



UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

Adaptação de mapas geográficos e planeamento de rotas na simulação de tráfego de veículos em ambiente citadino

Roberto Jorge Ferreira Machado

Coimbra, 2009

Adaptação de mapas geográficos e planeamento de rotas na simulação de tráfego de veículos em ambiente citadino

Sob Orientação de:

Prof. Doutor Urbano José Carreira Nunes

e co-orientação de:

Eng. Pedro Fernandes

Juri:

Prof. Dr. Urbano José Carreira Nunes

Prof. Dr. Rui Alexandre de Matos Araújo

Prof. Dr. Luís Alberto da Silva Cruz

Tese submetida para Mestrado Integrado em Engenharia Electrotécnica e de
Computadores

Departamento de Engenharia Electrotécnica e de Computadores

Faculdade de Ciências e Tecnologia, Universidade de Coimbra

Setembro 2009

À minha família e àqueles que se interessem pelo tema.

Agradecimentos

O meu agradecimento ao Dr. Urbano Nunes, pela sua disponibilidade, clarificação dos objectivos a cumprir e a imensa preocupação com o desenvolvimento deste trabalho.

Um muito obrigado ao Engenheiro Pedro Fernandes, ajuda fundamental no desenrolar do projecto, com a transmissão de todo o seu conhecimento e sabedoria, tornando cada reunião semanal numa autêntica aula, tanto de conteúdos relacionados com o trabalho, como de partilha de opiniões sobre os mais variados temas.

Ao Engenheiro Daniel Krajzewicz, investigador do *Institute of Transportation Systems* do *German Aerospace Center*, por todos os esclarecimentos sobre o simulador SUMO e sobre as suas próprias investigações.

À Fundação para a Ciência e a Tecnologia (FCT), através do projecto PmITS06 (Métodos de Percepção para um Sistema de Transporte Inteligente usando informação sensorial do Veículo e da Infraestrutura), PTDC/EEA-ACR/72226/2006 (projecto financiado pela FCT e co-financiado pelo FEDER, Fundo Europeu de Desenvolvimento Regional), e ao ISR-UC que permitiu as condições necessárias à realização deste trabalho.

Aos meus amigos por toda a ajuda disponibilizada, compreensão pela falta de tempo e por todo o apoio prestado.

Por último, um agradecimento enorme à minha família, em especial aos meus pais e irmãos, que com todo o seu apoio prestado, tornou mais fácil a transição dos obstáculos que me foram surgindo.

Um abraço especial a todos, porque sem vocês nada disto seria possível.

Resumo

A presente dissertação apresenta o estudo de simuladores de tráfego, em ambiente urbano. Nesse âmbito, é efectuada a criação de mapas abertos no projecto *OpenStreetMap* (OSM) e exportação de áreas geográficas em vários formatos diferentes, com especial relevo para o formato *Extensible Markup Language* (XML). Através da análise dos dados exportados foram detectadas e corrigidas algumas falhas actualmente existentes numa área da cidade de Coimbra. A criação de redes no formato do simulador SUMO (Simulation of Urban MObility), através da descrição de nodos, arestas e conexões, foi abordada em conjunto com o processo de definição de veículos e rotas a percorrer.

São elaboradas várias simulações de percursos de tráfego baseados em tráfego real, com análise de resultados e estudo das funcionalidades do SUMO. Uma análise detalhada do algoritmo de roteamento *Dynamic User Assignment* (DUAROUTER) e do dispositivo de roteamento dinâmico é efectuada, sendo descritas com precisão as suas vantagens e desvantagens. Também os restantes algoritmos de roteamento existentes no SUMO, *Junction Turning Ratio* (JTRROUTER) e *Detector Flows* (DFROUTER), são abordados e os modos de funcionamento apresentados.

A dissertação descreve uma nova abordagem de atribuição, aos veículos, de rotas equilibradas, onde os resultados obtidos através de variadas simulações exibem as vantagens e limitações desta nova abordagem.

Abstract

This thesis presents a study of traffic simulators in urban environment. Open maps of the *OpenStreetMap* (OSM) project are created, and geographical areas in different formats are exported, specially in the *Extensible Markup Language* (XML) format. Through the exported data analysis some existing flaws of a specific area of the city of Coimbra were detected and corrected. Moreover, the creation of networks in SUMO (Simulation of Urban MObility) format, through nodes, edges and connections description were conducted, as well as vehicle and routes definition process.

Several traffic simulations based on real cases are held, followed by the resulting data analysis and the study of the SUMO functionalities. A detailed analysis of the *Dynamic User Assignment* (DUAROUTER) routing algorithm and of the dynamic routing device are made, and a precise description of its advantages and disadvantages are presented. Furthermore, the remaining routing algorithms existing in SUMO, such as *Junction Turning Ratio* (JTRROUTER) and *Detector Flows* (DFROUTER), are studied and their operation modes are presented.

A new implementation of vehicle balanced route attribution is presented. The obtained results through several simulations show the advantages and limitations of this new approach.

Índice

Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Simuladores - Estado da Arte	1
1.2 Enquadramento do trabalho	9
1.3 Objectivos da dissertação	9
1.4 Trabalho realizado e contribuições	9
1.5 Organização da tese	10
2 Conversão de mapas abertos para mapas definidos em XML	11
2.1 Mapas abertos	11
2.2 OpenStreetMap	12
2.3 Conversão de dados OSM para dados XML	14
2.4 Resultados obtidos	16
3 Produção de percursos de veículos em ambiente de simulação urbana	19
3.1 Introdução ao SUMO	19
3.2 Criação de redes no SUMO	21
3.3 Criação de rotas no SUMO	22
3.4 Resultados obtidos	23

4	Planeamento de Rotas no SUMO	27
4.1	Introdução	27
4.2	Funcionamento do DUAROUTER	28
4.3	Funcionamento do JTRROUTER	32
4.4	Funcionamento do DFROUTER	32
5	Aperfeiçoamento do DUAROUTER	35
5.1	Funcionamento do dispositivo de roteamento dinâmico	35
5.2	Nova abordagem ao DUAROUTER	38
5.3	Implementação e Resultados obtidos	40
6	Futuras implementações	49
6.1	Introdução ao sistema de comutação de semáforos do SUMO	49
6.2	Sistema de semáforos dinâmico	50
6.3	Detector de acidentes	52
7	Conclusões	55
A	Algoritmos de Roteamento	i
A.1	Algoritmo de Encaminhamento de Dijkstra	i
A.2	Algoritmo de Bellman-Ford-Moore	ii
A.3	Condições de equilíbrio de Wardrop	ii
B	Simulation of Urban MObility	v
B.1	Configuração de nodos no SUMO	v
B.2	Configuração de arestas no SUMO	vi
B.3	Configuração de conexões de arestas no SUMO	vii
B.4	Ficheiro de Configuração de uma rede SUMO	vii

B.5 Criação de rotas no SUMO	viii
B.6 Sistema de comutação do sinal dos semáforos no SUMO	x
B.7 Definição de emissores no SUMO	xii
B.8 Definição de detectores pelo DFROUTER	xiii
B.9 Definição de rotas segundo o algoritmo JTRROUTER	xv
B.10 Ficheiro de Configuração de uma simulação SUMO	xv
B.11 Manual de utilização da GUI do SUMO	xvi

Índice Remissivo**xxiv**

Lista De Acrónimos e Glossário

VISSIM	<i>Verkehr In Stadten - SIMulationsmodell</i>
3D	Três Dimensões
GIS	<i>Geographic Information System</i>
2D	Duas Dimensões
ITS	<i>Intelligent Transport Systems</i>
CORSIM	<i>CORredor SIMulation</i>
TSIS	<i>Traffic Software Integrated System</i>
GUI	<i>Graphical User Interface</i>
OD	Origem - Destino
UAF	<i>Urban Analytics Framework</i>
SUMO	<i>Simulation of Urban MObility</i>
XML	<i>Extensible Markup Language</i>
API	<i>Application Programming Interface</i>
GPS	<i>Global Positioning System</i>
CATS	<i>Chicago Area Transportation Study</i>
FHWA	<i>Federal HighWay Administration</i>
BPR	<i>Bureau of Public Roads</i>
ISR-UC	Instituto de Sistemas e Robótica
OSM	<i>OpenStreetMap</i>
GPX	<i>Exchange Format</i>
JSON	<i>JavaScript Object Notation</i>
NASA	<i>National Aeronautics and Space Administration</i>
NPE	<i>New Popular Edition</i>
JOSM	<i>Java OSM Editor</i>
HTML	<i>HyperText Markup Language</i>
PNG	<i>Portable Network Graphics</i>
JPEG	<i>Joint Photographic Experts Group</i>
SVG	<i>Scalable Vectorial Graphics</i>
PDF	<i>Portable Document Format</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
FTP	<i>File Transfer Protocol</i>
NETCONVERTER	NETwork CONVERTER
NETGEN	NETwork GENerator

DUAROUTER	<i>Dynamic User Assignment ROUTER</i>
JTRROUTER	<i>Junction Turning Ratio ROUTER</i>
DFROUTER	<i>Detector Flows ROUTER</i>
POLYCONVERT	<i>POLYgon CONVERT</i>
GUI	<i>Graphical User Interface</i>
GPL	<i>General Public License</i>
OPENGL	<i>OPEN Graphics Library</i>
TIGER	<i>Topologically Integrated Geographic Encoding and Referencing</i>
ELMAR	<i>ELECTronicMARKeting</i>
CSV	<i>Comma Separated Value</i>
ITS-GAC	<i>Institute of Transportation Systems do German Aerospace Center</i>
CPUC	<i>Central Processing Unit</i>

Lista de Figuras

1.1	Representação dos vários modelos de simulação.	8
2.1	Processo de criação de mapas no OSM.	12
2.2	Representação da área do mapa de Coimbra exportada.	17
2.3	Rede SUMO do Pólo II e zona envolvente.	18
3.1	Funcionamento geral do simulador SUMO.	20
3.2	Criação de uma rede de tráfego no simulador SUMO.	21
3.3	Aspecto gráfico da rede utilizada na primeira simulação.	24
4.1	Diferentes processos de criação de rotas.	28
5.1	Aspecto geral da rede de testes de roteamento.	41
5.2	Gráfico com os tempos médios de viagem de simulação na rede 1.	43
5.3	Gráfico com os tempos médios de viagem de simulação na rede 2.	45
5.4	Gráfico com os tempos médios de viagem de simulação na rede 3.	45

Lista de Tabelas

1.1	Vantagens e desvantagens de alguns simuladores de tráfego.	4
5.1	Tempos médios de viagem da rede 1.	43
5.2	Tempos médios de viagem da rede 2.	44
5.3	Tempos médios de viagem da rede 3.	46

Capítulo 1

Introdução

ESTE capítulo apresenta inicialmente as motivações e o enquadramento do trabalho nas áreas de investigação científica actuais. De seguida são enumerados os objectivos da dissertação, bem como a descrição do trabalho realizado. Por fim é apresentada a organização da tese.

1.1 Simuladores - Estado da Arte

Um dos mais sérios problemas da actualidade consiste no controlo do tráfego de veículos nas grandes cidades, sobretudo nas horas de ponta. Nos últimos anos, várias companhias têm apostado na criação de simuladores de tráfego. Esta aposta deve-se a um aumento na procura de soluções e previsões por parte de várias entidades relacionadas com o tráfego urbano.

As simulações geradas por estes simuladores constituem uma ferramenta útil na análise e procura de soluções para o aumento do tráfego nas zonas urbanas. A selecção de rotas alternativas mais apelativas, com optimização dos tempos de viagem, comutação mais eficiente dos sinais dos semáforos e desempenho global das redes, são algumas das funcionalidades disponibilizadas pelos actuais simuladores.

Os resultados gerados por simulação podem ser úteis a entidades governamentais na análise de risco da construção de redes viárias, de modo a garantir o máximo conforto e segurança aos seus futuros utilizadores. A previsão e o impacto que as novas redes viárias terão no tráfego, assim como comparação dos níveis de poluição ambiental gerados pelos veículos, são outras das vantagens da realização de simulações.

As características dos simuladores variam muito entre os diferentes projectos, existindo por isso projectistas de redes a utilizar vários simuladores de tráfego.

Actualmente o simulador de tráfego líder no mercado mundial é o VISSIM, abreviatura de *Verkehr In Städten - SIMulationsmodell*, que significa em português, modelo de simulação de tráfego citadino. Desenvolvido pela *PTV Planung Transport Verkehr AG*[2], VISSIM permite a simulação de vários tipos de tráfego, tais como veículos, transportes públicos, velocípedes e peões. Possui opções de apresentação de simulações em 3D e um modelo realístico do comportamento de peões, tanto em transportes públicos como nas redes viárias.

O *TransModeler* [9], desenvolvido pela *Caliper Corporation*, é um simulador de tráfego muito procurado globalmente, que se destaca pela sua versatilidade e fidelidade na análise em detalhe de variadas redes viárias, desde zonas urbanas a auto-estradas. Possui um ambiente *Geographic Information System* (GIS) em 2D e 3D, útil para ilustrar dinâmicas de fluxo de tráfego, operações de *Intelligent Transport Systems* (ITS) e comutação do sinal de semáforos. O *TransModeler* permite a importação de dados de outros simuladores de tráfego, tais como o *CORredor SIMulation* (CORSIM) [33] e *Sim Traffic* [35], e possui vários casos de estudo para situações de emergência.

Desenvolvido pela *McTrans Moving Technology*, o simulador de tráfego microscópico CORSIM possibilita a simulação de redes (NETSIM) e simulação de auto-estradas (FRESIM). Dentro do pacote de software do CORSIM encontra-se um ambiente de desenvolvimento que permite aos utilizadores a análise e gestão de operações de tráfego, a ferramenta *Traffic Software Integrated System* (TSIS).

O simulador de tráfego *Sim Traffic*, implementado pela *TrafficWare*, possui cenários de redes urbanas em 3D, com tabelas de emissões de gases e acelerações actualizadas em tempo real durante a simulação. *Sim Traffic* foi desenhado para modelar redes com cruzamentos controlados, ou não, por semáforos. Gera relatórios relativos a uma única intersecção, faixa, zona ou trajectórias de veículos, exibindo vários indicadores, tais como densidade e ocupação de determinada via.

Possuindo uma poderosa plataforma integrada, o *Paramics* [27] da *Quadstone Paramics*, contém vários módulos que colocam este software como um dos melhores simuladores no mercado. *Paramics Modeller* fornece, através de um *Graphical User Interface* (GUI), a possibilidade de criar, simular e visualizar modelos de tráfego. *Paramics Analyser* constitui uma ferramenta útil na análise e investigação de dados. A ferramenta *Paramics Processor* é usada para testes, experiências e análise de processos, enquanto o módulo *Paramics Estimator* organiza e calcula as melhores rotas, de acordo com a tabela Origem - Destino (OD). *Paramics* permite a criação e edição de modelos 3D através da ferramenta *Designer*, conversão de vários formatos de dados pelo módulo Converter e o desenvolvimento de novas funções com o *Paramics Programmer*. Por fim, *Paramics Monitor* calcula os níveis de poluição ambiental gerados pelos veículos e, o estudo da integração dos peões no trânsito citadino pode ser estudado pelo módulo *Urban Analytics Framework* (UAF).

O SUMO [16], abreviatura de *Simulation of Urban MObility*, é um simulador de tráfego microscópico *open source*, construído essencialmente para gerir extensas redes viárias. Possui várias aplicações que permitem a geração e criação de redes, roteamento e geração de relatórios. Utiliza o *Extensible Markup Language* (XML) [8] como formato padrão de entrada e saída de dados.

No geral, as simulações efectuadas por estes simuladores consistem na circulação de diferentes tipos de veículos e, em alguns casos, peões, entre um local de origem e um local de destino. Designa-se por roteamento o processo que determina uma ou mais possíveis rotas para um condutor que pretenda deslocar-se de um local da rede para outro. O roteamento dos veículos efectuado pelos simuladores de tráfego pode seguir uma abordagem adaptativa

ou estática.

Simulador	Vantagens	Desvantagens
VISSIM	Animações 3D, incluindo perspectivas dos edifícios e ambiente rodoviário Recolha de dados flexível, tais como medidas dos tempos de viagem, velocidades, comprimentos das filas de trânsito, entre outros	Complicado de manusear por ser um programa complexo Execuções em 3D e em largas redes podem tornar-se lentas
TransModeler	Gera facilmente trajectos para os veículos Facilidade no funcionamento com dados GIS	Problemas no controlo de densidades de veículos Apenas disponível para plataformas Windows
CORSIM	Fornece uma ferramenta útil para conduzir as simulações em vários estudos de tráfego Capaz de modelar redes extensas e semáforos	Comportamentos diferentes dos condutores necessitam de ser processados em separado, aumentando o número de execuções Manuseamento complicado das ferramentas disponíveis
Sim Traffic	Fácil de utilizar	Ausência de funções <i>Application Programming Interface</i> (API) Apresenta muita informação da rede mas pouca sobre o estado dos veículos
Paramics	Vários ficheiros de dados tornando conveniente o ajuste de parâmetros Funções reais para criar curvas, rotundas e passeios	Complicado de manusear Processo de criação de passeios moroso
SUMO	Portabilidade Simulador <i>open source</i>	Não possui vertente 3D Utilização do XML não é muito útil para gerar informação de saída

Tabela 1.1: Vantagens e desvantagens de alguns simuladores de tráfego.

