Pós-Graduação em Ciência da Computação

“Pilgrim: UM SISTEMA PARA GERAÇÃO E CLASSIFICAÇÃO DE ROTAS DE ÔNIBUS SENSÍVEL AO CONTEXTO”

Por

Felipe Silveira Mello Borgiani

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, OUTUBRO/2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FELIPE SILVEIRA MELLO BORGIANI

"Pilgrim: UM SISTEMA PARA GERAÇÃO E CLASSIFICAÇÃO DE ROTAS DE ÔNIBUS SENSÍVEL AO CONTEXTO"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADORA: Patrícia Cabral de Azevedo Restelli Tedesco

RECIFE, OUTUBRO/2013

Catalogação na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Borgiani, Felipe Silveira Mello
Pilgrim: um sistema para geração e classificação de rotas de ônibus sensível ao contexto / Felipe Silveira Mello Borgiani. - Recife: O Autor, 2013.
96 f.: il., fig., tab., quadro

Orientador: Patrícia Cabral de Azevedo Restelli Tedesco.
Dissertação (mestrado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2013.

Inclui referências.

1. Inteligência Artificial. 2. Computação Sensível ao Contexto. I. Tedesco, Patrícia Cabral (orientadora). II. Título.

006.3 CDD (23. ed.) MEI2013 – 152

Dissertação de Mestrado apresentada por **Felipe Silveira Mello Borgiani** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Pilgrim: Um Sistema para Geração e Classificação de Rotas de Ônibus Sensível ao Contexto**" orientada pela **Profa. Patrícia Cabral de Azevedo Restelli Tedesco** e aprovada pela Banca Examinadora formada pelos professores:

_____

Prof. Ricardo Bastos Cavalcante Prudêncio
Centro de Informática / UFPE

_____

Profa. Jeane Cecília Bezerra de Melo
Departamento de Estatística e Informática /UFRPE

_____

Profa. Patrícia Cabral de Azevedo Restelli Tedesco
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 6 de setembro de 2013.

_____

Prof. Edna Natividade da Silva Barros
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Dedico este trabalho a meus pais, pelo seu amor incondicional, pelo esforço para me propiciar uma educação de qualidade, e por sempre me incentivarem e apoiarem.

Aos meus avós, pelo amor imenso, cuidado, esforço e lições, sem os quais não seria quem sou hoje.

À minha amada esposa, pelo carinho, cuidado, incentivo, e ajuda necessários para que eu pudesse concluir este trabalho.

Aos meus amigos, antigos e novos, por sempre me apoiarem e me darem forças, e pelo carinho inestimável.

Finalmente, aos tantos mestres que tive a honra de encontrar, com quem aprendi tantas lições de sabedoria, e que me ensinaram a lutar por meus sonhos.

AGRADECIMENTOS

Foram muitos os desafios enfrentados e as dificuldades superadas durante o desenvolvimento desta dissertação de mestrado, e tomo este espaço para agradecer formalmente a algumas pessoas e instituições, cuja ajuda e apoio foram fundamentais para a viabilização deste trabalho.

Agradeço primeiramente à minha orientadora, Professora Patrícia Tedesco, pela confiança em mim depositada, pelo incentivo e pelas sempre sábias recomendações e indicações, que me guiaram no desenvolvimento deste trabalho.

Também agradeço aos colegas do projeto UbiBus, professores e alunos, por sua garra, seu apoio, e suas contribuições que se mostraram de grande importância para esta dissertação.

Agradeço aos colegas de trabalho, em especial Anderson Paulo e Danielle Nathália, pelas sugestões, críticas construtivas, e acima de tudo pelo apoio, que me guiaram durante esta etapa.

Por fim, agradeço também aos Professores Ricardo Prudêncio, Jeane Melo e Patrícia Tedesco, por terem aceito participar de minha banca de defesa, cujas contribuições serão certamente de grande valia para esta pesquisa.

“Busque ser um homem de valor, em vez de ser um homem de sucesso.”

(Albert Einstein)

RESUMO

O trânsito caótico das grandes cidades, especialmente em países em desenvolvimento, impacta diretamente na qualidade de vida dos cidadãos, principalmente daqueles que necessitam diariamente dos sistemas de transporte público rodoviários. Diversas abordagens para amenizar esses problemas vem sido apresentadas por pesquisadores do mundo todo, na forma de diferentes propostas de sistemas de transporte inteligentes. Tendo em vista o desenvolvimento tecnológico dos últimos anos, aliado à expansão e popularização do uso de dispositivos computacionais portáteis como smartphones e tablets, este trabalho propõe-se a apresentar um sistema, componente do Sistema de Transporte Inteligente UbiBus, denominado Pilgrim, capaz de gerar e classificar rotas de ônibus em tempo real, e seja sensível ao contexto que permeia o sistema de transporte público rodoviário, em especial o trânsito. Para tanto, serão propostas duas abordagens utilizando técnicas de otimização da inteligência artificial, Algoritmos Genéticos e Hill-Climbing com Reinício Aleatório, para o desenvolvimento do sistema, e apresentadas a arquitetura, a modelagem e os detalhes da implementação. Este trabalho ainda apresenta os experimentos realizados para avaliar o desempenho de ambas as abordagens, comparando-as, e também uma pesquisa feita com potenciais usuários do sistema UbiBus, com o objetivo de avaliar a qualidade das rotas geradas pelo sistema Pilgrim.

Palavras-chave: Sistemas de Transporte Inteligente (ITS), Sistemas Avançados de Transporte Público (APTS), Sistemas de Planejamento de Viagens, Contexto Computacional; UbiBus; Hill-Climbing, Algoritmos Genéticos.

ABSTRACT

The chaotic traffic of large cities, especially in developing countries, directly affects the quality of life of citizens, especially those who depend on road public transport systems. Several approaches to alleviate these problems has been made by researchers worldwide in the form of different proposals for intelligent transport systems, and given the technological development of recent years, coupled with the expansion and popularization of the use of portable computing devices such as smartphones and tablets, this work proposes to present a system, a component of the Intelligent Transportation System UbiBus, called Pilgrim, that is able to generate and classify bus routes in real time, and is sensitive to the context that permeates the road public transportation system, particularly transit. For this purpose, two approaches are presented for the development of the system using the Genetic Algorithms and Random Restart Hill-Climbing artificial intelligence techniques, as well as architecture, modeling and implementation details. This work also presents some experiments conducted to evaluate the performance on both approaches, comparing them, and also survey with potential users of the UbiBus system, aiming to evaluate the quality of the routes generated by the Pilgrim system.

Keywords: Intelligent Transportation Systems (ITS), Advanced Public Transportation Systems (APTS), Trip Planning Systems, Computational Context; UbiBus; Hill-Climbing, Genetic Algorithms.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Esta dissertação apresenta um sistema para geração de rotas de ônibus sensível ao contexto, denominado Pilgrim, que utiliza técnicas de otimização para a criação e classificação de rotas, sendo capaz de operar em qualquer região metropolitana do país, havendo dados suficientes sobre a malha viária e sobre o sistema rodoviário local. O sistema é parte da camada de middleware do Sistema de Transporte Inteligente (ITS[1]) UbiBus[2], estando disponível como um serviço para todas as aplicações desse ITS, especialmente para o seu Sistema de Recomendação de Rotas.

Serão apresentadas as principais referências presentes na literatura dentro do contexto de Sistemas de Transporte Inteligente, Contexto Computacional e Sistemas de Planejamento de Viagens, bem como alguns trabalhos relacionados. Será também apresentado o projeto UbiBus, e a posição deste trabalho como seu componente, detalhando também a arquitetura e a implementação do sistema Pilgrim, e explorando duas técnicas de otimização para a funcionalidade de geração e classificação de rotas, a saber: Algoritmos Genéticos e Hill-Climbing com Reinício Aleatório [Russell and Norvig 2009].

1.1 MOTIVAÇÃO

Grandes cidades estão sofrendo cada vez mais com o problema de mobilidade urbana. Muitas dificuldades, como constantes engarrafamentos, falta de investimentos em infraestrutura rodoviária e a baixa qualidade dos serviços oferecidos pelo transporte público tornam a vida diária dos cidadãos mais estressante e complicada [Chaves et al. 2011].

Ao mesmo tempo, o desenvolvimento tecnológico tem permitido a rápida proliferação de dispositivos móveis com alto poder de processamento, como smartphones e tablets, e um maior número de usuários com acesso móvel à internet diretamente desses dispositivos. Aliado a isso, o crescente volume de informações produzidas sobre aspectos relacionados ao trânsito, quer sejam provenientes de fontes oficiais ou de usuários através de redes sociais e interações com diversos sistemas, permite que novos sistemas capazes

[1] Do inglês, *Intelligent Transportation System*

[2] Mais informações em: http://www.cin.ufpe.br/~ubibus/

de consumir essas informações e gerar conteúdo útil para a população sejam desenvolvidos e se tornem cada vez mais acessíveis.

Nesse cenário, os Sistemas de Transporte Inteligente se mostram como alternativa mais que plausível para tentar amenizar os efeitos negativos da desorganização do trânsito, podendo ser aplicados diretamente no sistema de transporte público rodoviário, aumentando sua eficiência e melhorando a qualidade do serviço para o usuário final [Silva 2000].

Em 2010, Ferris et al. [2010b] constataram em sua pesquisa que a existência de um ITS que forneça informações em tempo real sobre ônibus melhora de forma geral a satisfação dos usuários do transporte público. Garantir a consistência dessas informações entretanto, é bastante difícil quando o ITS é aplicado em uma grande cidade como observam Padmanaban et al. [2010] e Vieira et al. [2011]. O trânsito desorganizado, as características climáticas, acidentes e outras ocorrências interferem no tempo de chegada dos ônibus nas paradas e consequentemente no tempo total da viagem.

Tendo em vista a característica desorganizada do trânsito em grandes cidades, especialmente em países em desenvolvimento, destaca-se como um serviço de grande utilidade para os usuários de transporte público rodoviário um sistema de planejamento de viagens de ônibus capaz de informar, dados os pontos de origem e destino e algumas preferências do usuário, um conjunto de rotas mais adequadas considerando a situação atual do trânsito.

Os Sistemas de Planejamento de Viagens são sistemas que auxiliam viajantes e potenciais viajantes em vários aspectos de suas viagens, incluindo informações que ajudem a decidir o meio de transporte a ser utilizado, o horário da viagem e quais rotas seguir. Utilizam dados estáticos e dinâmicos sobre o contexto do trânsito e dos transportes, especialmente em tempo real, para aprimorar a qualidade das informações passadas aos usuários [McCormack and Roberts 1996].

Diante desse cenário, foi proposto um projeto cujo objetivo é investigar, especificar e implementar soluções tecnológicas que incluam modelos, algoritmos e ferramentas para facilitar o acesso a informações de transporte público aos usuários, em tempo real, baseado em informações dinâmicas de contexto, integrados em um sistema de transporte público inteligente, ubíquo e sensível ao contexto, denominado UbiBus [Vieira et al. 2012].

O UbiBus é composto de vários componentes abertos, sejam modelos, algoritmos, ferramentas ou middleware, e tem como foco principal o suporte aos passageiros de transporte público, especialmente usuários de ônibus urbanos. O sistema tem uma característica fortemente ubíqua, visando ser acessível de diversos dispositivos e locais, e

é integrado a redes sociais já existentes, utilizando a inteligência coletiva para prover melhores serviços aos usuários. Como um todo, o sistema é sensível ao contexto que permeia o uso de transporte público, abrangendo o contexto do usuário, do trânsito e dos próprios veículos, utilizando de informações estáticas e dinâmicas, capturadas em tempo real, sobre vias, veículos, dispositivos e passageiros.

Dentro do sistema UbiBus faz-se necessária a existência de um módulo capaz de gerar rotas de ônibus sensíveis ao contexto do trânsito, dos veículos e dos usuários, baseadas em informações fornecidas pelos usuários, como ponto de origem, ponto de destino, necessidade de veículos com acessibilidade, entre outras preferências.

1.2 OBJETIVOS

A seguir serão apresentados os objetivos gerais e específicos para esta pesquisa.

1.2.1. OBJETIVO GERAL

Este trabalho tem como objetivo geral apresentar um modelo de sistema sensível ao contexto para geração de rotas de ônibus em uma região metropolitana, que utilize de técnicas de otimização da inteligência artificial para realizar uma busca pelas rotas mais adequadas às solicitações feitas por usuários. O sistema desenvolvido estará disponível como um serviço na camada de middleware do UbiBus, sendo utilizado principalmente pelo seu Sistema de Recomendação de Rotas.

1.2.2. OBJETIVOS ESPECÍFICOS

Como objetivos específicos, estre trabalho propõe-se a:

- Desenvolver um serviço, integrado ao UbiBus, capaz de gerar e classificar rotas de ônibus sensíveis ao contexto, dados uma origem, um destino, e um conjunto de preferências, como horário de início da viagem e número máximo de trocas de ônibus;
- Explorar duas técnicas da inteligência artificial, algoritmos genéticos e hill-climbing com reinício aleatório, no processo de geração e classificação de rotas de ônibus, comparando-as.

- Descrever a modelagem do problema de planejamento de viagens de ônibus sensíveis ao contexto aplicada no desenvolvimento do sistema;

1.3 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma:

O Capítulo 2 apresenta a fundamentação teórica essencial para o desenvolvimento e contextualização desta pesquisa e uma breve descrição dos trabalhos relacionados a Sistemas de Planejamento de Viagens.

O Capítulo 3 apresenta a proposta para o sistema de geração de rotas desenvolvido nesta pesquisa, explicando sua fundamentação, arquitetura e construção, indicando a forma como ele interage e se inter-relaciona com outros componentes do UbiBus.

O Capítulo 4 apresenta os experimentos realizados e os resultados deste trabalho, apontando aspectos como desempenho e integração a outros sistemas, e apontando os resultados de uma pesquisa feita com potenciais usuários.

O Capítulo 5 apresenta algumas considerações finais sobre este trabalho, apontando as principais lições aprendidas durante sua execução e indicando sugestões de possíveis trabalhos futuros relacionados a este.

2 REVISÃO DA LITERATURA

Neste capítulo serão discutidos os aspectos mais relevantes das áreas de Sistemas de Transporte Inteligente e de Sistemas de Planejamento de Viagens, e também os conceitos principais da área de Contexto Computacional, apontando como essas três áreas estão conectadas. Serão também discutidas algumas técnicas de otimização e resolução de problemas, e como estas vem sendo utilizadas em sistemas de transporte inteligente e sistemas de planejamento de viagens.

2.1 SISTEMAS DE TRANSPORTE INTELIGENTE

Os Sistemas de Transporte Inteligente (ITS[3]), soluções que integram tecnologia e avanços em sistemas computacionais, comunicação, sensores e métodos matemáticos à infraestrutura convencional de transporte de superfície, objetivam a melhoria da qualidade dos sistemas de transporte, promovendo, por exemplo, a redução de congestionamentos e a melhoria na segurança no trânsito [Sussman 2005].

Sussman [1996] apresentou um histórico dos ITS, onde cita que em 1986, um grupo de acadêmicos, oficiais de transporte federal e estadual americanos, e representantes do setor privado (que mais tarde se tornou o grupo "*Mobility* 2000") começou a discutir o futuro do sistema de transporte de superfície nos Estados Unidos. A iniciativa, que vislumbrou o que viria a ser chamado de Sistemas de Veículos e Estradas Inteligentes (IVHS[4]) e eventualmente Sistemas de Transporte Inteligente, tinha um conceito essencial: fundir as tecnologias existentes na área de sistemas inteligentes com o cenário de infraestrutura de transporte de superfície convencional. Segundo ele, essa junção poderia prover uma capacidade além dos limites do concreto e do aço, melhorando também a segurança dos transportes através da melhoria tecnológica e de um melhor entendimento dos fatores humanos.

Ainda no mesmo estudo, Sussman [1996] também apresenta uma classificação dos ITS em seis áreas funcionais (a Figura 1 representa um esquema das áreas funcionais dos ITS, relacionando-as a alguns trabalhos que serão apresentados em seguida):

[3] Do inglês, *Intelligent Transportation Systems*

[4] Do inglês, *Intelligent Vehicle-Highway Systems*

- Sistemas Avançados de Gerenciamento de Tráfego (ATMS[5]): integram o gerenciamento de várias funcionalidades das rodovias, prevendo congestionamentos e provendo instruções de rotas alternativas para veículos em áreas regionais, melhorando a eficiência da rede viária e mantendo prioridades para veículos de alta-ocupação. Trabalham coletando e disseminando dados em tempo real, permitindo que sistemas de controle de tráfego dinâmicos possam responder a mudanças de condições (e.g.: roteando os motoristas em volta de acidentes);

- Sistemas Avançados de Informação ao Viajante (ATIS[6]): proveem dados aos viajantes em seus veículos, residências ou locais de trabalho. As informações podem incluir: localização de incidentes, problemas climáticos, condições das vias, rotas ótimas, restrições de faixa de rodagem e sinalização dentro do veículo. As informações podem ser providas tanto para motoristas quanto para usuários do trânsito e até para pessoas antes de viajarem, permitindo que possam se planejar melhor;

- Sistemas Avançados de Controle de Veículos (AVCS[7])*:* aprimoram o controle do motorista sobre o veículo, tornando a viagem mais segura e eficiente. Os AVCS incluem sistemas de aviso de colisão, frenagem automática ou mesmo a capacidade do veículo automaticamente desviar de uma colisão;

- Operações de Veículos Comerciais (CVO[8]): operadores privados de caminhões, vans e taxis já começaram a adotar tecnologias ITS para aprimorar a produtividade de suas frotas e a eficiência de suas operações, como cálculo automático de rotas de caminhões, utilizando da tecnologia Localização Automática de Veículos (AVL[9]);

- Sistemas Avançados de Transporte Público (APTS[10]): utiliza das tecnologias acima para melhorar a acessibilidade de informação para os usuários do transporte público, como situação atual e tabelas de horários dos veículos, bem como aprimorar o agendamento dos veículos de transporte público e a utilização de frotas de ônibus. Também podem oferecer a indicação de rotas e linhas e sobre as paradas atendidas por uma linha, incidentes no transporte público e indicações sobre possíveis atrasos;

---

[5]Do inglês, *Advanced Traffic Management Systems*

[6]Do inglês, *Advanced Traveler Information Systems*

[7]Do inglês, *Advanced Vehicle Control Systems*

[8] Do inglês, *Commercial Vehicle Operations*

[9] Do inglês, *Automatic Vehicle Location*

[10] Do inglês, *Advanced Public Transportation Systems*

- Sistemas Avançados de Transporte Rural (ARTS[11]): fornecem principalmente tecnologias de segurança para viagens em áreas de baixa densidade populacional;

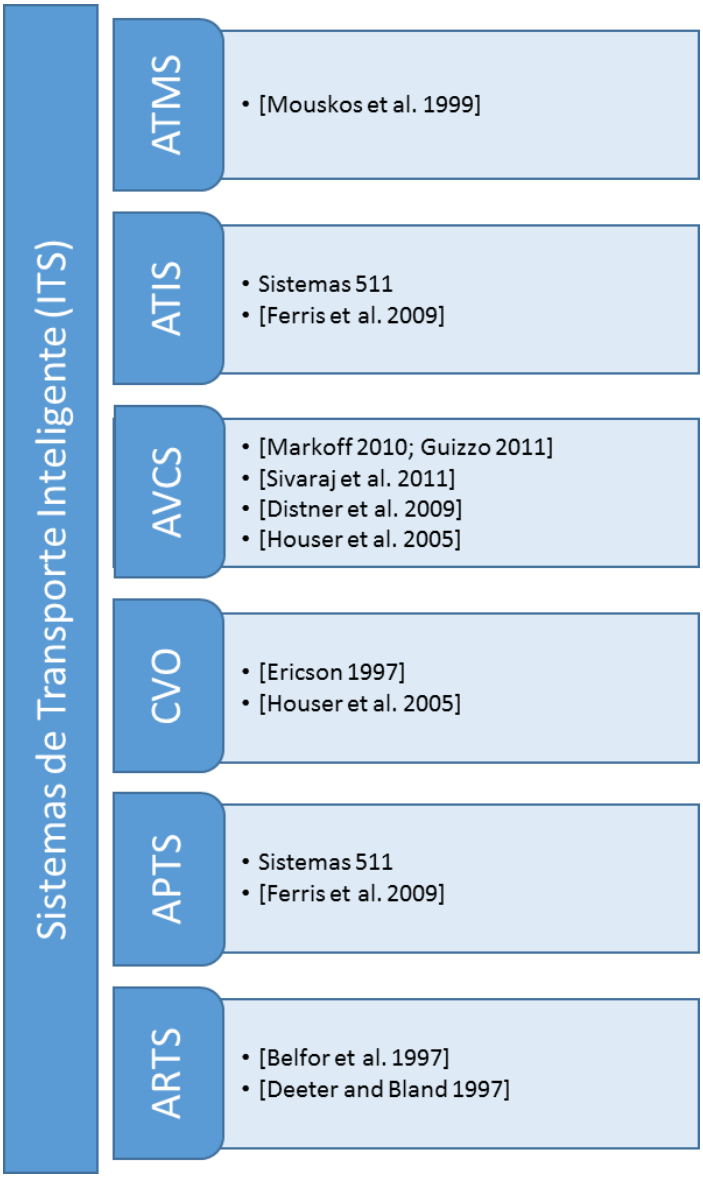

Figura 1 – 2.1 – Áreas Funcionais dos Sistemas de Transporte Inteligentes

Na região metropolitana de Nova Iorque, Nova Jérsei e Connecticut, nos Estados Unidos, existe um comitê de coordenação de operações de transporte denominado TRANSCOM[12], composto por 16 agências de transporte e segurança pública, como o departamento de transportes do estado de Nova Iorque e o departamento de polícia do estado de Nova Jérsei, que atua desde 1986 visando prover uma abordagem cooperativa e

[11]Do inglês, *Advanced Rural Transportation Systems*

[12] Disponível em: http://www.xcm.org

coordenada ao gerenciamento de transportes da região. Um dos sistemas provenientes da iniciativa é um sistema para gerenciamento de incidentes e tráfego, denominado TRANSMIT[13], que é um ATMS que utiliza equipamentos de pedágio eletrônico e gerenciamento de tráfego (ETTM[14]) para obter dados sobre os veículos passantes, e utiliza um algoritmo de detecção de incidentes que se baseia em uma comparação estatística de estimativas em tempo real dos tempos de viagem com um histórico constantemente atualizado de tempos de viagem para o mesmo período e tipo do dia (e.g. dia útil, sábado, feriado, entre outros) [Mouskos et al. 1999].

A cidade de Nova Iorque também é abrangida por um sistema denominado 511NY[15], também iniciativa do comitê TRANSCOM. O 511NY é um ITS que oferece vários serviços, como um mapa com a situação em tempo real do trânsito (ATIS), um sistema de planejamento de viagens que inclui o sistema de transporte público da região (ATIS e ATPS), um sistema para monitoramento das vias durante o inverno, um sistema integrando as câmeras de monitoramento do tráfego (ATMS), entre outros.

Sistemas de transporte inteligentes semelhantes ao 511NY existem também em outras regiões dos Estados Unidos, incluindo São Francisco (511 SF Bay[16]), Los Angeles (Go511[17]), San Diego (SANDAG 511[18]), entre outros, oferecendo serviços como planejamento de viagens (incluindo transporte público), monitoramento em tempo real da situação do trânsito e avisos de incidentes nas vias. A implantação de sistemas "511" é coordenada pela Administração Federal de Rodovias (FWHA[19]) do Departamento de Trânsito americano, estando funcional na maioria dos estados americanos (Na Figura 2, abaixo, as regiões marcadas como "*Active*" já possuem sistemas em operação, enquanto que as regiões marcadas como "*Assist*" já receberam verba do Departamento de Trânsito).

[13] Do inglês, *Transportation Operations Coordinating Committee's System for Managing Incidents and Traffic*

[14] Do inglês, *Electronic Toll and Traffic Management*

[15] Disponível em: http://www.511ny.org

[16] Disponível em: http://511.org

[17] Disponível em: http://www.go511.com

[18] Disponível em: http://www.511sd.com/

[19] Do inglês, *Federal Highway Administration*. Disponível em: http://www.fhwa.dot.gov/

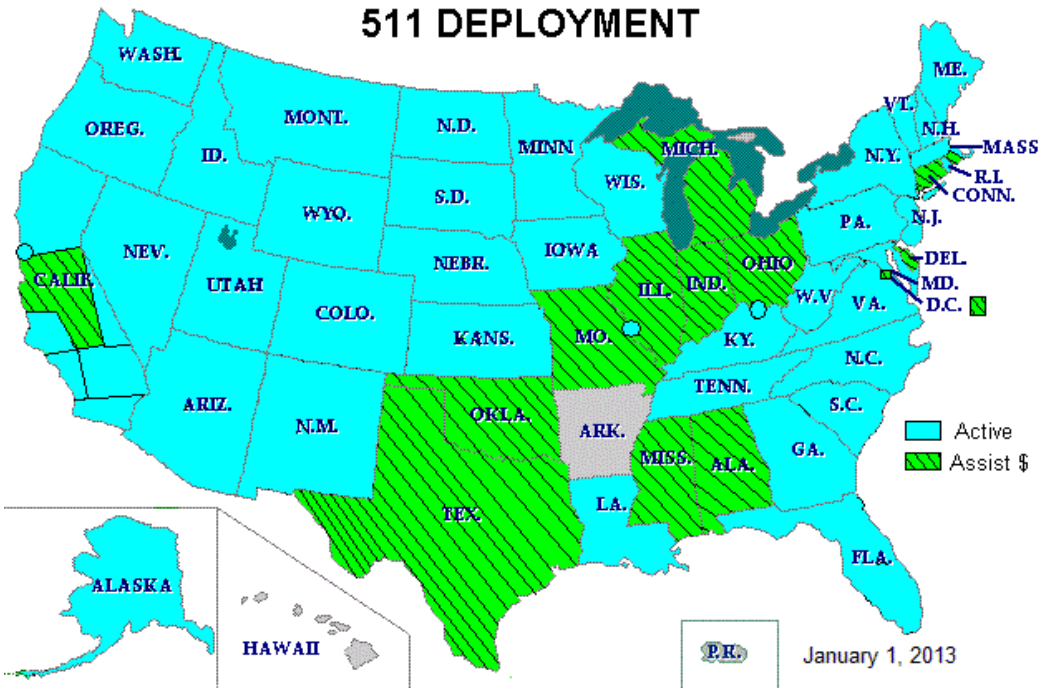

Figura 2 – 2.2 – Implantação dos Sistemas "511" no Estados Unidos
Fonte: [Federal Highway Administration 2013]

Na área de sistemas avançados de controle de veículos (AVCS), vem surgindo iniciativas voltadas para o desenvolvimento de veículos autônomos, capazes de se guiar sem a necessidade de um motorista. Um exemplo é o projeto de veículos autônomos do Google, que utilizam de diversos sensores como câmeras, radares, receptores GPS e um telêmetro a laser para mapear o ambiente à sua volta, permitindo que os veículos consigam trafegar em situações comuns de trânsito (e.g. semáforos, travessia de pedestres, entre outros) [Markoff 2010; Guizzo 2011].

Também na área de veículos autônomos, [Sivaraj et al. 2011] propõem uma arquitetura de controle para direção, aceleração e frenagem automáticas em um veículo autoguiado, capaz de seguir uma linha preta sobre uma superfície branca, através de um conjunto de sensores infravermelho. O veículo mantém sua trajetória sobre a linha minimizando a distância percebida pelos sensores, que os autores denominam erro lateral, através de um controlador PID que utiliza as informações capturadas pelos sensores e tratadas por um filtro de Kalman.

Ainda tratando-se de AVCS, a fabricante de veículos Volvo[20] oferece em alguns modelos de veículos um sistema denominado "*City Safety*" [Distner et al. 2009], que é um sistema de prevenção e mitigação de colisões em baixa velocidade, visando reduzir os riscos de colisões traseiras em veículos de passeio em situações de distração ou falta de atenção dos motoristas. O sistema opera através de um sensor laser infravermelho integrado ao topo do para-brisas do veículo, que constantemente mede a distância de veículos em até 10 metros à frente do veículo com o sistema, calculando a aceleração e velocidade relativas, permitindo a detecção de situações de risco de colisão, onde o sistema de frenagem do veículo é automaticamente ativado, evitando ou mitigando a colisão (Figura 3).

