

Otimização de uma Rede V2V Através de uma Abordagem Multi-Objetiva

Davino Mauro Tenório da Silva Júnior
Centro de Informática
Universidade Federal de Pernambuco
Email: dmts@cin.ufpe.br

Victor Hugo Sabino dos Santos Araújo
Centro de Informática
Universidade Federal de Pernambuco
Email: vhssa@cin.ufpe.br

Resumo—

I. INTRODUÇÃO

Os Sistemas de Transporte Inteligentes (ITS), de acordo com [1], lidam com as questões de tráfego como congestionamento e disseminação de informações, e estão cada vez mais em destaque com o avanço das tecnologias de comunicação sem fio e de automóveis. As Redes *Ad-Hoc* Veiculares (VANETs), que derivam das redes *Ad-Hoc* Móveis (MANETs), permitem que veículos em movimento estejam conectados e se comuniquem sem fio. Esta comunicação pode ser entre dois veículos (V2V) ou entre um veículo e uma infraestrutura (V2I) [2].

Para estabelecer uma comunicação segura e eficiente entre veículos, é necessário que haja uma infraestrutura de comunicação adequada. O padrão IEEE 802.11p apresentado em [3] de Comunicação Dedicada de Curto Alcance (DSRC) foi desenvolvido especificamente para requisitos das VANETs como: auto-organização, configuração automática, topologia dinâmica e alta mobilidade.

Para que a comunicação sem fio opere em tempo real, existem restrições associadas que precisam ser gerenciadas na camada física, como: o tempo de disseminação de dados (*delay*), a perda de pacotes e a taxa de transferência (*throughput*). [4] diz que para garantir um desempenho otimizado em tempo real, é necessário que ambas as restrições sejam atendidas simultaneamente, o que não é trivial devido à limitação de banda e o ambiente veicular altamente dinâmico.

Dentre os vários desafios de VANETs abordados na literatura, [6] destaca o cenário dinâmico e a necessidade de protocolos específicos para redes veiculares que sejam menos afetados pelas frequentes mudanças na rede. [5] aponta o problema de terminal escondido como a principal causa de perda de pacotes em VANETs e propõe uma solução de otimização adaptativa baseada em lógica *Fuzzy* para resolvê-lo. Já [7] faz uma análise do impacto do alcance de transmissão no desempenho de VANETs em termos de taxa de entrega de pacotes e *delay*, enquanto [8] relaciona a potência de transmissão de acordo com seu alcance. [5] também destaca a importância de uma janela de contenção dinâmica, que seja otimizada de acordo com o congestionamento no canal. **[BUSCAR REFERÊNCIA SOBRE A IMPORTÂNCIA DA RELAÇÃO SINAL-RUÍDO MAIS INTERFERÊNCIA]**

As principais contribuições deste artigo são: primeiramente, simular um cenário real de nós em movimento para extrair informações da rede como: *delay*, perda de pacote e *throughput*. Em seguida, definir um espaço amostral de parâmetros de entrada para a simulação, tendo como variáveis a potência de transmissão, o tamanho da janela de contenção (CW_{min} e CW_{max}) e a Relação Sinal-Ruído mais Interferência (SINR). Com isso, é possível fazer uma simulação exaustiva com todo o espaço amostral e realizar um estudo comparativo com o resultado obtido através da aplicação de uma técnica de otimização baseada no algoritmo genético multi-objetivo *Fast Non-dominated Sorting Algorithm* (NSGA-II).

O restante do artigo está organizado da seguinte forma: na Seção II, é apresentado o modelo do sistema a ser simulado; na Seção III, é formulado o problema multi-objetivo e apresentado o algoritmo NSGA-II; na Seção IV, é construído o modelo de simulação e a avaliação de seu desempenho; a seção V conclui o trabalho e são discutidos possíveis trabalhos futuros.

II. MODELO DO SISTEMA

Neste estudo, é considerado um ambiente urbano comumente encontrado em regiões metropolitanas. A Figura 1 representa o cenário utilizado para a simulação, retirado do *OpenStreetMaps*, o ambiente é localizado no centro da cidade do Recife, nas mediações da Av. Agamenom Magalhães. O número de veículos dentro da rede aumenta conforme o tempo de simulação, e são aleatoriamente colocados no mapa e com velocidade variando de acordo com o tráfego do local, tornando o modelo mais realista. **[VERIFICAR COMO FOI FEITA A SIMULAÇÃO DO SUMO]**

Assume-se que todos os veículos estão equipados com uma Unidade de Controle de Bordo (OBU), que dá suporte a comunicações V2I e V2V. Adicionalmente, os veículos são capazes de transmitir e receber informações atualizadas enquanto trafegam sobre o tráfego, acidentes, e informações baseadas em localização.