Na abordagem adaptativa, como o nome indica, o veículo procura adaptar-se ao estado actual da rede, verificando periodicamente as velocidades dos veículos, fluxo de tráfego, acidentes ou outros problemas na rede. O sistema verifica periodicamente quais são as melhores rotas existentes, comutando o veículo para estas, caso sejam melhores alternativas.

Actualmente a ligação entre os sistemas informáticos de telecomunicações ao trânsito rodoviário é cada vez maior. A informação prestada por presença de câmaras de filmar em

vias estratégicas, *Global Positioning System* (GPS), telemóveis, *internet* ou mesmo pelos meios de comunicação social, permite aos utilizadores das estradas escolherem a melhor rota para efectuarem a viagem. Como estes factores têm um papel muito importante nos sistemas de tráfego actuais, o estudo de algoritmos de roteamento adaptativos é uma peça fundamental para a melhoria da qualidade dos resultados de simulação de tráfego.

Um dos algoritmos mais utilizados nas abordagens de roteamento dinâmico, o algoritmo Vector-Distância, consiste na implementação de uma tabela com os melhores custos das arestas, sendo essa tabela actualizada periodicamente com informação fornecida pelos nodos da rede.

Inversamente, a abordagem estática define a melhor rota para os veículos, normalmente procurando o caminho mais curto, antes de estes iniciarem a marcha, não permitindo a alteração a rota durante a viagem, excepto se ocorrerem falhas na rede. Muitos sistemas de roteamento adaptativo utilizam roteamentos estáticos, para escolherem as melhores rotas, quando se encontram no processo de verificação da rede à procura de rotas alternativas.

Utilizado frequentemente em diversas áreas de roteamento, o algoritmo estático proposto por *Edsger Dijkstra* [11] consiste no cálculo do custo mínimo das ligações de todos os nodos a um nodo referência, podendo não resolver por completo as situações em que algumas arestas apresentam custos negativos. Quando encontra o custo mínimo de ligação ao nodo de referência, atribui o respectivo custo a este, escolhendo de seguida outro nodo de referência, processo que se repetirá até ser alcançado o fim da rede.

Também muito utilizado em tarefas de planeamento de rotas, a abordagem baseada no algoritmo *Bellman-Ford-Moore* [4], tem um procedimento muito semelhante ao algoritmo de *Dijkstra*, excepto que resolve as situações em que algumas arestas apresentam custos negativos. O pseudo-código destes dois algoritmos encontra-se em anexo, nas secções A.1 e A.2.

Apesar dos algoritmos de *Dijkstra* e *Bellman-Ford-Moore* fornecerem as rotas com caminho mais curto para um determinado destino, num sistema de tráfego nem todos os

veículos podem efectuar a rota mais curta. Caso isso sucedesse, os congestionamentos ocorreriam muito mais rapidamente, e rotas que à partida têm um custo maior, seriam agora melhores alternativas. No sentido de evitar situações como esta foram investigadas e implementadas diversas abordagens para uma distribuição equilibrada dos veículos pelas rotas existentes, segundo os custos das arestas da rede.

Através de processos heurísticos, investigadores da *Chicago Area Transportation Study* (CATS) [32] desenvolveram um algoritmo para atribuição de rotas equilibradas aos veículos inseridos em diversas situações de tráfego. Inicialmente, utilizando o algoritmo *Bellman-Ford-Moore*, atribui o caminho mais curto para cada veículo. De seguida divide a rede em várias partes e em cada uma dessas partes calcula os tempos de viagem resultantes, efectuando uma nova redistribuição consoante os resultados.

Com o intuito de resolver o mesmo problema, a abordagem foi efectuada de modo diferente pela *Federal Highway Administration* (FHWA) dos EUA [1]. Inicialmente inseriam os veículos no tráfego existente, com as rotas definidas através do processo de atribuição do caminho mais curto segundo o algoritmo *Bellman-Ford-Moore*. De seguida recalculavam os tempos de viagem resultantes da inserção de novo tráfego. O tempo de tráfego seguinte seria igual ao último tempo de viagem com um peso de 25%, mais o penúltimo tempo de viagem calculado com um peso de 75 %.

Os valores dos tempos de viagem resultantes dos métodos anteriores não eram propriamente exactos, o que levou algumas instituições a desenvolver novas funções de cálculo do custo de ligação das arestas da rede. Investigadores da *Bureau of Public Roads* (BPR) [21], actual FHWA, apresentaram a seguinte função de cálculo do custo de viagem dos *links* de uma rede:

$$\tau_a = \tau_a^0 \left(1 + \alpha \left(\frac{X_a}{C_a}\right)^\beta\right) \quad (1.1)$$

Os parâmetros têm o seguinte significado:

- $\tau_a \rightarrow$ Custo de viagem do *link* a
- $\tau_a^0 \rightarrow$ Tempo de viagem no *link* a em fluxo livre

- $X_a \rightarrow$ Fluxo de veículos no *link* a
- $C'_a \rightarrow$ Capacidade prática do *link* a
- $\alpha \rightarrow$ Parâmetro de modelação. Usualmente com o valor de 0,15.
- $\beta \rightarrow$ Parâmetro de modelação. Usualmente com o valor de 4.

Também *Davidson* [10] obteve uma função para o cálculo do custo de viagem dos *links* da rede:

$$\tau_a = \tau_a^0 \left(1 + J \frac{X_a}{C_a - X_a} \right) \quad (1.2)$$

Os parâmetros têm o seguinte significado:

- $J \rightarrow$ Parâmetro que necessita de ser estimado através de medições no terreno.
- $C_a \rightarrow$ Capacidade do *link* a .

Em qualquer um dos processos referidos, após a obtenção dos custos para cada *link* efectua-se a atribuição das rotas aos veículos, sendo necessário garantir que a atribuição decorre de forma equilibrada, ou seja, de acordo com as condições de equilíbrio definidas por *Wardrop* [39]. As condições de equilíbrio de *Wardrop* são descritas no anexo A.3 . A abordagem efectuada por estas técnicas de roteamento é denominada de estática, visto que, tanto a matriz Origem - Destino (OD) bem como os fluxos nos *links* não dependem do tempo.

Uma abordagem de atribuição de rotas dinâmica, ou seja, em que os fluxos nos *links* dependem do tempo, foi implementada por *Gawron* [14], que, basicamente, consiste em atribuir maior probabilidade de um *link* pertencer a uma rota quanto menor for o custo deste, sendo esse custo calculado a partir dos tempos de simulação nas iterações anteriores. Esta abordagem é revista com mais pormenor no capítulo 4.

Além da escolha da abordagem de atribuição de rotas a utilizar, também o(s) tipo(s) de simulação diferencia os vários simuladores de tráfego existentes. Na simulação de

tráfego microscópica a abordagem dinâmica de atribuição de rotas é a mais acolhida, visto que neste tipo de simulação o movimento de cada veículo é analisado ao detalhe, sendo a atribuição de rotas e futuras escolhas baseadas nesses movimentos. Em sentido contrário, na simulação macroscópica o funcionamento do sistema é analisado com base no fluxo de tráfego, velocidade e densidade nos *links*, não havendo cálculos ao nível do funcionamento individual dos veículos. A simulação mesoscópica consiste num misto entre as duas simulações referidas anteriormente, havendo alturas que a análise ao sistema é efectuada globalmente, e alturas em que a análise é efectuada detalhadamente, veículo a veículo. A representação dos modelos de simulação é ilustrada na figura 1.1.

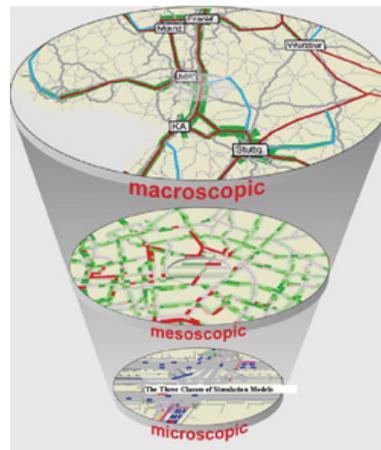


Figura 1.1: Representação dos vários modelos de simulação.

As redes de tráfego disponíveis para efectuar as simulações, são obtidas de diversas formas. Em alguns simuladores o utilizador tem a oportunidade de criar as suas próprias redes, mas noutros simuladores o fornecimento de mapas por outras entidades ou mesmo a sua compra torna-se por vezes necessário. De modo a poder contornar esse problema a criação de mapas abertos tem ultimamente sofrido um grande avanço, permitindo, além da visualização dos mapas criados, a exportação em vários formatos para que os simuladores sejam capazes de os aceitar.

1.2 Enquadramento do trabalho

A investigação de meios de transporte mais seguros e eficientes é actualmente um tema ao qual é dedicado um grande esforço de investigação. No ISR-UC, os Sistemas de Transporte Inteligente (ITS) constituem uma área estratégica; a equipa de investigação tem participado em projectos internacionais: Cybercars, Cybermove, Cybercars2 e CyberC3. Contudo, algumas das propostas para melhorar os sistemas de transporte não são facilmente testáveis em situações reais, tornando necessário o uso de simuladores. Esta dissertação enquadra-se nos trabalhos de investigação em curso em torno da utilização e melhoria de simuladores de tráfego para estudos de mobilidade em contexto citadino.

1.3 Objectivos da dissertação

O objectivo principal desta dissertação consiste na exploração de simuladores de tráfego, designadamente:

- Estudo e melhoria de algoritmos de planeamento de rotas.
- Conversão de mapas abertos para mapas definidos em XML, para utilização num simulador de tráfego, por exemplo o simulador SUMO.
- Produção de percursos de veículos em ambiente de simulação urbana.

1.4 Trabalho realizado e contribuições

No âmbito desta tese de mestrado foram realizados os seguintes trabalhos:

- Análise do processo de criação de mapas abertos no *OpenStreetMap* (OSM)[23] e exportação de diversas áreas geográficas em formato OSM XML. Conversão de dados OSM XML para dados XML.

- Análise detalhada do funcionamento do simulador de tráfego SUMO . Criação e conversão de redes de tráfego, implementação de rotas e realização de várias simulações.
- Obtenção e comparação de resultados fornecidos pelos vários algoritmos de roteamento existentes no SUMO.
- Implementação de uma nova abordagem de roteamento dos veículos, de forma a garantir melhores resultados que os algoritmos existentes, no que diz respeito ao tempo de viagem dos veículos e à rapidez na obtenção de rotas equilibradas.

1.5 Organização da tese

A dissertação encontra-se dividida em seis partes essenciais. Inicialmente, no primeiro Capítulo (**Introdução**), efectua-se a introdução aos vários simuladores de tráfego existentes e é efectuada uma análise aos vários algoritmos de planeamento de rotas. Este Capítulo aborda ainda diversos tipos de simulação existentes e apresenta uma breve introdução ao conceito de mapas abertos. O Capítulo 2 (**Conversão de mapas abertos para mapas definidos em XML**) apresenta uma análise de vários projectos de mapas abertos e detalha a criação e exportação de mapas abertos OSM. A terceira parte (**Produção de percursos de veículos em ambiente de simulação urbana e Planeamento de Rotas no SUMO**) apresenta além da descrição da criação de redes e rotas, uma análise aos vários algoritmos de roteamento presentes no simulador SUMO. O Capítulo 5 (**Aperfeiçoamento do DUAROUTER**) introduz o dispositivo de roteamento dinâmico existente no SUMO e descreve a nova implementação efectuada e respectivos resultados. O capítulo 6 (**Futuras implementações**) apresenta propostas de trabalhos futuros em diferentes contextos de gestão de tráfego. Finalmente, o capítulo 7 (**Conclusões**) apresenta as conclusões do trabalho desenvolvido.

Capítulo 2

Conversão de mapas abertos para mapas definidos em XML

2.1 Mapas abertos

ACTUALMENTE existem vários locais na Internet onde podemos visualizar e criar mapas abertos e descarregá-los em vários formatos diferentes. O facto da maior parte dos mapas existentes na *web* terem restrições legais levou à necessidade de criar projectos gratuitos, em que é possível tanto a criação como a edição de mapas rodoviários, conduzindo assim à democratização do acesso a dados geográficos.

Para criar e editar um mapa é necessário recolher os dados das vias pretendidas (normalmente por intermédio de um GPS) e carregá-los para o projecto pretendido, onde ocorrerá a edição do mapa.

Um dos projectos abertos mais conhecidos e utilizado hoje em dia é o OSM, que consiste na criação e edição de mapas gratuitos de todo o mundo. Porém, outros projectos de disponibilização de mapas abertos encontram-se acessíveis a todos os utilizadores.

Criado por um grupo de voluntários, o projecto *OpenMaps* [22] permite que, no seu sítio

original, sejam criados e disponibilizados mapas geográficos. Actualmente este projecto disponibiliza, para exportação, apenas mapas da Europa Central.

O projecto *CloudMade* [7] permite ao utilizador a criação de rotas entre dois pontos geográficos, definidos nos mapas fornecidos pelo OSM. O utilizador pode escolher o meio de transporte preferido (veículo a motor, velocípede ou a pé) e o tipo de rota pretendida (mais rápida ou mais curta). *CloudMade* fornece dois tipos de formatos de exportação de rotas, *GPS Exchange Format* (GPX) e *JavaScript Object Notation* (JSON).

2.2 OpenStreetMap

Iniciado em Julho de 2004, o projecto OSM conta actualmente com cerca de 130000 colaboradores e mais de 95000 milhões de pontos geográficos [24] carregados para na base de dados. O OSM permite a criação, edição e visualização de mapas de todo o mundo por todos os utilizadores, que apenas necessitam de se registarem no sítio oficial [23] de forma gratuita. O processo de criação de mapas pode ser visualizado na figura 2.1

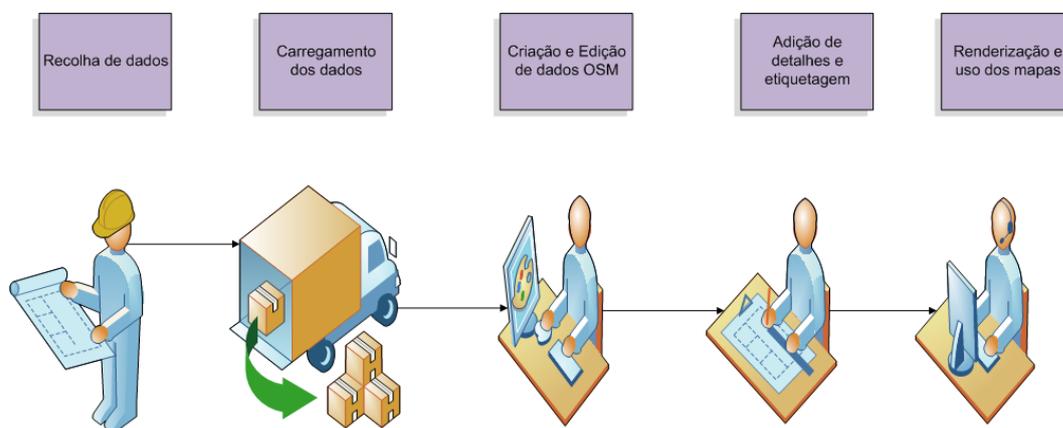


Figura 2.1: Processo de criação de mapas no OSM.

A primeira etapa do processo de criação de mapas no OSM consiste na recolha de dados, geralmente efectuada por intermédio de um GPS. Além dos pontos geográficos,

também o nome das estradas e outros detalhes devem ser adicionados. Outros métodos podem ser utilizados além do GPS, tais como extracção de dados de mapas fornecidos pela *Yahoo!* [40], imagens livres do satélite *Landsat* [18] da *National Aeronautics and Space Administration* (NASA) e cópia dos dados dos mapas da *New Popular Edition* (NPE) [20].

A utilização de mapas pessoais, necessariamente livres, e de dados previamente recolhidos são outros métodos de contribuição disponíveis.

Findo o processo de recolha de dados, é necessário gravar os ficheiros no formato GPX, carregá-los para o OSM e descarregá-los para o *Java OSM Editor* (JOSM) [34]. JOSM é um editor para o OSM que suporta dados GPX, disponibilizando a edição de estradas e nodos. Além do JOSM, também programas como *Potlatch* [28] e *Merkaartor* [30] permitem a edição e adição de detalhes de mapas existentes.

O processo de edição dos dados consiste na representação dos nodos e arestas por determinados elementos. A representação dos nodos é efectuada por intermédio de pontos e as arestas são identificadas por setas que conectam dois pontos, em que o sentido da seta indica o sentido de fluxo. O OSM apresenta também um tutorial que explica como efectuar a representação de curvas.

A seguir à edição dos dados é necessária a adição de etiquetas, ou seja, informação detalhada sobre a rede, tal como nome e tipo das vias, número de faixas ou vias proibidas a determinado tipo de veículos. Terminado o processo de edição é necessário carregar os dados para o OSM, podendo o utilizador visualizar as alterações no sítio oficial do OSM ou renderizar gráficos de mapas de *bits*.

OSM disponibiliza três plataformas de renderização, *Kosmos* [5], *Osmarender* e *Mapnik*. A apresentação dos mapas na página *web* do OSM é efectuada, por definição, pela ferramenta *Mapnik*, mas outras opções de apresentação encontram-se disponíveis, por intermédio das plataformas *Osmarender*, *CycleMap* [12] e *No Name*.

O OSM disponibiliza a exportação, para uso pessoal, dos dados dos mapas anteriormente criados e colocados na base de dados do sistema. O utilizador poderá escolher a

área geográfica que pretende descarregar e, por intermédio da opção **Export**, escolher o tipo de formato pretendido. O OSM disponibiliza vários formatos para exportação, nomeadamente dados de formato OSM XML, imagens *Mapnik* [19], imagens *Osmarender* [26] e *HyperText Markup Language* (HTML) embebido.