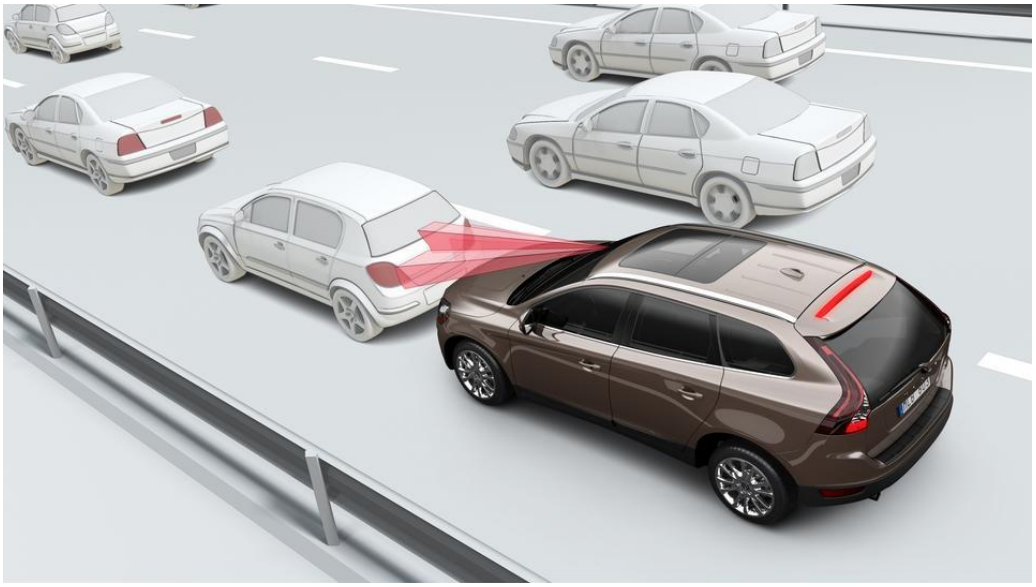

Figura 3 - 2.3 - Sistema City Safety
Fonte: [Distner et al. 2009]

Na área de Operação de Veículos Comerciais, [Ericson 1997] apresentou em um artigo na revista "*Public Roads*" uma iniciativa da Administração Federal de Rodovias americana (FHWA) para disseminar as novas tecnologias de CVOs pelos Estados Unidos, onde um caminhão batizado de "*Technology Truck*" viajou de cidade em cidade, passando por campus universitários, centros de convenções e agências governamentais, apresentando demonstrações das novas tecnologias CVO disponíveis no mercado. Em seu interior, estavam disponíveis uma sala de aula e uma área de demonstração com computadores, quiosques interativos, e uma cabine de caminhão em escala real para simulação de novos produtos de comunicação e segurança.

[20] Disponível em: http://www.volvocars.com

A FHWA também oferece um guia de produtos[21] para segurança em veículos comerciais, que podem ser classificados como sistemas AVCS e CVO, como sistemas de alerta de colisões (CWS[22]) com controle de cruzeiro adaptativo (ACC[23]) [Houser et al. 2005], que monitoram a rodovia à frente do veículo e alertam o motorista sobre potenciais riscos de colisão, controlando a velocidade do veículo para reduzir os riscos de acidentes.

Na área de APTS, além dos sistemas "511" citados, existe um sistema denominado OneBusAway [Ferris et al. 2009], que opera nas regiões de Atlanta, Nova Iorque, Puget Sound e Tampa nos Estados Unidos, oferecendo serviços como informações em tempo real sobre a posição dos ônibus, tempo de chegada dos veículos nas paradas, e um sistema de planejamento de viagens.

Como parte do programa ARTS do Departamento de Trânsito americano, a Administração Federal de Rodovias americanas criou o ARTS *Compendium* como uma ferramenta para acompanhar as tecnologias aplicadas relacionadas às áreas rurais, e para auxiliar na identificação de outras áreas com necessidades de maiores pesquisas [Belfor et al. 1997]. O projeto é uma compilação de uma variedade de tipos de projetos de ITS, desde estudos em desenvolvimento até testes operacionais de campo financiados com verba federal, incluindo projetos sendo planejados ou implementados em áreas rurais e urbanas, entretanto todos os projetos têm implicações no programa ARTS. Os projetos foram classificados de acordo com as Áreas Críticas do Programa (CPAs[24]) de ITS definidas no plano nacional do programa ITS [United States Department of Transportation 1997].

Um relatório preparado pela FHWA [Penney 1999] apresenta as principais necessidades e desafios da implantação de Sistemas Avançados de Transporte Rural (ARTS) nas áreas rurais dos Estados Unidos, que representam 80% do comprimento total de rodovias americanas. Esses sistemas se referem à porção dos ITS que têm como foco as necessidades de viajantes e motoristas em áreas não-urbanizadas, envolvendo assim viagens interestaduais, pequenas comunidades, entre outros. As principais necessidades do transporte rural são apresentadas no relatório sendo classificadas de acordo com as CPAs:

---

[21] Disponível em: http://www.fmcsa.dot.gov/facts-research/art-productguides.aspx

[22] Do inglês, *Collision Warning System*

[23] Do inglês, *Adaptive Cruise Control*

[24] Do inglês, *Critical Program Areas*

- Segurança do Viajante e Tecnologias de Segurança, que utilizam sensores no próprio veículo e sistemas de informação para alertar os motoristas sobre condições de risco e perigos. Esta área também engloba disseminação em área de avisos e alertas específicos de uma região;
- Tecnologias de Serviços de Emergência, utilizam satélites e sistemas avançados de comunicação para automaticamente alertar autoridades próximas, como polícia ou bombeiros, em caso de colisões ou outras emergências;
- Serviços de Turismo e Informação ao Viajante, que utilizam sistemas de navegação no interior do veículo e sistemas de comunicação ao longo das vias para prover informações aos viajantes que não são familiares com a área local.
- Serviços de Mobilidade e Transporte Público, que aprimorem a eficiência do transporte público e sua acessibilidade aos residentes rurais, incluindo melhorias nos horários de partida dos veículos, pagamentos através de *smartcards*, entre outros;
- Tecnologias de Manutenção e Infraestrutura de Operações, aprimorando a capacidade de trabalhadores realizarem manutenção e repararem vias em zonas rurais, incluindo sistemas de informações sobre o clima severo (e.g. nevascas, ventanias) e a capacidade de detecção e alerta de perigos aos funcionários das obras.
- Sistemas de Manutenção e Operações de Frotas, melhorando a eficiência de veículos de transporte público em zonas rurais e outras frotas, como removedores de gelo e veículos de autoridades, através de rastreamento de veículos e sistemas de monitoramento de equipamentos on-board.
- Operações de Veículos Comerciais, utilizando satélites, computadores e sistemas de comunicação para gerenciar a movimentação e a logística de veículos comerciais, e para localizar veículos em casos de emergência ou falha mecânica.

Mesmo antes do relatório de [Penney 1999], a FHWA já havia preparado um relatório contendo informações detalhadas de 14 sistemas de transportes inteligentes que foram selecionados como os mais prováveis para serem replicados para outras regiões. Os sistemas selecionados abrangem as áreas críticas do programa de ITS citadas anteriormente, incluindo sistemas de informações aos viajantes na internet, sistemas de controle de velocidade e sistemas de detecção de veículos de baixo custo [Deeter and Bland 1997].

Em 2009, [Passos and Rossetti 2009] apresentam os Sistemas de Transporte Inteligente sob uma perspectiva da computação ubíqua, afirmando que os ITS são inerentemente ubíquos. Segundo os autores, os sistemas de transporte são uma parte

essencial de qualquer economia baseada no comércio, e apontam que os sistemas de transporte contemporâneos passaram três grandes revoluções durante sua existência.

A primeira revolução, segundo Passos e Rossetti, veio com a introdução da comunicação eletromagnética, permitindo a existência de uma rede de informação que habilitou uma maior integração do sistema para a troca de informações de forma mais rápida e eficiente.

Ainda segundo os autores citados, a segunda revolução dos sistemas de transporte se iniciou com o advento dos sistemas digitais, fornecendo serviços de menor custo. Os computadores então começaram a desempenhar um papel imperativo nos transportes, aprimorando a eficiência do controle e coordenação do tráfego, assim como do planejamento dos transportes. A coordenação centralizada do tráfego pode, entretanto, ter seu desempenho limitado pela escala, i.e. pela quantidade de unidades processadas pelo sistema. Tal limitação representa um problema sério, visto que os sistemas de transporte estão se tornando cada vez maiores, tanto em termos de estrutura como de dimensão, e o processo de aquisição e processamento de informações de todas as fontes, provendo respostas em tempo hábil, é uma tarefa bastante árdua. Isso se torna ainda mais complexo com as restrições em tempo real e com a presença de entidades heterogêneas.

Por fim, a terceira revolução apontada por Passos e Rossetti se tornou evidente com o surgimento dos Sistemas de Transporte Inteligente (ITS), onde o usuário é o aspecto central do sistema de transporte, forçando as arquiteturas a se tornarem adaptáveis e acessíveis de diferentes maneiras, atendendo diferentes requisitos e uma grande variedade de propósitos. Segundo os autores, algumas das principais características do transporte inteligente incluem, entre outras: a computação automatizada, visto que os sistemas de transporte do futuro precisam tomar decisões automaticamente, analisando informações de entrada e agindo de acordo, disparando ações coordenadas para melhorar o desempenho do sistema; a flexibilidade e liberdade de escolha por parte do usuário, fornecendo serviços personalizados; a precisão também é importante, i.e. informações precisas e atualizadas devem ser entregues em tempo hábil, já que a precisão das reações e respostas a serviços requisitados está ligada diretamente a erros e outras situações de falha. Assim, a latência também deve ser reduzida através de arquiteturas distribuídas, reduzindo o tempo de resposta.

O conceito de computação ubíqua, utilizado por [Passos and Rossetti 2009], foi primeiro apontado por Mark Weiser [1991], onde ele argumenta que as tecnologias mais profundas são aquelas que desaparecem, que se tecem cada vez mais no tecido da vida diária, até que se tornam indistinguível dela. Weiser [1991], junto com seus colegas no Xerox

PARC[25], descrevem uma nova forma de se pensar sobre os computadores, levando em consideração o ambiente natural humano, permitindo que os computadores em si desapareçam na paisagem. Essa visão de computação ubíqua era baseada em recursos computacionais e no fornecimento de informações e serviços para as pessoas quando e onde elas desejassem ou precisassem. Segundo ele, o conceito de invisibilidade que ele propõe para os computadores significa que eles devem se tornar uma ferramenta que não interfira na consciência de seus usuários, permitindo que eles mantenham seu foco em suas tarefas, e não nos computadores [Weiser 1994].

2.2 CONTEXTO COMPUTACIONAL

O termo "*Context-Aware System"* (sistema ciente de contexto) foi primeiro introduzido por Schilit et al. [1994] como sendo software que "percebe e reage a mudanças no ambiente em que está situado", se referindo ao contexto como localização, identidades de pessoas e objetos próximos, e mudanças a esses objetos. Desde então, vários pesquisadores vêm propondo diferentes definições para o conceito. Em 2005, Bazire and Brézillon [2005] apresentaram e examinaram 150 definições diferentes de contexto, em diferentes áreas.

Dey [2001] afirma que definições de Contexto Computacional apresentadas por meio de exemplos são difíceis de serem aplicadas, justificando que não é claro como se pode usar essas definições para decidir se um determinado tipo de informação, que não esteja citado nelas, é Contexto ou não. Assim, ele apresenta uma nova definição para contexto: "Contexto é qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar, ou objeto que seja considerado relevante para a interação entre um usuário e uma aplicação, incluindo os próprios usuário e aplicação".

[25] *Do inglês, Xerox Palo Alto Research Center*

Dey também define que “um sistema é ciente de contexto se ele utiliza contexto para prover informação relevante e/ou serviços ao usuário, onde a relevância depende da tarefa do usuário”. Segundo ele, existem três categorias de funcionalidades que uma aplicação ciente do contexto pode apoiar [Dey 2001]:

- Apresentação de informações e serviços para um usuário;
- Execução automática de um serviço para um usuário;
- Marcação do contexto relativo a uma informação para apoiar futuro resgate;

Apontando dificuldades operacionais por conta da definição generalizada de Dey, Zimmermann et al. [2007] apresentaram uma extensão a ela, onde sugerem a divisão dos elementos utilizados para descrever essas informações do contexto em cinco categorias: individualidade (informações contextuais sobre a entidade à qual o contexto está vinculado); atividade (atividades nas quais a entidade está e irá futuramente estar envolvida, podendo ser descrito em termos de objetivos, tarefas e ações); localização (informações sobre a posição física ou virtual de uma entidade, bem como outras informações espaciais como velocidade e orientação); tempo (seja a hora atual, ou algum tempo virtual, e também informações sobre fuso horário) e relações (informações sobre relações que a entidade possa ter com outras entidades, sejam elas pessoas, coisas, dispositivos, serviços ou informações).

Em 2008, Vieira [2008] apresentou um framework independente de domínio para o projeto de Sistemas Sensíveis ao Contexto (CSS[26]), visando facilitar o desenvolvimento dos CSS. Segundo a autora, o Contexto é dinâmico, e as informações nele devem ser interpretadas para poderem ser utilizadas, o que introduz um problema de relevância, já que o que é considerado relevante para uma pessoa realizando uma tarefa pode não ser considerado por outra. Em outro trabalho, Vieira et al. [2009] apresentam em sua visão distinguindo os conceitos de Contexto e Elemento Contextual: Um elemento contextual (CE[27]) é qualquer dado ou informação associada a uma Entidade no domínio da aplicação que possa ser utilizada para caracterizar essa entidade; e o contexto da interação entre um agente e uma aplicação, para executar alguma tarefa, é o conjunto de elementos contextuais instanciados que são necessários para apoiar a tarefa atual.

Algumas características dos problemas que aplicações cientes ao contexto, apontadas pelos pesquisadores supracitados enfrentam, têm pontos de interseção com as

[26] Do inglês, *Context-Sensitive Systems*

[27] Do inglês, *Contextual Element*

dificuldades encontradas na implementação de Sistemas de Transporte Inteligente. Em 2002, [Bertolotto et al. 2002] apresentaram um protótipo de um APTS sensível ao contexto voltado para a cidade de Dublin, Irlanda. Seu principal objetivo era oferecer aos usuários "informações sobre transporte público relacionadas às necessidades do viajante individual". Eles planejaram utilizar como informações contextuais: localização das paradas de ônibus, localização em tempo real dos ônibus, hora do dia e condições climáticas, deixando para trabalhos futuros a utilização de histórico do usuário, avaliação de preço das passagens, atrações turísticas e personalização da linguagem da interface. Com o protótipo desenvolvido, eles realizaram diversos testes e avaliaram seu sistema com um grupo de voluntários e, apesar das limitações técnicas identificadas, os resultados foram satisfatórios.

O uso de Contexto em ITS também foi explorado por [Harrington and Cahill 2004], que apresentaram um protótipo de um sistema de perfil de rotas. Seu sistema utilizava dados de sensores para estimar a velocidade do fluxo de tráfego na rede de rodovias da Irlanda, e então gerava a partir desses dados um histórico de tempos de viagem entre seções da rede. Esses históricos de tempos de viagem eram então marcados com informações contextuais relevantes que caracterizassem as condições sob as quais o tempo de viagem foi gravado. Os autores identificaram como informações contextuais relevantes à aplicação: tempo, clima, condições da superfície da via e padrões de uso da rodovia mais predominantes (e.g. padrões de tráfego associados a grandes shows ou operações de manutenção na rodovia). O sistema então consumia essas informações em um processo de indexação dos históricos dos tempos de viagem.

As informações contextuais consideradas relevantes por Harrington e Cahill [Harrington and Cahill 2004] respondem a questões sobre tempo (*when*), local (*where*) e situação (*what*). Essas questões fazem parte de seis dimensões semânticas citadas em [Bulcao Neto 2006], conhecidas como 5W+H. As cinco primeiras dimensões (5W) foram inicialmente discutidas por [Abowd and Mynatt 2000] para auxiliar na especificação, modelagem e estruturação de informações de contexto em aplicações, e a sexta dimensão (H) foi apresentada em [Truong et al. 2001], originada do domínio de aplicações de captura e acesso. São elas:

- *Who* (quem): Aplicações sensíveis a contexto devem controlar a identificação de todas as entidades participantes de uma atividade no intuito de atender às necessidades dos usuários. Informações de contexto de identificação podem incluir, entre outras, nome, e-mail, senha, e dados biométricos como voz e impressão digital.
- *Where* (onde): A localização em ambientes físicos é normalmente associada a outras dimensões, como a dimensão temporal (*When* – quando). A combinação

dessas duas dimensões, por exemplo, pode fornecer informações sobre a orientação e a velocidade de um objeto móvel. Informações de contexto de localização incluem, entre outras, latitude, longitude, altitude, cidade e posição relativa a outras entidades.

- *When* (quando): Informações de contexto temporais podem ser usadas para situar eventos em uma linha do tempo, ou auxiliar na interpretação de atividades humanas e no estabelecimento de padrões de comportamento, e incluem, entre outras, data, hora, intervalos de tempo, dia da semana, mês e ano.
- *What* (o quê): A obtenção de informações de contexto que possibilitem a interpretação correta da atividade que está sendo desenvolvida por uma entidade é um grande desafio, especialmente para aplicações em que atividades, não previstas pelo projeto da aplicação, podem ser realizadas de forma concorrente. Informações de contexto de atividades variam de acordo com o domínio da aplicação e podem ser, por exemplo, escrever em um quadro branco, anotar em um caderno, trabalhar em grupo, participar de uma reunião, palestra ou operação cirúrgica.
- *Why* (por quê): A obtenção e interpretação de informações de contexto que indiquem o motivo da ação de uma determinada entidade é ainda mais difícil que identificar a atividade sendo desenvolvida. Em geral, as informações de contexto de atividade (*what*) e motivação (*why*), por sua complexidade, são obtidas por meio da combinação de informações de outras dimensões. Aplicações sensíveis ao contexto podem obter, através de sensores, informações que possam ser indicativos do estado emocional de um usuário, e.g. expressões faciais, batimento cardíaco e níveis de pressão arterial. Além disso, informações sobre a motivação do usuário podem ser fornecidas por ele mesmo, e.g. o usuário pode informar seu humor [Waze Mobile 2013].
- *How* (como): São Informações referentes a ferramentas e processos relevantes às entidades do contexto, que apontem a maneira como uma atividade está sendo desenvolvida. Essas informações podem ser, por exemplo, o software de navegador que um usuário está utilizando para visitar uma página da internet, a orientação do dispositivo móvel que ele está utilizando (i.e. horizontal ou vertical), entre outras.

Um tipo de ITS onde o uso de contexto auxilia bastante na melhoria dos serviços oferecidos aos usuários são os sistemas de planejamento de viagens, apresentados na seção a seguir.

2.3 SISTEMAS DE PLANEJAMENTO DE VIAGENS

Segundo [McCormack and Roberts 1996], Sistemas de Planejamento de Viagens (*Trip Planning Systems*, em inglês) podem ser definidos como sistemas para auxiliar viajantes ou potenciais viajantes em todos os aspectos de suas viagens, incluindo decisões sobre o meio de transporte utilizado, o horário da viagem e quais rotas seguir. As decisões da viagem podem ser baseadas em dados estáticos, como tabelas de horários, ou dinâmicos como informações sobre atrasos de veículos, congestionamentos e cancelamentos. Em especial, usuários demonstram particular interesse em sistemas que utilizem de informações em tempo real para auxiliar no planejamento de viagens.

O planejamento de viagens em redes de transporte público complexas é uma decisão frequentemente vivenciada por muitos passageiros pois, apesar de os recentes avanços nas tecnologias de comunicações e informações sem fio terem permitido o desenvolvimento de sistemas de navegação e informações a passageiros, ainda existem várias deficiências nos serviços de planejamento de viagens oferecidos. As informações relacionadas a horários de veículos e rotas do transporte público estão geralmente fragmentadas, e obtiveis por fontes diversas. Além disso, a maior parte dos serviços existentes abordam decisões de planejamento de viagens com um único critério, ignorando o fato que o processo de tomada de decisão associado envolve a consideração de múltiplos critérios de viagem simultaneamente, incluindo o número de trocas de veículos, o tempo e a distância percorridos a pé, e o custo da viagem [Zografos et al. 2009].

Em um estudo sobre as principais necessidades de informações por usuários de transporte público na região de Dublin, Irlanda, [Caulfield and O'Mahony 2007] apresentam os resultados de uma pesquisa feita com 248 pessoas que trabalhavam no centro da cidade de Dublin e possuíam acesso à internet, onde identificam que 73% dos respondentes classificam a existência de um sistema para planejamento de viagens como importante ou muito importante, e 90% dos respondentes indicam como importante ou muito importante a existência de um mapa contendo a localização em tempo real dos veículos. Além disso, a pesquisa realizada também apontou que, dentre os respondentes que são usuários de transporte público, 72% consideram muito frustrante a incerteza sobre o horário de chegada do transporte, e 55% consideram muito frustrante não saberem se o ônibus/trem que esperam já passou pela parada.

Existem alguns sistemas de planejamento de viagens urbanas em operação, implantados por organizações metropolitanas de transporte público [Zografos et al. 2009],

como os sistemas Calgary Transit[28], que opera na cidade de Calgary, Canadá; OPTI-TRANS[29], que opera na cidade de Atenas, Grécia e o sistema TransLink[30], que atua em Vancouver, Canadá.

Indo além do planejamento de viagens, o Google Transit[31] foi uma ferramenta web desenvolvida pelo Google que surgiu em Dezembro de 2005 como parte do programa Google Labs, e hoje está atualmente diretamente integrada ao produto Google Maps[32]. O sistema pode ser acessado por uma grande variedade de dispositivos, e oferece serviços relacionados ao trânsito para usuários em várias localidades do planeta, como informações em tempo real sobre o estado das vias (i.e. se uma determinada rua ou avenida está engarrafada - Figura 8), e um sistema de planejamento de viagens, inclusive de transporte público, que está disponível em mais de 500 cidades pelo mundo. No Brasil, o sistema oferece o serviço de planejamento de viagens apenas em algumas cidades, com uma abrangência restrita [Google 2013].

Além de sua utilidade para viajantes de todo o mundo, o Google Transit também ofereceu uma contribuição significante por estabelecer um padrão *de facto* para a representação de informações referentes a planejamento de trânsito: a Especificação Geral sobre Feeds de Transporte Público (GTFS[33]).

Devido a isso, muitas agências de transportes que participam do programa Google Transit também passaram a fornecer suas informações como horários e veículos no formato GTFS para que desenvolvedores possam utilizar, o que ajudou a evoluir o ecossistema de desenvolvimento de soluções que utilizem informações de trânsito [Ferris et al. 2010a].

Diversos sistemas têm aderido ao formato GTFS para representação de informações de trânsito, e muitas instituições disponibilizam informações a desenvolvedores que desejem criar aplicações que consumam esses dados. O sistema TransLink, citado anteriormente, disponibiliza uma página[34] com recursos para desenvolvedores, incluindo informações sobre o sistema de transporte local no formato GTFS, e uma API pública que

[28] Disponível em: http://hastinfoweb.calgarytransit.com

[29] Disponível em: http://athens.optitrans.net/

[30] Disponível em: http://www.translink.ca

[31] Disponível em: http://www.google.com/transit

[32] Disponível em: http://maps.google.com

[33] Maiores informações: https://developers.google.com/transit/gtfs/reference?hl=pt-BR

[34] Disponível em: http://www.translink.ca/en/Schedules-and-Maps/Developer-Resources.aspx

permite o acesso a alguns serviços como informações sobre paradas, ônibus, rotas e tempo estimado de chegada do próximo ônibus nas paradas.

Em 2009, foi lançada uma plataforma de código aberto para planejamento multimodal de viagens e análises de redes transportes, chamada OpenTripPlanner[35] (OTP). Desde então, a plataforma tem sido utilizada por sistemas em doze países, incluindo Estados Unidos, Canadá, Reino Unido, Espanha, Índia, entre outros. Entre os sistemas de planejamento de viagens que utilizam a plataforma [OpenTripPlanner 2013], pode-se citar os sistemas OPTI-TRANS, mencionado anteriormente, que também está em funcionamento na cidade de Londres[36], e o sistema OneBusAway, que atua nas regiões de Atlanta, Nova Iorque, Puget Sound e Tampa nos Estados Unidos [Ferris 2011].

O OTP permite que usuários combinem informações sobre viagens a pé, de bicicleta ou no trânsito através de uma interface web, permitindo também que aplicações de terceiros utilizem uma API para fornecer os serviços. A plataforma opera com padrões de dados abertos para informações sobre o trânsito e sobre a malha viária, permitindo que qualquer agência com um conjunto de dados no formato GTFS possa utilizá-la. Desde o lançamento da versão 0.9 [Emory 2012], o OTP oferece a possibilidade de utilizar um algoritmo denominado RAPTOR [Delling et al. 2012] para o processo de geração de rotas, onde anteriormente a plataforma oferecia apenas uma implementação do algoritmo A*.

Ainda no contexto de sistemas de planejamento de viagens, [Peng 1997] apresenta um método para projetar um sistema de informações ao viajante de transporte público para prover aos usuários serviços de melhor qualidade. Segundo ele, o propósito do sistema é oferecer aos usuários de transporte público itinerários ótimos baseados em suas origens, destinos e horários dos ônibus e/ou informações em tempo real sobre a localização dos ônibus, sendo uma viagem considerada ótima aquela com o menor tempo de deslocamento entre a origem e o destino, incluindo o caminho a ser percorrido a pé, o tempo esperando nas paradas, as trocas de ônibus e o tempo dentro dos veículos.

O autor ainda aponta que uma das principais diferenças no planejamento de uma viagem em uma rede de trânsito normal e o planejamento em uma rede de transporte público é que cada parada de ônibus pode fazer parte de mais de uma linha, e mais de uma linha de ônibus pode compartilhar um mesmo conjunto de trechos, e afirma assim que um sistema

[35] Maiores informações: http://opentripplanner.com

[36] Disponível em: http://london.optitrans.net/

de planejamento de viagens para transporte público precisa possuir informações sobre o relacionamento entre as linhas, as paradas e malha viária.

A partir das informações sobre esse relacionamento entre as partes componentes do sistema de transporte público, Peng sugere que o processo de geração de rotas inicie-se com a determinação das paradas de ônibus mais próximas aos pontos de origem e de destino e que, a partir dessas paradas, sejam encontradas as linhas de ônibus que passem por elas. Caso existam rotas que passem tanto pelos pontos de origem e de destino, isso indica que a viagem é direta e que não serão necessárias trocas de ônibus. Consequentemente, caso não existam linhas ligando diretamente os pontos de origem e destino, isso significa que as rotas deverão prever uma ou mais trocas de ônibus.

Os sistemas de planejamento de viagens abordados nesta seção utilizam de diferentes técnicas para a geração das rotas a serem fornecidas a seus usuários e, dessa forma, na seção seguinte serão discutidas algumas técnicas de otimização e de resolução de problemas presentes na literatura.

2.4 TÉCNICAS DE OTIMIZAÇÃO E RESOLUÇÃO DE PROBLEMAS

Nesta seção serão apresentadas algumas técnicas de otimização e de resolução de problemas, vindas da inteligência artificial, e sua relação com os sistemas de planejamento de viagens, apontando sua utilidade em um cenário de geração de rotas para transporte público.

Antes de expor diretamente as técnicas utilizadas no processo de geração de rotas de ônibus, é interessante expor alguns conceitos que permeiam o contexto de resolução de problemas, apresentados em [Russell and Norvig 2009].

Um problema pode ser definido por cinco componentes: (i) o **estado inicial** em que o agente solucionador de problemas (e.g. um algoritmo de busca) se encontra; (ii) uma descrição do conjunto de **ações** (ou operadores) que um agente pode executar em um determinado estado; (iii) uma descrição dos resultados provenientes da execução de uma certa ação em um dado estado (e.g. uma dada ação a, ao ser aplicada em um determinado estado e, resultará em um novo estado e'), denominada **modelo de transição**; (iv) o **teste de término**, que determina se um dado estado é um estado objetivo; e (v) o **custo do caminho**, que é uma função capaz de designar um valor numérico a todos os caminhos que possam ser percorridos pelo agente.

Juntos, o *estado inicial*, o *conjunto de ações* e o *modelo de transição* implicitamente definem o **espaço de estados** do problema, i.e. o conjunto de todos os estados alcançáveis a partir do estado inicial por qualquer sequência de ações. Denomina-se sucessor um estado que seja alcançável a partir de um outro estado (antecessor), executando-se uma única ação.

Dessa forma, uma **solução** para um problema é uma sequência de ações que, a partir do estado inicial, levam para um estado objetivo. A qualidade da solução é medida através da função de custo do caminho, e assim uma solução ótima tem o menor custo dentre todas as soluções existentes.

Os algoritmos de busca, utilizados na resolução de problemas, geralmente atuam expandindo nós do espaço de estados, navegando pelo espaço em procura de um estado objetivo. O conjunto de estados disponíveis para serem expandidos em um dado momento é denominado fronteira.

O comportamento de um algoritmo de busca pode ser classificado segundo um conjunto de quatro critérios, sendo: **completude**, i.e. um algoritmo é considerado completo quando for capaz de encontrar uma solução, quando houver; **otimalidade**, i.e. um algoritmo é considerado ótimo quando for capaz de encontrar a melhor solução possível dentro do espaço de estados**; custo de memória** e **custo de tempo**, que indicam o fator de crescimento do uso de memória e do tempo de processamento de um algoritmo de acordo com algumas características do domínio do problema, e.g. o fator de expansão dos nós.

As técnicas de otimização e resolução de problemas que serão discutidas a seguir podem ser classificadas como técnicas de busca exaustiva (também conhecida como busca cega), busca heurística e busca local.

Busca Exaustiva

As técnicas da inteligência artificial para resolução de problemas, onde não se possui informações adicionais sobre os estados além do que é conhecido a partir da definição do problema, são denominadas estratégias de busca exaustiva (ou busca cega). As estratégias de busca exaustiva se diferenciam na ordem pela qual os nós são expandidos, como as estratégias de busca em largura, em profundidade e a busca de custo uniforme [Russell and Norvig 2009].