III. FORMULAÇÃO DO PROBLEMA

Baseado nas informações anteriores, o objetivo deste estudo é minimizar a perda de pacotes e o *delay* de disseminação, e aumentar o *throughput* de dados de uma rede

VANET considerando apenas a comunicação entre veículos (V2V). Portanto, pode-se propor uma técnica de Otimização Multi-Objetiva (MOO) para resolver este problema. A MOO é um campo que diz respeito a otimização de duas ou mais funções objetivas [9]. A solução deste problema envolve a localização de um conjunto de soluções candidatas chamadas de não dominadas. O conjunto de soluções não dominadas ótimas é chamado de pareto ótimo. Todas as soluções que estão no pareto ótimo pertencem ao conjunto de paretos, e os pontos representados em relação a cada objetivo no espaço são chamados de pareto *front*. A complexidade dos problemas MOO são diretamente ligadas as dependências entre as variáveis de decisão em relação a seus objetivos, no caso em que existem conflitos entre estas variáveis, é necessário um *trade-off* para analisar qual objetivo deve ser priorizado.

O presente trabalho adotou o algoritmo genético multi-objetivo denominado *Fast Non-dominated Sorting Genetic Algorithm* (NSGA-II) proposto por [11]. Este algoritmo aplica o conceito de dominância ao classificar elementos da população de maneira elitista em fronteiras (*fronts*), de acordo com o grau de dominância. A cada geração, os indivíduos classificados no primeiro *front* são considerados melhores, ou, dominantes. Após a classificação em *fronts*, os indivíduos também são ordenados através de um operador de diversidade baseado num indicador de densidade (*crowding distance*), que relaciona a distância de cada indivíduo para seus vizinhos imediatos com a distância entre as soluções extremas do mesmo *front*. O pseudocódigo para o NSGA-II é apresentado no Algoritmo 1 a seguir.

IV. AVALIAÇÃO DO DESEMPENHO

Neste trabalho, os cenários descritos na seção III foram simulados utilizando a ferramenta Veins [Davino:[Descrever o veins?]] de forma exaustiva, tendo como resultado um arquivo de resultados. Para avaliação do algoritmo multi-objetivo em busca da melhor configuração de parâmetros a serem utilizados no simulador, foi utilizada uma implementação em Ruby do algoritmo NSGA-II com funções objetivo adaptadas para o problema proposto. As funções foram baseadas no problema multi-objetivo introduzido por Schaffer e apresentado em [11], e visam maximizar o *throughput* e minimizar a perda de pacotes e *delay*.

Foi utilizada uma implementação do algoritmo NSGA-II na linguagem de programação Ruby [9], adaptando a mesma para o problema e propósito deste trabalho.

A. Parâmetros de Configuração

[Davino:[Victor, por favor descreve de forma breve os parâmetros utilizados (cwmmin, cwmax, slotlength, txpower)]]

a) *Simulação*: A representação do indivíduo no algoritmo NSGA-II se dá por um mapa de 16 bits. Cada indivíduo é dividido de forma a representar os quatro parâmetros de configuração utilizados no simulador e descritos na seção IV-A, a saber (1) CW_{min} , (2) CW_{max} , (3) slot length, e (4) txPower, como mostra a figura 1. A simulação consiste

em efetuar o *parse* desse indivíduo para uma dada configuração, isto é, um valor para cada parâmetro, e buscar essa configuração no arquivo resultante da busca exaustiva. Uma vez encontrados os resultados da simulação realizada com os parâmetros de configuração do indivíduo, isto é, *throughput*, perda de pacotes e *delay*, são calculadas as respectivas funções objetivo buscando determinar se o indivíduo é ótimo, isto é, se condizem com os objetivos propostos na seção IV. O algoritmo NSGA-II busca então pelo pareto *front* com os melhores resultados de simulação para um dado indivíduo (configuração), executando o ranqueamento dos paretos não dominados, e os passos de reprodução e *crossover*.

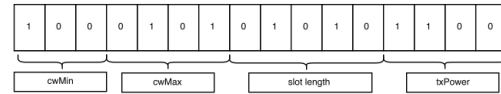


Figura 1. Indivíduo utilizado no algoritmo representando uma dada configuração para simulação.

V. RESULTADOS

Para execução do algoritmo NSGA-II, foi utilizada uma taxa de *crossover* de 0.98%, como consta em [9], bem como 50 gerações. A população utilizada variou de 5 a 50, incrementando de 5 em 5. Os resultados em termos de *throughput*, perda de pacotes e *delay* podem ser encontrados nas figuras 2, 3 e 4.