A aplicação *Mapnik* permite, além da escolha de vários formatos de imagem – *Portable Network Graphics* (PNG), *Joint Photographic Experts Group* (JPEG), *Scalable Vectorial Graphics* (SVG), *Portable Document Format* (PDF) e *PostScript* – a possibilidade de definição da escala da imagem.

O *Osmarender* disponibiliza os formatos PNG e JPEG para exportação de imagens e permite definir o *zoom* da área a exportar.

A opção de exportação HTML embutido fornece o código HTML de uma determinada área definida pelo utilizador, sendo um método muito utilizado para exibir uma área mais reduzida de um mapa, em páginas *web*.

Tanto as imagens fornecidas como a lista de vectores de dados estão disponíveis para descarregar através de uma licença da *Creative Commons Attribution-ShareAlike 2.0*.

2.3 Conversão de dados OSM para dados XML

O OSM permite vários formatos de exportação de dados. Destes, o formato de exportação para dados OSM XML é o que revela maior interesse no âmbito deste projecto, devido ao facto da interacção entre os utilizadores e o simulador de tráfego SUMO ser efectuada por intermédio de código XML.

O processo de exportação para dados OSM XML é iniciado pela selecção da área do mapa a exportar. Como muita informação é disponibilizada durante a exportação, esta opção apenas é disponibilizada para áreas reduzidas, no máximo de 2300 km². Assim que a área pretendida esteja seleccionada ou as coordenadas da área definidas, a exportação é efectuada premindo a opção *Export*, sendo os dados guardados num ficheiro de extensão

“.osm”.

Outra forma de exportação das imagens do OSM consiste na utilização do software livre *Wget* [31], que fornece ficheiros usando *HyperText Transfer Protocol* (HTTP), *HyperText Transfer Protocol Secure* (HTTPS) e *File Transfer Protocol* (FTP). A exportação dos dados OSM para um ficheiro XML é descrita através de:

```
wget.exe "my_map_web_page" -o my_osm_net.osm.xml
```

Os parâmetros têm o seguinte significado:

- **wget.exe** → Inicia a aplicação *Wget* com o intuito de efectuar a exportação
- **my_map_web_page** → Endereço *web* da área do mapa pretendida para exportar. Deve conter as coordenadas geográficas da referida área.
- **my_osm_net.osm.xml** → Nome do ficheiro, em formato XML, que contém os dados exportados do mapa OSM

O ficheiro que contém os dados do mapa possui informação variada. Na representação de um nodo é fornecida a sua identificação, coordenadas de latitude e longitude, versão de alteração, identificação do utilizador responsável pela edição, dia em que foi efectuada e qual a plataforma utilizada.

A representação de uma aresta é um pouco mais complexa do que a representação de um nodo, contendo, além da informação idêntica à que identifica um nodo, todos os nodos que a aresta conecta, o tipo de aresta e o número de vias.

Outros dados podem ser adicionados, tais como nome das arestas e nodos, identificação de túneis e pontes, ou mesmo definição de rios.

Depois de gravar o ficheiro de dados, é necessário utilizar o comando de conversão de redes, *NETworkCONVERTER* (NETCONVERTER), para converter os dados do mapa para uma rede SUMO, utilizando a seguinte linha de comandos [25]:

```
sumo-netconvert --osm my_osm_net.osm.xml -o map.net.xml --use-  
projection
```

Os parâmetros têm o seguinte significado:

- **sumo-netconvert** → Aplicação disponibilizada pelo SUMO para conversão de redes
- **osm** → Comando que efectua a leitura do mapa OSM importado
- **map.net.xml** → Nome do ficheiro, atribuído pelo utilizador, que representa a rede SUMO convertida a partir do mapa OSM
- **use-projection** → Opção necessária para converter coordenadas geográficas em cartesianas

Apesar de muita informação sobre o mapa ser convertida, algumas características da rede ainda não são inseridas na conversão entre dados OSM e dados XML. Dessas características destaca-se a ausência de semáforos e a importação de pontos de interesse, entre outros.

2.4 Resultados obtidos

Com o intuito de criar simulações de tráfego numa rede mais familiar, exportou-se do sítio da OSM uma área do mapa representativo do Pólo II e zona envolvente. Como foi referido no capítulo anterior, existem duas alternativas para efectuar essa exportação. A primeira exportação consistiu na selecção manual da área a exportar, escolha do formato de exportação em dados OSM XML, selecção do botão *Export* e por fim gravação do ficheiro de dados. Na segunda exportação foi utilizado o programa de software *Wget*, tendo a exportação da respectiva zona sido efectuada da seguinte forma:

```
wget.exe "www.openstreetmap.org/?lat=40.18804&lon=-8.41156&zoom=15&layers=B000FTFT" -o polo2.osm.xml
```

A imagem resultante da exportação do mapa do Pólo II encontra-se representada na figura 2.2.

De seguida, utilizou-se o comando NETCONVERT para converter os dados em formato OSM XML, contidos no ficheiro “polo2.osm.xml”, para dados XML, sendo utilizada a seguinte linha de comandos:

```
sumo-netconvert --osm polo2.osm.xml -o polo2.net.xml --use-projection
```

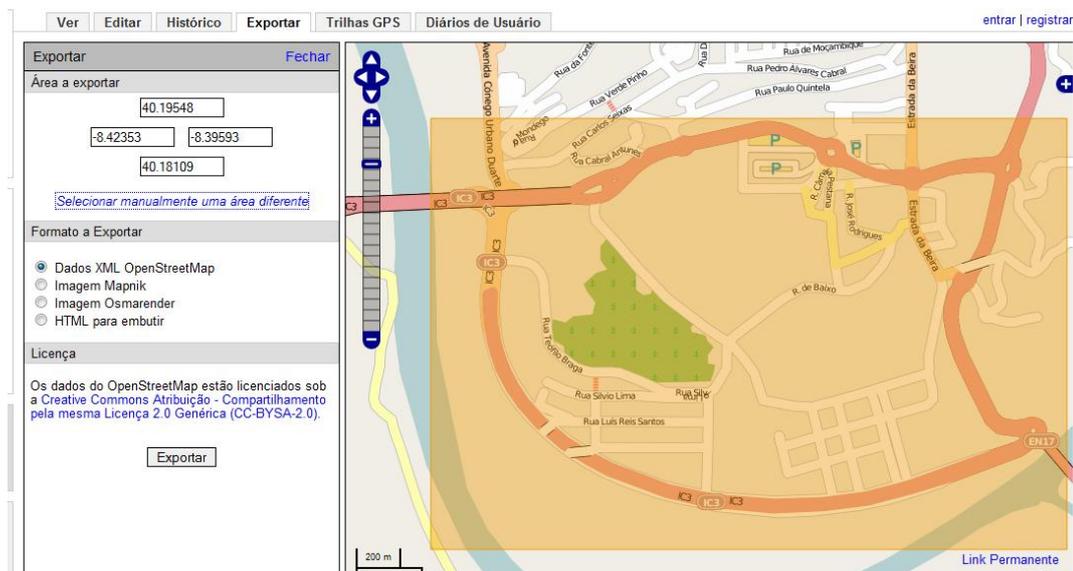


Figura 2.2: Representação da área do mapa de Coimbra exportada.

A rede SUMO representativa da área do Pólo II e zonas envolventes, resultante da conversão anterior, pode ser visualizada na figura 2.3.

Os dados da rede criada encontram-se todos definidos no ficheiro “polo2.net.xml”. O

comando de conversão de redes NETCONVERT possibilita a separação desses dados em três ficheiros separados, representando nodos, arestas e conexões de arestas, respectivamente. A conversão, utilizando o respectivo ficheiro “polo2.osm.xml” foi efectuada do seguinte modo:

```
sumo-netconvert --osm polo2.osm.xml --plain-output polo2 --use-projection
```

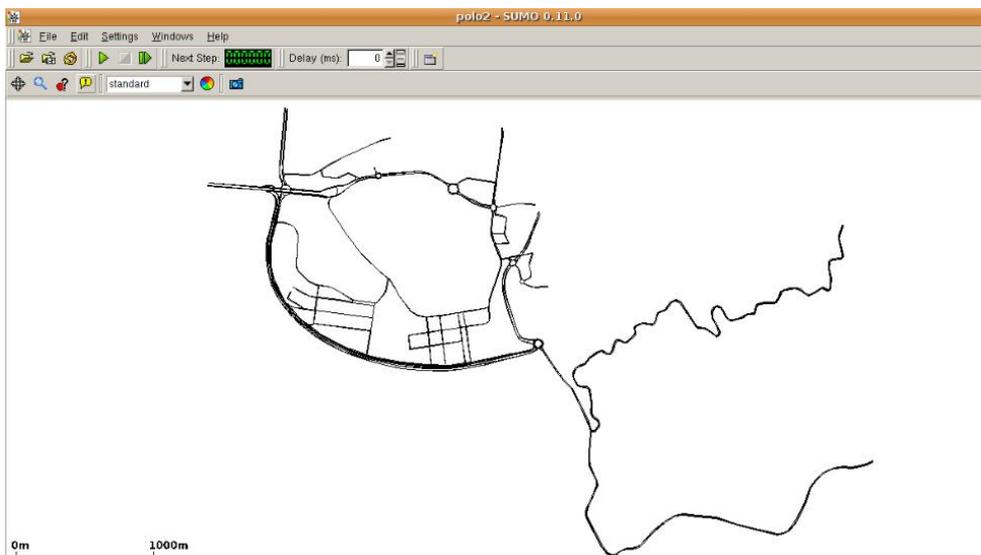


Figura 2.3: Rede SUMO do Pólo II e zona envolvente.

O parâmetro **plain-output** cria então os respectivos ficheiros de descrição de redes, nomeadamente, “polo2.nod.xml”, “polo2.edg.xml” e “polo2.con.xml”.

Concluído o processo de separação do tipo de ficheiros, foi possível renomear as arestas com os respectivos nomes das vias.

Capítulo 3

Produção de percursos de veículos em ambiente de simulação urbana

3.1 Introdução ao SUMO

O simulador de tráfego utilizado, SUMO, é um projecto aberto que consiste na criação, edição e gestão de redes de tráfego. A opção de tornar o seu conteúdo *open source*, teve como objectivo facultar aos seus utilizadores a liberdade para modificar o respectivo código fonte, possibilitando a personalização das funcionalidades do simulador. Como exemplo, pode referir-se a implementação de optimizações dos sistemas de comutação de semáforos. Dentro do pacote de software disponível para *download* na página do simulador, encontram-se várias aplicações, como por exemplo:

- Simulação em linha de comandos (SUMO) e com um GUI
- Constructor/Conversor de redes (NETCONVERT)
- Constructor/Gerador de redes (NETGEN)
- Planeadores de rotas

- Importador de polígonos (POLYCONVERT)

Durante uma simulação, tanto a rede como os veículos que a vão percorrer, são modelos microscópios em tempo discreto no espaço contínuo. O SUMO usa uma licença *General Public License* (GPL) e é um sistema portátil, visto que utiliza apenas funções padrão da linguagem C++ e bibliotecas padrão. Outras vantagens do simulador consistem num rápido *Open Graphics Library* (OPENGL) - GUI e no suporte à importação de vários formatos de redes (*VISUM*[2], *VISSIM*[2], *Arc View*[13], *XML - Descriptions* [8]). O SUMO possibilita ainda a gestão de redes com mais de 10000 vias, percorridas por vários tipos de veículos, circulando com rotas pré-definidas ou com roteamento dinâmico. As diferentes aplicações fornecidas pelo simulador SUMO são representadas na figura 3.1.

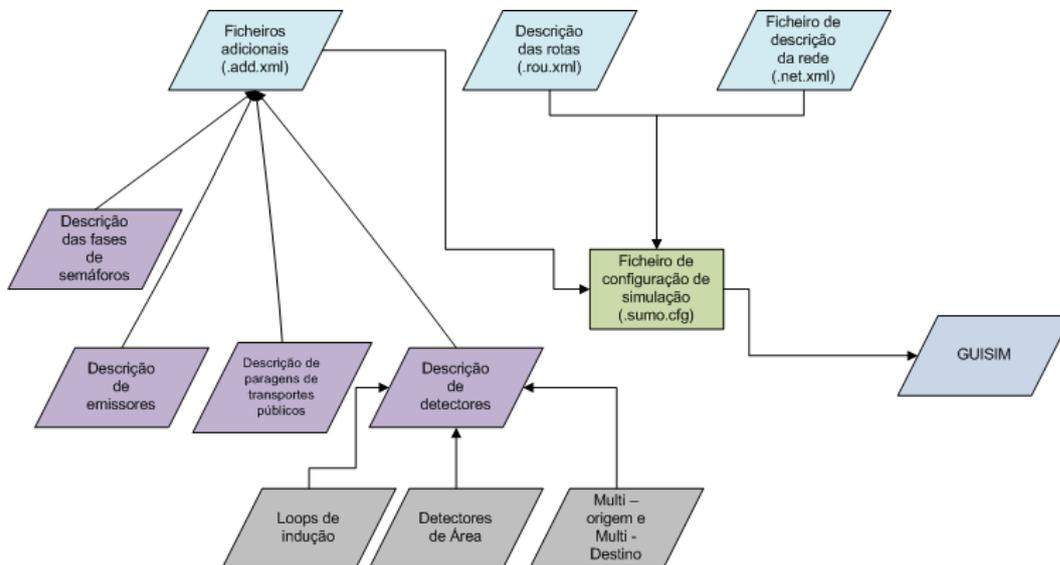


Figura 3.1: Funcionamento geral do simulador SUMO.

3.2 Criação de redes no SUMO

O SUMO utiliza a sua própria descrição de redes viárias, através de *XML - Descriptions*, e, dessa forma, as redes devem ser convertidas a partir de uma lista de dados já existente. As redes do SUMO podem ser geradas através de mapas existentes ou utilizando o *NETworkGENerator* (NETGEN) para gerar mapas rodoviários básicos.

Através da ferramenta NETCONVERTER podem converter-se os dados que descrevem uma rede viária numa descrição de redes que o SUMO seja capaz de ler. Esta ferramenta é capaz de converter vários formatos de redes, como por exemplo, OSM, VISUM, VISSIM, *ArcView* e descrições XML. A importação de bases de dados *Topologically Integrated Geographic Encoding and Referencing* (TIGER) [6] e ficheiros *NavTech* convertidos da *ELectronicMARKeting* (ELMAR) [29] também pode ser efectuada com a ferramenta NETCONVERT. O processo de criação de uma rede no simulador SUMO pode ser visualizado na figura 3.2.

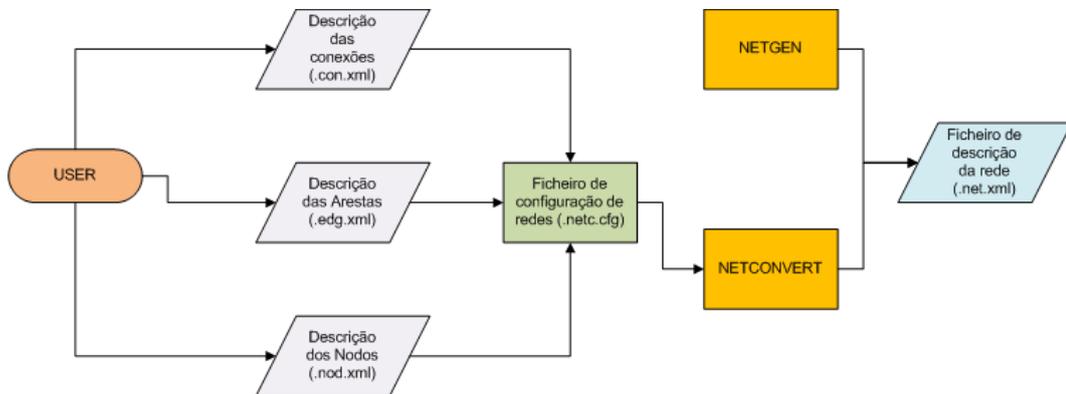


Figura 3.2: Criação de uma rede de tráfego no simulador SUMO.

Para desenhar uma rede através de descrições XML são necessários pelo menos dois ficheiros, um para os nodos e outro para as arestas (estradas) que interligam esses nodos. Adicionalmente, existe a possibilidade de unir as arestas por tipo e definir explicitamente conexões entre rotas. A descrição destes ficheiros é apresentada em anexo.

Desde a versão 0.9.4 do SUMO que se pode fornecer coordenadas x e y usando geo-coordenadas. Neste caso, as coordenadas serão interpretadas como longitude/latitude em graus.

Através de um ficheiro de configuração, o utilizador introduz como parâmetros de entrada o nome dos ficheiros de descrição dos nodos, das arestas e, opcionalmente, das conexões (descrição efectuada nos anexos B.1, B.2 e B.3). Como saída é introduzido o nome pretendido para a rede SUMO a ser criada. Utilizando a ferramenta NETCONVERT, o utilizador pode criar a rede respectiva introduzindo o comando seguinte na consola:

```
sumo-netconvert -c <NET_CONFIGURATION_FILE>
```

O protótipo de um ficheiro de configuração de uma rede é exibido em anexo, na secção B.4 .