A estratégia de busca em largura consiste em expandir primeiro todos os nós de um mesmo nível na árvore de busca, antes de avançar para os próximos níveis. Nessa

estratégia, a fronteira é gerenciada como uma fila FIFO, onde os novos nós são inseridos no fim da fronteira.

A busca em largura é completa se o objetivo mais próximo da raiz estiver em uma profundidade d finita, e ótima quando a função de custo do caminho for não-decrescente de acordo com a profundidade do nó, i.e. os nós mais profundos sempre têm um custo maior que os nós mais próximos à raiz. O cenário mais comum onde uma busca em largura é ótima ocorre quando todas as ações têm o mesmo custo.

Dado que um determinado problema tenha fator de expansão (i.e. o número de nós gerados a partir de cada nó) b, e o primeiro objetivo esteja na profundidade d, o custo de tempo da estratégia de busca em largura é de $O(b^d)$, igual ao custo de memória, pois todos os nós expandidos e os nós na fronteira ficam armazenados na memória enquanto o algoritmo está em execução.

Diferentemente da estratégia de busca em largura, a busca em profundidade consiste em expandir primeiro o nó mais profundo na fronteira da árvore de busca, em um processo onde a busca avança para o nível mais profundo da árvore, onde os nós não possuem mais sucessores. Conforme esses últimos nós são expandidos, caso não sejam estados objetivos, são removidos da fronteira, e a busca retorna para o próximo nó mais profundo que ainda tem sucessores inexplorados. A fronteira em uma busca em profundidade é gerenciada como uma pilha LIFO, onde os novos nós são inseridos no início da fronteira.

A estratégia de busca por profundidade não é completa, nem ótima. Seu custo de tempo, dado um fator de expansão b e a maior profundidade de um nó m, é de $O(b^m)$, onde m pode ser muito maior que a profundidade do primeiro objetivo d, mais próximo à raiz. O custo de memória da estratégia é de $O(bm)$, visto que uma vez que um nó tenha sido expandido, e todos seus sucessores explorados, ele pode ser removido da memória.

Por último, a estratégia de busca de custo uniforme (BCU), que se originou a partir do algoritmo proposto por Dijkstra [Dijkstra 1959], é semelhante à estratégia de busca em largura, diferenciando-se em que ao invés de expandir sempre o nó mais próximo da raiz, a BCU sempre expande o nó n com o menor custo do caminho $g(n)$. Isso acontece ordenando-se a fronteira de acordo com o custo dos nós expandidos.

Além da ordenação da fronteira, outra diferença entre a BCU e a busca em largura consiste no momento em que o teste de término é aplicado: na busca de custo uniforme, o

teste é aplicado a um nó quando ele é escolhido para ser expandido, ao invés de quando é gerado, pois o primeiro nó objetivo que for gerado pode não estar em um caminho ótimo.

A BCU é completa, e ótima quando o custo do caminho é dado por uma função não decrescente, de forma que dado um nó n e seu sucessor n', o custo $g(n')$ é sempre superior ou igual a $g(n)$. O custo de memória e de tempo da estratégia, no pior cenário, é de $O(b^{1+C^*/\varepsilon})$, onde b é o fator de expansão, C^* é o custo da solução ótima e ε é o menor custo de uma ação. Em um cenário onde todos os custos de caminho são iguais, o custo de memória e de tempo da BCU passa a ser $O(b^{d+1})$.

O algoritmo de Dijkstra [Dijkstra 1959], mencionado anteriormente, foi desenvolvido para resolver problemas onde se deseja encontrar o menor caminho entre dois nós de um grafo. Em sistemas de planejamento de viagens, muitas vezes é necessário que sejam apresentadas ao usuário um conjunto de melhores rotas para um dado par origem-destino, de modo que a técnica utilizada para a geração das rotas deva ser capaz de encontrar não apenas a melhor rota para o problema, mas as n melhores rotas. Esse problema é conhecido na literatura como problema de **k menores caminhos** (*k shortest paths*, em inglês) [Eppstein 1998].

Em [Schulz et al. 2000], os autores avaliam variações do algoritmo de Dijkstra para encontrar a menor rota entre dois pontos em um cenário de transporte público ferroviário, onde o tempo de resposta é crucial. Os dados utilizados são estáticos (tabelas de horários), e o único critério usado para avaliar as rotas é o tempo total de viagem. As variações apresentadas incluem técnicas como: restrição de ângulo, onde as estações a serem expandidas pelo algoritmo são limitadas a uma região geográfica delimitada por um ângulo a partir de um dado estado; seleção de estações, que consiste em priorizar durante a busca alguns nós específicos (no caso, estações com grande número de ligações – estações centrais); busca direcionada ao objetivo, onde os valores das arestas entre os nós do grafo de busca são modificados de maneira otimizar a fronteira de busca (no caso, os valores das arestas foram modificados de acordo com as velocidades máximas dos trens; e busca bidirecional, onde duas instâncias do algoritmo são executadas simultaneamente, uma iniciando no ponto de origem com objetivo de alcançar o ponto de destino, e outra no sentido contrário, partindo do ponto de destino e percorrendo até o ponto de origem.

Diversos trabalhos exploram o problema de encontrar os k menores caminhos para gerar rotas em sistemas de planejamento de viagens: em [Xu et al. 2012], os autores apresentam um algoritmo para encontrar os melhores caminhos em uma rede de trânsito baseada em tabelas de horários, com rotas limitadas a até duas trocas de veículos durante

o percurso, e utiliza fatores de conversão para transformar todos os elementos que compõem a função-objetivo em unidades temporais (e.g. converter o custo monetário da viagem em segundos).

Já em [Jariyasunant, Work, et al. 2011] e [Jariyasunant, Mai, et al. 2011] os autores apresentam um sistema de planejamento de viagens para transporte público acessível em dispositivos móveis que utiliza um algoritmo para encontrar os k menores caminhos a serem percorridos pelos usuários, considerando apenas informações sobre distância e tempo de chegada dos ônibus nas paradas e limitado a até quatro trocas de ônibus, que opera em duas etapas: primeiro é gerado um grafo com todas as rotas possíveis entre todos os pontos da rede, que utiliza informações estáticas sobre a rede de transporte, e é armazenado em um banco de dados; a segunda etapa consiste em atualizações no grafo a partir de informações dinâmicas obtidas de fontes externas ao sistema, como informações sobre atraso de ônibus.

Em [Zhao et al. 2008], os autores utilizam um algoritmo para a recomendação de rotas ônibus em uma malha de trânsito estocástica. O algoritmo é composto de duas partes: um algoritmo para encontrar o menor tempo gasto com trocas de ônibus, que fornece valores que servirão de heurística para a segunda parte; e um algoritmo para encontrar os k menores caminhos para as rotas, que leva em consideração, além do tempo total das viagens, outros aspectos como o tempo gasto com trocas de ônibus.

Busca Heurística

Enquanto as estratégias de busca exaustiva se baseiam apenas nas informações sobre os estados obtidas da formulação do problema, as estratégias de busca heurística utilizam conhecimento específico do problema, além das informações obtidas em sua formulação, para resolvê-lo. A abordagem geral é denominada busca pela melhor escolha (*best-first search*, em inglês), onde os nós são selecionados para expansão de acordo com uma função de avaliação $f(n)$, que é construída como uma estimativa do custo, então o nó com a menor avaliação é expandido primeiro. Sua implementação é idêntica à implementação da busca de custo uniforme, apresentada anteriormente, com a diferença que a fronteira é ordenada de acordo com a função f, ao invés da função g [Russell and Norvig 2009].

A escolha de f determina a estratégia de busca, e a maioria dos algoritmos de busca pela melhor escolha incluem uma função heurística como componente de f, denominada $h(n)$, de maneira que $h(n)$ seja o custo estimado do menor caminho partindo do estado no nó n até um estado objetivo.

A estratégia de busca gulosa é uma busca pela melhor escolha que consiste em expandir o nó mais próximo do objetivo, baseado que esse comportamento provavelmente levará o algoritmo rapidamente a uma solução. Dessa forma, a estratégia avalia os nós utilizando somente uma função heurística como componente de f, i.e. $f(n) = h(n)$.

Essa estratégia é incompleta, pois o algoritmo pode ficar preso em um laço caso não detecte a expansão de estados repetidos, e não é ótima. Seu custo de tempo e espaço, no pior cenário, é de $O(b^d)$, sendo b o fator de expansão e d a profundidade do primeiro objetivo, pois todos os nós expandidos são mantidos na memória.

Proposto em [Hart et al. 1968], o algoritmo A* também é uma forma de busca pela melhor escolha, onde a função de avaliação f dos nós a serem expandidos é composta pelo custo para se alcançar um nó n, $g(n)$, e pelo custo estimado do menor caminho partindo de n até um estado objetivo, $h(n)$, de forma que $f(n) = g(n) + h(n)$.

A busca A* é completa, e ótima quando duas condições são atingidas: primeiro, a função $h(n)$ deve ser uma heurística admissível, i.e. a heurística nunca superestima o custo real para se atingir o objetivo; e em segundo, a heurística $h(n)$ deve ser consistente, i.e. o valor da função heurística h não deve ser decrescente em um mesmo caminho, de forma que o custo estimado de um nó n', sucessor de n, nunca deve ser inferior ao custo estimado de n: $h(n') \geq h(n)$. O custo de tempo do algoritmo é exponencial de acordo com o comprimento da solução e dependente da qualidade da função heurística, e o custo de memória é de $O(b^d)$, pois o algoritmo armazena todos os nós expandidos em memória.

O algoritmo A* foi utilizado no sistema de planejamento de viagens Open Trip Planner [Emory 2012] para a geração de rotas, sendo substituído em 2012 por um novo algoritmo, denominado RAPTOR, apresentado em [Delling et al. 2012].

Uma variação do A*, denominada SAS (*Sparse A* Search*), foi apresentada e explorada em [Szczerba et al. 2000] para o planejamento de rotas de aviões, levando em consideração algumas restrições como comprimento mínimo a ser percorrido em linha reta pela aeronave antes de realizar uma curva (L), ângulo máximo de virada, distância total máxima e ângulo de aproximação. Seu funcionamento consistia em dividir um mapa em um formato de grade e, a partir de um ponto (uma célula da grade) de origem, criar um leque com ângulo igual ao dobro do ângulo máximo de virada e comprimento igual a L, dividindo o leque em um número determinado de setores.

A cada iteração, o algoritmo calcula os vetores de menor custo, partindo do ponto de origem do leque e com comprimento L, para cada setor, e os insere em um *min-heap* (i.e. uma árvore binária cujas chaves satisfazem a propriedade de que a chave de um dado nó é menor ou igual às chaves de seus nós-filhos). Em seguida, o algoritmo expande o nó com menor custo do *min-heap*, repetindo o processo de forma que o ponto de origem seja o nó expandido, até que o nó removido do *min-heap* esteja a uma distância L do destino da aeronave.

Guzolek e Koch [1989] apresentaram uma adaptação do algoritmo A* para o planejamento de rotas de veículos de passeio em rodovias convencionais, levando em consideração restrições de tempo e de memória através de: (i) melhoria dos critérios de otimização, de maneira que o custo g de um nó incluísse uma combinação ponderada de informações como limite de velocidade de um segmento da rota, condições do trânsito em diferentes horários do dia, comprimento dos segmentos, entre outros; (ii) variações na função heurística h em ocasiões onde não há necessidade que o algoritmo encontre uma rota ótima, como quando é necessário replanejar uma rota enquanto ela está sendo percorrida; e (iii) aplicação de métodos para poda do grafo de busca através do uso de heurísticas quase admissíveis, que forçam o direcionamento da busca visando minimizar o número de nós expandidos, buscas em estágios, dividindo a busca em um conjunto de sub-objetivos, ou negligenciando os caminhos com baixa probabilidade (i.e. caminhos que estão no final da fronteira).

Em um cenário de planejamento de viagens, os estados explorados pelas técnicas de busca exaustiva e de busca heurística representam rotas que podem estar incompletas, e.g. um estado a ser explorado por um algoritmo A* pode representar uma rota de ônibus cuja parada final não seja a parada de destino, mas sim uma parada anterior a ela. Dessa forma, conforme o algoritmo expande os nós e explora o espaço de busca, os estados vão se tornando mais completos.

Tendo em vista que as estratégias de busca exaustiva e busca heurística geralmente têm um custo de memória ou de tempo que escala de maneira exponencial com o espaço de estados, conforme foi discutido, sistemas que lidem com um grande número de informações e, consequentemente com um grande espaço de estados, e possuam restrições que limitem o tempo de resposta demandam estratégias que sejam capazes de encontrar soluções de forma rápida e com baixo custo temporal e de memória. As estratégias de busca local, apresentadas a seguir, se propõem a tentar minimizar os custos mencionados, mantendo bom desempenho.

Otimização

Estratégias de otimização, como busca local, geralmente consistem em algoritmos que operam sobre um único nó do espaço de estados, caminhando por estados vizinhos em busca de soluções melhores. Apesar de não serem sistemáticos, esses algoritmos possuem duas principais vantagens sobre técnicas sistemáticas: utilizam muito pouca memória, geralmente um valor constante, pois não armazenam informações sobre o caminho percorrido para atingir um estado objetivo, mas sim os estados que estão sendo evoluídos no momento; e muitas vezes conseguem encontrar soluções razoáveis em espaços de estados grandes ou infinitos (contínuos), onde algoritmos sistemáticos são impróprios.

Essas estratégias, então, trabalham sobre a ideia de, a partir de um estado ou de um conjunto de estados iniciais, navegar pelo espaço de estados, evoluindo os estados durante suas iterações, em busca de um estado objetivo. Problemas de otimização, dessa forma, consistem em evoluir estados completos, i.e. soluções, em busca de soluções melhores.

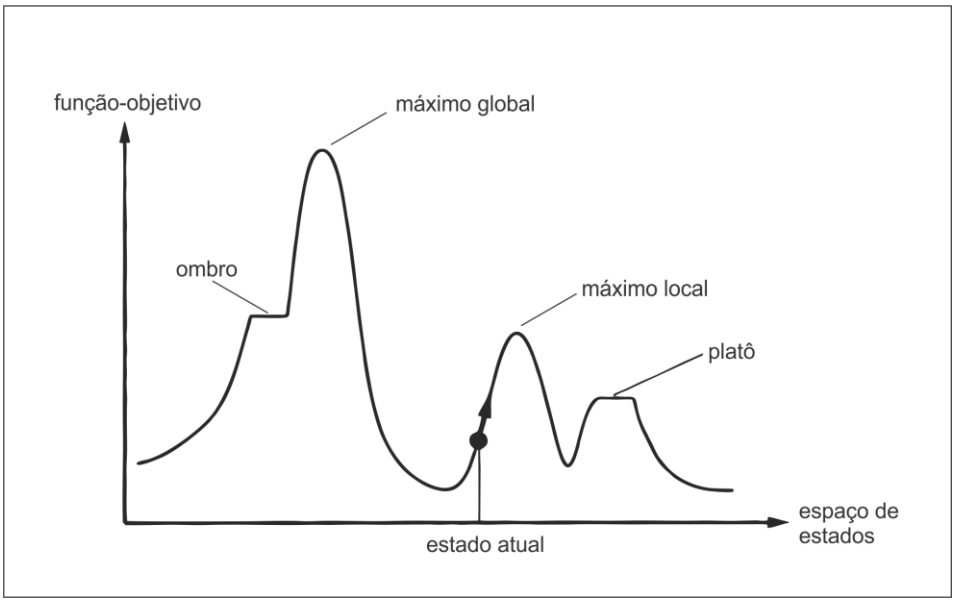

Figura 4 - 2.4 - Exemplo de Topologia do Espaço de Estados de uma Busca Local
Fonte: [Russell and Norvig 2009]

Os estados percorridos pelas estratégias de busca local podem ser representados sobre uma superfície (Figura 4), onde a altura de um determinado ponto corresponde à avaliação do estado naquele ponto. Dessa forma, no espaço de estados podemos identificar os pontos (estados) que possuam maiores e menores avaliações: o ponto com maior avaliação em todo o espaço de estados é conhecido como **máximo global**, e consiste no estado objetivo ótimo; um ponto que possua avaliação superior aos pontos a seu redor (vizinhos) é denominado **máximo local**; consequentemente, um ponto que possua

avaliação inferior aos pontos vizinhos é denominado **mínimo local**; e por fim o ponto com menor avaliação em todo o espaço de estados é conhecido como **mínimo global**.

O valor de um determinado estado é calculado utilizando-se uma função-objetivo, e geralmente representa sua aptidão, de forma que um algoritmo de otimização possa buscar pelo estado mais apto dentro do espaço de estados. Assim, o objetivo de uma otimização pode ser maximizar ou minimizar o valor calculado pela função-objetivo de um estado, dependendo do problema.

A técnica Hill-Climbing (Subida da Encosta, em português) [Russell and Norvig 2009] trata-se de algoritmo de busca local que consiste em um laço que continuamente se move em direção ao estado de maior aptidão, ou encosta acima. O algoritmo encerra quando alcança um “pico” onde não há nenhum estado vizinho com aptidão superior ao estado atual, e não mantém uma árvore de busca durante sua execução, de forma que a estrutura de dados do nó atual somente precisa armazenar o estado e o valor da função-objetivo.

O Hill-Climbing se assemelha a uma busca gulosa, pois escolhe um estado vizinho sem levar em consideração o caminho a seguir posteriormente, e geralmente progride rapidamente em direção a uma solução, pois frequentemente é fácil aprimorar um estado inicial ruim.

Os principais desafios encontrados durante a execução do algoritmo de Hill-Climbing, que podem fazer com que ele fique preso, i.e. não encontre uma solução são: (i) máximos locais, que são estados que possuem valor superior a todos seus estados vizinhos, porém inferior ao máximo global; (ii) picos, que são situações onde o algoritmo está avançando em direção ao máximo global, porém não existem operadores válidos que permitam ao algoritmo avançar mais; e (iii) platôs, que são áreas planas na topologia do espaço de estados, onde os valores calculados pela função-objetivo são iguais, impedindo que o algoritmo avance, e podem ser classificados em dois tipos: máximos locais planos, onde não existem estados superiores no platô; e ombros, onde em alguma parte do platô o algoritmo consegue continuar a avançar para estados superiores.

Visando melhorar o resultado do algoritmo de Hill-Climbing, a técnica de Reinício Aleatório consiste na execução de uma série de buscas Hill-Climbing a partir de estados iniciais gerados aleatoriamente, até que o objetivo seja alcançado. Dessa forma, o algoritmo é considerado completo, visto que eventualmente um estado objetivo será gerado como estado inicial. Para evitar que o algoritmo fique preso em um laço infinito, pode-se determinar que cada busca seja executada até que um número máximo de iterações aconteça, ou até

que os resultados encontrados não apresentem uma melhora significativa. Após as várias buscas terem sido executadas, o algoritmo escolhe o melhor resultado encontrado.

Quando aplicado a problemas de otimização, o algoritmo de Hill-Climbing é considerado completo, visto que cada nó visitado pelo algoritmo é obrigatoriamente um estado completo (solução), e pode ser considerado ótimo quando iterações suficientes forem permitidas em sua execução, utilizando a técnica de reinício aleatório. O sucesso do Hill-Climbing depende muito da topologia do espaço de estados: se houverem poucos máximos locais e platôs, um Hill-Climbing com reinício aleatório irá encontrar uma boa solução em poucas iterações.

Em [Jacobson et al. 2006], os autores exploram o uso de algoritmos de Hill-Climbing generalizados simultâneos (que são uma variação da técnica de Hill-Climbing onde múltiplas instâncias do algoritmo são executadas paralelamente e compartilham entre si informações sobre as soluções, melhorando seu desempenho) para o planejamento de recursos (e.g. veículos aéreos não tripulados (UAV[37]) e helicópteros) e rotas em operações militares de reconhecimento, vigilância e busca e resgate.

Outros trabalhos também estudam o uso de um algoritmo de Hill-Climbing para o planejamento de viagens. Em [Zaheer 2006], o autor apresenta um comparativo entre diferentes técnicas de inteligência artificial para o planejamento multimodal de viagens, considerando informações como distância total percorrida, custo monetário e tempo de espera.

Em [Chiong et al. 2008] os autores apresentam um comparativo entre seis diferentes técnicas para planejamento de viagens, incluindo busca pela melhor escolha, A* e Hill-Climbing, considerando apenas a distância total do percurso como critério de otimização. Os autores também apresentam os resultados de experimentos realizados com as técnicas em cenários com diferentes quantidades de cidades envolvidas no planejamento de viagens, e apontam que o algoritmo de Hill-Climbing obteve bom desempenho nos cenários com 15 ou menos cidades, onde o número de conexões entre os nós é grande.

Existem outras abordagens de algoritmos de otimização, que se baseiam na teoria da evolução das espécies de [Darwin 1859], denominados Algoritmos Evolucionários (AEs) [Eiben and Smith 2007], que buscam por soluções de um problema partindo de uma ou mais soluções (indivíduos) iniciais, e atuam através de um processo iterativo evoluindo um

[37] Do inglês, *Unmanned Aerial Vehicle*

conjunto de indivíduos, que é denominado população, através de uma série de operadores, como operadores que recombinem indivíduos ou que modifiquem suas estruturas. Entre as técnicas de Algoritmos Evolucionários, existem os Algoritmos Genéticos (AGs), que são inspirados nos processos naturais de evolução (especialmente através de conceitos da biologia evolutiva) [Goldberg 1989].

Os Algoritmos Genéticos, assim como os AEs atuam sobre uma população de indivíduos (denominados cromossomos), onde cada cromossomo é uma codificação de uma possível solução para o problema que o AG está trabalhando, e possui uma aptidão associada [Silva 2011]. Os cromossomos são codificados como estruturas de dados, geralmente vetores ou cadeias de bits, e representam um conjunto de parâmetros da função-objetivo que o AG deve maximizar ou minimizar. O conjunto de todas as configurações possíveis que um cromossomo pode assumir forma o espaço de busca do Algoritmo Genético, de forma que para uma codificação de cromossomos que represente *n* parâmetros de uma função-objetivo, o espaço de busca do AG será um espaço com *n* dimensões [Lacerda and Carvalho 1999].

Seu funcionamento é baseado em *gerações* de indivíduos, onde em cada geração a população passa por um processo de *seleção*, *cruzamento* e/ou *mutação*, através de operadores genéticos inspirados em fenômenos da natureza, como a mutação genética e a reprodução sexuada [Linden 2012].

O processo de seleção é responsável por aprimorar a qualidade da população, determinando quais indivíduos de uma determinada geração possuirão maior chance de sobreviverem (i.e. estarem presentes) na próxima geração [Blickle and Thiele 1996]. O cruzamento é responsável por realizar a transmissão de características através da combinação entre os indivíduos, criando novos cromossomos de forma que eles possuam algumas características dos cromossomos originais. A mutação, por sua vez, é responsável por modificar a estrutura dos indivíduos, e ocorre de acordo com uma probabilidade denominada *taxa de mutação* [Goldberg 1989].

Existem vários métodos utilizados para o processo de seleção de indivíduos [Silva 2011], como seleção por Roleta e por Torneio. No método de seleção por Roleta [De Jong 1975], os indivíduos são selecionados com uma probabilidade proporcional a seus valores de aptidão, de forma que os mais aptos tenham maiores chances de estarem presentes em novas gerações, porém mantendo uma possibilidade para que os indivíduos menos aptos também prosperem [Lacerda and Carvalho 1999]. É importante evitar o descarte completo dos indivíduos menos aptos, pois estes podem conter características (“material genético”)

fundamentais para a construção de cromossomos mais aptos, garantindo assim a diversidade genética.

Caso uma população passe por muitos processos de seleção sem que se tome cuidado para manter uma boa diversidade genética, as características dos indivíduos começaram a convergir, tornando a população cada vez mais homogênea (todas as soluções serão muito parecidas), reduzindo a capacidade do Algoritmo Genético de encontrar novas soluções [Blickle and Thiele 1995].

O método de seleção por Torneio [Miller and Goldberg 1995] funciona selecionando-se *n* indivíduos aleatórios da população, que são postos em uma disputa entre si, onde aquele com maior aptidão vence e é colocado na próxima geração. Esse processo se repete até que a população da nova geração esteja preenchida.

O operador genético de *cruzamento* é responsável por criar novos indivíduos (descendentes) com características de seus pais, combinando as representações cromossômicas que representam os indivíduos. Um cromossomo $\boldsymbol{a}$, ao cruzar com outro cromossomo $\boldsymbol{b}$, gera um novo indivíduo composto por características (*genes*) dos dois cromossomos geradores (*pais*). O novo indivíduo pode ser formado pela metade de $\boldsymbol{a}$ e outra metade de $\boldsymbol{b}$ ou, dependendo da aplicação, ser formado por diversos fragmentos de $\boldsymbol{a}$ e $\boldsymbol{b}$. Dessa maneira, a forma mais usual consiste em dividir os dois pais, gerando as partes esquerda e direita de cada um $\{\boldsymbol{a_e}, \boldsymbol{a_d}$ e $\boldsymbol{b_e}, \boldsymbol{b_d}\}$, então um filho pode ser formado pela parte esquerda do indivíduo $\boldsymbol{a}$ e pela parte direita do indivíduo $\boldsymbol{b}$ $\{\boldsymbol{a_e}, \boldsymbol{b_d}\}$, ou pela parte esquerda do indivíduo $\boldsymbol{b}$ e pela parte direita do indivíduo $\boldsymbol{a}$ $\{\boldsymbol{b_e}, \boldsymbol{a_d}\}$ [Silva 2011].

A *mutação* é um operador que modifica a codificação interna de um cromossomo, alterando sua estrutura, de acordo com a *taxa de mutação*. Existem diversas formas de mutação, que são dependentes da codificação utilizada para representar uma solução em um cromossomo, como por exemplo a mutação baseada em posição, onde dois genes do cromossomo são escolhidos, aleatoriamente, e o segundo é colocado antes do primeiro; e a mutação baseada em ordem, onde dois genes do cromossomo, também escolhidos de maneira aleatória, têm suas posições trocadas [Lacerda and Carvalho 1999].

Em [Kanoh and Hara 2008], os autores abordam a utilização de uma versão de um algoritmo genético multi-objetivo para o planejamento de veículos de passeio em um cenário real com informações de trânsito, considerando a distância percorrida, o tempo total da viagem e um critério denominado facilidade de direção pelos autores. A geração da população inicial é realizada através de um algoritmo de Dijkstra focado em criar rotas que

passem pelas vias principais da região, e a estratégia de cruzamento genético utilizada é de cruzamento em um ponto. O fitness dos indivíduos é calculado através de duas funções-objetivo, que levam em consideração a distância e o tempo totais, e um valor gerado a partir de um conjunto de restrições, que quando são quebradas pelos indivíduos, geram penalidades. Essas restrições dizem respeito a: redução do número de cruzamentos na rota, preferência por vias arteriais e/ou largas, evitar congestionamentos e reduzir o número de curvas.

Já em [Pellazar 1994], um AG foi utilizado para gerar rotas de veículos aéreos militares considerando restrições como distância total, uso de combustível e nível de perigo da viagem. A codificação dos cromossomos adotada representava as soluções como uma lista de instruções de controle do veículo, indicadas por um valor entre 1 e 27, que diziam respeito a inclinação, giro e velocidade, e o intervalo de tempo entre as instruções era dado por uma constante determinada no início da execução do sistema. O cálculo do fitness dos indivíduos era dado por um conjunto de parâmetros de custo, como distância percorrida, número de instruções diferentes, custo de sobrevoar regiões proibidas, entre outros, associados a pesos para cada parâmetro.

AGs também são utilizados em [Kumar et al. 2009], onde os autores apresentam seu uso no processo de busca por nós adjacentes durante a execução de algoritmos de busca como Dijkstra e A*, comparando seu impacto no desempenho dos algoritmos, e concluem que o uso de AGs na busca por nós adjacentes melhorou o tempo de execução em todos os algoritmos, sendo que o algoritmo A* obteve o melhor resultado.

Em [Peng and Huang 2000], os autores apontam que a maioria dos programas e algoritmos de busca de caminho (*pathfinding*, em inglês) são projetados para uso na malha viária convencional, i.e. a malha viária percorrida por veículos de passeio e comerciais comuns, como carros e caminhões, e que alguns problemas surgem quando são aplicados ao transporte público, pois a malha viária do transporte público tem características significantemente diferentes da malha convencional. Segundo eles, diversos trabalhos, como [Le Clerq 1972], [Chriqui and Robillard 1975] e [Wong and Tong 1998], apontam que os algoritmos de busca de caminhos utilizados em malhas viárias convencionais não são adequados para resolver problemas de encontrar menores rotas em malhas de transporte público, visto que tratando-se de transporte público, primeiramente a quantidade de veículos em operação pode variar de acordo com o horário do dia. Segundo, um único trecho de uma rua pode servir a várias linhas de ônibus, e várias linhas podem passar pela mesma parada de ônibus. Em terceiro lugar, diferentemente de como geralmente acontece em cálculos de rotas em malhas viárias convencionais, as melhores rotas de transporte público para um

dado par origem-destino não são simétricas em ambos os sentidos, i.e. as rotas que partem da origem ao destino geralmente não são simétricas às que partem do destino à origem.

