

## HEURÍSTICAS BASEADAS EM SIMULAÇÃO PARA O PROJETO DE REDES DE TRÁFEGO<sup>1</sup>

André Elias Zanella<sup>2</sup>, Marcelo de Souza<sup>3</sup>

<sup>1</sup> Vinculado ao projeto intitulado “Projeto automático de algoritmos”.

<sup>2</sup> Acadêmico(a) do Curso de Engenharia de Software – CEAVI – Bolsista PIVIC.

<sup>3</sup> Orientador, Departamento de Engenharia de Software – CEAVI – marcelo.desouza@udesc.br.

O problema do Projeto de Redes Viárias Urbanas (UNDP – *Urban Network Design Problem*) é um problema bastante comum no planejamento da infraestrutura viária urbana. O problema tem por objetivo aprimorar redes de transporte urbano, a fim de minimizar congestionamentos e enfrentar a crescente demanda por mobilidade. O congestionamento, que já não é exclusividade de grandes centros, é um problema que afeta a mobilidade urbana e a qualidade de vida, gerando diversos problemas, como aumento do tempo de viagem e impactos ambientais. Diante disso, o estudo e desenvolvimento de abordagens eficazes para projetar e adaptar redes viárias de maneira eficiente é fundamental para o desenvolvimento das cidades.

Este trabalho tem por objetivo explorar técnicas e desenvolver algoritmos para auxiliar na otimização da topologia de redes viárias urbanas. A metodologia envolve a integração entre algoritmos de otimização e o simulador SUMO (*Simulation of Urban Mobility*), visando identificar as melhores intervenções em redes viárias. O trabalho explora duas alternativas de intervenção possíveis: (i) ampliar a capacidade de uma via existente, ou seja, a criação de uma nova pista na mesma rua; ou (ii) construir novas vias, que devem ter as mesmas características (velocidade, capacidade etc.) das vias próximas. Para essa última intervenção, a via construída não pode cruzar nenhuma via já existente. Caso isso aconteça, deve ser definido um cruzamento entre as vias associadas (ou seja, a criação de um novo vértice). Além disso, ainda há a restrição de orçamento (*budget*), onde o mesmo não pode ser ultrapassado. O *budget* é definido em termos do comprimento máximo da intervenção, seja na ampliação ou na construção de vias.

Uma vez definida e implementada a integração entre algoritmos de otimização e o simulador SUMO, o trabalho foca no desenvolvimento de algoritmos heurísticos para a construção e refinamento de soluções para o problema (i.e., intervenções na rede viária). Após a leitura do grafo que representa a rede de tráfego, da demanda de viagens e do *budget*, esses algoritmos executam os seguintes passos, iterativamente: (i) geração das modificações, ampliação ou criação de vias, enquanto tiver orçamento disponível; (ii) invocação do simulador SUMO usando a plataforma de integração desenvolvida previamente; (iii) leitura do tempo médio de viagem medido pela simulação. Com base nesse funcionamento geral, foram projetados e implementados três algoritmos heurísticos de otimização. O primeiro foi chamado de *baseline*, visto que executa uma amostragem aleatória simples, enquanto o segundo e terceiro algoritmos são abordagens baseadas em *heurísticas* de otimização.

O algoritmo *baseline* implementa uma abordagem simplificada, com o objetivo de fornecer uma base de comparação para algoritmos mais sofisticados, bem como fornecer soluções iniciais para outras abordagens. Esse algoritmo repetidamente gera modificações aleatórias na rede, com base no *budget* fornecido. Para cada solução (conjunto de modificações), a simulação é usada para avaliar sua qualidade, i.e. o tempo de viagem médio dos veículos. O critério de parada desse algoritmo é dado por um número máximo de iterações, que representa o número de soluções amostradas aleatoriamente. Ao final, a melhor solução produzida é retornada.

O segundo algoritmo, chamado *busca local restrita*, implementa uma busca local com vizinhança restrita. O algoritmo inicia gerando uma solução inicial aleatória. A cada iteração, o algoritmo seleciona um par de vértices aleatoriamente e avalia as possíveis modificações envolvendo esses vértices, selecionando uma modificação (i.e., uma solução vizinha) que melhora a qualidade da solução anterior. Para selecionar a solução vizinha, o algoritmo usa uma entre duas estratégias. Na *primeira melhoria* (PM), o algoritmo seleciona a primeira solução encontrada que melhora o tempo médio de viagem. Na *melhor melhoria* (MM), o algoritmo analisa todas as soluções vizinhas, selecionando aquela cuja melhoria no tempo de viagem seja maior. O critério de terminação desse algoritmo é dado por um número máximo de iterações.

Finalmente, a terceira abordagem é um algoritmo *guloso iterado* que implementa uma estratégia de reconstrução baseada em busca. Iterativamente, o algoritmo destrói parte da solução, removendo modificações da rede até liberar perto de metade do *budget* disponível. Essa solução parcial é reconstruída por avaliar toda possível modificação que satisfaz o *budget* restante (liberado pelo procedimento de destruição parcial). Novamente, são aplicadas as estratégias PM ou MM, retornando à primeira ou melhor melhorias, respectivamente. A solução reconstruída é usada na próxima iteração. Ao final do número máximo de iterações, a melhor solução encontrada retorna.

Os algoritmos desenvolvidos foram testados em dois cenários. O primeiro (*grid 4x4*) consiste em uma rede no formato de grade. O segundo (*Sioux Falls*) consiste em um cenário amplamente usado na literatura para simulação de tráfego, mas ainda não explorado para o projeto de redes. Os resultados são apresentados na Tabela 1, onde o baseline amostrou 50 soluções e os outros algoritmos foram executados por 5 iterações. Ambos cenários foram simulados com 50 veículos e com um budget de 5 km de intervenções. Os melhores valores de cada cenário estão destacados em negrito.

Tabela 1: Avaliação dos métodos propostos.

Cenário	Algoritmo	Estratégia	Tempo de viagem
grid 4x4	baseline	–	8,02
	busca local restrita	PM	8,18
		MM	8,27
		–	–
	guloso iterado	PM	7,50
MM		<b>7,16</b>	
Sioux Falls	baseline	–	<b>41,40</b>
	busca local restrita	PM	43,50
		MM	43,88
		–	–
	guloso iterado	PM	43,26
MM		42,79	

Como pode-se observar nos resultados, o algoritmo guloso iterado apresentou o melhor resultado no cenário artificial (*grid*), enquanto o algoritmo de *baseline* não foi superado no cenário *Sioux Falls*. A avaliação exaustiva de soluções em alguns componentes dos algoritmos de busca local e guloso faz com que eles tenham um tempo de processamento elevado e tenham que ser avaliados com poucas iterações. Como trabalhos futuros, serão realizadas melhorias nesses algoritmos para aumentar seu desempenho. Além disso, serão implementadas políticas de interrupção prematura na avaliação de soluções com tempo de viagem superior ao apresentado pela melhor solução encontrada.

**Palavras-chave:** Projeto de redes. Otimização heurística. Busca local.