3.3 Criação de rotas no SUMO

Terminado o processo de criação de redes viárias, é necessário descrever os percursos que os veículos irão efectuar dentro de uma simulação. O simulador SUMO possibilita várias alternativas para efectuar este processo de criação de rotas, designadamente:

- **Definição de viagens:** O conceito de viagem consiste na definição de um local de partida, um local de chegada e do instante em que o veículo inicia a sua viagem.
- **Definição de fluxos:** Método semelhante ao de definições de viagens, excepto que se podem agregar veículos com a mesma rota.
- **Definição de fluxos e taxas de comutação:** Processo em que se utilizam taxas de comutação nas junções, em substituição das arestas de destino do fluxo.

- **Matrizes OD:** Tabelas que contêm as origens e destinos das viagens a efectuar. Em primeiro lugar efectua-se a conversão dos valores da tabela para definições de viagem, e, por fim, procede-se à criação das rotas, tendo como base as definições de viagem recentemente criadas.
- **Definição de rotas pelo utilizador:** O utilizador pode definir as rotas que pretende num ficheiro de configuração com a extensão “.rou.xml”.
- **Rotas aleatórias:** Processo mais célere de criação de rotas, apesar de não ser o mais fiável, pois as rotas são criadas aleatoriamente.
- **Importação de rotas disponíveis:** Tarefa que consiste na importação de rotas dos simuladores VISSIM E VISUM.

Com o intuito de processar as rotas, o SUMO disponibiliza quatro aplicações, *Dynamic User Assignment* (DUAROUTER), *Junction Turning Ratio* (JTRROUTER), *Detector Flows* (DFROUTER) e OD2TRIPS. O seu funcionamento é explicado em pormenor no capítulo 4. A criação de rotas no SUMO e definição da implementação de veículos é apresentada no anexo B.5.

3.4 Resultados obtidos

Com o intuito de testar grande parte das potencialidades do SUMO e descobrir algumas das suas limitações, procedeu-se à realização de várias simulações. Na primeira simulação foi criada uma rede (figura 3.3) que permitisse verificar várias aplicações de gestão de tráfego fornecidas pelo SUMO.

Para simular o tráfego nesta rede foram criadas quatro rotas. A primeira percorre apenas as vias do lado direito, duas rotas percorrem apenas as vias do lado esquerdo e a quarta quase toda a rede.

Nesta simulação foi possível testar a comutação dos sinais de semáforos, implementando

um sistema de comutação mais equilibrado, de modo que o tempo de espera dos veículos nesse nodo fosse menor. Também foram criados dois sistemas de comutação que actuam consoante as fases do dia. Durante a manhã e a tarde, quando o fluxo de veículos na via principal é mais elevado, a luz verde está accionada num maior período de tempo de modo a permitir que um maior número de veículos abandone o cruzamento. No sistema de comutação implementado para actuar durante o período nocturno, o sinal verde na via principal está menos tempo accionado, pois o fluxo de veículos é inferior ao verificado durante o dia. O sistema de comutação dos sinais dos semáforos encontra-se descrito em anexo, na secção B.6.

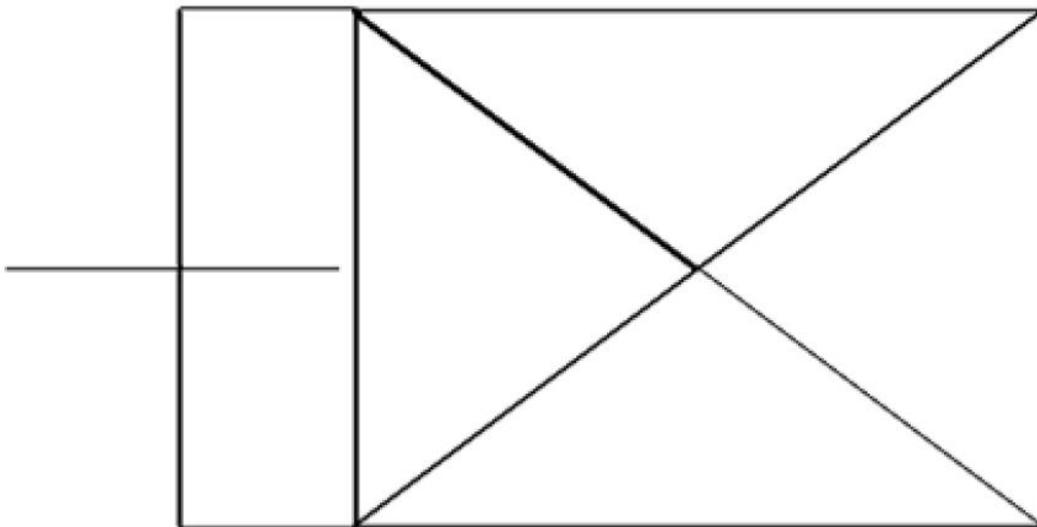


Figura 3.3: Aspecto gráfico da rede utilizada na primeira simulação.

Além das quatro rotas, foram também implementados vários emissores ao longo da rede. Um emissor é um dispositivo que injecta veículos na rede a partir de determinada posição e num tempo definido. O modo de funcionamento dos emissores é descrito em anexo, na secção B.7.

Na segunda simulação foi utilizada a rede do Pólo II e zona envolvente, criada no

Capítulo 2.4, onde se implementou vários percursos entre as estradas da respectiva rede. Desse modo, foram criadas rotas idênticas aos transportes públicos municipais que circulam nessa zona, com paragens nos respectivos locais durante um determinado período de tempo. Com esta simulação foi possível verificar o efeito que os transportes públicos provocam nesta zona da cidade.

Analisando os resultados obtidos da simulação efectuada nesta rede pode-se concluir que, de um modo geral, o tempo de paragem dos autocarros nas paragens, para entrada e saída de passageiros, não provoca algum tipo de congestionamento nas vias. Isto deve-se ao facto de parte das paragens não se encontrar na via, mas sim numa zona de apoio à direita da via. Nas paragens que se encontram na via, na maioria dos casos os outros veículos têm outra faixa, podendo facilmente ultrapassar o autocarro parado, não sendo assim necessário esperar que este inicie a marcha de novo. As excepções a estes casos anteriores são as paragens da Rua Miguel Bombarda, pois caso o autocarro pare numa dessas paragens, os veículos que o seguem têm de aguardar o reiniciar da sua marcha, devido ao SUMO impossibilitar a ultrapassagem em estradas com apenas uma via. Neste caso também não ocorrerão grandes problemas de congestionamento, devido ao fluxo de veículos nesta estrada não ser muito elevado, ao que se deve o facto de existirem várias alternativas de entrada e saída do Pólo II.

Capítulo 4

Planeamento de Rotas no SUMO

4.1 Introdução

ACTUALMENTE o simulador de tráfego SUMO tem disponíveis quatro aplicações para a criação de rotas, DUAROUTER, JTRROUTER, DFROUTER e OD2TRIPS. O DUAROUTER atribui maior probabilidade de escolha a uma rota com menor custo, sendo esse custo calculado com base no algoritmo de *Dijkstra*. JTRROUTER é utilizado quando se pretende modelar o tráfego estatisticamente, atribuindo probabilidades de escolha a cada aresta conectada à aresta de origem do fluxo. DFROUTER efectua a atribuição de rotas segundo o fluxo recolhido por detectores que se encontram colocados em vários locais da rede. A aplicação OD2TRIPS consiste na conversão de matrizes OD em viagens e seguidamente em rotas. OD2TRIPS é apenas utilizado para importar dados com o formato VISUM/VISION/VISSIM. Os diferentes processos de criação de rotas são ilustrados na figura 4.1.

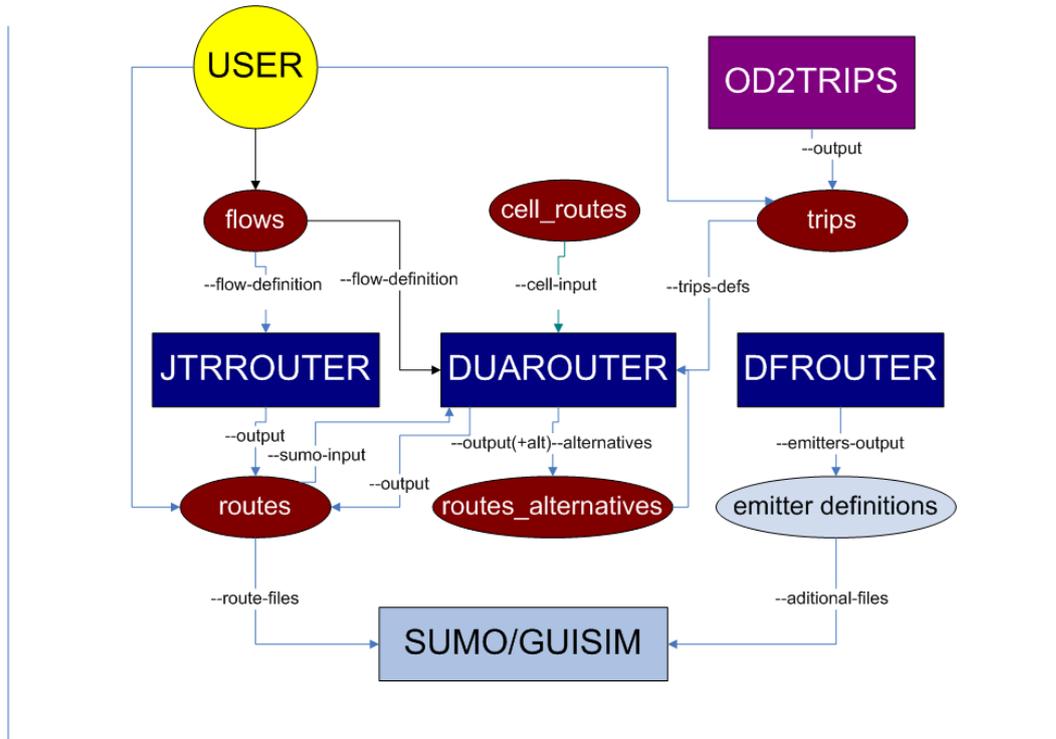


Figura 4.1: Diferentes processos de criação de rotas.

4.2 Funcionamento do DUAROUTER

O objectivo desta aplicação de planeamento de rotas consiste na criação das melhores rotas possíveis para os condutores que pretendam efectuar determinada viagem.

Teoricamente, a melhor rota é aquela que apresenta menor custo de passagem para o condutor, ou seja, a rota que permite ao condutor demorar menos tempo a percorrer o percurso desejado. Na prática, esta análise não é tão linear, pois caso todos os utilizadores da rede escolham a rota mais curta, existe grande probabilidade de ocorrerem congestionamentos numa via, especialmente se esta terminar num cruzamento. Actual-

mente o trânsito citadino encontra-se de certo modo equilibrado, pois os condutores já têm um conhecimento prévio das rotas mais curtas ou, caso estas estejam congestionadas, das melhores rotas alternativas. A procura deste equilíbrio é, no fundo, o objectivo do DUAROUTER, que vai procurando as melhores rotas alternativas e distribuindo os veículos por estas, baseado em iterações anteriores. Verificando o tempo de viagem de cada veículo durante a simulação, o DUAROUTER atribui maior probabilidade de escolha de determinada rota quanto menor foi o tempo de simulação desta numa iteração anterior.

Utilizando como base a origem e destino da viagem, DUAROUTER começa por calcular o custo menor, ou seja, a rota que, teoricamente, o condutor demorará menos tempo a efectuar. O custo de cada aresta é calculado do seguinte modo:

$$C_a = \frac{l}{v_{\max}} \quad (4.1)$$

em que os parâmetros têm o seguinte significado:

- $C_a \rightarrow$ Custo da aresta a
- $l \rightarrow$ Comprimento da aresta a
- $v_{\max} \rightarrow$ Velocidade máxima permitida nessa aresta

Para calcular o custo da rota é necessário somar o custo de cada aresta que pertence a essa rota:

$$C_T(r) = \sum_{a=1}^{a=N} C_a(r) \quad (4.2)$$

em que os parâmetros têm o seguinte significado:

- $C_T(r) \rightarrow$ Custo total da rota r
- $C_a(r) \rightarrow$ Custo de cada aresta a que pertence à rota r
- $N \rightarrow$ Número máximo de arestas que pertencem à rota r

Após calcular o custo de cada aresta, a aplicação escolherá como rota a utilizar na primeira simulação, aquela que apresentar menor custo, com base no algoritmo de *Dijkstra*.

No fim de cada simulação, o sistema procura rotas alternativas às que foram simuladas. Caso seja encontrada alguma, é atribuída uma probabilidade teórica de roteamento inicial $\rho_d=1$, sendo de seguida adicionada à lista de alternativas.

Caso exista mais do que uma rota disponível, é necessário recalculer a probabilidade teórica de cada uma, através de:

$$\rho_d \leftarrow \rho_d \frac{N_a - 1}{N_a} \quad (4.3)$$

em que os parâmetros têm o seguinte significado:

- ρ_d → Probabilidade teórica da rota
- N_a → Número de rotas alternativas

No fim de cada simulação é necessário actualizar os custos de cada rota. Segundo a abordagem de *Gawron* [14], o custo actual de cada rota vai depender do custo dessa rota antes da simulação e do tempo que o veículo demorou a percorrer essa rota durante a simulação:

$$\tau_c(r) = \beta \tau_n(r) + (1 - \beta) \tau_o(r) \quad (4.4)$$

em que os parâmetros têm o seguinte significado:

- $\tau_c(r)$ → Custo actual da rota r
- β → Parâmetro de modelação. Tem por defeito o valor de 0.9
- $\tau_n(r)$ → Tempo de simulação da rota r , ou seja, tempo que o veículo demorou a transitar a rota r . $\tau_n(r)$ é igual à soma do tempo de simulação de cada aresta que pertence à rota r .

- $\tau_o(r) \rightarrow$ Custo da rota r antes da simulação. Caso não tenha ocorrido ainda uma simulação, $\tau_o(r) = C_T(r)$. No caso em que já tenha ocorrido pelo menos uma simulação, $\tau_o(r) = \tau_c(r)$.

Após a actualização dos custos das rotas é necessário actualizar as probabilidades de escolha das rotas. Novamente segundo *Gawron*, a diferença relativa de custos entre duas rotas consecutivas pertencentes à lista de alternativas é dada por:

$$\delta_{rs} = \frac{\tau_c(s) - \tau_c(r)}{\tau_c(s) + \tau_c(r)} \in [-1, 1] \quad (4.5)$$

em que parâmetros têm o seguinte significado:

- $\delta_{rs} \rightarrow$ Custo relativo entre as rotas r e s
- $\tau_c(r) \rightarrow$ Custo actual da rota r
- $\tau_c(s) \rightarrow$ Custo actual da rota s

As probabilidades actualizadas das rotas r e s , respectivamente $\rho_a(r)$ e $\rho_a(s)$, serão dadas por:

$$\rho_a(r) = f(\delta_{rs}) \quad (4.6)$$

$$\rho_a(s) = \rho_d(r) + \rho_d(s) - \rho_a(r) \quad (4.7)$$

A função $f(\delta_{rs})$ é uma função diferenciável monótona, dada por:

$$f(\delta_{rs}) = \frac{\rho_d(s)(\rho_d(r) + \rho_d(s))g(\delta_{rs})}{\rho_d(r)\delta_{rs} + \rho_d(s)} \quad (4.8)$$

em que:

$$g(\delta_{rs}) = \exp\left(\frac{\alpha\delta_{rs}}{1 - \delta_{rs}}\right) \quad (4.9)$$

em que α é um parâmetro de modelação que tem por defeito o valor de 0.5.

Para garantir que ρ_a seja uma distribuição de probabilidades, tem de se assegurar as seguintes condições:

$$f(-1) \geq 0 \quad (4.10)$$

$$f(1) \leq \rho_d(r) + \rho_d(s) \quad (4.11)$$

$$f(0) = \rho_d(r) \quad (4.12)$$

O processo de actualização de custos e probabilidades tem de ser sempre efectuado após o término de uma simulação, até que o sistema atinja o equilíbrio. Na prática, o DUAROUTER não consegue indicar quando é que o sistema atingiu o equilíbrio. Porém, durante a realização de uma série de simulações, quando se observa que os valores dos custos das rotas e das probabilidades são sempre iguais, pode-se indicar que o sistema se encontra equilibrado.

4.3 Funcionamento do JTRROUTER

Ao contrário do algoritmo DUAROUTER, que em cada nodo é indicada qual a próxima aresta a seguir, o algoritmo JTRROUTER define, em cada nodo, uma probabilidade de escolha para cada aresta que o utilizador pode seguir depois de abandonar o nodo. Enquanto na definição de viagens era necessário declarar a sua origem e destino, nesta aplicação é apenas necessário definir a origem. A cada aresta que está conectada à aresta de origem é atribuída uma probabilidade diferente e, baseado nessas probabilidades, o sistema determina qual a aresta seguinte. De realçar que a soma das probabilidades de todas as arestas conectadas tem necessariamente de ser igual a 1. A sintaxe a utilizar para criar uma lista de rotas utilizando o JTRROUTER encontra-se representada em anexo, na secção B.9.

4.4 Funcionamento do DFROUTER

De um modo geral, a aplicação DFROUTER utiliza os dados resultantes de detectores de fluxo, que se encontram colocados por toda a rede, para criar as melhores rotas possíveis.

Inicialmente é necessário definir qual o tipo de detector a utilizar, processo que o DFROUTER utiliza usando como parâmetros de entrada a rede a utilizar e um ficheiro que indica o local da rede onde se encontram os detectores.

De seguida o DUAROUTER calcula as rotas entre os detectores criados anteriormente. O ficheiro de rotas resultante apenas contém informação das rotas, não definindo os veículos que irão circular na rede. De um modo geral, apenas as rotas que se encontram entre um detector *source* e um detector *sink* são criadas. Para criar rotas iniciadas num detector *between* é necessário a utilização do comando **--routes-for-all** durante a compilação das rotas.

