a (new) start time for the various network states.

Delete O/D matrix: deletes O/D matrix (not possible if the matrix is open).

Close O/D matrix: closes the current O/D matrix.

Vehicle types: edits the vehicle types in the current O/D matrix (same as with result container).

Statements: sets the different time periods for a time sliced matrix, as well as setting the vehicle flows from each origin to each destination.

Path Definition: opens a dialog window for the definition of the desired paths from the selected origin centroids to the selected destination centroids.

Path Assignment: once the chosen matrix has been opened, it opens a dialog window for the assignment of the desired proportions of each vehicle type to the different paths defined with the previous menu option.

Import from Emme/2: Imports O/D matrices from EMME/2 data bank.

Export to Emme/2: Exports O/D matrices to EMME/2 data bank.

These two last options are available only in those versions equipped with the EMME/2 interface (GETRAM kit 2).

8.1.9 Transyt Menu

Associated with the Transyt Interface. Please refer to the Transyt Interface user manual

8.1.10 PT

PTLines: browser for the PTLines on the model.

PTPlans...: opens the dialog window in which you can enter combinations of timetables for future use with AIMSUN.

Vehicle Types...: opens the dialog window with which to associate vehicle types to be used with a PT Plan.

Import Transit Line from EMME/2: opens a browser to the EMME/2 transit file to be imported.

8.1.11 Windows

Show Map: opens navigator window.

Show Object Info: opens information window for selected objects.

Show Log Window: opens a window with the working session's messages.

Hide Status Bar: hides message area.

Hide Toolbar: hides tool bar.

8.1.12 Help

Licenses: opens a window containing the contents of the license.dat file.

About: the TEDI intro window.

We at TSS-Transport Simulation Systems have tried to make the information contained in this manual as accurate as possible. We cannot, however, guarantee that it is error-free.

Copyright

Copyright © 1992-2004 TSS-Transport Simulation Systems

All rights reserved. TSS-Transport Simulation Systems products contain certain trade secrets and confidential and proprietary information of TSS-Transport Simulation Systems. Use of this copyright notice is precautionary and does not imply publication or disclosure.

Trademark

AIMSUN and GETRAM are trademarks of TSS-Transport Simulation Systems.

Other brand or product names are trademarks or registered trademarks of their respective holders.