Conforme os trabalhos apresentados nesta seção, vemos que diferentes técnicas de otimização e resolução de problemas vêm sendo utilizadas em processos de geração e classificação de rotas tanto em sistemas de planejamento de viagens como para o planejamento de recursos. Na próxima seção, será apresentada uma revisão sistemática da literatura, realizada em 2012, buscando identificar como informações contextuais vem sendo utilizadas por sistemas de informação ao usuário de transporte público, que incluem sistemas de planejamento de viagens.

2.5 REVISÃO SISTEMÁTICA DA LITERATURA

Foi realizada em 2012 uma revisão sistemática da literatura [Tito et al. 2012] buscando identificar como informações contextuais são utilizadas por Sistemas de Informação ao Usuário de Transporte Público, e teve como objetivo responder a três questões de pesquisa:

Q1. Quais informações contextuais são utilizadas por sistemas de informação a usuários de transporte público?
Q2. Como as informações contextuais utilizadas por sistemas de informação a usuários de transporte público são adquiridas?
Q3. Como os sistemas de informação a usuários de transporte público utilizam essas informações contextuais?

A estratégia de pesquisa consistiu de duas etapas: uma busca manual em Conferências, Workshops e Periódicos, com objetivo de encontrar artigos completos tratando de sistemas de informação a usuários de transporte público (ônibus) com informações contextuais, escritos em inglês ou português; e uma busca automatizada baseada em strings de busca que foram utilizadas em algumas bases de dados de trabalhos acadêmicos mais relevantes.

O processo de análise de relevância dos trabalhos encontrados durante a etapa de pesquisa tomou como critérios de inclusão (I) e exclusão (E):

- (I) Somente trabalhos acadêmicos;
- (I) Sistemas de Transporte Público para Ônibus que operam com informações contextuais;

- (E) Sistemas de Transporte Público para Ônibus que não operam com informações contextuais;
- (E) Trabalhos duplicados ou não escritos em português ou inglês;
- (E) Sistemas de Transporte Inteligente não voltados para transporte público por ônibus;
- (E) Sistemas de Informação ao Usuário que não utilizem informações contextuais.

Após essa etapa, foi realizada uma análise qualitativa sobre os trabalhos restantes, seguindo um novo conjunto de critérios, e ao final foram selecionados um total de 10 trabalhos. Estes foram então classificados de maneira a responder cada uma das três questões de pesquisa: informações contextuais utilizadas (Quadro I); método para aquisição das informações contextuais (Quadro II); e serviços providos aos usuários utilizando informações contextuais (Quadro III).

Quadro I - 2.1 - Informações Contextuais Utilizadas pelos Trabalhos Selecionados

Tipo de I.C.	Informação Contextual	Frequência
Dinâmica	Localização do Veículo	8
	Velocidade dos Veículos	2
	Feedback do Usuário	1
	Incidentes no Trajeto	1
	Intensidade do Tráfego	1
	Localização do Usuário	2
Estática	Localização das Paradas	8
	Distância entre as Paradas	8

Fonte: [Tito et al. 2012]

Quadro II - 2.2 - Aquisição das Informações Contextuais nos Trabalhos Selecionados

Aquisição das Informações Contextuais	Frequência
Equipamento GPS no Ônibus	6
Feedback do Usuário através de smartphone	1
Equipamento GPS do usuário	1
Informações históricas	1
Informações inferidas pela aplicação	1
Informações históricas previamente catalogadas	9
Não discutido	1

Fonte: [Tito et al. 2012]

Quadro III - 2.3 - Informações Oferecidas aos Usuários Utilizando Informações Contextuais

Informação Oferecida	Frequência
Tempo de chegada dos Veículos nas Paradas	8
Visualização da Rota do Ônibus	2
Criação de Rota personalizada	2
Indicação de Rota	2
Notificação de próximos destinos	1
Qualidade do Trânsito	1
Informações personalizadas	1
Localização dos Veículos	2

Fonte: [Tito et al. 2012]

Como resultado da revisão sistemática, foram identificadas algumas oportunidades de pesquisa, sendo elas: (i) desenvolvimento de Sistemas de Informação ao Usuário (SIU) que façam um maior uso de informações contextuais dinâmicas e inferidas; (ii) uso de novas fontes de informações contextuais que possam fornecer dados em tempo real para apoiar SIUs, como por exemplo redes sociais; (iii) exploração do uso de aplicações móveis, visto o aumento do número de usuários de smartphones; (iv) desenvolvimento de novas funcionalidades projetadas para os usuários, buscando prover serviços personalizados mais relevantes para as suas necessidades; (v) análise dos relacionamentos entre as informações contextuais, buscando enriquecer as funcionalidades oferecidas pelos SIUs.

2.6 TRABALHOS RELACIONADOS

Nesta seção serão apresentados alguns trabalhos encontrados na literatura e alguns sistemas comerciais já em operação, que atendam a pelos menos dois requisitos do conjunto especificado a seguir:

Requisito 1 (R1): sistemas que atuem com transporte público por ônibus;

Requisito 2 (R2): sistemas que utilizem informações em tempo real sobre a situação do trânsito;

Requisito 3 (R3): sistemas que ofereçam uma funcionalidade de cálculo de rotas;

Requisito 4 (R4): sistemas que utilizem de contexto computacional em alguma etapa de seu funcionamento.

Sistema ANTARES:

Em 2010, [Bastos and Jaques 2010] apresentaram em seu trabalho um sistema denominado ANTARES (Figura 5), que através de uma interface web permitia que usuários realizassem uma busca por rotas de ônibus, atuando como um sistema de planejamento de viagens. O sistema utilizava-se da técnica de busca heurística A* para calcular a melhor trajetória para uma dada origem e destino, levando em consideração apenas a distância total percorrida, sem considerar informações contextuais como tempo da viagem e situação do trânsito na região.

O sistema desenvolvido não utiliza nenhuma informação obtida em tempo real em sua operação, se baseando apenas em informações estáticas armazenadas em seu banco de dados. Além disso, nenhum tipo de informação contextual é utilizada durante a execução do sistema.

Com relação à modelagem do domínio de transporte público utilizada no sistema ANTARES, somente as linhas de ônibus (denominadas “ônibus” pelos autores) e as paradas são consideradas.

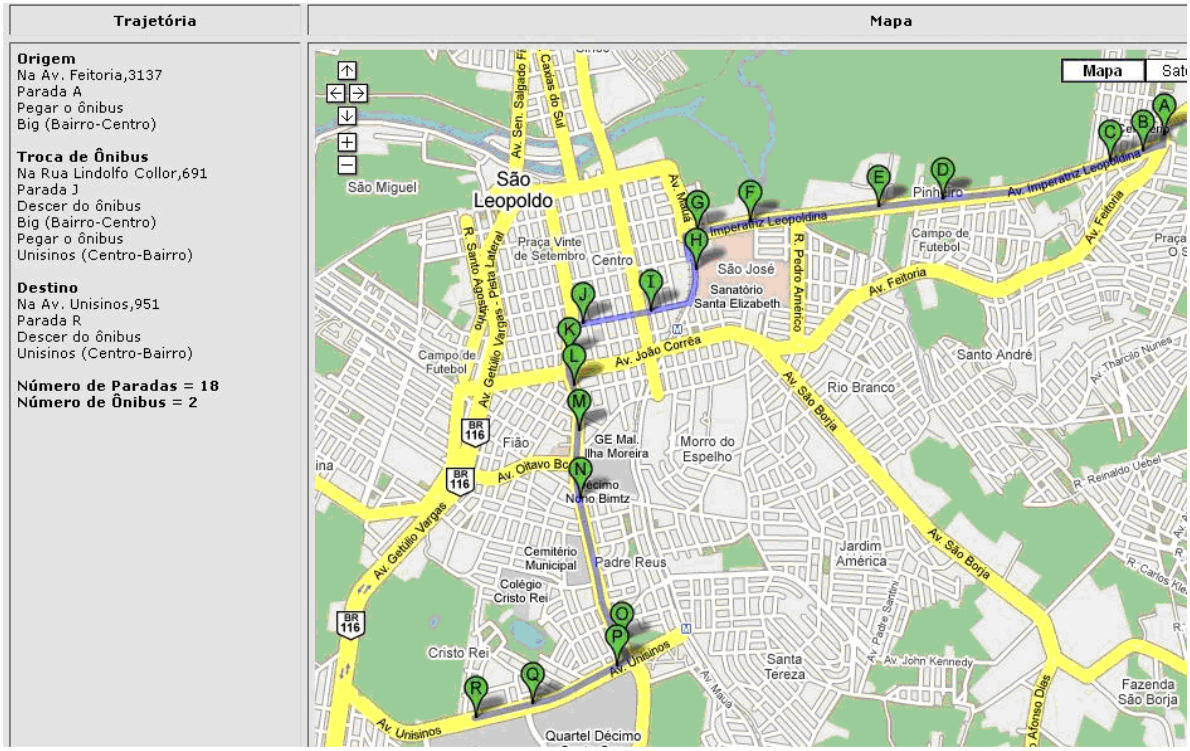

Figura 5 - 2.5 - Sistema ANTARES
Fonte: [Bastos and Jaques 2010]

Identifica-se, portanto, que o sistema ANTARES apresentado atende aos requisitos R1 e R3, não atendendo os requisitos restantes, R2 e R4.

OneBusAway:

Há em funcionamento nas regiões de Atlanta, Nova Iorque, Puget Sound e Tampa, nos Estados Unidos, um sistema de informações ao usuário denominado OneBusAway [Ferris et al. 2009]. Essa ferramenta provê informações em tempo real sobre o trânsito de ônibus nas regiões onde atua, oferecendo aos usuários informações sobre paradas de ônibus, tempo de chegada dos ônibus nas paradas, além de um sistema de planejamento de viagens.

O sistema disponibiliza seis diferentes interfaces para os usuários, sendo elas: (i) interface web, disponível através da página do OneBusAway; (ii) unidade de resposta audível (URA) por telefone; (iii) interface de mensagens SMS; (iv) aplicativo para dispositivos iPhone (Figura 6); (v) aplicativo para dispositivos Android; e (vi) um aplicativo para dispositivos Windows Phone.

As informações em tempo real sobre a movimentação dos veículos utilizadas pelo OneBusAway são obtidas de uma entidade pública externa, sendo somente tratadas e processadas para serem utilizadas no sistema. O sistema utiliza informações contextuais ligadas à localização do usuário para apresentar alertas sobre problemas em seu trajeto.

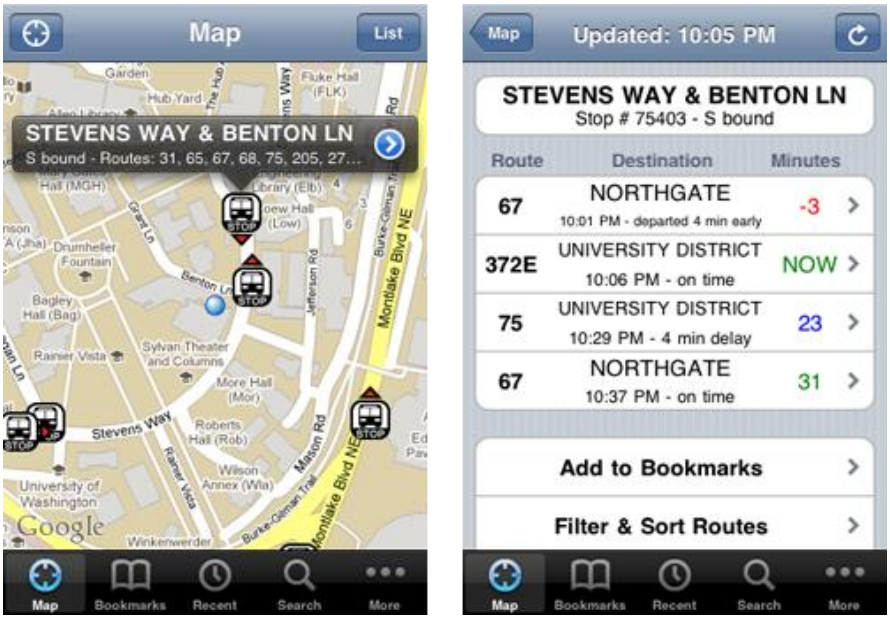

Figura 6 - 2.6 - Interface para iPhone do sistema OneBusAway
Fonte: [Ferris 2011]

Em [Ferris et al. 2010b], os autores apresentam os resultados provenientes de uma pesquisa realizada via internet com usuários do sistema OneBusAway. Ela revelou a opinião de usuários sobre o impacto do uso de um sistema de informações em tempo real sobre ônibus em suas vidas. Os pesquisadores concluíram, a partir dos dados obtidos, que os respondentes da pesquisa em geral sentiram maior satisfação com o trânsito, fazem mais viagens de ônibus por semana, passam menos tempo esperando pelos ônibus, sentem-se mais seguros ao utilizar o meio de transporte público e tendem a caminhar mais quando utilizam os ônibus.

Segundo os requisitos especificados no início desta seção, observa-se que o sistema OneBusAway atende a todos.

Sistema de Informações de Trânsito para a Cidade de Calgary, Canada

Hoar [2010], por sua vez, descreveu em 2010 um sistema de informação de trânsito para a cidade de Calgary, no Canadá, baseado na web que apresenta para os usuários algumas informações sobre linhas de ônibus, localização das paradas de ônibus e dos veículos, bem como o horário de chegada dos ônibus nas paradas. O sistema ainda permite que os usuários compartilhem informações sobre o estado atual de um dado ônibus ou trem através da submissão de relatórios quando um veículo deixa a parada, permitindo o

rastreamento dos veículos em tempo real. Através desses relatórios, ele é capaz de ajustar a posição relativa dos ônibus e publicar para os usuários as novas informações.

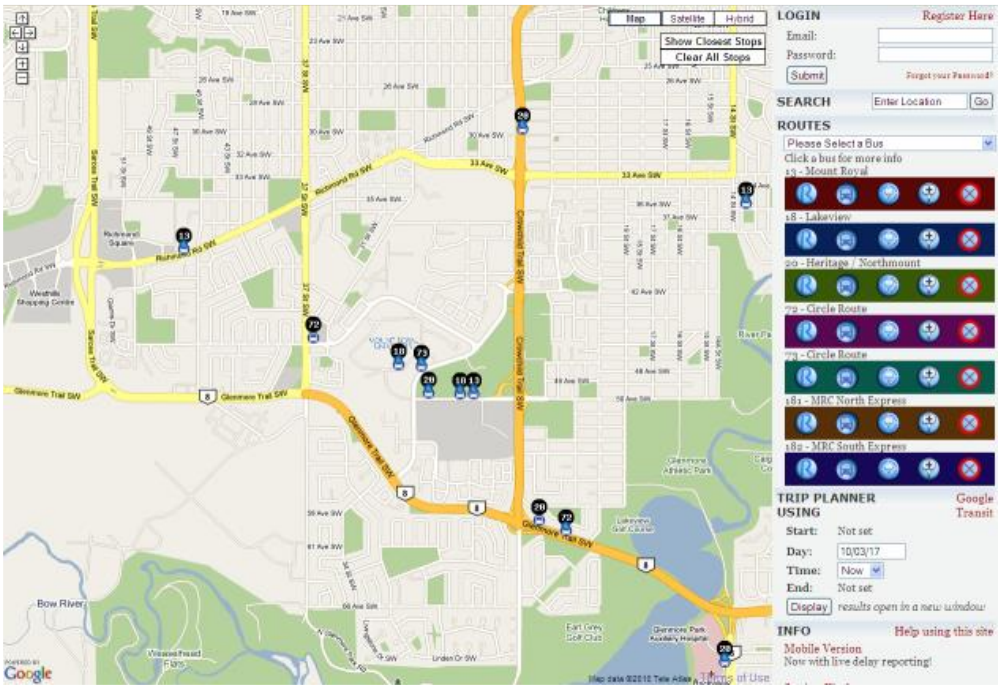

Figura 7 - 2.7 - Interface Web do Sistema de Hoar
Fonte: [Hoar 2010]

O sistema de Hoar foi implementado sobre um framework baseado em *web services*, facilitando o design e implementação de interfaces para múltiplas plataformas, apresentando três interfaces: uma interface web (Figura 7); uma para quiosques de informação em estações de ônibus, universidades, entre outros lugares; e uma interface para dispositivos móveis.

O mecanismo apresentado que permite que os usuários compartilhem se um determinado veículo partiu adiantado, atrasado ou em seu tempo correto, não é pervasivo: ele exige que o usuário tenha uma interação direta com a interface do sistema, informando a situação da partida do veículo. As informações coletadas ajudam no processo de previsão de chegada dos ônibus nas paradas, mas o sistema não leva em consideração as condições climáticas e de trânsito do local e do momento.

Além disso, o sistema de Hoar ainda permite que os usuários se cadastrem, possibilitando que mantenham uma lista de rotas e paradas favoritas, facilitando o acesso nas próximas vezes em que utilizarem o sistema. Essa funcionalidade foi desenvolvida prevendo uma futura integração com o restante do sistema, permitindo que os usuários configurem em seus perfis informações que possam facilitar o planejamento de viagens personalizado, como distância máxima a ser percorrida a pé.

De acordo com os requisitos elencados no início desta seção, observa-se que o sistema apresentado em [Hoar 2010] atende aos requisitos R1, R2 e R4, porém não atende ao requisito R2 pois o sistema em si não oferece uma funcionalidade para o cálculo de rotas, que é realizada utilizando o sistema de planejamento de viagens do Google Transit.

Google Transit

O sistema Google Transit, apresentado anteriormente na seção 2.3 , foi uma ferramenta web desenvolvida pelo Google em 2005 e que mais tarde foi integrada ao sistema produto Google Maps[38]. A ferramenta oferece entre seus serviços, um serviço de planejamento de viagens, e um serviço para visualização em tempo real da situação do trânsito (Figura 8). O sistema de planejamento de viagens do Google Maps abrange várias modalidades, incluindo deslocamentos a pé, de bicicleta, em veículos de passeio ou por transporte público, e leva em consideração algumas informações contextuais como a situação do trânsito e o histórico de utilização do sistema para classificar as rotas geradas.

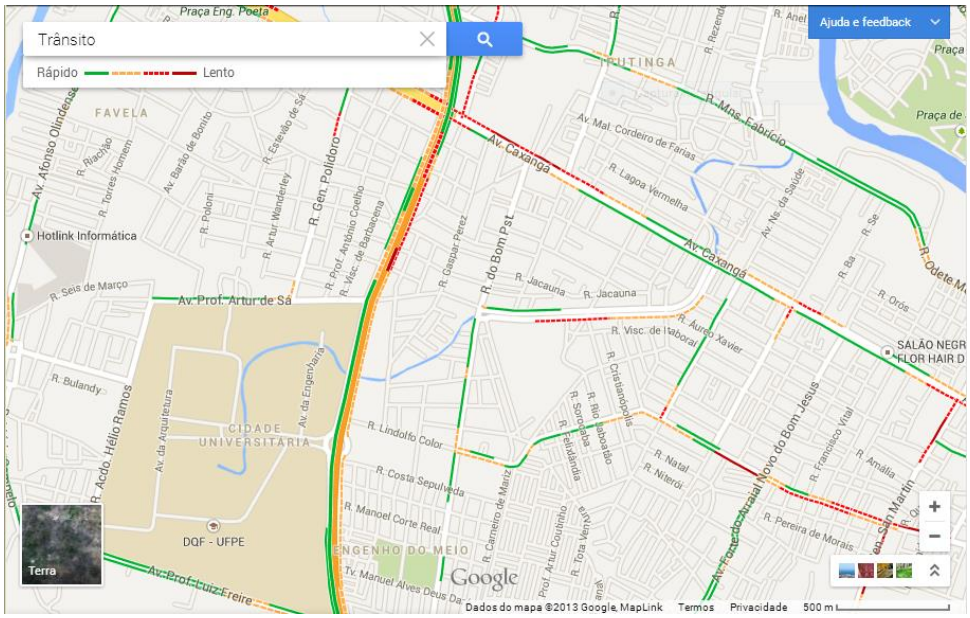

Figura 8 - 2.8 - Visualização do Trânsito no Google Maps

Segundo os requisitos elencados no início desta seção, observa-se que o sistema Google Maps atende a todos.

Além dos trabalhos apresentados acima, podem-se citar alguns sistemas de planejamento de viagens já detalhados na seção 2.3 , como os sistemas Calgary Transit[39],

[38] Maiores informações em: http://www.google.com/maps/about/explore/

[39] Disponível em: http://www.calgarytransit.com/

OPTI-TRANS[40], Translink[41], e os sistemas “511” que operam em diversos estados dos EUA. Todos esses sistemas atendem aos requisitos apresentados no início desta seção.

A partir dos sistemas apresentados, identifica-se como oportunidade de pesquisa o desenvolvimento de sistemas de planejamento de viagens para transporte público por ônibus que sejam capazes de operar sem a necessidade de informações externas que sejam vinculadas a empresas e/ou aos governos, especialmente informações em tempo real sobre o trânsito e sobre os veículos.

Pode-se identificar também como oportunidade de pesquisa a aplicação do contexto do usuário e do trânsito nos processos de geração de rotas, que na maioria dos sistemas apresentados acontece de forma generalizada, i.e. as rotas geradas pelos sistemas de planejamento de viagens citados não são, em geral, personalizadas para o usuário.

Nota-se também que nenhum dos trabalhos apresentados propõe estratégias para personalizar, de maneira automatizada, suas interfaces e resultados conforme as preferências do usuário, como em situações onde um usuário possua alguma deficiência visual ou motora. Assim, cita-se como oportunidade de pesquisa a exploração de soluções capazes de adaptar interfaces e resultados de APTS de acordo com o contexto deus usuários, e.g. interfaces adaptadas para deficientes visuais e rotas personalizadas para usuários com deficiência motora.

Tendo em vista as oportunidades identificadas, no capítulo seguinte será apresentada uma proposta para um sistema de geração e classificação de rotas de ônibus sensível ao contexto, denominado Pilgrim.

[40] Disponível em: http://www.optitrans.net/

[41] Disponível em: http://www.translink.ca/

3 PILGRIM: UM SISTEMA DE GERAÇÃO E CLASSIFICAÇÃO DE ROTAS DE ÔNIBUS SENSÍVEL AO CONTEXTO

Dentro do contexto do projeto UbiBus, este trabalho propõe-se a apresentar um sistema capaz de gerar e classificar rotas de ônibus levando em consideração o contexto dos veículos, dos usuários e do trânsito em geral. Desta forma, este capítulo apresenta o sistema Pilgrim, componente da camada de middleware do sistema UbiBus, expondo seus requisitos e as soluções, tecnologias, arquiteturas e algoritmos utilizados e explorados em seu desenvolvimento. Serão apresentados também o UbiBus e duas propostas de implementação para o sistema Pilgrim, explorando as técnicas de Algoritmos Genéticos e Hill-Climbing com Reinício Aleatório para o processo de geração e classificação das rotas.

Este capítulo está estruturado da seguinte forma: a seção 3.1 apresenta o projeto UbiBus; a seção 3.2 apresenta uma visão geral do sistema Pilgrim, sua arquitetura, seus requisitos, as tecnologias utilizadas e a modelagem do sistema; a seção 3.3 apresenta os detalhes gerais da implementação, independentes dos algoritmos de classificação utilizados; a seção 3.4 apresenta uma proposta para a implementação do sistema utilizando a técnica de Algoritmos Genéticos; e a seção 3.5 apresenta uma proposta para a implementação do sistema utilizando a técnica de Hill-Climbing com Reinício Aleatório.

3.1 PROJETO UBIBUS

Em 2011, [Vieira et al. 2011] apresentaram uma proposta para um sistema de transporte público ubíquo e sensível ao contexto, chamado UbiBus. O UbiBus visa especificar e implementar soluções técnicas (e.g. modelos, algoritmos e ferramentas) para facilitar o acesso a informações sobre o transporte público. Ele se usa de informações de contexto dinâmicas em tempo real coletadas de diferentes fontes, considerando a mobilidade dos veículos e passageiros, bem como fatores dinâmicos que podem afetar o transporte. Diferentes serviços são providos aos passageiros, como previsão e recomendação de rotas, estimativa de tempo de chegada, entre outros. Tais serviços devem ser acessíveis de diferentes dispositivos (e.g. desktops, terminais nas paradas de ônibus, monitores dentro dos ônibus, dispositivos móveis).

O UbiBus investiga diferentes aspectos relacionados ao desenvolvimento de: (i) técnicas de aquisição, processamento e gerenciamento de informações contextuais, tanto estáticas quanto dinâmicas; (ii) algoritmos e modelos matemáticos para calcular tempos de viagem e para definir melhores rotas de ônibus; (iii) técnicas para visualizar e interpretar

informações geográficas, considerando grandes volumes de dados de transportes e rotas; (iv) um *middleware* para auxiliar a construção de aplicações ubíquas sensíveis ao contexto; (v) diferentes aplicações sensíveis ao contexto para auxiliar os passageiros, adaptáveis e disponíveis em diferentes dispositivos; e (vi) sistemas de recomendação de rotas e veículos.

A Figura 9 abaixo apresenta uma visão-geral da arquitetura do UbiBus com suas 5 camadas principais: Dados, Comunicação, Aquisição, Processamento e Aplicação [Vieira et al. 2012].

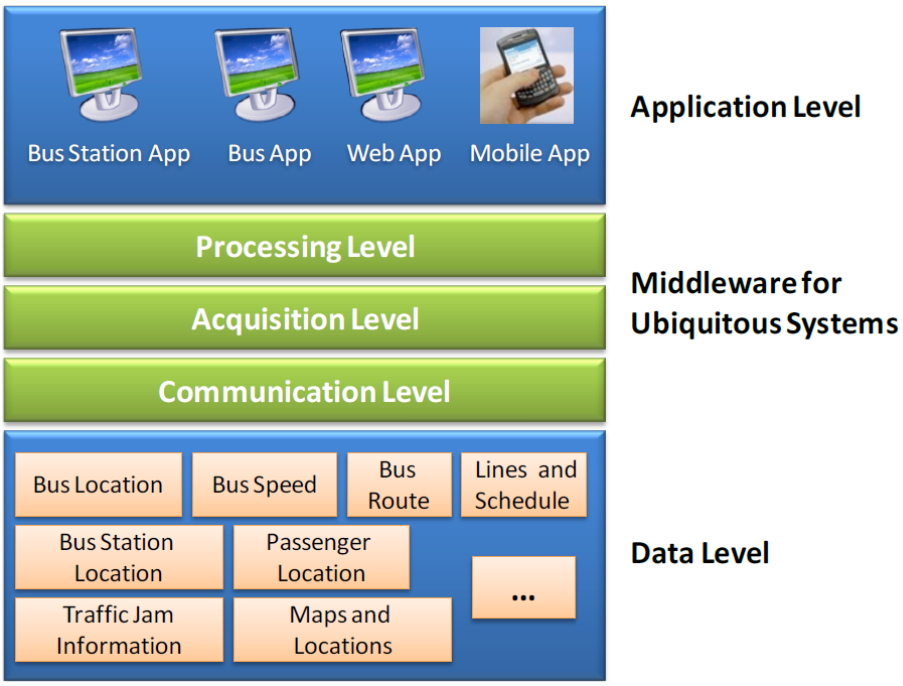

Figura 9 - 3.1 - Visão Geral da Arquitetura do UbiBus
Fonte: [Vieira et al. 2012]

A camada de Dados é responsável pelo gerenciamento dos dados processados pelo sistema, incluindo representação de dados, armazenamento e recuperação. O banco de dados contém contexto espacial georeferenciado para as trajetórias dos objetos móveis (e.g. ônibus, carros, metrôs, trens e pessoas). Diferentes dados e informações podem ser consideradas: localização, velocidade e rotas dos ônibus, localização das paradas de ônibus, localização dos passageiros, informações sobre congestionamentos, mapas e localizações, linhas e itinerários, entre outros. Os dados sobre os objetos móveis são atualizados constantemente, de maneira a representar seus movimentos durante o tempo.

A camada de Comunicação permite a conexão entre elementos estáticos e dinâmicos que compõem a infraestrutura de transporte, permitindo a troca de informações em tempo real entre gerentes, operadores, usuários, motoristas, veículos e outros elementos que permeiam as rodovias. Os avanços e a padronização de tecnologias de

comunicação sem fio (e.g.: WiFi, Bluetooth, WiMAX) permitem comunicações de curto e longo alcance. Informações sobre condições do tráfego e tabelas de tempo estimado de viagem podem ser oferecidas diretamente aos usuários, considerando a grande disponibilidade de conectividade baseada em tecnologias de comunicação sem fio entre dispositivos móveis e laptops.

A camada de Aquisição é responsável por coletar informações contextuais estáticas e dinâmicas de diferentes fontes, enviando-as para a camada de Dados. A localização e velocidade dos ônibus, por exemplo, podem ser coletadas através de sistemas GPS ou RFID disponíveis nos ônibus ou nas paradas de ônibus; informações de tráfego podem ser provenientes de câmeras, informações de redes sociais, monitoramento de sistemas de tráfego ou por inferência através do uso de modelos matemáticos; informações climáticas podem ser obtidas por sistemas meteorológicos externos; informações dos passageiros (e.g. localização atual ou próximo destino) podem ser obtidos através de um dispositivo móvel pessoal ou pelos próprios usuários.