Utilizando as rotas calculadas e uma lista de fluxos criada anteriormente pelo utilizador, DFROUTER implementa de seguida os fluxos que circularão na rede. A lista de fluxos tem necessariamente de estar inserida num ficheiro de formato *Comma Separated Value* (CSV). Com as rotas e fluxos definidos, o DFROUTER termina o seu processo e os veículos podem então iniciar a sua rota nos detectores *source*, sendo injectados para a rede através de emissores. A definição do tipo de detectores e da lista de fluxos encontra-se descrita em anexo, na secção B.8.

Capítulo 5

Aperfeiçoamento do DUAROUTER

5.1 Funcionamento do dispositivo de roteamento dinâmico

O simulador de tráfego SUMO fornece um dispositivo de roteamento que atribui a melhor rota, no momento em que os veículos (com o dispositivo implementado) iniciam as suas viagens e em cada período de roteamento (caso esse período esteja definido), consoante as velocidades actuais das arestas.

Inicialmente é atribuída uma rota válida para cada veículo. O sistema calcula o custo de esforço de cada aresta e, utilizando o algoritmo de *Dijkstra*, identifica a rota com menor custo. O custo de cada aresta é efectuado utilizando a equação 4.1.

No instante temporal em que um veículo, com dispositivo de roteamento implementado, inicia a sua rota, a velocidade média de cada via de cada aresta é verificada. A velocidade média de cada via é obtida através da média das velocidades instantâneas dos veículos que nesse instante circulam nessa via:

$$s_{med}(l) = \frac{\sum_{v=1}^{v=nv} s_{inst}(v)}{nv(l)} \quad (5.1)$$

em que os parâmetros têm o seguinte significado:

- $s_{med}(l)$ → Velocidade média da via l
- $s_{inst}(v)$ → Velocidade instantânea do veículo v
- $nv(l)$ → Número de veículos que percorrem a via l nesse instante temporal

A velocidade média de cada aresta será igual à soma das velocidades médias das vias dessa aresta a dividir pelo número de vias:

$$v_{med}(a) = \frac{\sum_{l=1}^{l=nl} s_{med}(l)}{nl(a)} \quad (5.2)$$

em que os parâmetros têm o seguinte significado:

- $v_{med}(a)$ → Velocidade média da aresta a
- $nl(a)$ → Número de vias da aresta a

O esforço de cada aresta, no momento em que um veículo com um dispositivo de roteamento implementado inicia a sua viagem, será igual a:

$$c_{corrente}(a) = \frac{l(a)}{v_{med}(a)} \quad (5.3)$$

em que os parâmetros têm o seguinte significado:

- $c_{corrente}(a)$ → Custo corrente da aresta a
- $l(a)$ → Comprimento da aresta a

O custo corrente da aresta a será comparada com o custo teórico que o veículo teria na mesma aresta, sendo o custo real igual ao máximo destes dois valores. O custo que teoricamente o veículo v iria ter na aresta a , é dado por:

$$c_{th}(v,a) = \frac{l(a)}{v_{\max}(v)} \quad (5.4)$$

em que os parâmetros têm o seguinte significado:

- $c_{th}(v,a)$ → Custo teórico do veículo v na aresta a
- $v_{\max}(v)$ → Velocidade máxima que o veículo v pode atingir

O custo real da aresta, $C(a)$, é portanto:

$$C(a) = \max(c_{corrente}(a), c_{th}(v,a)) \quad (5.5)$$

Como os veículos variam constantemente de velocidade, é necessário adaptar constantemente os pesos das arestas. Através do ficheiro de configuração existe a possibilidade de definir um peso de adaptação, W_a . O custo adaptado da aresta a será calculado a partir de:

$$C(a) \leftarrow C(a)W_o + c_{corrente}(a)W_n \quad (5.6)$$

em que os parâmetros têm o seguinte significado:

- W_o → Peso de adaptação antigo, sendo o seu valor igual a W_a
- W_n → Peso de adaptação novo, sendo o seu valor igual a $1 - W_a$

Com o custo de cada aresta calculado e actualizado, o dispositivo de roteamento utiliza o algoritmo de *Dijkstra* para atribuir a melhor rota ao veículo que inicia a sua viagem.

Além de ser possível efectuar roteamento quando os veículos com o dispositivo implementado iniciam a rota, também é possível efectuar roteamento em etapas temporais separadas por um período de tempo, definido no ficheiro de configuração. Sempre que se

alcance essa etapa temporal, o dispositivo efectua os cálculos da melhor rota utilizando as equações definidas anteriormente. Caso o dispositivo calcule uma rota melhor do que a que se encontrava definida, o veículo vai alterar a sua rota para a nova e melhor alternativa. O valor do período terá necessariamente que ser superior a 0.

Os veículos com o dispositivo instalado possuem então uma vantagem em relação aos outros utilizadores da rede que não tenham implementado o dispositivo, pois aqueles sabem no início e ao longo da viagem quais os melhores percursos a efectuar. Como com bastante frequência vai existir uma alteração da rota dos veículos com dispositivo, os tempos de viagem dos veículos sem dispositivo vai ser diferente daquele em que não fosse possível a alteração da rota durante a viagem. Esta alteração dos tempos de viagem deve-se ao facto de que novos veículos surgirão e desaparecerão em alturas distintas, interferindo com o fluxo que se efectuava anteriormente. Segundo testes realizados em várias redes viárias diferentes (descritas no capítulo seguinte) pode-se constatar que, apesar disso, a introdução de veículos com dispositivo melhora o tempo médio global de viagem bem como a grande maioria dos tempos de viagem dos veículos sem o dispositivo implementado. Apenas um número muito reduzido de utilizadores sem o dispositivo resultará prejudicado com a introdução de veículos com dispositivo. Assim, a sua implementação constitui uma excelente ferramenta na gestão de tráfego.

5.2 Nova abordagem ao DUAROUTER

O *Institute of Transportation Systems* do *German Aerospace Center* (ITS-GAC) [15], desenvolveu uma investigação[3], utilizando o simulador de tráfego SUMO, de modo a testar diferentes tipos de roteamento em diversas redes viárias, num tipo de simulação microscópico. Foram efectuadas diversas medições, em duas redes criadas pelos investigadores e numa rede real, baseada na cidade alemã de *Magdeburg*. A investigação consistiu na análise dos algoritmos DUAROUTER (secção 4.2), roteamento e re-roteamento dinâmico (secção 5.1) e abordagem de *Dijkstra*.

Analisando os resultados obtidos da performance da rede chegou-se à conclusão que os algoritmos DUAROUTER e de roteamento dinâmico apresentam melhores resultados em relação ao algoritmo de *Dijkstra*, em vários parâmetros, tais como velocidade média, tempo de viagem médio, tempo de espera para início da viagem, entre outros.

Para serem obtidos estes resultados foi necessário efectuar 50 iterações quando se realizou o teste segundo o algoritmo DUAROUTER. Na maior parte dos casos, para atingir um certo equilíbrio na rede não são necessárias tantas iterações. A realização destes cálculos requer muito tempo de processamento do *Central Processing Unit* (CPU) e os vários ficheiros gerados, como contêm muitos dados de informação, ocupam demasiado espaço de armazenamento.

Para melhorar os resultados obtidos pelo ITS-GAC, ou seja, para reduzir o número de iterações a efectuar para atingir um certo equilíbrio e diminuir os tempos de viagem das iterações iniciais, elevados em resultado do sistema ainda se encontrar à procura de todas as rotas alternativas, foi analisada durante a realização deste projecto uma nova abordagem, baseada numa junção parcial entre o algoritmo DUAROUTER e o de roteamento dinâmico.

O conceito fundamental desta nova implementação consiste em equilibrar o tráfego nas redes viárias com determinadas percentagens de veículos sem dispositivo de roteamento, aumentando gradualmente essa percentagem. Os restantes veículos têm instalado o dispositivo de roteamento que lhes permite escolher a melhor rota consoante o fluxo na rede.

A abordagem consiste na realização de uma primeira iteração em que os todos os veículos, que irão percorrer a rede, têm instalado o dispositivo de roteamento. A percentagem de veículos com o dispositivo inserido vai diminuir nas iterações seguintes. Os veículos que nessas iterações não possuam roteamento dinâmico irão efectuar uma rota atribuída com base na utilização do algoritmo DUAROUTER. Apesar do tempo de viagem dos veículos, sem dispositivo, ser elevado nas primeiras iterações, a compensação é efectuada pelo reduzido tempo de viagem dos veículos com o dispositivo implementado, que se baseiam no estado actual da rede para calcular a melhor rota.

Quando se atinge a iteração em que deixam de existir veículos com o dispositivo implementado, o sistema funciona puramente como um sistema que se guia pelo algoritmo DUAROUTER.

5.3 Implementação e Resultados obtidos

Com o intuito de testar a nova implementação foi utilizada uma rede exemplo disponível no pacote do software SUMO. A rede é constituída por um nodo de origem denominado de “beg” e um de destino denominado de “last”. Entre a origem e o destino existem três rotas possíveis, que percorrem as arestas centrais, do lado esquerdo e lado direito da rede. A rota central, a mais curta, engloba as arestas “beg”, “middle”, “end” e “last”. A rota que percorre o lado esquerdo da rede inclui as arestas “beg”, “beg2left”, “left”, “left2end”, “end” e “last”. Por fim a rota que percorre o lado direito da rede engloba as arestas “beg”, “beg2right”, “right”, “right2end”, “end” e “last”. O nodo “end” que conecta as arestas “left2end”, “middle” e “right2end” à aresta “end”, tem implementado um sistema de semáforos. Além dos testes efectuados nesta rede, alteraram-se alguns aspectos desta de modo a poder testar a nova abordagem em diferentes situações. Na segunda rede de testes inseriu-se mais uma via nas arestas “beg2left”, “left” e “left2end” e na terceira rede de testes inseriu-se mais uma via nas arestas “beg2right”, “right” e “right2end”. O aspecto global da rede pode ser visualizado na figura 5.1.

Inicialmente criou-se um ficheiro de definição de viagens de 125 veículos entre a aresta origem “beg” e a aresta destino “last”. De seguida foi também criado um sistema de comutação do sinal dos semáforos mais equilibrado do que o definido por defeito pelo simulador.

De modo a poder comparar resultados entre o algoritmo DUAROUTER e a nova implementação, obtiveram-se inicialmente os tempos médios de viagem do sistema DUAROUTER e de seguida os da nova abordagem.

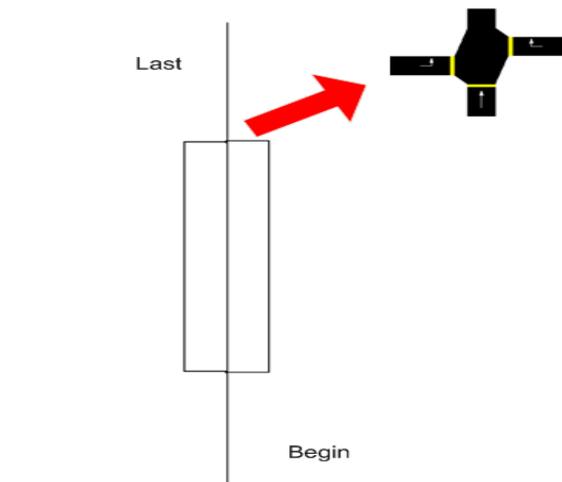


Figura 5.1: Aspecto geral da rede de testes de roteamento.

O sistema DUAROUTER recebe como entrada, na primeira iteração, o ficheiro da rede e o ficheiro das viagens dos veículos e, partindo destes, obtém o ficheiro das rotas a percorrer e respectivas alternativas. De seguida, através de um ficheiro de configuração, efectua uma simulação em que os veículos irão percorrer a rede com as rotas anteriormente definidas pelo sistema DUAROUTER. Na segunda iteração o sistema DUAROUTER analisa os tempos de viagem da simulação anterior e das rotas alternativas e, com base nestes valores, calcula a probabilidade de cada rota ser escolhida na próxima simulação. Este processo repete-se até que seja obtido um determinado equilíbrio. Nos testes efectuados definiu-se que o sistema se encontrava em equilíbrio quando o tempo médio de viagem dos veículos, em 10 simulações consecutivas, é igual.

Obtidos os resultados para o sistema DUAROUTER, implementou-se a nova abordagem na mesma rede, com o mesmo número de veículos e ordem de saída igual.

Na primeira iteração introduziu-se de novo como entrada o ficheiro da rede do sistema e ficheiro de definição das viagens para a aplicação DUAROUTER atribuir uma rota válida para cada veículo. No ficheiro de configuração da primeira simulação, cada veículo tem definida uma rota, calculada pelo sistema DUAROUTER, mas tem também implementado o dispositivo de roteamento que irá verificar o estado da rede no início da sua viagem e

num período de minuto a minuto de simulação (período pode ser definido pelo utilizador). Caso nestas etapas temporais seja encontrada uma melhor rota para o veículo, este comuta então para a melhor alternativa. Na simulação seguinte a percentagem de veículos com o dispositivo de roteamento implementado foi alterada de 100% para 60%, e os restantes 40% dos veículos percorreram as rotas atribuídas pelo sistema DUAROUTER. Na terceira simulação a percentagem de veículos com o dispositivo implementado baixou para 30% e a restante percentagem de veículos percorreu as rotas atribuídas pelo sistema DUAROUTER que foram calculadas na iteração anterior. Na quarta simulação nenhum veículo tinha implementado o dispositivo, ou seja, todos os veículos efectuaram o percurso baseado nas rotas definidas pelo sistema DUAROUTER. Nesta altura uma elevada percentagem de veículos já tem rotas definidas mais equilibradas, que foram calculadas baseadas nos tempos de simulação das iterações anteriores.

Também nesta abordagem se definiu que o sistema se encontrava em equilíbrio quando o tempo médio de viagem dos veículos, em 10 simulações consecutivas, era igual. Os valores das percentagens de veículos com o dispositivo implementado foram os valores que proporcionaram melhores tempos médios de viagem durante a realização de testes a esta nova abordagem. Nas simulações com uma percentagem de veículos com roteamento dinâmico, esses veículos foram escolhidos aleatoriamente.

Os resultados obtidos, durante a realização dos testes nos diferentes tipos de rede e utilizando as duas abordagens de roteamento, foram os seguintes:

Na rede 1, utilizando o algoritmo DUAROUTER, o equilíbrio da rede é obtido na 5^a iteração, com um valor de tempo médio de viagem igual a 94,34 segundos de simulação. Utilizando a nova abordagem o equilíbrio da rede é obtido na 7^a iteração, com um valor de tempo médio de viagem igual a 86,39 segundos de simulação. O gráfico com os resultados deste 1^o caso de estudo podem ser visualizados na figura 5.2.

	DUAROUTER	Nova Implementação
Iteração_1	429,85	137,12
Iteração_2	154,21	122,58
Iteração_3	90,44	94,17
Iteração_4	95,66	87,41
Iteração_5	94,34	91,14
Iteração_6	94,34	87,41
Iteração_7	94,34	86,39

Tabela 5.1: Tempos médios de viagem da rede 1.

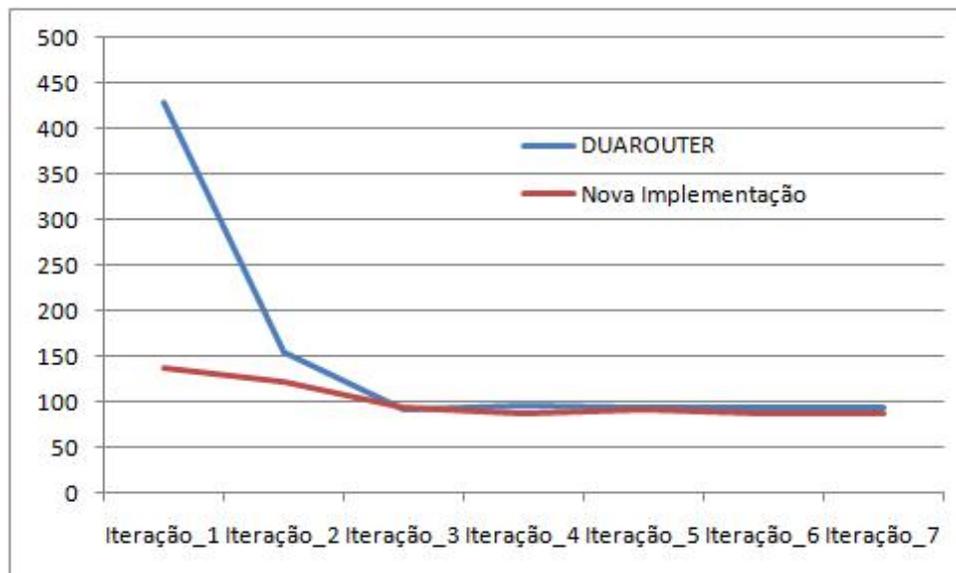


Figura 5.2: Gráfico com os tempos médios de viagem de simulação na rede 1.

Na rede 2, utilizando o algoritmo DUAROUTER, o equilíbrio da rede é obtido na 3^a iteração, com um valor de tempo médio de viagem igual a 90,94 segundos de simulação. Utilizando a nova abordagem o equilíbrio da rede é obtido na 14^a iteração, com um valor

de tempo médio de viagem igual a 88,92 segundos de simulação. O gráfico com os resultados deste 1º caso de estudo podem ser visualizados na figura 5.3.