A camada de Processamento utiliza informações contextuais coletadas junto com soluções matemáticas e algorítmicas para calcular o tempo estimado de chegada dos ônibus. Esta camada também é responsável por pré-processar as informações coletadas, garantindo a compatibilidade entre os diferentes sistemas de posicionamento (e.g. GPS e RFID) e pela incorporação de semântica às trajetórias geradas pelas unidades de transporte móvel.

A camada de Aplicação contém os diferentes tipos de aplicações construídas sobre a infraestrutura do UbiBus. Estas aplicações devem ser planejadas para diferentes plataformas e dispositivos (e.g. Web, desktop, dispositivos móveis, terminais nas paradas de ônibus ou monitores dentro dos ônibus). As funcionalidades oferecidas por cada uma das aplicações processam os dados da camada de Dados e proveem informações atualizadas para o usuários sobre rotas e ônibus que eles pretendem utilizar. As aplicações devem ser adaptáveis, visto que diferentes dispositivos possuem diferentes requisitos e interfaces, limitações e/ou requisitos de usabilidade e conteúdo.

3.2 VISÃO GERAL DO SISTEMA PILGRIM

O Pilgrim é um sistema para geração e classificação de rotas de ônibus sensível ao contexto, que a partir de um conjunto de informações fornecidas em uma requisição, gera um conjunto de rotas classificadas de acordo com alguns critérios como tempo total da viagem, necessidade de veículo com acessibilidade e número de trocas de ônibus.

As informações que compõem uma requisição são: (i) o ponto de origem da viagem, que pode ser uma parada de ônibus específica ou uma coordenada geográfica; (ii) o ponto de destino da viagem, que também pode ser uma parada de ônibus ou uma coordenada geográfica; (iii) o horário em que a viagem irá acontecer; (iv) o número máximo de trajetos nas rotas geradas, i.e. cada trajeto adicional representa uma nova troca de ônibus; (v) necessidade de veículos com acessibilidade; e (vi) número máximo de rotas geradas.

As rotas geradas pelo sistema são retornadas para o requisitante em formato JSON, facilitando a interoperabilidade entre sistemas desenvolvidos com tecnologias diferentes, o que é um fator fundamental visto que o UbiBus prevê aplicações para múltiplas plataformas.

O sistema Pilgrim é disponibilizado na camada de middleware do UbiBus como um serviço, provendo uma interface (Pilgrim API) para outras aplicações ou serviços do UbiBus que precisem acessá-lo. A Figura 10, abaixo, representa uma visão geral da arquitetura do Pilgrim dentro do sistema UbiBus, indicando as interações entre os módulos em um cenário onde um usuário faz uma requisição a uma aplicação de recomendação de rotas do UbiBus.

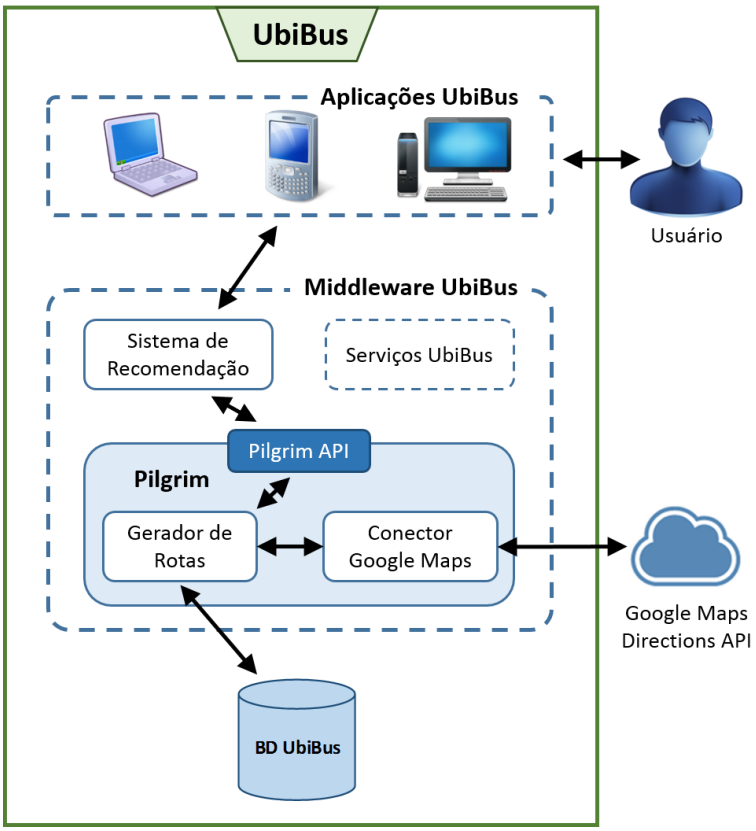

Figura 10 - 3.2 - Visão Geral da Arquitetura do Pilgrim

O Pilgrim acessa diretamente o banco de dados do UbiBus, e utiliza um serviço oferecido pelo Google Maps para o cálculo das rotas a serem percorridas a pé. Outros

sistemas e serviços do UbiBus se comunicam com o sistema Pilgrim exclusivamente através de sua interface pública: Pilgrim API.

Serão exploradas duas técnicas de inteligência artificial para o processo de geração e classificação das rotas, a saber: Algoritmos Genéticos, escolhidos por já terem sido utilizados em processos de geração e classificação de rotas para veículos de passeio e veículos militares [Pellazar 1994; Kanoh and Hara 2008; Kumar et al. 2009]; e Hill-Climbing com reinício aleatório, escolhido por seus baixos custos de tempo e memória, e por ser capaz de rapidamente aprimorar a qualidade uma rota inicial [Jacobson et al. 2006; Zaheer 2006; Chiong et al. 2008].

3.2.1. REQUISITOS DO SISTEMA

Nesta seção serão apresentados os requisitos para o sistema Pilgrim, que servirão como base para a especificação do sistema e para a implementação das técnicas a serem exploradas citadas na seção anterior.

Há algumas diferenças quando se trata de geração e recomendação de rotas para transporte público – mais especificamente ônibus, principal foco deste trabalho – e para veículos de transporte particular. Primeiramente pode-se citar a diferença com relação à liberdade de percurso, i.e. ao utilizar veículos particulares, um usuário tem poucas limitações quanto ao percurso percorrido em sua viagem, enquanto que ao utilizar ônibus, o percurso dos veículos é limitado a linhas determinadas, fazendo com que mudanças no percurso sejam atreladas a trocas de ônibus.

Em segundo lugar, do ponto de vista da recomendação de rotas, os pontos a serem analisados para a classificação de diferentes rotas também é diferente: quando um usuário está utilizando um transporte público como um ônibus, o custo do transporte praticamente não varia de acordo com a distância percorrida, mas sim com outros fatores como, entre outros, quantidade de trocas de veículos e duração da viagem, vistos em casos onde, por exemplo, uma passagem de ônibus é válida por um determinado período de tempo e/ou para uma quantidade de trocas de veículos fixa [SPTRANS 2013]; já quando um usuário utiliza um meio de transporte particular, existem outros fatores que podem influenciar na recomendação como, por exemplo, a distância percorrida que impacta diretamente no gasto de combustível do veículo (ou diretamente no custo da viagem, caso o transporte seja um taxi) e as condições da via, i.e. se existem muitos buracos, se a sinalização é boa, entre outros.

Dessa forma, foram elaborados um conjunto de requisitos funcionais e não-funcionais para o sistema Pilgrim, apresentados abaixo:

Requisitos Funcionais:

RF01: Gerar, a partir de informações passadas durante a requisição ao sistema, um conjunto de rotas classificadas de acordo com sua qualidade, levando em consideração o contexto do trânsito e dos veículos para o horário da viagem;

RF02: Permitir que os pontos de origem e destino sejam informados através de coordenadas geográficas ou paradas de ônibus específicas;

RF03: Permitir que o número máximo de trocas de ônibus das rotas seja especificado no momento de uma requisição;

RF04: Permitir que o número máximo de rotas geradas seja especificado no momento de uma requisição;

RF05: Disponibilizar uma interface pública de aplicação (API) que possa ser acessada por outros sistemas;

RF06: Representar as rotas geradas em formato JSON;

Requisitos Não-Funcionais:

RNF01: O tempo de resposta do sistema não deve ultrapassar 7 segundos;

RNF02: As informações armazenadas no banco de dados do UbiBus não devem ser modificadas pelo sistema;

3.2.2. TECNOLOGIAS UTILIZADAS

O banco de dados do UbiBus opera utilizando o SGBD objeto-relacional PostgreSQL[42], estendido pelo complemento para bancos de dados espaciais PostGIS[43].

O módulo Gerador de Rotas, principal componente do sistema, foi implementado utilizando a linguagem de programação C# (“C Sharp”), utilizando o ambiente de

[42] Disponível em: http://www.postgresql.org/

[43] Disponível em: http://postgis.net/

desenvolvimento Microsoft Visual Studio 2012. O driver para conexão com o SGBD utilizado foi o Npgsql[44], e os caminhos a pé percorridos pelos usuários são calculados utilizando um serviço disponível na *Directions API*[45] do Google Maps.

O sistema Pilgrim é disponibilizado como um serviço dentro da camada de middleware do UbiBus, através de um *wrapper* desenvolvido na linguagem de programação Java, onde a comunicação entre o *wrapper* e o sistema é implementada utilizando o framework JNI (*Java Native Interface*). O desenvolvimento do *wrapper* mencionado não faz parte do escopo deste trabalho, portanto não será detalhado.

3.2.3. MODELAGEM DO SISTEMA

Tendo em vista os requisitos apresentados na seção 3.2.1, será apresentada a seguir a modelagem do sistema Pilgrim. A Figura 11 apresenta o diagrama de classes simplificado do sistema, e cada uma das estruturas representadas será explicada em seguida.

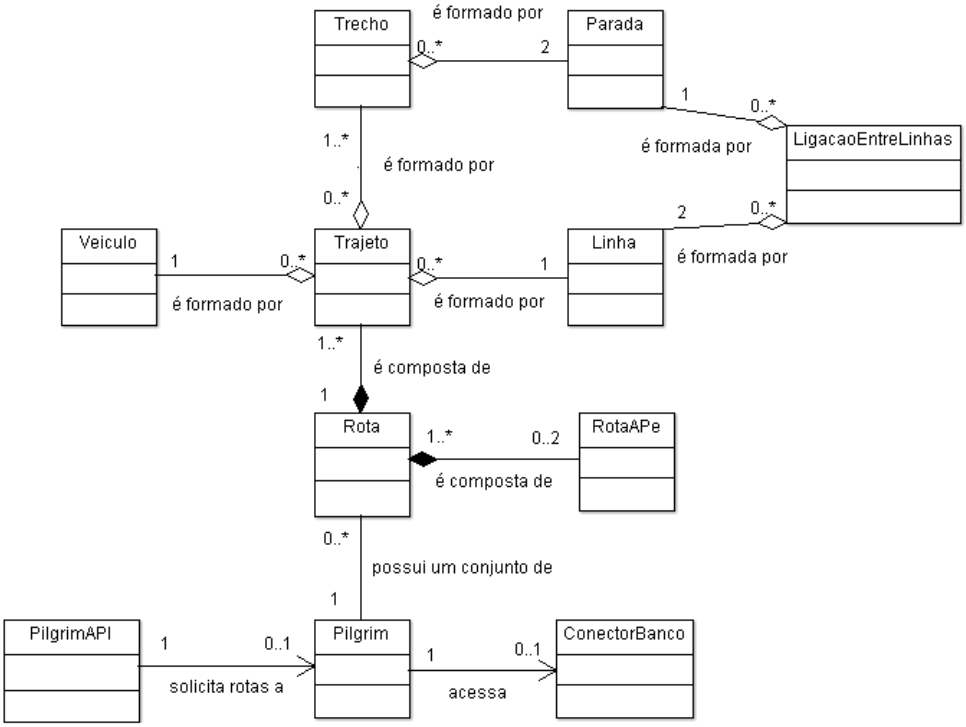

Figura 11 - 3.3 - Diagrama de Classes Simplificado do Pilgrim

[44] Disponível em: http://npgsql.projects.pgfoundry.org/

[45] Mais informações em: https://developers.google.com/maps/documentation/directions/

As 11 classes que compõem o sistema Pilgrim são:

- **PilgrimAPI:** é a interface pública da aplicação, responsável por expor os métodos a serem acessados através do middleware do UbiBus, encapsulando todo o sistema Pilgrim.
- **Pilgrim:** é a classe responsável pelo processamento das informações extraídas do banco de dados, gerando e classificando as possíveis rotas para determinados pontos de origem e de destino. Esta é a classe principal do sistema, contendo a maior parte da lógica responsável pelo seu funcionamento, incluindo a geração e classificação das rotas, e contém também o módulo que se comunica com o Google Maps para geração de rotas a pé.
- **ConectorBanco:** é a classe responsável por encapsular toda a comunicação com o banco de dados do UbiBus, obtendo todas as informações referentes a paradas, linhas, veículos, trechos e situação do trânsito.
- **Rota:** é uma classe que representa uma rota completa a ser percorrida por um usuário, partindo do ponto de origem e indo até o ponto de destino. Cada rota é composta por um conjunto de trajetos e por até duas rotas a pé, opcionais, uma ligando o ponto de origem à primeira parada de ônibus e outra ligando a última parada de ônibus ao ponto de destino, e armazena também informações sobre a distância total em metros e o tempo total estimado da rota. A existência de mais de um trajeto em uma rota significa que, para cada trajeto adicional, haverá uma troca de ônibus.
- **RotaAPe:** é uma classe que representa o caminho a ser percorrido a pé pelo usuário para chegar de uma certa origem a um certo destino, e armazena informações sobre sua distância total em metros, as coordenadas geográficas da rota e o tempo total estimado.
- **Trajeto:** é uma classe que representa um trajeto percorrido por um usuário em um determinado veículo e em uma linha específica. O trajeto é composto por um veículo, uma linha e uma sequência de trechos.
- **Trecho:** é uma classe que representa um trecho, i.e. um conjunto de duas paradas, inicial e final, que são adjacentes em uma mesma linha de ônibus. Armazena informações sobre seu comprimento em metros, a velocidade média, a duração estimada para se percorrer o trecho, os horários em que o trânsito no trecho tende a engarrafar, e uma lista de coordenadas geográficas dos pontos do trecho.
- **Veiculo:** é uma classe que representa um ônibus, e armazena informações sobre suporte à acessibilidade e sobre a coordenadas geográficas da localização atual.

- **Parada:** é uma classe que representa uma parada de ônibus, e armazena informações sobre o nome da parada, i.e. o nome pelo qual a parada é conhecida na região, as coordenadas geográficas de sua localização e a distância entre a parada e os pontos de origem e de destino informados na requisição ao sistema.
- **Linha:** é uma classe que representa uma linha de ônibus. Armazena informações sobre o número e o nome da linha, a empresa permissionária que a opera, e possui uma lista ordenada das paradas e dos trechos pelos quais a linha passa, uma lista dos veículos que operam nela e uma lista de todas as ligações com outras linhas.
- **LigacaoEntreLinhas:** é uma estrutura que representa um ponto de conexão entre duas linhas, i.e. quando duas linhas de ônibus passam por uma mesma parada, essa parada é o ponto de conexão entre as linhas. Armazena informações sobre a linha e sobre a parada onde a conexão ocorre.

3.3 DETALHES GERAIS DA IMPLEMENTAÇÃO

Nesta seção serão apresentados os detalhes gerais da implementação do sistema Pilgrim que são independentes das técnicas utilizadas para a geração e classificação das rotas.

Processamento de Requisições:

A interface pública do Pilgrim, PilgrimAPI, disponibiliza um único método para outros serviços e sistemas através da camada de middleware do UbiBus. Este método recebe, conforme apontado na seção 3.2 , seis informações: (i) o ponto de origem da viagem, que pode ser uma parada de ônibus específica ou uma coordenada geográfica, representado de forma textual; (ii) o ponto de destino da viagem, que também pode ser uma parada de ônibus ou uma coordenada geográfica, representado de forma textual; (iii) o horário em que a viagem irá acontecer, representado de forma textual segundo o padrão estabelecido pela norma ISO 8601 [International Standards Organization 2004]; (iv) o número máximo de trajetos nas rotas geradas, i.e. cada trajeto adicional representa uma nova troca de ônibus, representado através de um número inteiro; (v) necessidade de veículos com acessibilidade, representado através de um número binário; e (vi) número máximo de rotas geradas, representado através de um número inteiro.

Os pontos de origem e destino, representados em forma textual no momento das requisições ao sistema, devem obedecer às seguintes regras: caso os pontos sejam

referentes a coordenadas geográficas, cada ponto deve ser representado por dois números reais, separados entre si por uma vírgula, sendo o primeiro referente à latitude e o segundo à longitude dos pontos (e.g.: “-7.117716, -34.890003”); e caso os pontos sejam referentes a paradas de ônibus específicas, cada ponto deve ser representado pelo caractere “p” seguido do número identificador da parada de ônibus (e.g.: “p305”).

Requisições com Coordenadas Geográficas:

Nos casos onde os pontos de origem e/ou destino são coordenadas geográficas, o sistema calcula diferentes rotas considerando diferentes paradas iniciais e/ou finais, e.g. em uma solicitação onde são passadas para o sistema Pilgrim uma coordenada como ponto de origem e uma parada específica como ponto de destino, o sistema irá gerar rotas partindo de cada uma das 5 paradas mais próximas à coordenada de origem, possibilitando que a melhor parada inicial seja encontrada.

Ao receber as coordenadas dos pontos de origem e/ou destino, o sistema calcula as distâncias entre todas as paradas de ônibus e os pontos, atualizando essas informações nas próprias paradas.

Requisições com Paradas Específicas:

Em situações onde os pontos de origem e/ou destino são paradas de ônibus específicas, o sistema realiza uma busca para encontrar as paradas de acordo com seus números identificadores únicos, e calcula as rotas utilizando as próprias paradas como origem e/ou destino. Dessa forma, as rotas geradas para uma requisição onde os pontos de origem e destino são paradas específicas não terão rotas a serem percorridas a pé.

Geração de Rotas a Pé:

Os caminhos a serem percorridos a pé pelo usuário entre seu ponto de origem e a primeira parada da rota e/ou entre a última parada da rota e o ponto de destino são obtidos através de uma requisição à API de cálculo de rotas do Google Maps (*Directions* API).

Os parâmetros enviados para a API são as coordenadas de origem e destino do caminho a ser calculado, e uma informação indicando que o caminho seja calculado considerando que será percorrido a pé. A API então processa essas informações e retorna um JSON contendo as informações do caminho, que é tratado pelo sistema e armazenado em um objeto RotaAPe.

Comunicação com o Banco de Dados:

O Pilgrim se comunica com o banco de dados do UbiBus assim que o sistema é inicializado, onde todas as informações são carregadas: paradas, linhas, veículos, entre outros. Todos os relacionamentos entre as entidades são mapeados e o sistema constrói um modelo próprio, como apresentado na seção 3.2.3.

Uma vez que todas as informações tenham sido carregadas, o sistema irá consultar o banco novamente somente quando necessitar de informações em tempo real atualizadas, como a posição atual dos veículos.

Desta forma, visto que a execução mais custosa (carregar todas as informações do BD durante a inicialização) é feita uma única vez logo no início do ciclo de vida do Pilgrim, o tempo de resposta do sistema às requisições recebidas não é afetado por operações de acesso ao banco de dados.

Conforme citado no início desta seção, os detalhes da implementação, específicos de cada técnica utilizada para a geração e classificação das rotas, serão apresentados nas próximas seções, junto com maiores informações sobre a forma como as técnicas exploradas são adaptadas para o domínio do UbiBus.

3.4 IMPLEMENTAÇÃO UTILIZANDO ALGORITMOS GENÉTICOS

Esta seção apresenta uma proposta de implementação para o sistema Pilgrim baseada na técnica de Algoritmos Genéticos, discutida na seção 2.4 , descrevendo a representação do domínio do UbiBus e os detalhes específicos da implementação da funcionalidade de geração e classificação de rotas utilizando essa técnica.

Para que o algoritmo genético seja capaz de calcular as rotas, é necessário que as informações sobre ônibus, linhas, paradas, rotas e outras entidades relevantes à aplicação sejam codificadas em um cromossomo. Considerando a modelagem proposta para o sistema Pilgrim, apresentada na seção 3.2.3, um cromossomo deve representar uma rota completa, i.e. uma rota contendo trajetos que levem um usuário de seu ponto de origem até seu destino.

O funcionamento de um algoritmo genético depende a existência de uma população inicial de indivíduos (cromossomos), que é iterativamente evoluída utilizando processos de seleção, cruzamento e mutação, através de operadores genéticos. Neste trabalho, a população inicial consiste de um conjunto de rotas completas, que a cada

iteração passam por um processo de seleção utilizando o método de torneio [Miller and Goldberg 1995] e por um processo de cruzamento genético, onde novos indivíduos são criados a partir de indivíduos já existentes na população.

Tendo em vista a natureza pouco flexível das rotas de ônibus, i.e. mudanças no percurso de uma rota são possíveis somente através de uma troca de ônibus, onde um passageiro muda para um ônibus que está trafegando em uma linha diferente, o operador de mutação genética foi considerado não apropriado e, assim, não é utilizado nesta proposta de implementação do sistema Pilgrim.

A seguir, serão propostas duas alternativas para a codificação dos cromossomos, indicando o funcionamento dos operadores genéticos em cada caso. Por fim, serão apresentados os detalhes da implementação da alternativa escolhida. Os experimentos realizados e seus resultados correspondentes são discutidos no Capítulo 4 .

PRIMEIRA PROPOSTA DE CODIFICAÇÃO E OPERADORES GENÉTICOS

Nesta proposta, para representar uma rota completa, foram escolhidas as informações sobre o veículo e as paradas que ele irá percorrer. Tendo em vista que o veículo já possui informações sobre a linha em que está trafegando, não é necessário que a linha de ônibus seja representada no cromossomo.

O cromossomo é subdividido em genes, de forma que cada gene representa um conjunto de paradas percorridas em um determinado veículo, conforme a Figura 12. O gene é composto por um veículo e uma sequência de paradas. A existência de mais de um gene em um cromossomo indica que há uma troca de ônibus durante a rota.

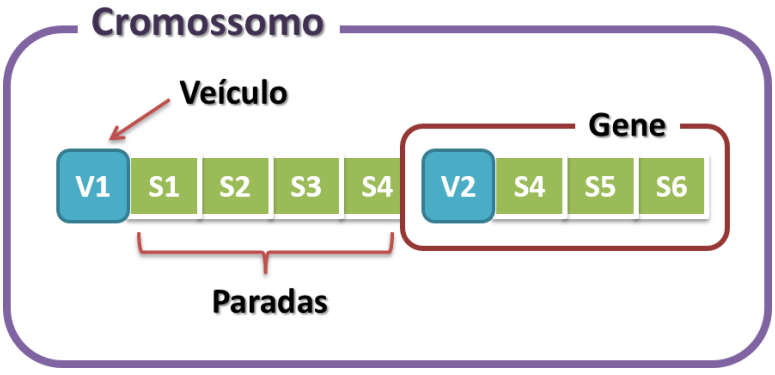

Figura 12 - 3.4 - Primeira Proposta de Cromossomo

Cruzamento Genético

O cruzamento genético dos cromossomos, para esta proposta, foi representado como sendo a troca de linhas de ônibus entre dois indivíduos durante a rota. Essa troca, entretanto, não pode ser feita em qualquer momento do percurso: é necessário que a parada onde a troca irá ocorrer seja parte do percurso (i.e. esteja na linha de ônibus) tanto do veículo origem quanto do veículo destino. Isso significa que o operador genético de cruzamento só pode ser aplicado a dois indivíduos I e I' caso ambos possuam paradas em comum em seus cromossomos.

O processo de cruzamento, representado na Figura 13 abaixo, ocorre da seguinte forma: dados dois indivíduos (pais) $P1$ e $P2$, e um ponto de cruzamento (PC), i.e. a parada onde a troca de ônibus irá ocorrer, são criados dois novos indivíduos (filhos) $F1$ e $F2$, de forma que o filho $F1$ seja composto de: todos os genes de $P1$ anteriores ao gene que contém o ponto de cruzamento, caso existam; um gene contendo o mesmo veículo do gene do pai $P1$ que contém o PC e as paradas desse gene anteriores ao PC, incluindo a própria parada que é o ponto de cruzamento; um gene contendo o mesmo veículo do gene do pai $P2$ que contém o PC e as paradas desse gene posteriores ao PC, incluindo a parada que é o ponto de cruzamento; e todos os genes de $P2$ posteriores ao gene que contém o ponto de cruzamento, caso existam.

Da mesma forma, o filho $F2$ é composto de: todos os genes de $P2$ anteriores ao gene que contém o ponto de cruzamento, caso existam; um gene contendo o mesmo veículo do gene do pai $P2$ que contém o PC e as paradas desse gene anteriores ao PC, incluindo a própria parada que é o ponto de cruzamento; um gene contendo o mesmo veículo do gene do pai $P1$ que contém o PC e as paradas desse gene posteriores ao PC, incluindo a parada que é o ponto de cruzamento; e todos os genes de $P1$ posteriores ao gene que contém o ponto de cruzamento, caso existam.

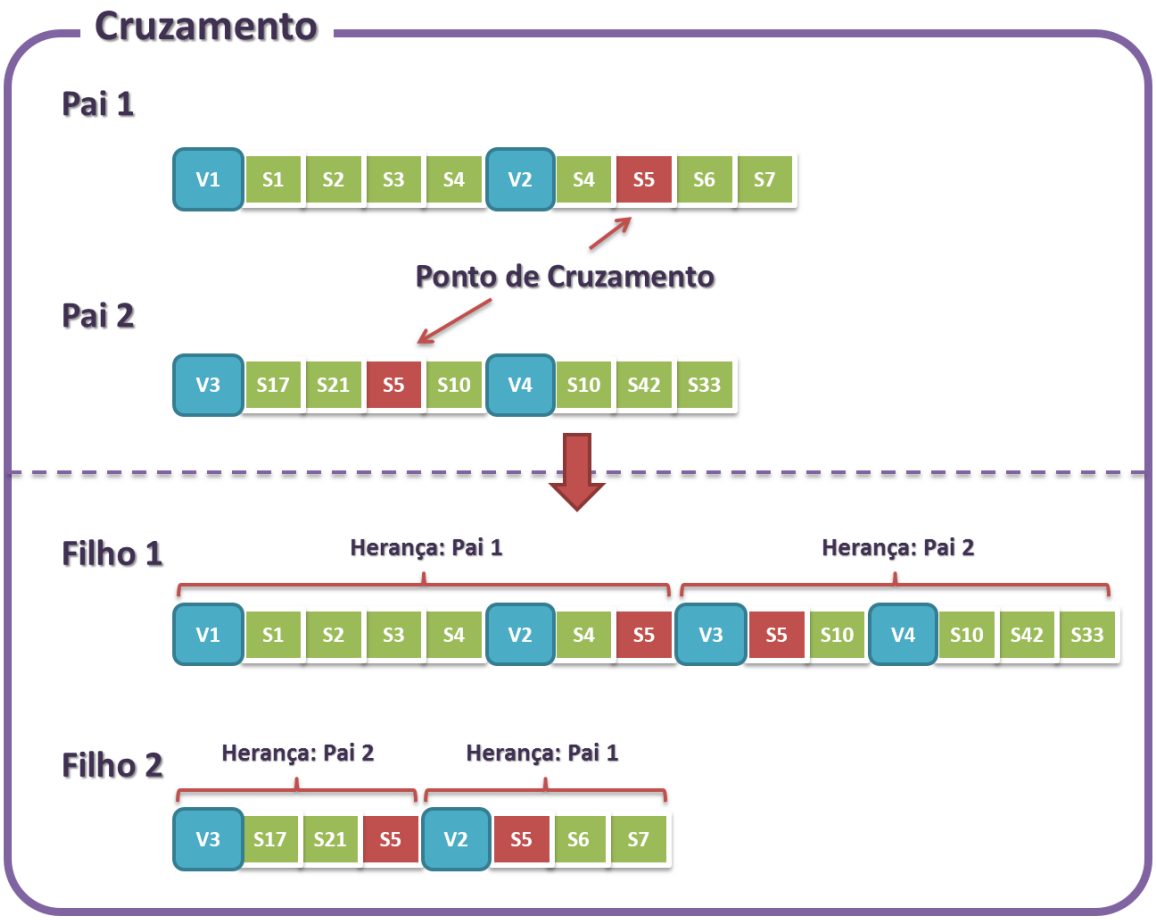

Figura 13 - 3.5 - Primeira Proposta de Cruzamento Genético

Foram realizadas algumas simulações utilizando esta proposta de codificação de cromossomos, e identificou-se que ela não é suficiente para representar o domínio da aplicação, pois durante as simulações notou-se que durante a execução do algoritmo, algumas paradas de ônibus representadas nas rotas ficavam fora de ordem, ou em ordem invertida. Essa mudança na ordenação das paradas invalida as rotas, visto que as rotas resultantes não seriam realizáveis no mundo real, pois os ônibus têm seu percurso definido e imutável durante o trajeto, i.e. os veículos percorrem as paradas em uma sequência, não sendo possível alterar essa ordem.