	DUAROUTER	Nova Implementação
Iteração_1	388,65	116,9
Iteração_2	143,63	97,9
Iteração_3	90,44	97,66
Iteração_4	90,94	92,11
Iteração_5	90,94	86,78
Iteração_6	90,94	86,78
Iteração_7	90,94	87,87
Iteração_8	90,94	87,78
Iteração_9	90,94	87,62
Iteração_10	90,94	88,3
Iteração_11	90,94	88,05
Iteração_12	90,94	91,3
Iteração_13	90,94	86,27
Iteração_14	90,94	88,92

Tabela 5.2: Tempos médios de viagem da rede 2.

Utilizando o algoritmo DUAROUTER, o equilíbrio da rede 3 é obtido na 11ª iteração, com um valor de tempo médio de viagem igual a 98,04 segundos de simulação. Utilizando a nova abordagem o equilíbrio da rede é obtido na 13ª iteração, com um valor de tempo médio de viagem igual a 93,71 segundos de simulação. O gráfico com os resultados deste 1º caso de estudo podem ser visualizados na figura 5.4.

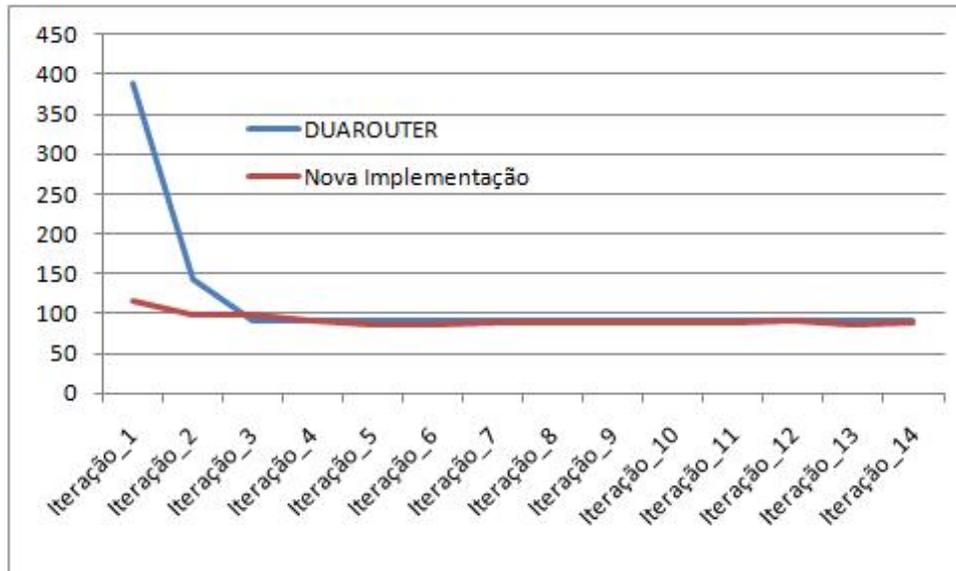


Figura 5.3: Gráfico com os tempos médios de viagem de simulação na rede 2.

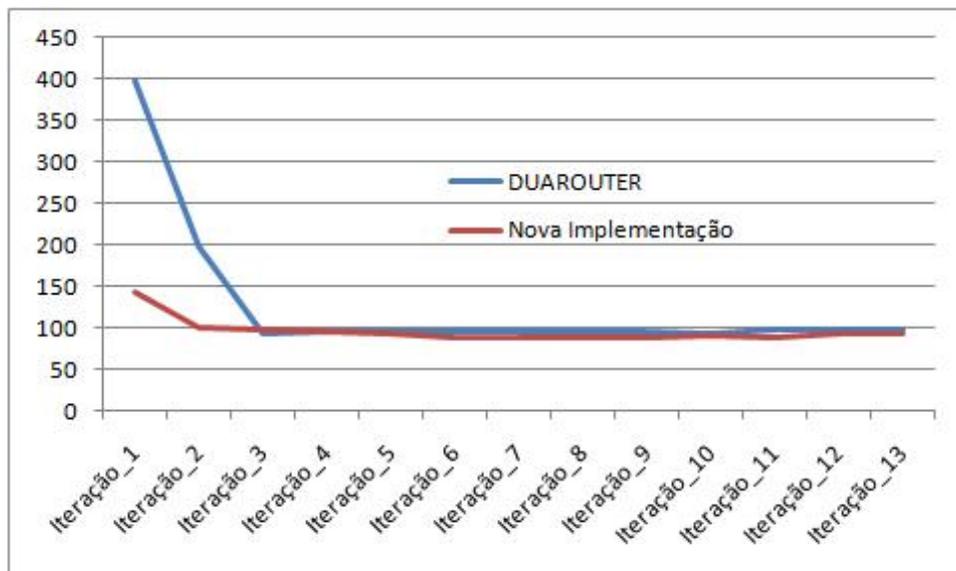


Figura 5.4: Gráfico com os tempos médios de viagem de simulação na rede 3.

	DUAROUTER	Nova Implementação
Iteração_1	397,53	142,97
Iteração_2	198,57	99,79
Iteração_3	94,43	98,74
Iteração_4	96,77	96,73
Iteração_5	96,66	92,48
Iteração_6	95,04	89,24
Iteração_7	95,04	89,24
Iteração_8	95,04	88,9
Iteração_9	95,04	88,9
Iteração_10	93,2	89,93
Iteração_11	98,04	87,81
Iteração_12	98,04	92,3
Iteração_13	98,04	93,71

Tabela 5.3: Tempos médios de viagem da rede 3.

Analisando os resultados chegou-se à conclusão que, em todas as redes, os tempos médios iniciais são inferiores utilizando a nova implementação. Para os utilizadores que pretendam uma lista de rotas equilibradas com rapidez, esta nova abordagem é uma boa solução. No que diz respeito ao equilíbrio da rede, este é obtido mais cedo utilizando o algoritmo de DUAROUTER, mas os tempos de equilíbrio são inferiores com a nova implementação. Como foi referido na secção 5.1, a inclusão de um dispositivo de roteamento em alguns veículos não prejudica os tempos de viagem da maioria dos veículos sem o dispositivo implementado.

Durante a implementação da nova abordagem, na 2ª iteração 60 % dos veículos têm instalado o dispositivo de roteamento e os restantes 40% têm rotas definidas. Comparando o tempo desses 40% de veículos na segunda iteração do DUAROUTER e da segunda

iteração da nova implementação, na rede 3, obteve-se um tempo médio de viagem dessa percentagem de veículos de 194,62 segundos de simulação com o algoritmo DUAROUTER e de 94,19 segundos de simulação com a nova implementação. Concluindo, o tempo médio destes veículos foi reduzido para menos de metade, mostrando que a inclusão de veículos com dispositivo de roteamento não prejudica os seus tempos de viagem.

Resumindo, a nova implementação é favorável em todos os factores, excepto nas iterações necessárias para se atingir o equilíbrio. Isto deve-se ao facto dos cálculos de análise da rede efectuados pelo dispositivo de roteamento apenas verificar periodicamente as velocidades médias das arestas, não estimando todos os veículos que se dirigem para essas arestas mas que ainda não entram para esses cálculos.

Capítulo 6

Futuras implementações

6.1 Introdução ao sistema de comutação de semáforos do SUMO

A O longo da realização deste projecto foram discutidas novas ideias que poderiam contribuir para o melhoramento da gestão dos sistemas de tráfego. Uma dessas ideias consistia na implementação de um sistema de semáforos em que a comutação do sinal deste fosse de acordo com o actual estado da rede.

O processo de implementação de um sistema de semáforos no SUMO inicia-se pela atribuição de *bits* a cada faixa rodoviária em que o seu fluxo de tráfego termina num nodo com semáforos. Para cada um dos sinais verde e amarelo, o *bit* 1 coloca esse sinal activo e o *bit* 0 desactiva esse sinal. Quando um sinal se encontra activo os outros sinais têm necessariamente de se encontrarem desactivados. O sistema inicia então um ciclo que começa por atribuir o *bit* 1 ao sinal verde de uma faixa, e esse *bit* vai rodar todas as arestas até atingir a aresta inicial, efectuando assim uma fase de comutação. O tempo em que cada *bit* se encontra activo no sinal verde é o tempo necessário que demora a transitar para a aresta seguinte. A definição de um sistema de comutação de semáforos é apresentada com mais pormenor em anexo, na secção B.6.

6.2 Sistema de semáforos dinâmico

Um dos problemas actuais do tráfego rodoviário consiste no tempo excessivo com que os veículos se deparam num cruzamento à espera que o sinal de semáforo comute para verde, que além de aumentar os seus tempos de viagem também aumenta o consumo de combustível. Apesar de em muitos cruzamentos surgirem constantemente, e em elevado número, veículos de todos os sentidos, em alguns nodos existem arestas com menor fluxo e outras com maior fluxo. A ideia base de um sistema de semáforos dinâmico consiste em fornecer mais tempo de sinal de semáforo verde às arestas com maior fluxo de tráfego e menos tempo às arestas com menor fluxo, possibilitando assim que um maior número de veículos abandone o cruzamento.

Muitas abordagens têm sido actualmente efectuadas no sentido de melhorar o controlo das comutações dos sinais de semáforos. Segundo a abordagem de investigadores da Universidade de *Utrecht*[36], cada veículo indica ao sistema de controlo de semáforos o destino da sua viagem e a sua posição na rede viária. Com base nesta informação o sistema estima os tempos de viagem de cada veículo e calcula quanto tempo de sinal verde deverá fornecer a cada aresta no sentido de minimizar todos os tempos de viagens.

Com o intuito de resolver o mesmo problema, investigadores do ITS-GAC apresentaram outra abordagem[17], que consistia basicamente em recolher o comprimento das filas de tráfego, localizadas em frente a um semáforo, e fornecer mais tempo de sinal verde às vias que apresentem filas com maior comprimento.

Comparando as duas abordagens, a abordagem do ITS-GAC é mais simples e exequível, pois é exigido menor quantidade de cálculos complexos.

No sentido de melhorar ainda mais os tempos de viagem obtidos, com a nova implementação descrita no capítulo 5, foi abordado, durante a realização deste projecto, um algoritmo para uma futura implementação de um sistema de comutação das fases de um semáforo, utilizando a ideia base do sistema do ITS-GAC referida anteriormente.

O algoritmo consiste na criação de vários programas de fases de comutação do sinal de semáforos, escolhendo o sistema qual o melhor programa a utilizar consoante o estado da rede. Inicialmente são criados dois programas, um para actuar durante o dia, altura em que existe maior fluxo de tráfego, e outro programa que actuará durante a noite, quando existe menor fluxo. Estes dois programas possuem fases de definição dos sinais equilibradas, não existindo nenhuma aresta com vantagem em relação às outras. De seguida, o sistema criará tantos programas de fases como possíveis combinações que existam entre arestas que se conectam ao cruzamento. O número de programas poderá ser elevado, visto que, todas as situações do estado do tráfego no nodo tem de ser satisfeitas, pois existem alturas com muito fluxo de tráfego não só em apenas numa aresta, mas também em duas ou mais arestas. Cada programa criado favorecerá então, com maior tempo de sinal de semáforo verde, a(s) respectiva(s) aresta(s).

Criados os vários programas efectua-se, de seguida, a implementação de um detector de área, existente no SUMO, para cada aresta que conecta aos nodos com semáforos. Cada detector recolherá o valor do comprimento da fila de tráfego à entrada do semáforo da aresta correspondente.

Quando se iniciar uma simulação, o primeiro programa de fases a funcionar será o programa equilibrado de dia ou o programa equilibrado de noite. Quando o programa inicial terminar, o detector indicará ao sistema o comprimento de cada fila de congestionamento existente. O sistema irá comparar então todos os comprimentos das filas e comuta de seguida para o programa de fases favorável às arestas com áreas congestionadas. Este processo repetir-se-á sempre que as fases de um determinado programa terminem, verificando o sistema qual o melhor programa a utilizar de seguida. No caso de nenhuma área ou todas as áreas terem comprimentos de filas pouco significativos ou iguais, o sistema irá comutar para o programa de fases equilibrado.

6.3 Detector de acidentes

Outro dos problemas actuais no tráfego rodoviário é o congestionamento provocado por um acidente. Em situações reais, quando acontece um acidente rodoviário, os condutores são avisados por mensagens electrónicas exibidas em ecrãs colocados ao longo das estradas ou por intermédio de meios de comunicação que constantemente informam a situação do trânsito rodoviário, dando a liberdade aos condutores de escolherem estradas alternativas.

Neste momento o simulador SUMO não possui nenhum sistema que alerte os condutores que ocorreu um acidente, o que pode originar vários congestionamentos na rede. Além disso, quando se pretende efectuar os cálculos das melhores rotas para um hipotético equilíbrio da rede, caso ocorra um acidente em determinada aresta, o tempo de simulação dos veículos que circulam nesta será muito elevado, o que irá originar numa probabilidade de escolha desta aresta menor na próxima iteração.

O dispositivo de roteamento utiliza as velocidades médias das arestas para escolher as melhores rotas e para calcular essas velocidades médias verifica a velocidade instantânea de cada veículo. Quando um veículo se encontra envolvido num acidente, a sua velocidade instantânea será igual a 0, mas como o dispositivo de roteamento recolherá a velocidade média da aresta, poderão existir inúmeros veículos a circular sem limitações, antes ou depois do acidente, resultando numa velocidade média da aresta com valor muito superior a 0, tornando esta aresta numa falsa via alternativa.

Uma das formas de limitar o acesso de veículos, a uma estrada congestionada em resultado de uma acidente rodoviário, consiste na recolha da velocidade instantânea de cada veículo em determinada área da aresta. Quando a velocidade instantânea de qualquer veículo que se encontre nessa respectiva área for igual a 0, significando que esse veículo não se encontra em andamento, define-se o custo dessa aresta com um valor muito elevado, tornando-a menos apetecível aos utilizadores. Para ser fornecida a informação aos veículos da rede que ocorreu um acidente é necessário algum método de comunicação. SUMO disponibiliza um dispositivo de comunicação entre veículos, C2C, embora o seu

funcionamento não seja o mais correcto. Apesar de ser possível efectuar a verificação das velocidades instantâneas ao longo de toda a estrada, a área de recolha das velocidades é limitada ao início da aresta. Esta opção deve-se ao facto de que caso o acidente ocorra mais próximo do fim da aresta, existe a possibilidade de os veículos, que nesse momento estão a escolher a rota, quando atingirem o local do acidente a situação já se encontrar resolvida. A área estimada donde serão recolhidas as velocidades instantâneas dos veículos seria então desde o início da aresta até uma determinada posição na estrada obtida através de:

$$lim = l(a) - 2\frac{l(a)}{3} \quad (6.1)$$

O atributo *lim* indica a posição limite, na aresta, de recolha de velocidades instantâneas dos veículos, enquanto que o parâmetro $l(a)$ representa o comprimento da aresta a . Este valor não é exacto, visto que o comprimento das estradas não é constante. O utilizador poderá definir o período de espaçamento temporal entre recolhas de velocidades, tendo esse período necessariamente de ser superior a 0. Logicamente quando menor for o período definido maior é a probabilidade de se detectar um acidente permitindo a muitos veículos evitar essa aresta.

No simulador SUMO não é permitido ultrapassar um veículo utilizando a faixa contrária, permitindo apenas a ultrapassagem em estradas com duas ou mais vias. Deste modo, em estradas com duas ou mais vias não é atribuído um custo elevado a essa aresta, pois veículos têm a possibilidade de ultrapassar o veículo que se encontra imóvel.

Capítulo 7

Conclusões

ESTE capítulo apresenta as principais conclusões aos resultados obtidos das análises e implementações realizadas, descritas nos capítulos anteriores.

No capítulo 2 foi abordado o processo de criação de mapas abertos no OSM e correspondente exportação para formato XML. Analisando os processos de criação de um mapa aberto chegou-se à conclusão que sem a ajuda de voluntários na recolha de dados, a possibilidade de disponibilização livre de mapas geográficos não era possível de efectuar. Ao nível da exportação, o OSM permite alguns formatos de dados, destacando-se o formato XML, o mesmo formato que efectua a ligação entre o utilizador e o simulador utilizado neste projecto, o SUMO.

Em virtude dos resultados obtidos aquando da exportação, e respectiva conversão de uma área do mapa de Coimbra, obteve-se algumas falhas na respectiva rede. No que diz respeito à zona do Pólo II, a rua Sílvio Lima era representada como uma via de sentido único, tendo sido necessário criar uma faixa paralela a esta com o sentido inverso. Na zona envolvente, as maiores modificações foram efectuadas na rotunda de acesso à Ponte Rainha Santa Isabel, local onde o fluxo de tráfego se efectuava em sentido contrário, ou seja, pela esquerda. Além da alteração do sentido de fluxo da rotunda, foi necessário criar uma ligação desta à avenida Mendes Silva e vice-versa. Como referido anteriormente, a

presença de semáforos ainda não é uma realidade nos mapas OSM, o que tornou a sua criação necessária na avenida Mendes Silva. A criação do bairro da Boavista e edição das falhas referidas anteriormente, são algumas das modificações que precisam de serem efectuadas na respectiva zona do mapa OSM.

No capítulo 3 e, em maior pormenor, no anexo B foram apresentadas todas as características do simulador de tráfego SUMO. A realização de várias simulações de teste permitiu analisar muitas funcionalidades existentes no SUMO, tais como criação de rotas, emissores de veículos, detectores de tráfego e sistemas de comutação da fase de semáforos.