Para se garantir que as paradas representadas em um cromossomo mantenham sua sequência durante a execução do algoritmo, é necessário que essa informação (a sequência correta das paradas) esteja também codificada no cromossomo. Dessa forma, identificou-se necessária a representação dos trechos nos cromossomos, visto que cada trecho representa duas paradas de ônibus adjacentes, uma inicial e outra final, formando uma sequência unidirecional. Isso garante que uma dada parada $S1$, que seja antecessora no percurso de uma linha de ônibus a outra parada $S2$, não possa ser representada no

cromossomo como sendo sucessora de $S2$ (o que seria equivalente, no mundo real, ao veículo trafegar na contramão).

Segunda Proposta de Codificação e Operadores Genéticos

Nesta segunda proposta de codificação, para representar uma rota completa, foram escolhidas as informações sobre o veículo, a linha de ônibus, e os trechos que o veículo irá percorrer, onde um trecho é um conjunto de duas paradas de ônibus (inicial e final) adjacentes em uma mesma linha.

O cromossomo é subdividido em genes, de forma que cada gene representa um conjunto de trechos percorridos por um determinado veículo, em uma linha de ônibus específica. Assim como na primeira proposta apresentada anteriormente, a existência de mais de um gene em um cromossomo indica que há uma troca de ônibus durante a rota. Na Figura 14 abaixo, que representa um cromossomo, os elementos "V1" e "V2" são veículos, "L1" e "L2" são linhas de ônibus, "C1" a "C6" são trechos e "S1" a "S7" são paradas.

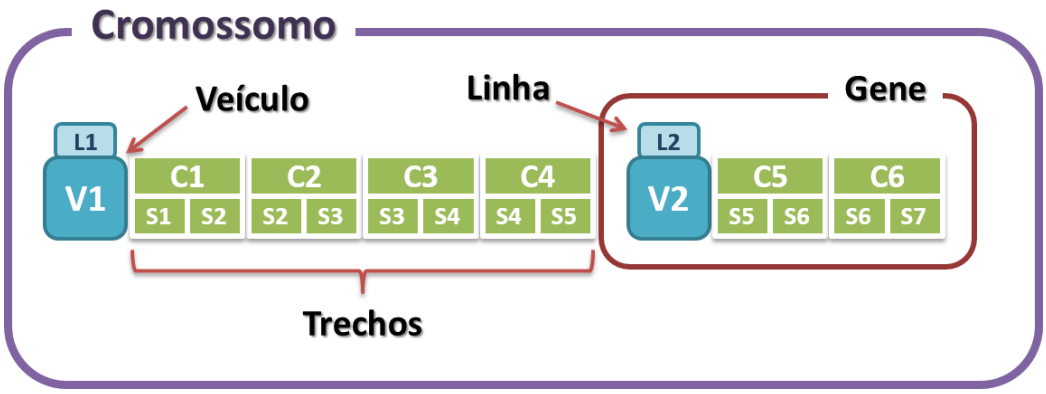

Figura 14 - 3.6 - Segunda Proposta de Cromossomo

Cruzamento Genético

O cruzamento genético dos cromossomos, para esta proposta, foi representado como sendo a troca de linha de ônibus entre dois indivíduos durante a rota, de maneira semelhante à apresentada na proposta anterior.

O processo de cruzamento, representado na Figura 15 abaixo, ocorre da seguinte forma: dados dois indivíduos (pais) $P1$ e $P2$, e um ponto de cruzamento (PC), i.e. a parada onde a troca de ônibus irá ocorrer, são criados dois novos indivíduos (filhos) $F1$ e $F2$, de forma que o filho $F1$ seja composto de: todos os genes de $P1$ anteriores ao gene que contém o ponto de cruzamento, caso existam; um gene contendo o mesmo veículo do gene do pai $P1$ que contém o PC, e os trechos desse gene anteriores ao PC, incluindo o trecho que

contém o ponto de cruzamento como parada final; um gene contendo o mesmo veículo do gene do pai $P2$ que contém o PC e os trechos desse gene posteriores ao PC, incluindo o trecho que contém o ponto de cruzamento como parada inicial; e todos os genes de $P2$ posteriores ao gene que contém o ponto de cruzamento, caso existam.

Da mesma forma, o filho $F2$ é composto de: todos os genes de $P2$ anteriores ao gene que contém o ponto de cruzamento, caso existam; um gene contendo o mesmo veículo do gene do pai $P2$ que contém o PC, e os trechos desse gene anteriores ao PC, incluindo o trecho que contém o ponto de cruzamento como parada final; um gene contendo o mesmo veículo do gene do pai $P1$ que contém o PC e os trechos desse gene posteriores ao PC, incluindo o trecho que contém o ponto de cruzamento como parada inicial; e todos os genes de $P1$ posteriores ao gene que contém o ponto de cruzamento, caso existam.

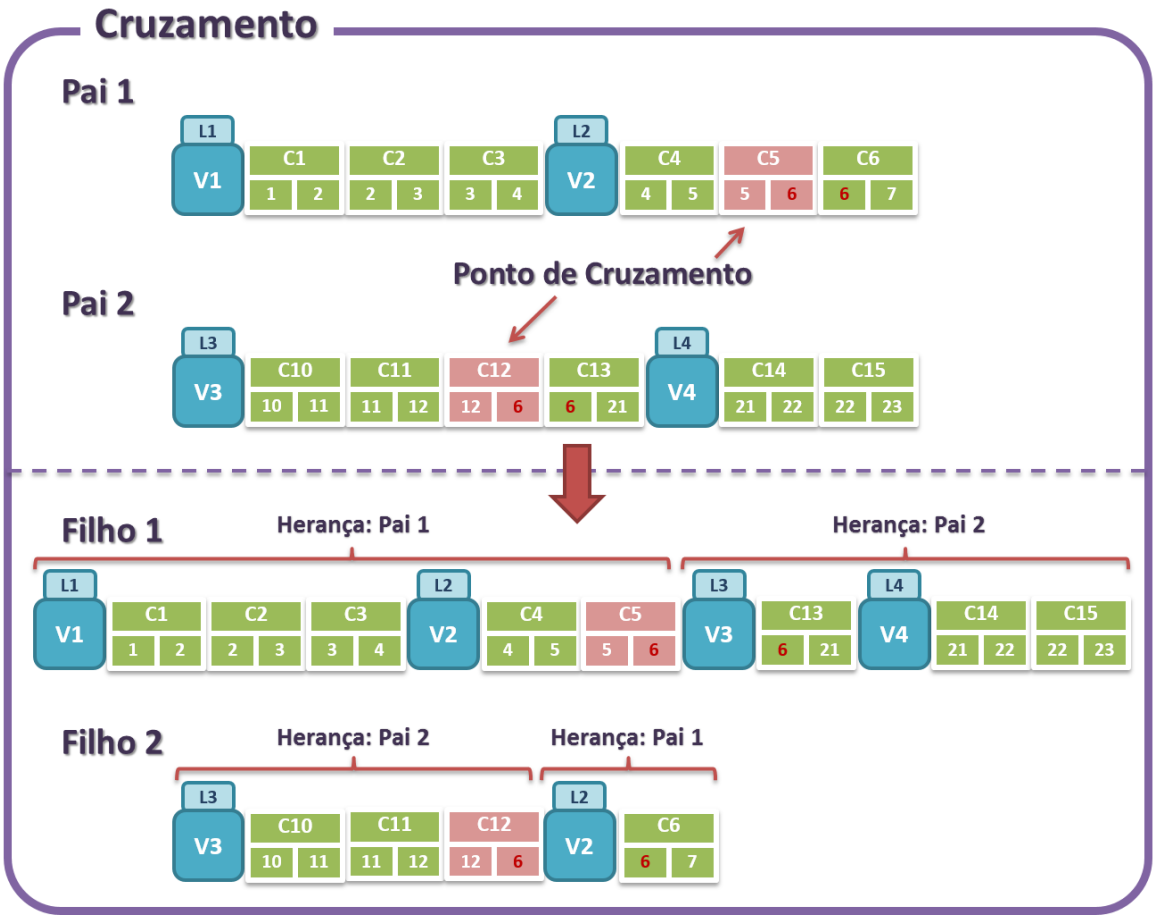

Figura 15 - 3.7 - Segunda Proposta de Cruzamento Genético

Em ambas as propostas, a informação sobre o número máximo de trajetos, passada durante a requisição ao Pilgrim, atua como limitadora no momento de geração de novos indivíduos: caso um cromossomo filho recém-gerado possua uma quantidade de genes superior ao limite de trajetos especificado, ele é descartado antes de ser inserido na população.

Foram realizadas algumas simulações com esta proposta de codificação de cromossomos e cruzamento genético, e identificou-se que o sistema foi capaz de gerar e classificar um conjunto de rotas válidas corretamente. Assim, esta segunda proposta de codificação foi escolhida para ser utilizada na implementação do algoritmo genético.

O processo de seleção por torneio, aplicado na criação de novas gerações de indivíduos, consiste em selecionar aleatoriamente dois indivíduos distintos da população, e eliminar o indivíduo que tiver maior aptidão, calculada por uma função-objetivo. Desta forma, os indivíduos mais aptos tendem a permanecer na população durante as gerações, enquanto que os menos aptos tendem a serem eliminados. Esse mecanismo faz com que a população de indivíduos evolua de geração em geração.

A função-objetivo utilizada leva em consideração elementos contextuais estáticos e dinâmicos relevantes à situação do veículo e do trânsito no momento da viagem para atribuir um valor para cada indivíduo (rota), abrangendo 3 dimensões contextuais segundo a classificação apresentada em [Bulcao Neto 2006; Abowd and Mynatt 2000; Truong et al. 2001]:

- *Where*: pontos de origem e destino, localização das paradas, localização dos veículos; Distância total da rota; Situação do Trânsito na região compreendida pela rota (e.g. engarrafado, livre);
- *When*: horário da solicitação (o banco de dados possui informações sobre os horários em que o trânsito fica ruim em determinados trechos); Tempo total da viagem; Tempo esperando pelo ônibus na parada; Condições climáticas no momento da solicitação;
- *How*: Número de trocas de ônibus; Suporte à acessibilidade pelo veículo;

Os elementos citados, provenientes do banco de dados ou de outros sistemas do UbiBus, são combinados de forma a serem utilizados pelo Pilgrim: os pontos de origem e destino, aliados à localização das paradas, permitem que o sistema calcule a distância total a ser percorrida a pé; o tempo total esperando pelos ônibus nas paradas é obtido a partir de um serviço separado do UbiBus, que utiliza a localização dos veículos, a situação do trânsito e o horário da solicitação em seu processamento; as condições climáticas, junto com o

horário da solicitação e com a situação do trânsito são usadas para determinar um valor chamado Modificador de Situação do Trânsito (MTi), que representa uma variação no tempo estimado para se percorrer cada trecho. Por fim, a informação sobre o suporte à acessibilidade pelo veículo atua como elemento excludente durante o processamento do sistema, de forma que caso uma rota possua veículos sem suporte à acessibilidade, esta será descartada, caso no momento da solicitação a acessibilidade seja informada como necessária.

Assim, a função-objetivo utilizada no sistema Pilgrim pode ser definida da seguinte forma. Sejam Dp a distância total percorrida a pé, Te o tempo total aguardando pelos ônibus nas paradas, Nt o número de trocas de ônibus na rota, T_i o tempo estimado de deslocamento do veículo no trecho i e MT_i o modificador de situação do trânsito no trecho i, então Tt é o tempo total estimado percorrido pelo veículo, dado por $Tt = \sum_{i=1}^{n} T_i \times MT_i$. Dessa forma, a função-objetivo utilizada para avaliar um estado I é dada por: $f(I) = (Dp * Cp) + (Te * Ce) + (Nt * Ct) + Tt$, onde Cp é um fator de transformação de distância em unidades temporais, Ce é uma constante de balanceamento e Ct é um fator para transformar o número de trocas de ônibus em unidades temporais.

A seguir, serão apresentados os detalhes da implementação, específicos da técnica de algoritmos genéticos proposta, para o sistema Pilgrim.

3.4.1. DETALHES DA IMPLEMENTAÇÃO UTILIZANDO ALGORITMOS GENÉTICOS

Nesta seção serão apresentados os detalhes da implementação do sistema Pilgrim que são específicos para a proposta de implementação utilizando a técnica de Algoritmos Genéticos, de acordo com a segunda proposta de codificação de cromossomos, que foi escolhida conforme apresentado na seção anterior.

Geração da População Inicial

A população inicial de indivíduos, que será evoluída pelo algoritmo genético (AG) durante suas iterações, consiste de um conjunto de rotas completas e válidas, i.e. rotas que se iniciam no ponto de origem informado na requisição e terminam no ponto de destino. A geração inicial dessa população é dada em três passos, sendo eles: geração de rotas diretas (sem trocas de ônibus), geração de rotas com até uma troca de ônibus e geração de rotas com até duas trocas. Salienta-se que, caso os pontos de origem e/ou destino fornecidos na

requisição ao sistema Pilgrim sejam coordenadas geográficas, serão geradas rotas partindo de cada uma das 5 paradas de ônibus mais próximas a eles.

O primeiro passo, responsável por gerar indivíduos com um único trajeto, se dá através de uma busca por linhas de ônibus que passem tanto pela parada de origem como pela de destino. Para cada linha encontrada que satisfaça essa condição, será gerado um novo indivíduo (rota) com um único trajeto, que se inicia na parada de origem e percorre os trechos da linha até a parada de destino. No momento de sua criação, o sistema associará um veículo para o trajeto, de acordo com o horário da viagem especificado na requisição e com a posição geográfica dos veículos.

Em seguida, no segundo passo, o algoritmo irá buscar por rotas que possuam até dois trajetos, i.e. rotas com uma única troca de ônibus. Para isso, primeiramente são identificadas todas as linhas de ônibus que passem pela parada de destino, e todas as linhas que passem pela parada de origem. Dentre as linhas que passem pelo destino encontradas, selecionam-se aquelas que possuam ligações com linhas que passem pela origem, i.e. linhas que possuam ao menos uma parada em comum. Para cada par de linhas encontradas que satisfaçam a essas condições, o algoritmo irá criar uma nova rota com dois trajetos: um que se inicia na parada de origem e percorre os trechos da linha que a contém (Lo) até o ponto onde Lo se liga com a linha que contém a parada de destino (Ld), e outro que se inicia na parada onde a ligação acontece, e percorre os trechos de Ld até a parada de destino. O sistema então associará um veículo para cada um dos trajetos criados no momento de sua geração.

Por fim, no terceiro passo, o algoritmo irá buscar por rotas que possuam até três trajetos, i.e. rotas com duas trocas de ônibus. Para tanto, primeiro são identificadas todas as linhas de ônibus que passem pela parada de destino, e todas as linhas que passem pela parada de origem. Em seguida, o algoritmo busca por linhas de ônibus que possuam ligações com alguma das linhas que passam pela origem (Lo) e também com alguma das linhas que passam pelo destino (Ld), que denominaremos linhas intermediárias (Li). Ao encontrar linhas que satisfaçam essas condições, para cada linha encontrada será criada uma rota com três trajetos: um que se inicia na parada de origem e percorre os trechos de Lo até a parada em que Lo se liga com Li; um trajeto que se inicia na parada em que a ligação anterior aconteceu, e percorre os trechos de Li até a parada em que Li se liga com Ld; e um trajeto que se inicia na parada em que ocorreu a ligação entre Li e Ld, e percorre os trechos de Ld até a parada de destino.

O conjunto de todas as rotas geradas nos três passos acima compõe, assim, a população inicial do AG.

Seleção

O mecanismo de seleção utilizado pelo algoritmo genético, responsável por selecionar os indivíduos da população que irão persistir de uma geração para outra, é um processo de seleção por torneio, e seu funcionamento consiste em escolher, aleatoriamente, dois indivíduos distintos da população, e compará-los de acordo com seus valores de aptidão atribuídos pela função-objetivo do AG. O indivíduo com maior aptidão será preservado para a próxima geração, e o com menor aptidão será eliminado.

A seleção de indivíduos ocorre a cada ciclo (geração) do algoritmo genético, e a cada geração o processo de seleção é realizado diversas vezes, de forma a manter o número total de indivíduos na população dentro de um limite máximo, definido durante a inicialização do sistema Pilgrim.

Cruzamento Genético

Conforme apresentado na seção anterior, o cruzamento genético entre dois cromossomos é representado como sendo a troca de linhas de ônibus entre os dois, de forma que dados dois indivíduos (pais) $P1$ e $P2$, que tenham ao menos uma parada de ônibus em comum (ponto de cruzamento - PC), serão gerados dois indivíduos filhos $F1$ e $F2$, onde $F1$ será composto da parte de $P1$ anterior ao ponto de cruzamento e da parte de $P2$ posterior ao PC, e $F2$ será composto da parte de $P2$ anterior a PC e da parte de $P1$ posterior ao ponto de cruzamento.

O processo de inicia escolhendo-se aleatoriamente um indivíduo qualquer da população ($P1$), e então identificando todos os indivíduos da população que são possíveis de serem cruzados com $P1$, i.e. rotas que possuam ao menos uma parada de ônibus em comum com $P1$. Dentre os indivíduos encontrados, seleciona-se aleatoriamente um segundo indivíduo $P2$.

Uma vez que $P1$ e $P2$ foram selecionados, determina-se o ponto de cruzamento entre eles como sendo uma das paradas que eles possuem em comum, escolhida de forma aleatória. Em seguida, cria-se uma nova rota ($F1$), composta por: todos os trajetos de $P1$ anteriores ao trajeto que contém o ponto de cruzamento; um trajeto, com o mesmo veículo do trajeto de $P1$ que contém o PC, contendo os trechos desse trajeto de $P1$ anteriores ao trecho que possui como parada final a parada onde o cruzamento acontece, inclusive o

próprio trecho; um trajeto, com o mesmo veículo do trajeto de $P2$ que contém o PC, contendo os trechos desse trajeto de $P2$ posteriores ao trecho que possui como parada inicial a parada onde o cruzamento acontece, inclusive o próprio trecho; e todos os trajetos de $P2$ posteriores ao trajeto que contém o ponto de cruzamento.

Por fim, é criada outra rota ($F2$), de maneira idêntica ao processo de criação de $F1$, trocando-se nesse processo $P1$ por $P2$, e $P2$ por $P1$.

3.5 IMPLEMENTAÇÃO UTILIZANDO HILL-CLIMBING

Esta seção apresenta uma proposta de implementação do sistema Pilgrim utilizando a técnica Hill-Climbing com Reinício Aleatório, discutida na seção 2.4 , descrevendo a representação do domínio utilizada e o processo necessário para a geração e classificação das rotas.

No problema de geração de rotas, o objetivo do algoritmo será minimizar o custo total das rotas, onde o custo de uma determinada rota é calculado através de uma função-objetivo que leva em consideração diferentes aspectos como distância total, número de trocas de ônibus, entre outros. Dessa forma, pode-se considerar um estado superior como sendo o que possuir menor valor obtido da função-objetivo.

A representação do domínio de rotas de ônibus é feita de acordo com a modelagem apresentada na seção 3.2.3, onde cada rota completa é considerada um estado, e o conjunto de todas as rotas válidas possíveis considerando as informações recebidas na requisição ao sistema Pilgrim é considerado o espaço de estados.

O processo de geração e classificação de rotas pode ser dividido em três etapas, sendo elas: (i) geração do um estado inicial; (ii) geração de rotas alternativas; e (iii) avaliação e classificação das rotas. Após a primeira etapa, o algoritmo irá continuamente executar as duas etapas seguintes, buscando pela melhor rota para o conjunto origem-destino.

Após a inicialização do sistema Pilgrim, novas requisições são processadas conforme descrito na seção 3.3 , e a primeira etapa do processo de geração de rotas se inicia: a partir das informações obtidas da requisição ao sistema, o algoritmo irá gerar um estado inicial, que será iterado pelo algoritmo de Hill-Climbing na busca pelas melhores rotas.

O estado inicial geralmente não é a melhor rota possível, e é gerado realizando-se uma busca com até três passos, de forma que caso um passo encontre uma rota válida, os

passos subsequentes não serão executados. O primeiro passo consiste em buscar no espaço de estados por rotas que conectem os pontos de origem e destino sem a necessidade de trocas de ônibus, i.e. serão procuradas linhas de ônibus que passem tanto pelos pontos de origem quanto de destino, gerando uma rota direta (Figura 16).

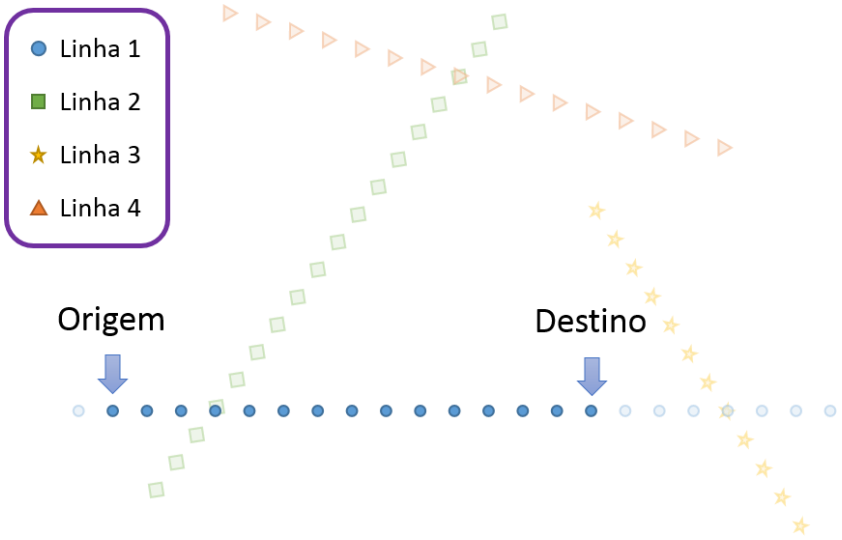

Figura 16 - 3.8 – Exemplo de Rota com um Único Trajeto

Caso o algoritmo não seja capaz de encontrar uma rota direta entre os pontos, no segundo passo serão buscadas rotas que possuam até uma troca de ônibus adicional, i.e. rotas com dois trajetos (Figura 17). Por fim, caso não tenha sido encontrada uma rota válida até então, no terceiro passo o algoritmo irá buscar por rotas com até duas trocas de ônibus (Figura 18). Uma vez que um estado inicial tenha sido encontrado, o algoritmo partirá para a segunda etapa do processo de geração de rotas.

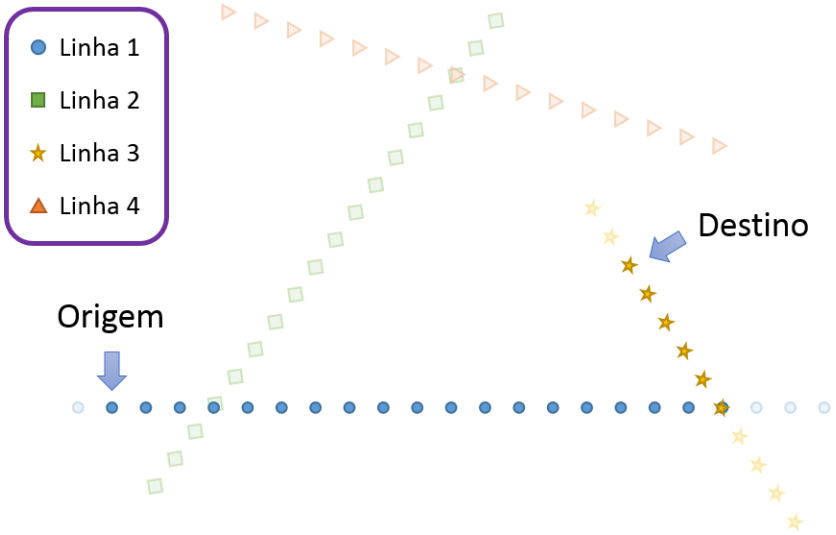

Figura 17 - 3.9 – Exemplo de Rota com Dois Trajetos

A segunda etapa consiste na geração de, a partir de uma rota inicial, um conjunto de rotas alternativas (estados vizinhos) a ela. Nesta etapa, o algoritmo irá criar novas rotas através da modificação da rota inicial, buscando por rotas com até dois trajetos adicionais, i.e. duas trocas de ônibus adicionais. O processo de geração de estados vizinhos será detalhado na seção 3.5.1, adiante.

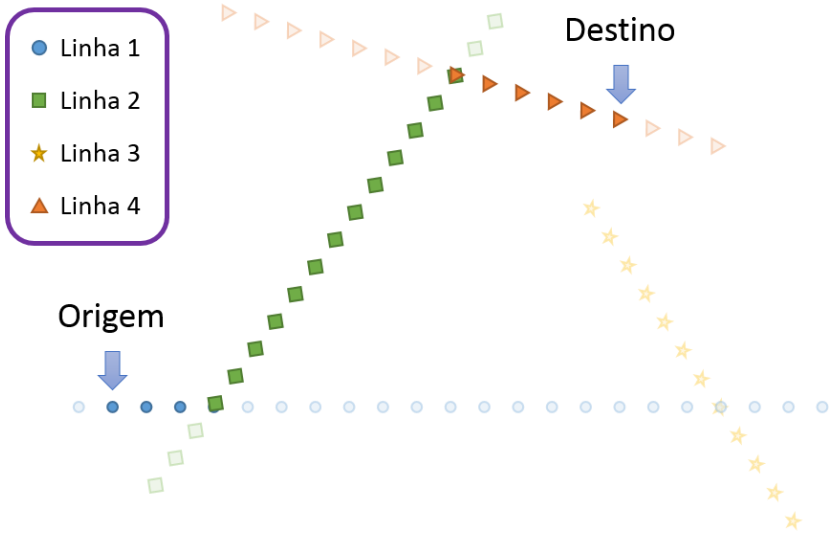

Figura 18 - 3.10 – Exemplo de Rota com Três Trajetos

Todo o conjunto de rotas vizinhas é então agrupado em uma única lista, e se inicia então a terceira etapa do algoritmo, avaliação e classificação. Nesta etapa, as rotas vizinhas geradas são avaliadas e classificadas de acordo com um valor gerado pela função-objetivo, que leva em consideração elementos contextuais estáticos e dinâmicos relevantes à situação do veículo e do trânsito no momento da viagem para atribuir um valor para cada rota.

O número máximo de trajetos em uma rota, fornecido durante a requisição ao sistema Pilgrim, atua como restrição no momento de criação de novos estados: se um novo estado possuir número de trajetos superior ao número máximo, ele é descartado.

A função-objetivo usada pelo algoritmo de hill-climbing é idêntica à utilizada na implementação utilizando algoritmos genéticos, apresentada anteriormente na seção 3.4 .

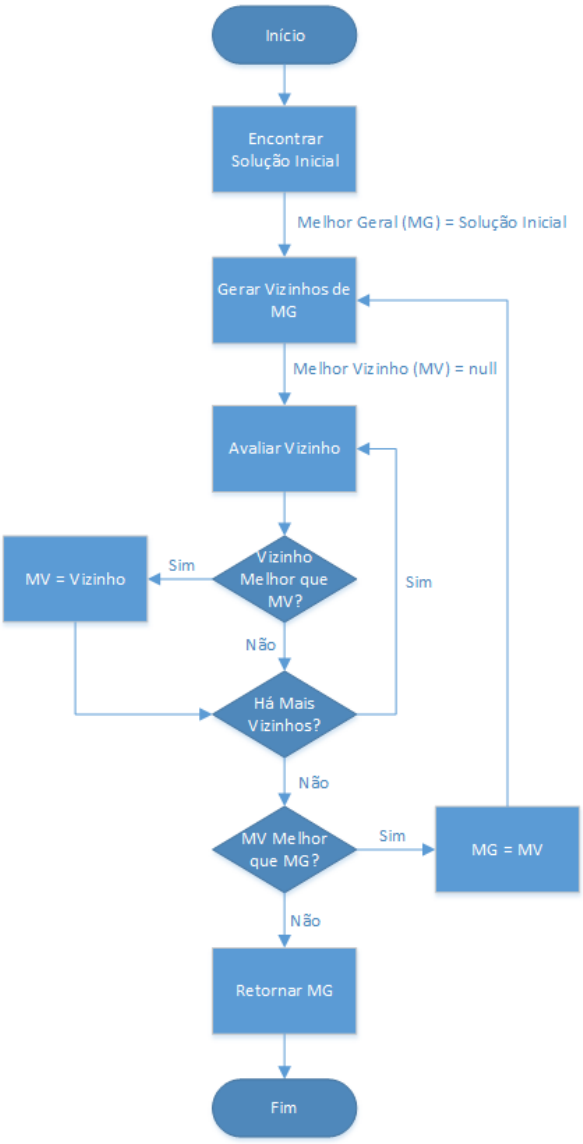

Figura 19 - 3.11 - Fluxograma do Algoritmo de *Hill-Climbing*

Seguindo o fluxo de execução do algoritmo de Hill-Climbing, representado na Figura 19, após a função-objetivo atribuir um valor a cada rota vizinha gerada, ela é comparado à melhor rota encontrada até o momento (se não houver uma melhor rota, a vizinha se torna a melhor) e, se seu valor atribuído for inferior, ela se torna a melhor. Quando uma rota é promovida à situação de melhor estado em um dado momento, ela é adicionada a uma lista de melhores rotas. O ciclo então se repete, utilizando a melhor rota encontrada como rota inicial para a segunda etapa do algoritmo, até que não sejam encontradas novas rotas com uma avaliação superior (i.e. menor valor atribuído pela função-objetivo) à da última rota eleita como melhor estado.