Os diversos algoritmos de roteamento foram abordados no capítulo 4 e, em maior pormenor, no anexo B. Durante a realização de vários testes foi possível verificar as diferenças de funcionamentos entre estes, destacando-se o algoritmo DUAROUTER na sua tarefa de atribuição de rotas equilibradas aos veículos.

No capítulo 5 foi introduzido o funcionamento básico do dispositivo de roteamento disponibilizado pelo SUMO, que atribui no início da viagem e num período de roteamento, as melhores rotas para os veículos com o dispositivo implementado. Segundos os resultados obtidos nas simulações efectuadas, concluiu-se que os veículos sem o dispositivo implementado não serão prejudicados com a presença de outros veículos com dispositivo, constatando-se que os tempos de viagem destes veículos é muito melhor, tirando alguns casos esporádicos.

No mesmo capítulo é apresentada a abordagem de roteamento idealizada neste projecto, tendo sido realizados testes de comparação ao algoritmo DUAROUTER em várias redes diferentes. Na análise desses resultados concluiu-se que os tempos de viagem das iterações iniciais e de equilíbrio são melhores utilizando a nova implementação, enquanto que o algoritmo DUAROUTER atinge primeiro esse equilíbrio.

Esta nova abordagem pode ser melhorada caso seja introduzida no dispositivo de roteamento a possibilidade de verificação do caminho que os veículos, que se encontram à frente de um veículo com dispositivo implementado, irão efectuar, possibilitando assim análise

periódica do estado da rede muito mais eficiente.

Outra implementação que foi investigada, para o melhoramento da nova abordagem, é descrita no capítulo 6, que consiste em fornecer mais tempo de sinal verde de semáforo às arestas mais congestionadas. No mesmo capítulo é apresentada uma possível implementação de um detector de acidentes.

Resumindo, pode-se concluir de que com a nova abordagem de roteamento se efectuou uma redução significativa dos valores dos tempos de viagem dos veículos em comparação com o algoritmo DUAROUTER. Estima-se que estes valores ainda possam ser melhorados caso sejam introduzidos no sistema as três implementações referidas anteriormente, constante verificação das arestas que os veículos irão percorrer, sistema equilibrado de comutação de fases de semáforo e detector de acidentes.

Apêndice A

Algoritmos de Roteamento

NESTE apêndice é ilustrado o funcionamento geral dos diversos algoritmos de encaminhamento abordados anteriormente.

A.1 Algoritmo de Encaminhamento de Dijkstra

Pseudo-código do algoritmo de Dijkstra [38] :

- 1º passo: iniciam-se os valores:

para todo $v \in V[G]$ $d[v] \leftarrow \infty$ $\Pi[v] \leftarrow \text{nulo}$ $d[s] \leftarrow 0$

$V[G]$ é o conjunto de vértices(v) que formam o Grafo G . $d[v]$ é o vector de distâncias de s até cada v . Admitindo-se a pior estimativa possível, o caminho infinito. $\Pi[v]$ identifica o vértice de onde se origina uma conexão até v de maneira a formar um caminho mínimo.

- 2º passo: temos que usar dois conjuntos: S , que representa todos os vértices v onde $d[v]$ já contém o custo do menor caminho e Q que contém todos os outros vértices.
- 3º passo: realizamos uma série de relaxamentos das arestas, de acordo com o código:

enquanto $Q \neq \emptyset$ $u \leftarrow \text{extraia-mín}(Q)$ $S \leftarrow S \cup u$ para cada v adjacente a u se $d[v] > d[u] + w(u, v)$ então $d[v] \leftarrow d[u] + w(u, v)$ $\Pi[v] \leftarrow u$

$w(u, v)$ é o peso (weight) da aresta que vai de u a v .

u e v são vértices quaisquer e s é o vértice inicial.

No final do algoritmo teremos o menor caminho entre s e qualquer outro vértice de G .

A.2 Algoritmo de Bellman-Ford-Moore

Pseudo-código do algoritmo de Bellman-Ford-Moore [37]:

- Passo 1: Inicializar o grafo para cada vértice v em vértices fazer: se v é origem então: $v.\text{distância} = 0$ senão: $v.\text{distância} := \text{infinito}$ $v.\text{anterior} := \text{nulo}$
- Passo 2: Ajustar as arestas repetidamente repetir tamanho (vértices) vezes: para cada aresta uv em arestas fazer: $u := uv.\text{origem}$ $v := uv.\text{destino}$ se $v.\text{distância} > u.\text{distância} + uv.\text{peso}$ então: $v.\text{distância} := u.\text{distância} + uv.\text{peso}$ $v.\text{anterior} := u$
- Passo 3: Verificar a existência de ciclos com peso negativo para cada aresta uv em arestas fazer: $u := uv.\text{origem}$ $v := uv.\text{destino}$ se $v.\text{distância} > u.\text{distância} + uv.\text{peso}$ então: erro "O grafo contém um ciclo de peso negativo."

A.3 Condições de equilíbrio de Wardrop

Segundo *Wardrop* o equilíbrio óptimo de roteamento pode ser encontrado através da resolução do seguinte problema de programação não-linear [39]:

$$\min \sum_a \int_0^{X_a} \tau_a(x) dx \quad (\text{A.1})$$

Esta condição encontra-se sujeita às seguintes condições:

$$X_a = \sum_i \sum_j \sum_r \alpha_{ij}^{ar} x_{ij}^r \quad (\text{A.2})$$

$$\sum_r x_{ij}^r = T_{ij} \quad (\text{A.3})$$

$$X_a \geq 0, x_{ij}^r \geq 0 \quad (\text{A.4})$$

x_{ij}^r representa o número de veículos num caminho r entre a origem i e o destino j . $\alpha_{ij}^{ar} = 1$ se um *link* a encontra-se num caminho r desde i até j . Caso contrário assume o valor de 0

Apêndice B

Simulation of Urban MObility

NESTE apêndice é apresentado o formato dos diversos ficheiros de configuração existentes no SUMO, as definições de nodos, arestas, conexões e a criação de rotas no SUMO. A implementação de emissores e semáforos, princípios básicos de funcionamento dos algoritmos JFRROUTER e DFROUTER e um manual do utilizador da GUI do SUMO também são apresentados.

B.1 Configuração de nodos no SUMO

Os ficheiros de descrição dos nodos têm a extensão “.nod.xml”, em que cada nodo é descrito normalmente por:

```
<node id="<STRING>" x="<FLOAT>" y="<FLOAT>" [type="<TYPE>"]/>
```

Os parâmetros têm o seguinte significado:

- *id* → Representa o nome do nodo
- *x* → Representa a posição *x* do nodo no plano, em metros
- *y* → Indica a posição *y* do nodo no plano, em metros

- **type** → Significa um tipo opcional de nodo. Existem dois tipos possíveis para o parâmetro *type*:
 - **priority** → Estabelece que dentro de um nodo é seguida a regra básica de prioridade do tráfego de veículos, ou seja, os veículos que se apresentam pela direita, têm prioridade sobre os outros.
 - **traffic_light** → Define se um nodo é controlado por semáforos.

B.2 Configuração de arestas no SUMO

Os ficheiros-arestas têm a extensão “.edg.xml”, sendo a descrição das estradas semelhante à descrição dos nodos, excepto em alguns parâmetros :

```
<edge id="<STRING>"(fromnode="<NODE_ID>"tonode="<NODE_ID>"|
xfrom="<FLOAT>"yfrom="<FLOAT>"xto="<FLOAT>"yto="<FLOAT>")
[type="<STRING>"| nolanes="<INT>"speed="<FLOAT>"priority="<INT>"
length="<FLOAT>")] [shape="<2D_POINT> [ <2D_POINT>] <2D_POINT>"]
[spread_type="center"] [function=( "source"| "sink"| "normal")]/>
```

Os parâmetros têm o seguinte significado:

- **id** → Representa o nome da aresta
- **fromnode** → Identifica o nodo de origem da aresta
- **tonode** → Define o nodo de destino da aresta
- **xfrom** → Identifica a posição *x* do nodo de origem da aresta
- **yfrom** → Identifica a posição *y* do nodo de origem da aresta
- **xto** → Representa a posição *x* do nodo de destino da aresta
- **yto** → Representa a posição *y* do nodo de destino da aresta
- **type** → Efectua a ligação a um determinado ficheiro que define o tipo de estradas. A descrição dos tipos de arestas também pode ser efectuada através dos parâmetros:
 - **nolanes** → Define o número de vias de cada estrada
 - **speed** → Define a velocidade máxima permitida
 - **priority** → Indica a prioridade das arestas

- *length* → Define o comprimento da estrada
- *shape* → Lista as posições da aresta
- *spread.type* → Consiste na descrição de como as estradas se propagam

B.3 Configuração de conexões de arestas no SUMO

O ficheiro conexão especifica quais as arestas, que abandonaram uma junção, podem ser alcançadas por uma determinada aresta que está a entrar nesta junção e, opcionalmente, quais as rotas que podem ser usadas em ambos os lados. Para estabelecer a ligação entre determinadas arestas e definir o número de rotas conectadas utiliza-se o seguinte código:

```
<connection from="<FROM_EDGE_ID>" to="<TO_EDGE_ID>"
lane="<INT_1>:<INT_2>" />
```

Os parâmetros têm o seguinte significado:

- *From* → Nome da aresta que os veículos abandonam
- *To* → Nome da aresta que os veículos tem de alcançar
- *Lane* → Número das rotas conectadas

B.4 Ficheiro de Configuração de uma rede SUMO

Protótipo base do ficheiro de configuração de uma rede SUMO. Os parâmetros de indicação dos ficheiros de descrição dos nodos e das arestas, bem como a indicação do ficheiro de saída, são obrigatórios.

```
<configuration>
<input
xml-node-files="<NODE_FILE>"
xml-edge-files="<EDGE_FILE>"
```

```

xml-connection-files=" <CONNECTION_FILE>"
/>
<output
output-file=" <NET_FILE>"
/>
</configuration>

```

B.5 Criação de rotas no SUMO

Para criar rotas através de um ficheiro de configuração “.rou.xml”, o utilizador terá de definir não só a origem e destino dessas rotas, bem como todas as arestas que as compõem. Antes desse processo, é necessário descrever o formato dos veículos que percorrerão essas rotas. Para definir o formato de determinado veículo utiliza-se a seguinte definição:

```

<vtype id=" <STRING>" accel=" <FLOAT>" decel=" <FLOAT>"
sigma=" <FLOAT>" length=" <FLOAT>" maxspeed=" <FLOAT>" />

```

Os parâmetros têm o seguinte significado:

- ***Id*** → Identificação do tipo de veículo
- ***Accel*** → Capacidade de aceleração do veículo ($(m/s)^2$)
- ***decel*** → Capacidade de desaceleração do veículo ($(m/s)^2$)
- ***sigma*** → Imperfeição do condutor (valor entre 0 e 1)
- ***length*** → Comprimento do veículo (metros)
- ***maxspeed*** → Velocidade máxima permitida ao veículo
- ***color*** → Parâmetro, opcional, que define a cor pretendida para este tipo de veículo. São necessários indicar três valores, de 0 a 1, que representam a intensidade das cores vermelha, verde e azul.

Findo este processo de criação de tipos de veículos, o utilizador pode criar de seguida os veículos e as rotas que estes irão percorrer. A sintaxe a utilizar para construir um veículo, de um determinado tipo definido anteriormente, é a seguinte:

```
<vehicle id="<STRING>" type="<STRING>" [route="<STRING>"]
depart="<FLOAT>" [period="<FLOAT>"] [repno="<INT>"]
color="<FLOAT>,<FLOAT>,<FLOAT>">
```

Os parâmetros têm o seguinte significado:

- *id* → Identificação do veículo
- *type* → Identificação do tipo de veículo a que este veículo pertence. É necessário definir o tipo anteriormente
- *route*: → Rota que este veículo irá efectuar. Parâmetro opcional, mas caso seja atribuído, a definição desta rota tem necessariamente de ser efectuada antes da definição deste veículo
- *depart* → Tempo em que o veículo inicia a rota
- *period* → Período de tempo em que é efectuado o lançamento deste veículos. Parâmetro opcional que funciona em conjunto com o parâmetro *repno*
- *repno* → Número de veículos idênticos que serão emitidos a um período definido anteriormente
- *color* → Parâmetro, opcional, que define a cor pretendida para este veículo. São necessários indicar três valores, de 0 a 1, que representam a intensidade das cores vermelha, verde e azul.

Por fim define-se a rota, que o veículo criado anteriormente irá percorrer, através de:

```
<route id="<STRING>" edges="<STRING>"
color="<FLOAT>,<FLOAT>,<FLOAT>" />
```

Os parâmetros têm o seguinte significado:

- *id* → Identificação da rota
- *edges* → Lista de arestas que constituem a rota. A ordem de listagem corresponderá à ordem das arestas que o veículo irá percorrer.

- **color** → Parâmetro, opcional, que define a cor pretendida para este veículo. São necessários indicar três valores, de 0 a 1, que representam a intensidade das cores vermelha, verde e azul.

B.6 Sistema de comutação do sinal dos semáforos no SUMO

Além de possuírem o seu sistema de comutação, SUMO permite aos utilizadores do *software* implementarem o seu próprio sistema de comutação. Essa definição pode ser efectuada na criação ou conversão da rede, ou num ficheiro adicional que será lido pelo ficheiro de configuração. A definição de um sistema de comutação de semáforos é efectuada da seguinte forma:

```

<tl-logic type="<TYPE>">
  <key><STRING></key>
  <subkey><INT></subkey>
  <phaseno><INT></phaseno>
  <offset><INT></offset>
  <phase duration="<INT>" phase="<BITS>" brake="<BITS>"
  yellow="<BITS>" />
</tl-logic>

```

Os parâmetros têm o seguinte significado:

- **tl-logic** → Indica o início da descrição de um sistema de comutação de semáforos. Recebe como atributo o tipo de algoritmo de comutação, *static*, *actuated* ou *agent-based*
- **key** → indica qual o nodo, com semáforo definido, onde será implementado este sistema de comutação
- **subkey** → Sub-chave da definição. Utilizado para identificar o programa de definição dos sinais dos semáforos
- **phaseno** → Número de fases que o sistema possui
- **offset** → Offset com o qual o semáforo deverá comutar para a fase seguinte

- **phase** → Definição da fase de comutação dos sinais de semáforo. Contém os seguintes parâmetros:
 - **duration** → Tempo de duração da fase
 - **phase** → Lista de *bits*, em que os *bits* a 1 indicam quais as arestas que neste momento encontram-se com sinal de semáforo verde. Os *bits* a 0 indicam quais as arestas que neste momento encontram-se com sinal de semáforo vermelho
 - **brake** → lista de *bits*, em que os *bits* a 1 indicam quais as arestas em que é necessário reduzir a velocidade caso ao entrar nestas
 - **yellow** → Lista de *bits*, em que os *bits* a 1 indicam quais as arestas que neste momento encontram-se com sinal de semáforo amarelo

O utilizador tem a possibilidade de definir vários sistemas de fases de comutação dos sinais e definir um determinado tempo de actuação de cada sistema de fases. A definição de comutação entre os diferentes programas é efectuada pelo sistema WAUT, sendo definida da seguinte forma:

```
<WAUT refTime="<INT>"id="<STRING>"startProg="<STRING>">
  <wautSwitch time="<INT>"to="<STRING>" />
</WAUT>
```

Os parâmetros têm o seguinte significado:

- **refTime** → Tempo de referência onde o sistema de comutação entre programas é iniciado
- **id** → Identificação do sistema WAUT
- **startProg** → Identificação do programa que iniciará o sistema
- **time** → Segundo da simulação em que ocorrerá uma comutação de programas
- **to** → Identificação do programa de fases que irá substituir o anterior programa

Adicionalmente é necessário definir o nodo onde será efectuada a comutação de programas e o algoritmo de comutação:

```
<wautJunction wautID="<STRING>"junctionID="<STRING>"
[procedure="<STRING>"] [synchron]
```

Os parâmetros têm o seguinte significado:

- *wautID* → Identificação do sistema WAUT
- *junctionID* → Identificação do nodo onde será efectuada a comutação de programas
- *procedure* → Parâmetro opcional que define o algoritmo de comutação a utilizar
- *synchron* → Atributo opcional que indica se a comutação deve ser efectuada de forma síncrona. Por defeito é falso

B.7 Definição de emissores no SUMO

O SUMO possibilita além da definição de veículos e suas rotas, a implementação de emissores de veículos colocados em posições da rede definidas pelo utilizador. Em cada emissor o utilizador define os veículos a injectar na rede e o momento de partida destes. O código a introduzir, num ficheiro adicional (lido posteriormente pelo ficheiro de configuração), para implementar um emissor é seguinte:

```
<emitter id="<STRING>"pos="<FLOAT>"lane="<LANE_ID>"
[friendly_pos="x"] file="<DEFINITION_FILE>" />
```

Os parâmetros têm o seguinte significado:

- *id* → Identificação do emissor
- *pos* → Posição, em metros, na faixa onde será colocado o emissor
- *lane* → Faixa da estrada onde será colocado o emissor
- *friendly_pos* → Parâmetro opcional que coloca o emissor no final da faixa, quando a posição definida pelo parâmetro *pos* encontra-se mal definida.
- *file* → Parâmetro que recebe o nome do ficheiro onde se encontram as definições de funcionamento do emissor

Dentro do ficheiro de definição do funcionamento do emissor é introduzida a informação respeitante ao(s) veículo(s) a emitir:

```
<triggeredsource>
  <emit [id=" <STRING>"] time=" <INT>" [type=" <STRING>"]
  route=" <STRING>" [speed=" <FLOAT>"] />
</triggeredsource>
```

Os parâmetros têm o seguinte significado:

- *id* → Identificação do veículo a emitir. Caso não se encontre definido, o sistema atribui uma identificação por defeito
- *type* → Parâmetro opcional que indica o tipo de veículos a que o veículo a emitir deve pertencer
- *time* → Período temporal em que será emitido o veículo
- *route* → Identificação da rota que o veículo a emitir irá percorrer
- *speed* → Velocidade máxima permitida ao veículo a emitir

B.8 Definição de detectores pelo DFROUTER

Como foi referido no capítulo 4.4, o algoritmo de roteamento DFROUTER utiliza os dados resultantes de detectores, que se encontram colocados por toda a rede, para criar as melhores rotas possíveis. A definição do local onde estão colocados os detectores, é efectuada da seguinte forma:

```
<detector_definition id=" <STRING>" lane=" <LANE_ID>" pos=" <FLOAT>" />
```

Os parâmetros têm o seguinte significado:

- *id* → Identificação do detector

- *lane* → Identificação da faixa onde será colocado o detector
- *pos* → Posição da faixa onde será colocado o detector

Como resultado desta operação, a aplicação indicará qual o tipo de detector a utilizar, ou seja, se o detector é do tipo *source*, *sink*, *between* ou *discarded*.

Um detector *source*, colocado no início de uma aresta, indica a posição em que os veículos iniciarão a sua rota, um detector *sink* determina onde estes abandonam a rede e um detector *between* é colocado noutra posição da aresta, servindo também como local de partida dos veículos. Quando a compilação apresenta como resultado um detector do tipo *discarded* significa que algum parâmetro se encontra mal definido.

Utilizando as rotas calculadas e uma lista de fluxos criada anteriormente pelo utilizador, DFROUTER implementa de seguida os fluxos que circularão na rede. A lista de fluxos tem necessariamente de estar inserida num ficheiro de formato CSV, com o aspecto seguinte:

Detector;Time;qPKW;qLKW;vPKW;vLKW

Os parâmetros têm o seguinte significado:

- **Detector** → Identificação do detector a utilizar
- **Time** → Período temporal em que é efectuada a detecção de veículos
- **qPKW** → Número de veículos ligeiros que são detectados durante o período de detecção
- **qLKW** → Número de veículos de transporte que são detectados durante o período de detecção
- **vPKW** → Velocidade média dos veículos ligeiros que são detectados durante o período de detecção
- **vLKW** → Velocidade média dos veículos de transporte que são detectados durante o período de detecção

B.9 Definição de rotas segundo o algoritmo JTRROUTER

A sintaxe a utilizar para criar uma lista de rotas utilizando o JTRROUTER é a seguinte:

```
jtrrouter -flows=<FLOW_DEFS> -turns=<TURN_DEFINITIONS>
-net=<SUMO_NET> -output-file=MySUMORoutes.rou.xml -b <INT>
-e <INT> -alternatives=MySUMORoutes.rou.alt.xml
```

Os parâmetros têm o seguinte significado:

- **-flows** → Opção que define qual o ficheiro que contém as definições de fluxo
- **-turns** → Indica qual o ficheiro em que se encontra a definição da aresta *sink*. A aresta *sink* é a aresta que o veículo utiliza para abandonar a rede.
- **-net** → Define qual a rede a utilizar
- **-output-file** → Indica qual o ficheiro de saída. No ficheiro de saída constará as rotas criadas por esta aplicação. Também um ficheiro com as rotas alternativas será gerado, tendo a extensão “.alt.xml”
- **-b** → Parâmetro opcional que indica o tempo em que o JTRROUTER inicia a criação de rotas
- **-e** → Parâmetro opcional que indica o tempo em que o JTRROUTER termina a criação de rotas
- **-alternatives** → Comando opcional, que força o JTRROUTER a ler um ficheiro que contém uma lista de rotas alternativas
- **-turn-defaults** → Parâmetro opcional que atribui automaticamente probabilidades às arestas que estão conectadas à aresta origem
- **-max-edge-factor** → Este comando é utilizado para limitar o tamanho da rota, pois não há garantias temporais de quando é que o veículo atinge a aresta de saída, ou seja, a aresta definida em *sink*

B.10 Ficheiro de Configuração de uma simulação SUMO

Protótipo base do ficheiro de configuração de uma simulação SUMO. Os parâmetros de indicação do ficheiro de descrição da rede a utilizar, bem como a indicação do ficheiro

de rotas a efectuar, são parâmetros obrigatórios. A definição do tempo inicial e final de simulação também é necessária. Todos os ficheiros adicionais são colocados, separados por vírgulas, no parâmetro **additional-files**.

```
<configuration>
<files
net-file=" <NET_FILE>"
route-files=" <ROUTE_FILE>"
additional-files=" <ADDITIONAL_FILES>"
/>
<time
begin=" <BEGIN_TIME>"
end=" <END_TIME>"
/>
</configuration>
```

B.11 Manual de utilização da GUI do SUMO

Dentro da GUI do SUMO encontram-se disponíveis várias ferramentas e opções de configuração e simulação. O acesso a essas opções pode ser efectuado através dos vários menus existentes:

- **Menu FILE**

- **Open Simulation** → Abre uma caixa de diálogo onde se pode escolher um ficheiro de configuração SUMO que descreve uma simulação completa.
- **Open Network** → Abre uma caixa de diálogo onde se pode escolher um ficheiro que ilustra uma rede SUMO
- **Reload Simulation** → Recarrega a última simulação aberta
- **Clear Recent Configurations** → Apaga a lista de configurações recentes
- **Clear Recent Networks** → Apaga a lista de redes recentes

- *Close* → Fecha a simulação carregada
- *Quit* → Fecha o simulador de tráfego SUMO
- **Menu EDIT**
 - *Edit Chosen* → Abre uma lista onde é possível carregar/gravar e editar itens de escolha
 - *Edit Breakpoints* → Permite editar, carregar e salvar *breakpoints*(pontos de paragem). Para já a simulação vai parar num dos *breakpoints* e pode ser continuada premindo o botão **Play**
- **Menu SETTINGS**
 - *Application Settings* → Define se a aplicação é fechada automaticamente quando a simulação carregada termina
- **Menu WINDOWS**
 - *Show Status Line* → Item onde se pode comutar o estado da linha entre *on* e *off*
 - *Show Message Window* → Item onde se pode comutar a janela de mensagens entre *on* e *off*
 - *Show Simulation Time* → Item onde se pode comutar a janela que exhibe o tempo de simulação entre *on* e *off*
 - *Show Simulation Delay* → Item onde se pode comutar a janela que exhibe o atraso da simulação entre *on* e *off*
 - *Show Tool Bar* → Item onde se pode comutar a barra de ferramentas entre *on* e *off*
 - *Tile Horizontally* → Reordena a posição das janelas
 - *Tile Vertically* → Reordena a posição das janelas
 - *Cascade* → Reordena a posição das janelas
 - *Close* → Fecha a janela mais elevada
 - *Clear Message Window* → Apaga todos os conteúdos da janela de mensagens
- **Menu HELP**
 - *About* → Exibe uma pequena janela com pequena informação sobre o SUMO, como por exemplo a sua versão
- **Barra de Ferramentas**
 - *Botão de Abrir Configuração* → Abre uma caixa de diálogo onde se pode escolher um ficheiro de configuração SUMO que descreve uma simulação completa

- **Botão de Abrir Rede** → Abre uma caixa de diálogo onde se pode escolher um ficheiro que ilustra uma rede SUMO
- **Botão de Recarregamento** → Recarrega a última simulação aberta
- **Botão de Iniciar** → Inicia a simulação. Se uma simulação carregada não tiver começado antes, esta inicia-se numa etapa descrita pelo parâmetro *begin* dentro do ficheiro de configuração carregado. Se a situação tiver começado e parar, ela depois vai continuar. Não é possível reiniciar a simulação, é necessário recarregá-la de novo
- **Botão de Paragem** → Pára uma aplicação que está a ser executada. Uma aplicação parada pode ser continuada usando o botão *play*
- **Botão de única etapa** → Executa uma única etapa da simulação
- **Campo de Informação Temporal** → Depois de a simulação carregada ter iniciado, a informação sobre o tempo da etapa corrente é indicado aqui
- **Controlo de Velocidade da Simulação** → Usando este controlo pode-se alterar o tempo que a aplicação espera entre duas etapas da simulação. Quanto maior o valor, mais lenta a aplicação vai correr
- **Nova Visualização Microscópica** → Abre uma nova janela que exhibe as ruas, e os veículos a moverem-se nestas
- **Localização de Junção** → Abre uma janela que permite a escolha de uma determinada junção da rede. Pressionando **OK** na junção pretendida, esta será exibido em pormenor
- **Localização de Aresta** → Abre uma janela que permite a escolha de uma determinada aresta da rede. Pressionando **OK** na aresta pretendida, esta será exibido em pormenor
- **Localização de Semáforo** → Abre uma janela que permite a escolha de uma determinado semáforo. Pressionando **OK** no semáforo pretendido, este será exibido em pormenor
- **Localização de uma estrutura adicional** → Abre uma janela que permite a escolha de uma estrutura adicional. Pressionando **OK** na estrutura adicional pretendida, esta será exibido em pormenor
- **Localização de uma forma de aresta** → Abre uma janela que permite a escolha de uma forma de aresta. Pressionando **OK** na forma de aresta pretendida, esta será exibido em pormenor

• Definições de Visualização

- **Visualização Recente** → Botão usado para restaurar uma vista que exhibe toda a rede. Depois de pressionar este botão, a vista vai ser a mesma que foi exibida a seguir ao carregamento da simulação. O factor *zoom* vai ser restaurado para um valor que permita a visualização da área total da rede e o centro da rede carregada vai ser colocado no centro da visualização

- **Controlos de visualização microscópica adicionais**

- **Definições de monitorização**

- * *Localização de Veículo* → Abre uma janela que permite escolher um nome de um veículo de uma lista de veículos que no momento estão a percorrer a pista. Pressionando **OK** o veículo escolhido vai ser ampliado. Esta função ainda está em desenvolvimento pelo que não é aconselhado o seu uso

- **Definições de Visualização**

- * *Change Vehicle Colouring Scheme* → Permite escolher um determinado parâmetro que vai alternar a cor dos veículos. Os seguintes esquemas de cor estão disponíveis para os veículos:
 - *by speed* → Cor do veículo alterna entre azul e vermelho enquanto a velocidade desce
 - *specified* → São usadas as cores do veículo carregado
 - *by type* → Veículos são coloridos consoante o seu tipo
 - *by route* → Veículos são coloridos consoante as rotas que vão utilizar
 - *random* → Cor dos veículos é alternada aleatoriamente. Existe dois métodos de alternar a cor aleatoriamente, aleatoriamente ou quando este muda de rota

Bibliografia

- [1] F. H. Administration. Fhwa. <http://www.fhwa.dot.gov/>, (consultado: 2 de Setembro de 2009).
- [2] P. P. T. V. AG. Vissim. <http://www.ptvag.com/>, (consultado: 2 de Setembro de 2009).
- [3] M. Behrisch, D. Krajzewicz, and Y.-P. Wang. Comparing performance and quality of traffic assignment techniques for microscopic road traffic simulations. 2008.
- [4] D. P. Bertsekas. A simple and fast label correcting algorithm for shortest paths. volume 23, pages 703–709, 1993.
- [5] I. Brejc. Kosmos. <http://wiki.openstreetmap.org/wiki/Kosmos>, (consultado: 2 de Setembro de 2009).
- [6] U. S. C. Bureau. Tiger. <http://www.census.gov/geo/www/tiger/>, (consultado: 2 de Setembro de 2009).
- [7] S. Coast and N. Black. Cloudmade. <http://www.cloudmade.com/>, (consultado: 2 de Setembro de 2009).
- [8] W. W. W. Consortium. Xml. <http://www.w3.org/XML/>, (consultado: 2 de Setembro de 2009).
- [9] C. Corporation. Transmodeler. <http://www.caliper.com/transmodeler/default.htm>, (consultado: 2 de Setembro de 2009).
- [10] K. B. Davidson. A flow-travel time relationship for use in transportation planning. In *In Proceedings of the Australian Road Research Board*, volume 3, 1966.

- [11] E. Dijkstra. *Numerische Mathematik*. 1959.
- [12] A. A. e Dave Stubbs. Cyclemap. <http://www.opencyclemap.org/>, (consultado: 2 de Setembro de 2009).
- [13] ESRI. Arcview. <http://www.esri.com/software/arcview/>, (consultado: 2 de Setembro de 2009).
- [14] C. Gawron. Simulation-based traffic assignment. Master's thesis, Universidade de Koln, Fevereiro 1998.
- [15] ITS-GAC. Its-gacs. <http://www.dlr.de/ts/en/>, (consultado: 2 de Setembro de 2009).
- [16] ITS-GAC. Sumo. <http://sumo.sourceforge.net/>, (consultado: 2 de Setembro de 2009).
- [17] D. Krajzewicz, E. Brockfeld, J. Mikat, J. Ringel, C. Rössel, W. Tuchscheerer, P. Wagner, and R. Woesler. Simulation of modern traffic lights control systems using the open source traffic simulation sumo. In *Proceedings of the 3rd Industrial Simulation Conference*, 2205.
- [18] Landsat.org. Landsat. <http://www.landsat.org/>, (consultado: 2 de Setembro de 2009).
- [19] Mapnik. Mapnik. <http://mapnik.org/>, (consultado: 2 de Setembro de 2009).
- [20] NPE. Mapas npe. <http://www.npemap.org.uk/>, (consultado: 2 de Setembro de 2009).
- [21] U. B. of Public Roads. *Traffic Assignment Manual*. U.S. Department of Commerce. U.S. Bureau of Public Roads, Washington, D.C., 1964.
- [22] OpenMapsProject. Openmaps. <http://openmaps.eu/>, (consultado: 2 de Setembro de 2009).
- [23] OpenStreetMap. Openstreetmap. www.OpenStreetMap.org, (consultado: 2 de Setembro de 2009).

-
- [24] OpenStreetMap. Openstreetmap-wiki. <http://wiki.openstreetmap.org/>, (consultado: 2 de Setembro de 2009).
- [25] OpenStreetMap. Osm export. <http://sourceforge.net/apps/mediawiki/sumo/index.php?title=OpenStreetMapImport>, (consultado: 2 de Setembro de 2009).
- [26] Osmarender. Osmarender. <http://wiki.openstreetmap.org/wiki/Osmarender>, (consultado: 2 de Setembro de 2009).
- [27] Q. Paramics. Paramics. http://www.paramics-online.com/product_estimator.php, (consultado: 2 de Setembro de 2009).
- [28] Potlach. Potlach. <http://potlatchosm.wordpress.com/>, (consultado: 2 de Setembro de 2009).
- [29] M. Power. Elmar. <http://www.marketingpower.com>, (consultado: 2 de Setembro de 2009).
- [30] M. Project. Merkaator. <http://www.merkaator.org/>, (consultado: 2 de Setembro de 2009).
- [31] W. Software. Wget. <http://wget.addictivecode.org/>, (consultado: 2 de Setembro de 2009).
- [32] C. A. T. Study. Cats. <http://www.catsmpo.com/>, (consultado: 2 de Setembro de 2009).
- [33] M. M. Technology. Corsim. <http://mctrans.ce.ufl.edu/>, (consultado: 2 de Setembro de 2009).
- [34] TRAC. Josm. <http://josm.openstreetmap.de/>, (consultado: 2 de Setembro de 2009).
- [35] TrafficWare. Sim traffic. <http://www.trafficware.com/>, (consultado: 2 de Setembro de 2009).

- [36] M. Wiering. Utrecht university - intelligent traffic light control. www.ercim.org/publication/Ercim_News/enw53/wiering.html, (consultado: 2 de Setembro de 2009).
- [37] Wikipedia. Algoritmo de bellman-ford-moore. http://pt.wikipedia.org/wiki/Algoritmo_de_Bellman-Ford, (consultado: 2 de Setembro de 2009).
- [38] Wikipedia. Algoritmo de dijkstra. http://pt.wikipedia.org/wiki/Algoritmo_de_Dijkstra, (consultado: 2 de Setembro de 2009).
- [39] Wikipedia. Route assignment. http://en.wikipedia.org/wiki/Route_assignment, (consultado: 2 de Setembro de 2009).
- [40] Yahoo! Yahoo! - maps. <http://maps.yahoo.com/>, (consultado: 2 de Setembro de 2009).

Índice

- Algoritmo de Bellman-Ford-Moore, ii, 5
- Algoritmo de Dijkstra, i, 5
- Algoritmo Vector-Distância, 5
- Bureau of Public Roads, 6
- CloudMade, 12
- CORSIM, 2
- Criação de rotas, 22
- Detector de acidentes, 52
- DFROUTER, 32
- DUAROUTER, 29
- Emissores, 24
- Gawron, 7, 30
- ITS-GAC, 38
- JOSM, 13
- JTRROUTER, 32
- Mapnik, 14
- NETCONVERTER, 21
- NETworkGENerator, 21
- Nova abordagem ao DUAROUTER, 39
- OpenMaps, 11
- OpenStreetMap, 11
- Osmarender, 14
- Paramics, 3
- Roteamento, 3
- Roteamento dinâmico, 35
- Semáforos, 49
- Sim Traffic, 3
- SUMO, v, 3, 19
- TransModeler, 2
- VISSIM, 2
- Wardrop, ii, 7
- Wget, 15