Uma vez que uma execução do algoritmo finalize, as rotas encontradas são armazenadas, e ele é reinicializado e retorna para a primeira etapa, onde será gerada uma

rota inicial diferente das geradas anteriormente. Esse processo se repete por um número de vezes, e ao final as rotas encontradas pelas execuções são comparadas e classificadas de acordo com a função-objetivo.

3.5.1. DETALHES DA IMPLEMENTAÇÃO UTILIZANDO HILL-CLIMBING

Nesta seção serão apresentados os detalhes da implementação do sistema Pilgrim que são específicos para a proposta utilizando a técnica de Hill-Climbing com Reinício Aleatório.

Geração do Estado Inicial

O estado inicial (rota inicial) que será evoluído pelo algoritmo deve ser calculado de forma rápida, e o processo para sua geração, conforme mencionado na seção anterior, consiste de três passos, idênticos aos do processo de geração da população inicial para a proposta de implementação do sistema Pilgrim utilizando algoritmo genético, apresentado na seção 3.4.1, diferenciando-se no fato que para a implementação utilizando hill-climbing, o primeiro estado a ser encontrado durante a execução dos três passos será utilizado como estado inicial para o algoritmo.

Geração de Estados Vizinhos

Um estado vizinho a uma rota é uma rota alternativa, criada a partir da rota original, podendo ter novas trocas de ônibus em seu percurso. O processo de geração de rotas vizinhas consiste em, para cada trajeto de uma rota, gerar rotas alternativas que tenham o mesmo destino da rota original, porém seguindo por linhas diferentes, i.e. novos trajetos são criados.

O sistema gera todas as rotas alternativas possíveis para uma dada rota inicial, limitadas a duas trocas de ônibus adicionais. Esse processo pode ser dividido em duas etapas, semelhantes às etapas da geração de estados iniciais: geração de rotas com uma troca de ônibus adicional e geração com duas trocas adicionais.

O processo de geração de estados vizinhos de uma rota, com uma única troca de ônibus adicional, é dado pelo algoritmo abaixo:

1. Para cada trajeto no estado atual (EA), encontrar, dentre as linhas com as quais a linha do trajeto tem ligações, ligações cujos percursos passem pela parada de destino.

2. Para cada ligação encontrada no passo anterior, criar uma nova rota, composta por:
    a. Todos os trajetos da EA anteriores ao trajeto atual, caso existam;
    b. Um novo trajeto, criado a partir do trajeto atual, cujo último trecho termina no ponto onde a ligação com a nova linha acontece.
    c. Outro novo trajeto, com um novo veículo associado, na linha onde a ligação acontece, cujo primeiro trecho inicia no ponto onde a ligação aconteceu e último trecho termina na parada de destino.

Na Figura 20, abaixo, que representa a criação de um novo estado vizinho de uma rota, os elementos “V1” e “V2” representam veículos, “L1” e “L5” representam linhas de ônibus, “T1” a “T23” representam trechos, e os elementos numerados logo abaixo dos trechos representam paradas de ônibus.

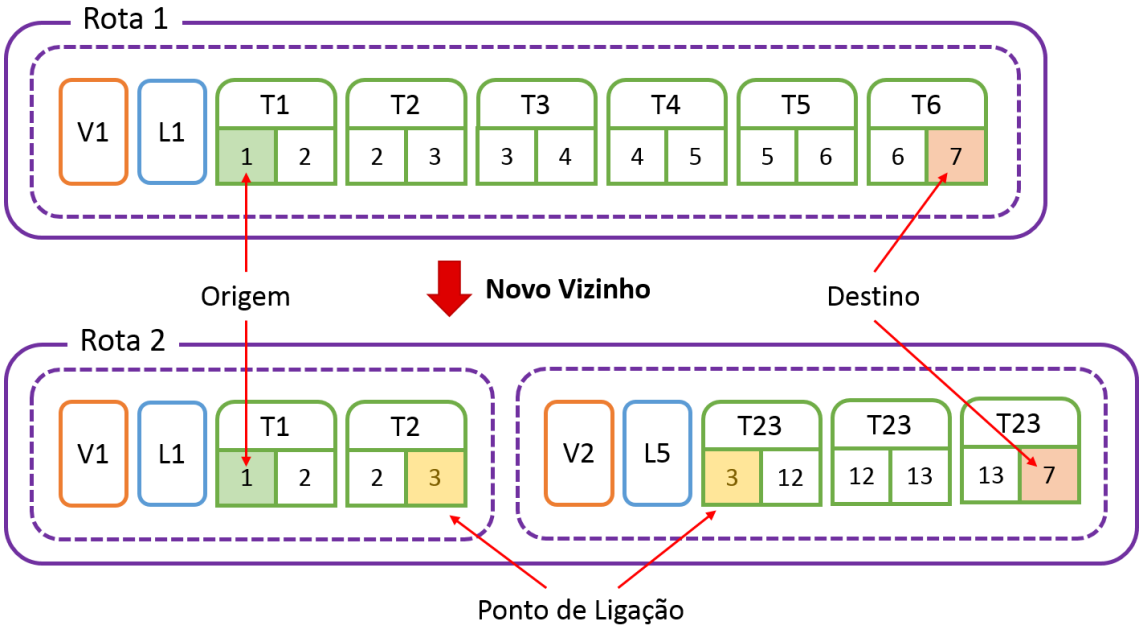

Figura 20 - 3.12 - Criação de um Novo Estado Vizinho de uma Rota

O processo de geração de um estado vizinho com duas trocas de ônibus adicionais é dado pelo algoritmo a seguir:

1. Para cada trajeto no estado atual (EA), determinar as linhas intermediárias entre a linha do trajeto atual e as linhas cujos percursos passem pela parada de destino.

2. Para cada ligação encontrada no passo anterior, criar uma nova rota, composta por:
    a. Todos os trajetos da EA anteriores ao trajeto atual, caso existam;
    b. Um novo trajeto, criado a partir do trajeto atual, cujo último trecho termina no ponto onde a ligação com a linha intermediária acontece.
    c. Outro novo trajeto, com um novo veículo e na linha intermediária, cujo primeiro trecho inicia no ponto onde a ligação acontece, e o último trecho termina no ponto onde a linha intermediária se liga à linha final.
    d. Um último trajeto, com outro novo veículo e na linha final, cujo primeiro trecho inicia no ponto onde a ligação com a linha intermediária aconteceu, e o último trecho termina na parada de destino.

Este capítulo apresentou o sistema Pilgrim, detalhando o contexto em que o sistema se encaixa e diversos aspectos relacionados ao seu desenvolvimento, incluindo duas propostas de implementação da funcionalidade de geração e classificação de rotas utilizando as técnicas de algoritmos genéticos e hill-climbing com reinício aleatório. No próximo capítulo, serão discutidos os experimentos realizados com ambas as propostas, os resultados encontrados, e será apresentada uma pesquisa feita com potenciais usuários do UbiBus.

4 EXPERIMENTOS E RESULTADOS

Neste capítulo, serão discutidos os experimentos realizados com as duas propostas de implementação do sistema Pilgrim, detalhadas no capítulo anterior, e os resultados encontrados. Também será apresentada uma pesquisa feita com potenciais usuários do UbiBus, a fim de validar um conjunto de rotas geradas pelo sistema Pilgrim.

Desta forma, este capítulo está estruturado da seguinte forma: na seção 4.1 , serão apresentados os experimentos realizados com as duas propostas de implementação do sistema; na seção 4.2 , serão expostos os resultados dos experimentos realizados; na seção 4.3 , serão discutidos os resultados expostos; e na seção 4.4 , será apresentada uma pesquisa feita com potenciais usuários do UbiBus.

4.1 EXPERIMENTOS

No capítulo 3 foram apresentadas suas propostas de implementação do sistema Pilgrim, uma utilizando a técnica de algoritmos genéticos e outra utilizando a técnica de hill-climbing com reinício aleatório. Nesta seção, serão apresentados os experimentos realizados com as duas propostas do sistema, com objetivo de identificar a proposta mais adequada.

Os experimentos consistiram em requisitar ao sistema, para cada proposta, a geração de rotas partindo de três pontos de origem distintos, com três destinos diferentes para cada ponto de origem, utilizando os pontos apresentados no Quadro IV, que são alguns pontos de referência da cidade de João Pessoa - PB. Para cada par origem-destino, foram feitas duas requisições, simulando horários distintos para a viagem: a primeira seria realizada no horário de 14:30, visto que esse é um horário com pouco trânsito, e a segunda seria no horário de 18:30, quando o trânsito na cidade de João Pessoa fica mais intenso. Dessa forma, ao total foram requisitadas, a cada uma das propostas do sistema, a geração de rotas para um total de dezoito cenários diferentes.

Em cada execução do sistema, serão coletados dados sobre: o tempo de execução total, o tempo gasto calculando as rotas a pé através da *Directions* API do Google Maps, e o pico de memória utilizada.

Na proposta do sistema Pilgrim baseada na técnica de algoritmos genéticos, a população máxima de indivíduos foi limitada a 100. Na proposta do sistema baseada na

técnica de hill-climbing com reinício aleatório, a quantidade de reinícios do algoritmos de hill-climbing foi definida em três reinícios por requisição.

Quadro IV - 4.1 – Pontos de Referência Utilizados como Origem e Destino

Legenda	Ponto de Referência
O1	Terminal Rodoviário de Integração
O2	Hiper Bompreço
O3	UFPB
D1	Hospital de Olhos da Paraíba
D2	Hospital São Luiz
D3	IFPB - Campus Jaguaribe

Tendo em vista que não faz parte do escopo deste trabalho a criação de uma interface para os usuários, foi desenvolvido um sistema de testes, capaz de interagir com o sistema Pilgrim, de forma a solicitar ao Pilgrim a geração das rotas descritas acima, calculando o tempo de execução levado para a geração de cada conjunto de rotas, e armazenando as rotas geradas (recebidas em formato JSON) em arquivo.

Os experimentos foram executados utilizando um banco de dados local contendo as mesmas informações do banco do UbiBus, sendo que a única comunicação com sistemas externos foi feita para consumir o serviço da *Directions* API do Google Maps, que é utilizado para calcular as rotas a pé. Todas as rotas geradas foram armazenadas em arquivos com nomenclatura sequencial, e a saída do aplicativo de testes foi registrada em um arquivo de log.

Todos os experimentos realizados foram executados em um computador com processador Intel® Core™ i7-4770 com clock de 3.40GHz, 32GB de memória RAM e sistema operacional Windows 8 Professional 64-bits.

4.2 ANÁLISE

A partir dos arquivos contendo as rotas e do arquivo de log mencionados na seção anterior, foi possível extrair algumas informações sobre o processo de geração de rotas. A Tabela I, abaixo, apresenta algumas informações sobre o processo de geração e classificação de rotas do sistema Pilgrim utilizando a técnica de Hill-Climbing com Reinício Aleatório (HCRA), incluindo o número de rotas geradas, o tempo total de execução (**T1**), o

tempo gasto calculando as rotas a pé através da *Directions* API do Google Maps (**T2**) e o tempo efetivo de execução (**Te**), calculado segundo a equação: $Te = T1 - T2$. Os pontos de origem e destino apresentados são referentes aos pontos apresentados no Quadro IV.

Tabela I - 4.1 – Tempos de Execução Utilizando HCRA

Origem	Destino	Horário	Rotas	T1 (ms)	T2 (ms)	Te (ms)
O1	D1	14:30	82	2.774,5	2.529,4	245,10
O1	D1	18:30	82	5.952,6	5.736,5	216,10
O1	D2	14:30	76	6.799,4	6.466,2	333,20
O1	D2	18:30	76	4.831,5	4.500,3	331,20
O1	D3	14:30	68	4.230,9	4.029,8	201,10
O1	D3	18:30	68	4.321,0	4.121,9	199,10
O2	D1	14:30	75	4.221,9	4.000,8	221,10
O2	D1	18:30	75	5.925,6	5.690,4	235,20
O2	D2	14:30	72	6.029,6	5.691,4	338,20
O2	D2	18:30	72	4.380,1	4.030,8	349,30
O2	D3	14:30	74	4.237,9	4.000,8	237,10
O2	D3	18:30	74	5.911,5	5.668,4	243,10
O3	D1	14:30	121	6.051,7	5.655,4	396,30
O3	D1	18:30	117	4.375,1	3.992,8	382,30
O3	D2	14:30	92	4.616,3	4.030,8	585,50
O3	D2	18:30	92	7.291,9	6.706,4	585,50
O3	D3	14:30	96	4.362,0	4.020,8	341,20
O3	D3	18:30	96	2.658,4	2.323,2	335,20

O sistema ainda levou 1.262,21 milissegundos para inicializar e carregar todas as informações do banco de dados local. O processo de inicialização foi executado uma única vez, no início do experimento, não impactando no tempo de execução da geração de rotas.

O pico de memória utilizada durante a execução do sistema Pilgrim utilizando a técnica de hill-climbing com reinício aleatório foi de 87.272 kilobytes.

A Tabela II, abaixo, apresenta algumas informações sobre o processo de geração e classificação de rotas do sistema Pilgrim utilizando a técnica de algoritmos genéticos (AG), incluindo o número de rotas geradas, o tempo total de execução (**T1**), o tempo gasto

calculando as rotas a pé através da *Directions* API do Google Maps (**T2**) e o tempo efetivo de execução (**Te**), calculado segundo a equação: $Te = T1 - T2$. Os pontos de origem e destino apresentados são referentes aos pontos apresentados no Quadro IV.

Tabela II - 4.2 – Tempos de Execução na Geração de Rotas Utilizando AG

Origem	Destino	Horário	Rotas	T1 (ms)	T2 (ms)	Te (ms)
O1	D1	14:30	100	3.768,61	3.496,35	272,26
O1	D1	18:30	100	3.536,39	3.311,17	225,22
O1	D2	14:30	100	5.230,02	4.993,79	236,23
O1	D2	18:30	100	5.796,56	5.412,19	384,37
O1	D3	14:30	100	6.233,98	4.995,79	1.238,19
O1	D3	18:30	100	4.575,39	3.704,55	870,84
O2	D1	14:30	100	3.473,33	3.199,07	274,26
O2	D1	18:30	100	5.715,48	5.015,81	699,67
O2	D2	14:30	100	6.891,61	5.084,88	1.806,73
O2	D2	18:30	100	5.151,94	3.621,47	1.530,47
O2	D3	14:30	100	4.309,13	3.162,03	1.147,10
O2	D3	18:30	100	6.163,91	3.420,28	2.743,63
O3	D1	14:30	100	11.779,30	3.691,54	8.087,76
O3	D1	18:30	100	11.405,94	3.581,43	7.824,51
O3	D2	14:30	100	7.224,93	4.003,84	3.221,09
O3	D2	18:30	100	10.931,49	3.176,04	7.755,45
O3	D3	14:30	100	8.287,95	3.350,21	4.937,74
O3	D3	18:30	100	12.419,92	3.184,05	9.235,87

O sistema ainda levou 1.413,05 milissegundos para inicializar e carregar todas as informações do banco de dados local. O processo de inicialização foi executado uma única vez, no início do experimento, não impactando no tempo de execução da geração de rotas.

O pico de memória utilizada durante a execução do sistema Pilgrim utilizando a técnica de algoritmos genéticos foi de 118.021 kilobytes.

Tendo em vista que as rotas geradas pela proposta do sistema Pilgrim baseado na técnica de algoritmos genéticos são as rotas presentes na última geração da população de

indivíduos, o número total de rotas geradas pelo sistema se manteve constante em todas as requisições feitas.

4.3 INTERPRETAÇÃO

Os resultados obtidos e apresentados na Seção 4.2 , tanto para a proposta do sistema Pilgrim utilizando algoritmos genéticos quanto para a proposta utilizando hill-climbing com reinício aleatório, apontam que o tempo total de execução (**T1**) do processo de geração de rotas é tomado em grande parte pelo tempo gasto nas chamadas feitas à API do Google Maps.

Isso acontece devido à limitação do serviço de *Directions* do Google, que impede que mais de 10 solicitações por segundo[46] sejam respondidas. Sendo assim, a cada vez que o número de solicitações atinge o limite, o sistema precisa aguardar por 1 segundo até que novas solicitações sejam enviadas. Durante o processo de geração de rotas, isso pode acontecer mais de uma vez, impactando consideravelmente o desempenho do sistema.

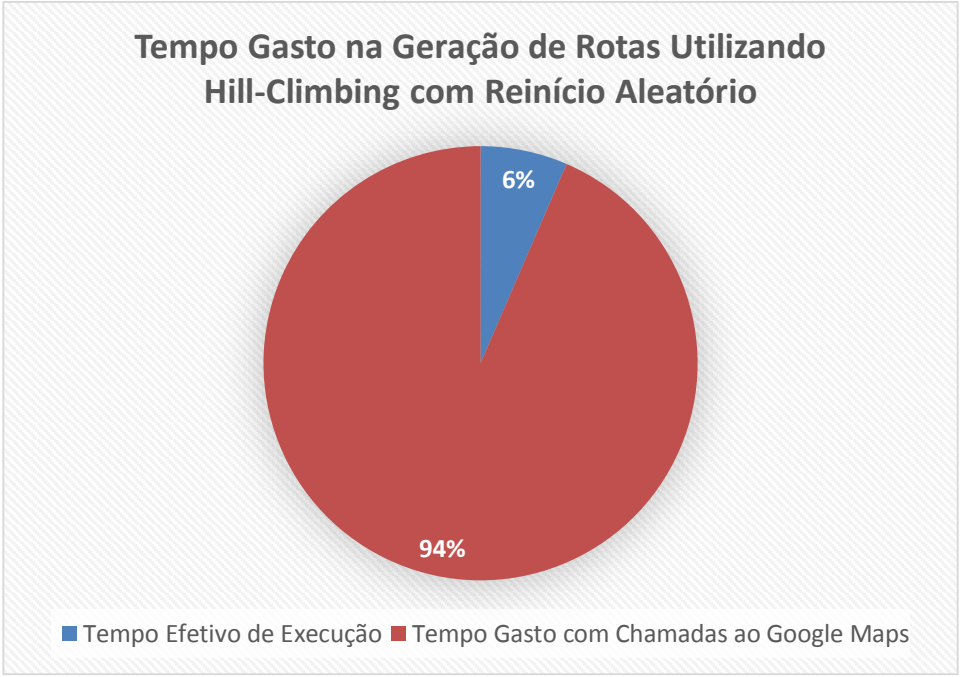

Figura 21 - 4.1 - Tempo Gasto na Geração de Rotas

Para os dados referentes aos experimentos com o sistema Pilgrim utilizando Hill-Climbing com Reinício Aleatório, apresentados na Tabela I, calcula-se que o tempo médio total de execução **T1** é de 4.942,9 milissegundos, que o tempo médio gasto com chamadas

[46] Mais informações em: https://developers.google.com/maps/documentation/business/faq#usage_limits

ao Google Maps **T2** é de 4.622,0 milissegundos e que consequentemente, o tempo médio efetivo de execução **Te** é de 320,88 milissegundos. Isso significa que, para o conjunto de rotas geradas durante o experimento, em média, gasta-se 93,51% do tempo de execução total exclusivamente com chamadas ao serviço de *Directions* do Google Maps (Figura 21).

O tempo total gasto calculando as rotas a pé é justificado pela quantidade de rotas que são requisitadas ao Google Maps: como nas requisições feitas ao sistema os pontos de origem e destino eram coordenadas geográficas, para cada ponto de origem, são calculadas as paradas de ônibus mais próximas a ele e, para cada parada, é requisitada ao serviço do Google uma rota ligando ela ao ponto de origem. O mesmo processo ocorre para os pontos de destino, sendo que as rotas requisitadas à API *Directions* têm como ponto de partida as paradas próximas ao destino, e como ponto de chegada o ponto de destino.

Os dados coletados também apontam que o tempo efetivo de execução é influenciado pelo número de trechos das rotas (Figura 22): as rotas com origem em O1, mais curtas, têm em média um tempo efetivo **Te** de 254,30 milissegundos, enquanto as rotas com origem em O3, mais longas, têm em média um tempo efetivo **Te** de 437,67 milissegundos.

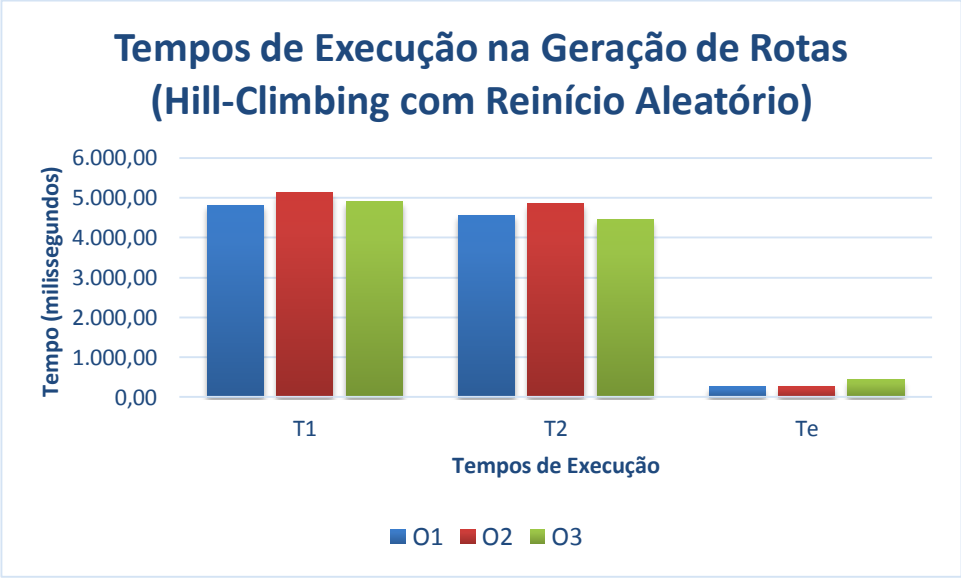

Figura 22 - 4.2 - Tempos de Execução por Origem Utilizando Hill-Climbing

Já para os dados referentes aos experimentos com o sistema utilizando a técnica de algoritmos genéticos, apresentados na Tabela II, calcula-se que o tempo médio total de execução **T1** é de 6.827,55 milissegundos, que o tempo médio gasto com chamadas ao Google Maps **T2** é de 3.911,36 milissegundos e que, consequentemente, o tempo médio efetivo de execução **Te** é de 2.916,19 milissegundos. Nesta situação, em média gasta-se 57,28% do tempo de execução total exclusivamente com chamadas ao serviço *Directions* do Google Maps.

Nota-se, da mesma forma que na proposta do Pilgrim utilizando Hill-Climbing com Reinício Aleatório, que o tempo efetivo de execução é influenciado pelo número de trechos nas rotas (Figura 23): as rotas com origem em O1, mais curtas, têm em média um tempo efetivo **Te** de 537,85 milissegundos, enquanto as rotas com origem em O3, mais longas, têm em média um tempo efetivo **Te** de 6.843,74 milissegundos.

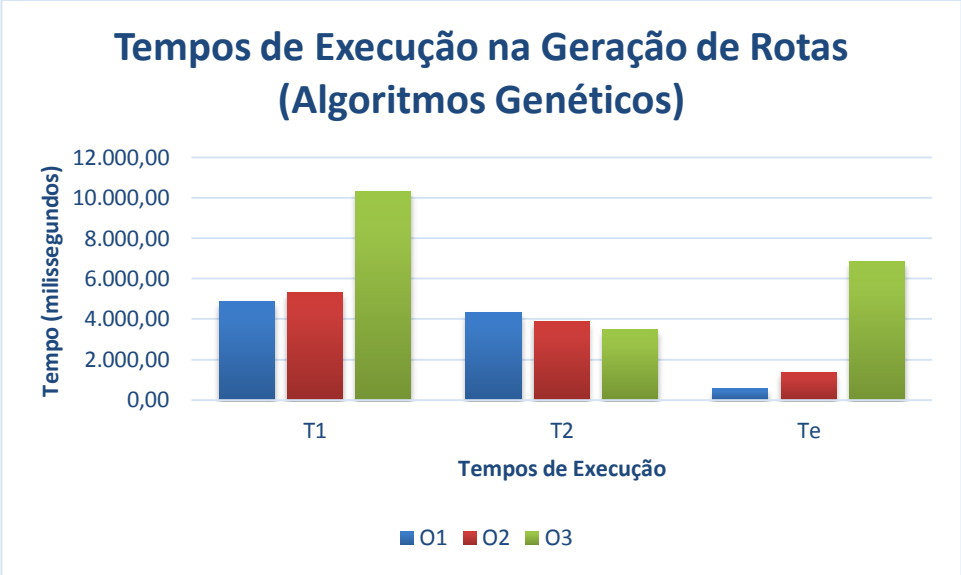

Figura 23 - 4.3 - Tempos de Execução por Origem Utilizando Algoritmos Genéticos

A Tabela III, abaixo, apresenta um comparativo entre as propostas para o sistema Pilgrim utilizando hill-climbing com reinício aleatório (HCRA) e algoritmos genéticos (AG), onde estão representadas as informações: tempo efetivo de execução médio das rotas com origem em O1, **TeO1**; tempo efetivo de execução médio das rotas com origem em O2, **TeO2**; tempo efetivo de execução médio das rotas com origem em O3, **TeO3**; tempo efetivo de execução médio em todas as rotas, **TeM**; e o pico de memória do sistema durante a execução dos experimentos.

Tabela III - 4.3 – Comparativo Entre as Propostas para o Sistema Pilgrim

Técnica	TeO1 (ms)	TeO2 (ms)	TeO3 (ms)	TeM (ms)	Memória (KB)
AG	537,85	1.366,98	6.843,74	2.916,19	118.021
HCRA	254,30	270,67	437,67	320,88	87.272

É interessante apontar que, em ambas as propostas para o sistema Pilgrim, as melhores rotas encontradas dentre todas as rotas geradas são idênticas.

Levando em consideração todos os pontos apresentados nesta seção, é evidente que o desempenho no processo de geração de rotas da proposta para o sistema Pilgrim

baseada na técnica de hill-climbing com reinício aleatório foi superior à proposta baseada em algoritmos genéticos, ressaltando-se entretanto que ambas as técnicas são capazes de gerar um conjunto de rotas para uma dada requisição feita ao sistema. Por essa razão, decidiu-se que a técnica a ser utilizada para a geração e classificação das rotas do sistema Pilgrim é a técnica de hill-climbing com reinício aleatório.

4.4 PESQUISA COM USUÁRIOS

Para avaliar a qualidade das rotas geradas pelo sistema, e a relevância da utilização de um Sistema de Recomendação de Rotas de Ônibus, foi aplicada no período entre 17 de Julho de 2013 e 22 de Julho de 2013 uma pesquisa online com potenciais usuários do UbiBus da cidade de João Pessoa – PB.

A pesquisa foi desenvolvida utilizando a ferramenta Qualtrics[47], e envolveu um total de 48 respondentes. Foram coletados dados demográficos sobre os participantes, incluindo informações sobre gênero, idade e escolaridade.

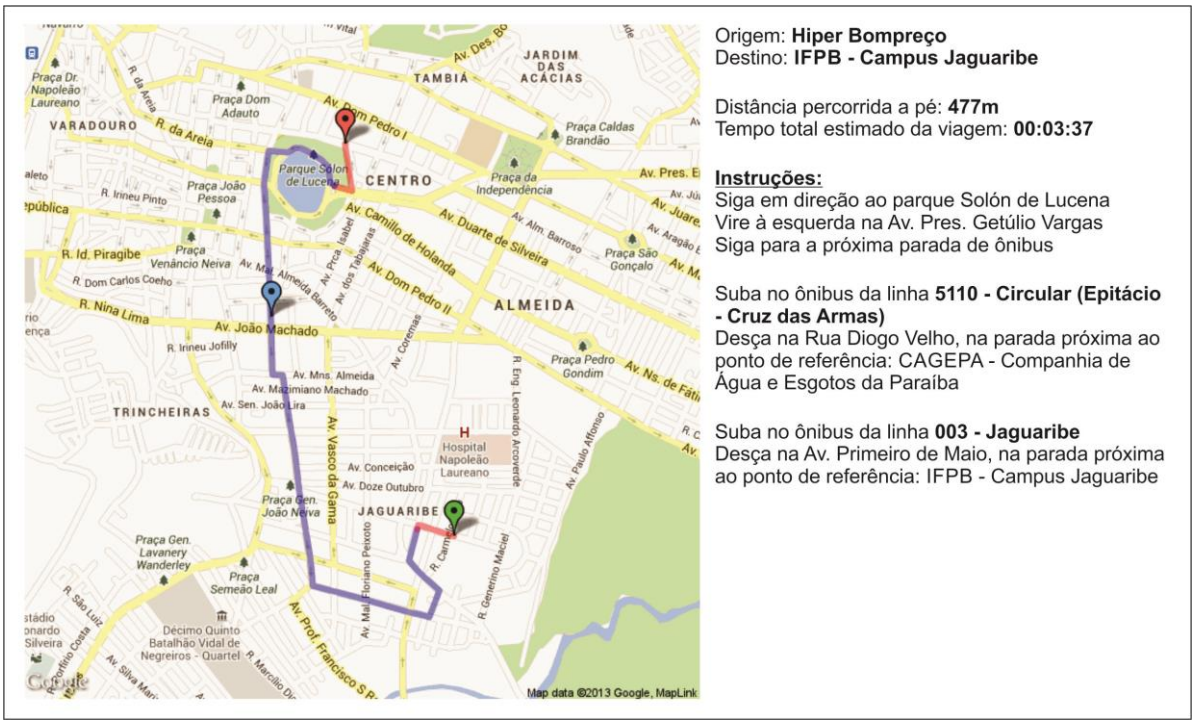

Figura 24 - 4.4 – Exemplo de Rota Gerada com Mapa e Instruções

[47] Disponível em: http://www.qualtrics.com

O questionário desenvolvido consiste em quatro etapas, que devem ser respondidas pelos participantes:

i. **Identificação e Informações Demográficas:** nesta etapa, o participante deve informar seu nome, e-mail, gênero, idade e escolaridade;
ii. **Seleção de Origem e Destino:** nesta etapa, o respondente deve indicar qual ponto de referência deseja utilizar como ponto de origem para sua rota, e qual ponto utilizar como destino;
iii. **Apresentação da Rota:** nesta etapa, o questionário apresenta para o participante, de acordo com as respostas da etapa anterior, uma rota gerada com os pontos de origem e destino escolhidos. São apresentadas uma visualização simplificada da rota e um conjunto de instruções para o usuário (Figura 24).
iv. **Avaliação da Rota:** nesta etapa o participante deve responder a algumas questões sobre sua opinião da importância de um Sistema de Recomendação de rotas e sobre a qualidade da rota apresentada. São elas:
    a. Questão de múltipla-escolha com título “Você acha interessante o uso de um Sistema de Recomendação de Rotas de Ônibus?”. São oferecidas duas opções de resposta: “Sim” e “Não (Explique)”, sendo que nesta última o participante tem a opção de escrever um comentário;
    b. Questão de avaliação em escala, onde o usuário deve avaliar cada critério com uma nota entre 1 (uma estrela) e 5 (cinco estrelas), com título “Por favor, avalie alguns critérios sobre a rota recomendada”. São oferecidos três critérios para serem avaliados: (i) “Caminho Percorrido pelo Ônibus”, (ii) “Caminho Percorrido a Pé” e (iii) “Trocas de Ônibus (caso existam)”.
    c. Questão de múltipla-escolha com título “Para chegar da origem ao destino, você usaria a rota recomendada?”. São oferecidas três opções de resposta: “Sim”, “Não (explique)” e “Talvez (explique)”, sendo que nas duas últimas o respondente tem a opção de escrever comentários;

As opções de pontos de origem (O1 a O3) e destino (D1 a D3) oferecidas aos respondentes estão listadas no Quadro IV. Devido ao número reduzido de linhas de ônibus cadastradas no banco de dados do UbiBus até a data de elaboração deste trabalho, todos os pontos mencionados são referentes a uma região restrita da cidade de João Pessoa – PB.

O questionário da pesquisa foi distribuído digitalmente através de listas de e-mail de alunos da graduação e da pós-graduação da UFPB e do Centro de Informática da UFPE, solicitando aos leitores que, caso possuíssem conhecimento suficiente sobre o sistema de transporte público rodoviário de João Pessoa, respondessem ao questionário. Também foi solicitado aos leitores que replicassem o questionário para conhecidos, visando expandir o número de pessoas que pudessem contribuir participando da pesquisa.

Os resultados da pesquisa, provenientes das respostas coletadas dos participantes, serão apresentados na próxima seção.

4.4.1. RESULTADOS DA PESQUISA

Os dados demográficos coletados através do questionário apresentado aos respondentes da pesquisa foram consolidados, e estão representados nas tabelas abaixo. Dentre os 48 participantes da pesquisa, 36 são do sexo masculino (75%) e 12 são do sexo feminino (25%).

Dentre os 48 respondentes, 24 (50%) declararam possuir idade entre 18 e 25 anos, 22 (45,8%) declararam ter entre 26 e 34 anos, e 2 (4,2%) declararam ter entre 35 e 54 anos.

Os níveis de escolaridade dos respondentes, representados na Figura 25 abaixo, são: dos 48 respondentes, 1 (2,1%) declarou sua escolaridade como sendo ensino médio completo, 19 (39,6%) declararam a escolaridade como sendo ensino superior incompleto, 24 (50%) declararam possuir nível superior completo, e 4 (8,3%) declararam possuir pós-graduação.

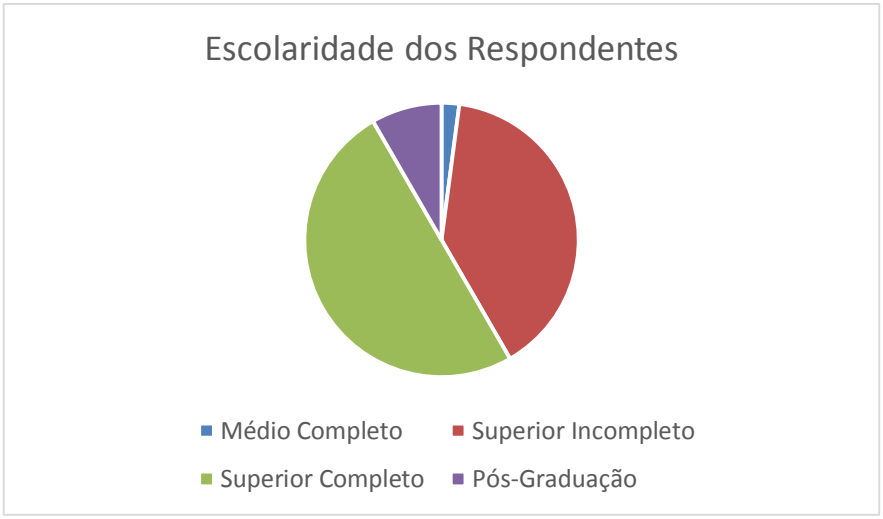

Figura 25 - 4.5 - Escolaridade dos Respondentes

Com relação aos pontos escolhidos como origem, dos 48 respondentes, 26 (54%) escolheram o ponto O1 como origem, 12 (25%) escolheram o ponto O2, e 10 (21%) escolheram o ponto O3 como ponto de origem para a rota.

Quanto ao ponto de destino, dos 48 respondentes, 15 (31%) escolheram o ponto D1 como destino, 13 (27%) escolheram o ponto D2, e 20 (42%) escolheram o ponto D3 como ponto de destino para a rota.

Para a pergunta “Você acha interessante o uso de um Sistema de Recomendação de Rotas de Ônibus?”, todos os 45 respondentes (100%) responderam “Sim”.

A Figura 26 abaixo apresenta os resultados da questão com título “Por favor, avalie alguns critérios sobre a rota recomendada”, indicando a pontuação dada pelos respondentes, entre 1 (menor) e 5 (maior) para três critérios: Qualidade da Rota (C1), Caminho Percorrido a Pé (C2), e Trocas de Ônibus (C3).

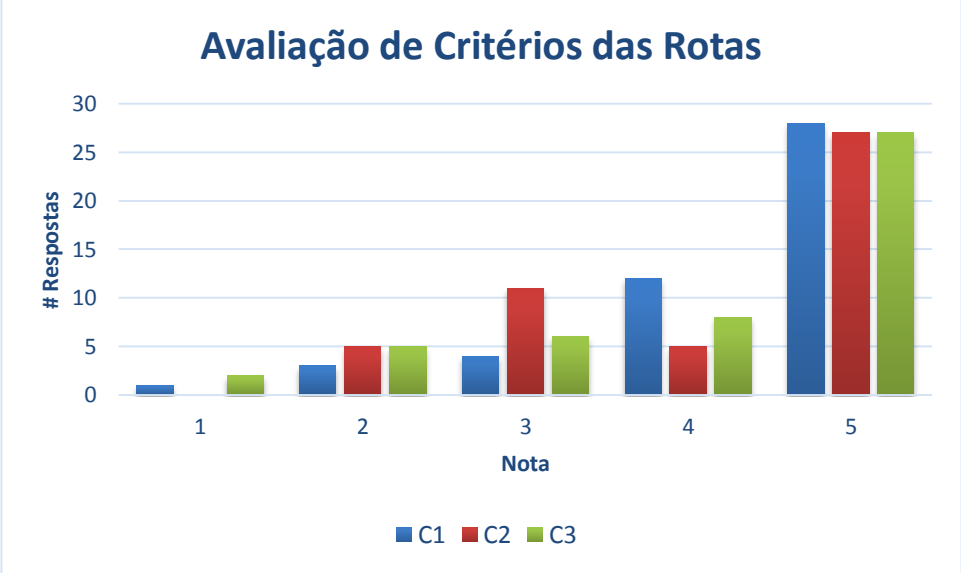

Figura 26 - 4.6 - Avaliação de Diferentes Critérios das Rotas

Dos 48 respondentes, 37 (77%) afirmaram que utilizariam a rota recomendada pelo sistema para a origem e o destino informados, 3 (6%) informaram que não a utilizariam e 8 (17%) informaram que talvez utilizariam.

A partir dos dados obtidos, acredita-se que os participantes envolvidos durante a pesquisa sejam em sua maioria estudantes de graduação ou pós-graduação, possivelmente vinculados às instituições de ensino onde a pesquisa foi divulgada. Sendo assim, os resultados apresentados representam a opinião de um conjunto restrito de potenciais usuários do sistema UbiBus, e servem como indicadores de que as rotas geradas pelo sistema Pilgrim são adequadas aos cenários apresentados na pesquisa feita.

5 CONSIDERAÇÕES FINAIS

Dentro do contexto do Sistema de Transporte Inteligente UbiBus, este trabalho tem como objetivo propor um sistema sensível ao contexto para geração de rotas de ônibus, que esteja disponível como um serviço para as demais aplicações e serviços do UbiBus através de sua camada de middleware. O sistema desenvolvido, denominado Pilgrim, deve ser capaz de, considerando as especificidades do domínio de transporte público rodoviário, gerar e classificar um conjunto de rotas de ônibus para um conjunto de informações de entrada, como ponto de origem e de destino, horário da viagem e número máximo de trocas de ônibus. Para isso, foram propostas duas abordagens para o sistema utilizando técnicas de inteligência artificial, a saber: Algoritmos Genéticos e Hill-Climbing com Reinício Aleatório.

De acordo com os resultados de experimentos apresentados no capítulo anterior, identificou-se que a técnica de Hill-Climbing com Reinício Aleatório obteve melhor desempenho em comparação à técnica de Algoritmos Genéticos, sendo desta forma utilizada para o desenvolvimento do sistema. No mesmo capítulo, também foram discutidos os resultados de uma pesquisa realizada com potenciais usuários do sistema UbiBus, que apresentou indicadores de que as rotas geradas pelo sistema Pilgrim foram consideradas adequadas aos cenários apresentados aos respondentes.

5.1 LIÇÕES APRENDIDAS

Faz-se necessário evidenciar a necessidade de uma preocupação grande com o desempenho de um sistema para geração de rotas de ônibus, tendo em vista que solicitações feitas por usuários devem ser atendidas com um baixo tempo de resposta. Assim, o sistema deve possuir mecanismos para otimizar seu funcionamento e economizar recursos.

Também é interessante apontar o impacto no desempenho geral do sistema ocasionado pelo uso de serviços externos ao sistema desenvolvido, como o consumo de um serviço para calcular rotas a pé, utilizado neste trabalho.

Por fim, evidencia-se a importância de uma quantidade de dados suficientemente grande e devidamente atualizados sobre as entidades relevantes do domínio de transporte público, incluindo informações sobre ônibus, paradas, linhas, situação do trânsito, entre

outros, para que sistemas que explorem soluções para transporte público rodoviário possam ser devidamente testados.

5.2 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Durante o desenvolvimento desta pesquisa, foi possível identificar alguns pontos que podem ser melhor explorados, dentro do contexto de Sistemas de Transporte Inteligente que atuem com transporte público. Cita-se, em primeiro lugar, a exploração de diferentes técnicas de otimização da inteligência artificial para o problema de geração e classificação de rotas de ônibus utilizando informações contextuais, como a técnica *Simulated Annealing* [Russell and Norvig 2009], que se propõe contornar algumas dificuldades encontradas na técnica de Hill-Climbing, como máximos locais, platôs, entre outros e técnicas baseadas em inteligência de enxames, como Otimização de Colônia de Formigas (ACO) [Manfrin 2004] e Enxame de Partículas (PSO) [Kennedy and Eberhart 1995], já utilizadas na literatura em problemas de busca pelo menor caminho.

Em segundo lugar, notou-se que o desempenho do sistema apresentado por este trabalho foi impactado pela necessidade de utilização de um serviço externo para o cálculo das rotas a pé. Assim, aponta-se como potencial objeto de estudo a incorporação de um sistema para geração de rotas a pé, que seja complementar ao sistema para geração de rotas de ônibus.

Por último, indica-se como objeto de exploração a escalabilidade de um sistema de geração e classificação de rotas de ônibus sensível ao contexto, tendo em vista que o volume de dados e de requisições tende a aumentar drasticamente com a inclusão de sistemas de transporte de novas regiões metropolitanas. Essa maior escalabilidade do sistema pode ser explorada através do uso de técnicas como computação paralela [Barney 2013], que possibilitem que um número grande de requisições sejam processadas simultaneamente pelo sistema.

REFERÊNCIAS

ABOWD, G.D. AND MYNATT, E.D., 2000. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1), pp.29–58.

BARNEY, B., 2013. Introduction to Parallel Computing. Available at: https://computing.llnl.gov/tutorials/parallel_comp/ [Accessed August 15, 2013].

BASTOS, R. AND JAQUES, P.A., 2010. ANTARES: Um sistema web de consulta de rotas de ônibus como serviço público. *Revista Brasileira de Computação Aplicada*, 2(1), pp.41–56.

BAZIRE, M. AND BRÉZILLON, P., 2005. Understanding context before using it. In A. K. Dey, B. Kokinov, D. Leake, & R. Turner, eds. *Proceedings of the 5th International and Interdisciplinary Conference CONTEXT 2005*. Paris, France: Springer Berlin Heidelberg, pp. 29–40.

BELFOR, G., CHEN, L., LIU, C., PISANO, P. AND SINGLETON, E., 1997. The ARTS Compendium: FHWA's Electronic Rural ITS Project Tracking System. *Public Roads*, p.23.

BERTOLOTTO, M., O'HARE, G., STRAHAN, R., BROPHY, A., MARTIN, A. AND MCLOUGHLIN, E., 2002. Bus catcher: a context sensitive prototype system for public transportation users. In *Proceedings of the Third International Conference on Web Information Systems Engineering (Workshops), 2002.* IEEE, pp. 64–72.

BLICKLE, T. AND THIELE, L., 1996. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation*, 4(4), pp.361–394.

BLICKLE, T. AND THIELE, L., 1995. *A Comparison of Selection Schemes used in Genetic Algorithms*, Zurich, Switzerland.

BULCAO NETO, R., 2006. *Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicaç oes sensíveis a contexto*. São Carlos, SP, Brazil: Universidade de São Paulo - USP.

CAULFIELD, B. AND O'MAHONY, M., 2007. An Examination of the Public Transport Information Requirements of Users. *IEEE Transactions on Intelligent Transportation Systems*, 8(1), pp.21–30.

CHAVES, A., STEINMACHER, I. AND VIEIRA, V., 2011. Social networks and collective intelligence applied to public transportation systems: A survey. In *VIII Simpósio Brasileiro de Sistemas Colaborativos*. Paraty, RJ, p. 8.

CHIONG, R., SUTANTO, J.H. AND JAP, W.J., 2008. A Comparative Study on Informed and Uninformed Search for Intelligent Travel Planning in Borneo Island. In *2008 International Symposium on Information Technology*. IEEE, pp. 1–5.

CHRIQUI, C. AND ROBILLARD, P., 1975. Common Bus Lines. *Transportation Science*, 9(2), pp.115–121.

LE CLERQ, F.P., 1972. A public transport assignment method. *Traffic Engineering Control*, (14), pp.91–96.

DARWIN, C., 1859. On the origins of species by means of natural selection. *London: Murray*, p.247.

DEETER, D. AND BLAND, C.E., 1997. *Technology in Rural Transportation: "Simple Solutions,"* McLean, Virginia.

DELLING, D., PAJOR, T. AND WERNECK, R.F., 2012. Round-Based Public Transit Routing. In *Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12)*. Kyoto, Japan: Society for Industrial and Applied Mathematics, pp. 130–140.

DEY, A.K., 2001. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1), pp.4–7.

DIJKSTRA, E.W., 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1), pp.269–271.

DISTNER, M., BENGTSSON, M., BROBERG, T. AND JAKOBSSON, L., 2009. City safety—a system addressing rear-end collisions at low speeds. In *Proceedings of the 21st (ESV) International Technical Conference on the Enhanced Safety of Vehicles*. Stuttgart, Germany: National Highway Traffic Safety Administration, pp. 1–7.

EIBEN, A. AND SMITH, J., 2007. *Introduction to Evolutionary Computing* 2nd ed., Springer.

EMORY, D., 2012. OpenTripPlanner 0.9 Released. Available at: http://opentripplanner.com/2012/10/opentripplanner-0-9-released/ [Accessed August 7, 2013].

EPPSTEIN, D., 1998. Finding the k Shortest Paths. *SIAM Journal on Computing*, 28(2), pp.652–673.

ERICSON, N., 1997. New CVO Technologies Hit the Road. *Public Roads*, pp.13–17.

FEDERAL HIGHWAY ADMINISTRATION, 2013. National Traffic and Road Closure Information. Available at: http://www.fhwa.dot.gov/trafficinfo/511.htm [Accessed August 4, 2013].

FERRIS, B., 2011. *OneBusAway: Improving the Usability of Public Transit*. Seattle, USA: University of Washington.

FERRIS, B., WATKINS, K. AND BORNING, A., 2010a. Location-Aware Tools for Improving Public Transit Usability. *IEEE Pervasive Computing*, 9(1), pp.13–19.

FERRIS, B., WATKINS, K. AND BORNING, A., 2009. OneBusAway : A Transit Traveller Information System. In *Proceedings of Mobicase 2009*. Sand Diego, CA, pp. 1–15.

FERRIS, B., WATKINS, K. AND BORNING, A., 2010b. OneBusAway: Results from Providing Real-Time Arrival Information for Public Transit. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA: ACM Press, pp. 1807–1816.

GOLDBERG, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning* 1st ed., Addison-Wesley Professional.

GOOGLE, 2013. Google Transit. Available at: http://www.google.com/intl/pt-BR/landing/transit/ [Accessed July 19, 2013].

GUIZZO, E., 2011. How Google's Self-Driving Car Works. *IEEE Spectrum*.

GUZOLEK, J. AND KOCH, E., 1989. Real-Time Route Planning in Road Networks. In *Conference Record of papers presented at the First Vehicle Navigation and Information Systems Conference (VNIS '89)*. Toronto, Ontario, Canada: IEEE, pp. 165–169.

HARRINGTON, A. AND CAHILL, V., 2004. Route Profiling: Putting Context to Work. In *Proceedings of the 2004 ACM symposium on Applied computing - SAC '04*. New York, New York, USA: ACM Press, p. 1567.

HART, P., NILSSON, N. AND RAPHAEL, B., 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), pp.100–107.

HOAR, R., 2010. A personalized web based public transit information system with user feedback. In *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 1807–1812.

HOUSER, A., PIEROWICZ, J. AND MCCLELLAN, R., 2005. *Concept of Operations and Voluntary Operational Requirements for Forward Collision Warning Systems (CWS) and Adaptive Cruise Control (ACC) Systems On-Board Commercial Motor Vehicles*, Buffalo, NY.

INTERNATIONAL STANDARDS ORGANIZATION, 2004. ISO 8601:2004(E) Data elements and interchange formats - Information interchange - Representation of dates and times. , 2004, p.40.

JACOBSON, S.H., MCLAY, L. A., HALL, S.N., HENDERSON, D. AND VAUGHAN, D.E., 2006. Optimal search strategies using simultaneous generalized hill climbing algorithms. *Mathematical and Computer Modelling*, 43(9-10), pp.1061–1073.

JARIYASUNANT, J., MAI, E. AND SENGUPTA, R., 2011. Algorithm for Finding Optimal Paths in a Public Transit Network with Real-Time Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2256, pp.34–42.

JARIYASUNANT, J., WORK, D.B., KERKEZ, B., SENGUPTA, R., GLASER, S. AND BAYEN, A., 2011. Mobile Transit Trip Planning with Real-Time Data. Available at: http://escholarship.org/uc/item/51t364vz [Accessed August 1, 2013].

DE JONG, K.A., 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. University of Michigan.

KANOH, H. AND HARA, K., 2008. Hybrid Genetic Algorithm for Dynamic Multi-Objective Route Planning with Predicted Traffic in a Real-World Road Network. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO '08*. New York, New York, USA: ACM Press, pp. 657–664.

KENNEDY, J. AND EBERHART, R., 1995. Particle Swarm Optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE, pp. 1942–1948.

KUMAR, A.J.S., ARUNADEVI, J. AND MOHAN, V., 2009. Intelligent Transport Route Planning Using Genetic Algorithms in Path Computation Algorithms. *European Journal of Scientific Research*, 25(3), pp.463–468.

LACERDA, E.G.M. DE AND CARVALHO, A.C.P.L.F. DE, 1999. Introdução aos Algoritmos Genéticos. In *Sistemas Inteligentes – Aplicações a Recursos Hídricos e Ciências Ambientais*. Porto Alegre, RS: EDUFRGS, pp. 87–148.

LINDEN, R., 2012. *Algoritmos Genéticos – Uma Importante Ferramenta da Inteligência Computacional* 3rd ed., Rio de Janeiro, RJ: Brasport.

MANFRIN, M., 2004. *Ant Colony Optimization for the Vehicle Routing Problem*. Université Libre de Bruxelles.

MARKOFF, J., 2010. Google Cars Drive Themselves, in Traffic. *The New York Times*.

MCCORMACK, J.E. AND ROBERTS, S. A., 1996. Exploiting object oriented methods for multi-modal trip planning systems. *Information and Software Technology*, 38(6), pp.409–417.

MILLER, B.L. AND GOLDBERG, D.E., 1995. Genetic Algorithms , Tournament Selection , and the Effects of Noise. *Complex Systems*, 9, pp.193–212.

MOUSKOS, K., NIVER, E., LEE, S., BATZ, T. AND DWYER, P., 1999. Transportation Operations Coordinating Committee System for Managing Incidents and Traffic: Evaluation of the Incident Detection System. *Transportation Research Record*, 1679(1), pp.50–57.

OPENTRIPPLANNER, 2013. OpenTripPlanner Demos. Available at: https://github.com/openplans/OpenTripPlanner/wiki/Demos [Accessed August 7, 2013].

PADMANABAN, R.P.S., DIVAKAR, K., VANAJAKSHI, L. AND SUBRAMANIAN, S.C., 2010. Development of a real-time bus arrival prediction system for Indian traffic conditions. *IET Intelligent Transport Systems*, 4(3), p.189.

PASSOS, L. AND ROSSETTI, R., 2009. Intelligent Transportation Systems: a ubiquitous perspective. In L. S. Lopes, N. Lau, P. Mariano, & L. M. Rocha, eds. *New Trends in Artificial Intelligence: the 14th Portuguese Conference on Artificial Intelligence, EPIA 2009*. Aveiro, Portugal, pp. 27–38.

PELLAZAR, M.B., 1994. Vehicle Route Planning with Constraints using Genetic Algorithms. In *Proceedings of National Aerospace and Electronics Conference (NAECON'94)*. Dayton, OH: IEEE, pp. 111–118.

PENG, Z.-R., 1997. A methodology for design of a GIS-based automatic transit traveler information system. *Computers, Environment and Urban Systems*, 21(5), pp.359–372.

PENG, Z.-R. AND HUANG, R., 2000. Design and development of interactive trip planning for web-based transit information systems. *Transportation Research Part C: Emerging Technologies*, 8(1-6), pp.409–425.

PENNEY, T., 1999. *Rural ITS Needs and Challenges*, McLean, Virginia.

RUSSELL, S. AND NORVIG, P., 2009. *Artificial Intelligence: a Modern Approach* 3rd ed., Upper Saddle River, New Jersey, USA: Prentice Hall.

SCHILIT, B., ADAMS, N. AND WANT, R., 1994. Context-aware computing applications. *Workshop on Mobile Computing Systems and Applications, 1994. WMCSA 1994*, pp.85–90.

SCHULZ, F., WAGNER, D. AND WEIHE, K., 2000. Dijkstra's Algorithm On-Line: An Empirical Case Study from Public Railroad Transport. *Journal of Experimental Algorithmics*, 5, p.12–es.

SILVA, A.P. DA, 2011. *Otimização de Redes Neurais Artificiais MultiLayer Perceptron através de Algoritmos Genéticos Celulares*. Universidade Federal de Pernambuco.

SILVA, D., 2000. *Sistemas inteligentes no transporte público coletivo por ônibus*. Porto Alegre, Brasil: Universidade Federal do Rio Grande do Sul.

SIVARAJ, D., KANDASWAMY, A., RAJASEKAR, V., SANKARGANESH, P.B. AND MANIKANDAN, G., 2011. Implementation of AVCS using Kalman Filter and PID Controller in Autonomous Self Guided Vehicle. *International Journal of Computer Applications*, 27(2), pp.1–8.

SPTRANS, 2013. Bilhete Único Comum. Available at: http://www.sptrans.com.br/bilhete_unico/comum.aspx [Accessed May 2, 2013].

SUSSMAN, J., 1996. ITS: A Short History and a Perspective on the Future. *Transportation Quarterly 75th Anniversary Special Issue*, pp.115–125.

SUSSMAN, J., 2005. *Perspectives on Intelligent Transportation Systems (ITS)*, Boston, MA: Springer US.

SZCZERBA, R.J., GALKOWSKI, P., GLICKTEIN, I.S. AND TERNULLO, N., 2000. Robust Algorithm for Real-Time Route Planning. *IEEE Transactions on Aerospace and Electronic Systems*, 36(3), pp.869–878.

TITO, A.O., BORGIANI, F.S.M., DOS SANTOS, R. A., TEDESCO, P.C. A. R. AND SALGADO, A.C., 2012. Contextual information in user information systems in public transportation: A systematic review. *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp.361–366.

TRUONG, K., ABOWD, G. AND BROTHERTON, J., 2001. Who, What, When, Where, How: Design Issues of Capture & Access Applications. In *UbiComp '01 Proceedings of the 3rd international conference on Ubiquitous Computing*. Springer-Verlag London, pp. 209–224.

UNITED STATES DEPARTMENT OF TRANSPORTATION, 1997. *Advanced Rural Transportation Systems (ARTS) Strategic Plan*,

VIEIRA, V., 2008. *CEManTIKA: A Domain-Independent Framework for Designing Context-Sensitive Systems*. Recife, PE, Brazil: Federal University of Pernambuco.

VIEIRA, V. ET AL., 2012. The UbiBus Project: Using Context and Ubiquitous Computing to build Advanced Public Transportation Systems to Support Bus Passengers. In *Anais do VIII Simpósio Brasileiro de Sistemas de Informação*. São Paulo, SP, p. 7.

VIEIRA, V., CALDAS, L.R. AND SALGADO, A.C., 2011. Towards an Ubiquitous and Context Sensitive Public Transportation System. *2011 Fourth International Conference on Ubi-Media Computing*, pp.174–179.

VIEIRA, V., TEDESCO, P. AND SALGADO, A., 2009. Modelos e Processos para o desenvolvimento de Sistemas Sensíveis ao Contexto. *André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski. (Org.). Jornadas de Atualização em Informática 2009 (JAI'09)*, pp.381–431.

WAZE MOBILE, 2013. Waze. Available at: http://www.waze.com/ [Accessed April 8, 2013].

WEISER, M., 1991. The computer for the 21st century. *Scientific American*, p.8.

WEISER, M., 1994. The World is not a Desktop. *Interactions*, 1(1), pp.7–8.

WONG, S.C. AND TONG, C.O., 1998. Estimation of Time-Dependent Origin–Destination Matrices for Transit Networks. *Transportation Research Part B: Methodological*, 32(1), pp.35–48.

XU, W., HE, S., SONG, R. AND CHAUDHRY, S.S., 2012. Finding the K shortest paths in a schedule-based transit network. *Computers & Operations Research*, 39(8), pp.1812–1826.

ZAHEER, K., 2006. *Artificial Intelligence Search Algorithms In Travel Planning*. Mälardalen University Västerås Sweden.

ZHAO, L., XIONG, Y. AND SUN, H., 2008. The K Shortest Transit Paths Choosing Algorithm in Stochastic Transit Network. In *Proceedings of the Third International Conference, RSKT 2008*. Chengdu, China: Springer Berlin Heidelberg, pp. 747–754.

ZIMMERMANN, A., LORENZ, A., OPPERMANN, R. AND AUGUSTIN, S., 2007. An Operational Definition of Context. , pp.558–571.

ZOGRAFOS, K.G., ANDROUTSOPOULOS, K.N. AND SPITADAKIS, V., 2009. Design and Assessment of an Online Passenger Information System for Integrated Multimodal Trip Planning. *IEEE Transactions on Intelligent Transportation Systems*, 10(2), pp.311–323.