Em termos de *throughput*, pode-se observar que os resultados convergem rapidamente, logo a partir das primeiras gerações, com poucas ocorrências de *outliers* (como visto na 35 geração). Pode-se observar também que, uma população menor que 30 indivíduos não apresenta convergência ou mesmo resultados ótimos. É importante notar que utiliza-se 1/*throughput* em virtude dos valores simulados do *throughput* serem muito baixos.

Em termos de perda de pacotes, os resultados apresentam convergir a partir de 35 gerações. Nota-se que, com uma população abaixo de 50 indivíduos, ocorre uma alta taxa de variação de resultados, sugerindo que populações com mais indivíduos podem ser utilizadas a fim de minimizar a perda de pacotes.

Em termos de *delay*, populações acima de 40 indivíduos obtêm resultados convergentes logo de início, o que sugere a otimização do algoritmo NSGA-II em obter resultados ótimos. Populações abaixo de 40 indivíduos mostraram-se ineficazes em minimizar o *delay*.

A. Simulações por Geração

Com a população variando de 5 a 50 indivíduos, o número de simulações aumenta proporcionalmente ao número de indivíduos e gerações, como pode ser observado na figura 8. Observa-se que o número de simulações chega a pouco

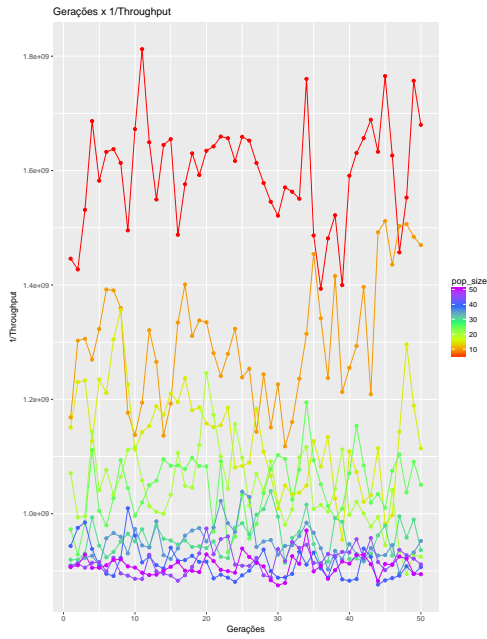


Figura 2. Resultado em termos de gerações por média de 1/Throughput alcançados nas simulações.

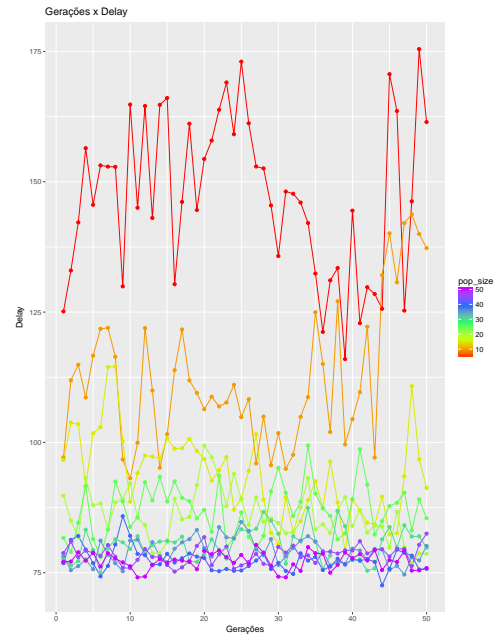


Figura 4. Resultado em termos de gerações por média de delay alcançados nas simulações.

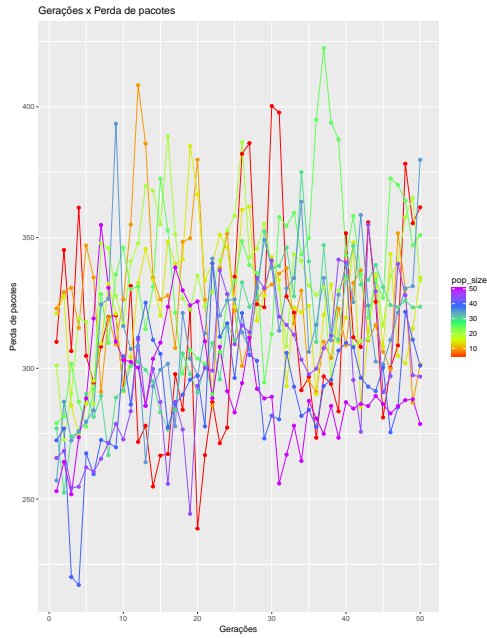


Figura 3. Resultado em termos de gerações por média de perda de pacotes alcançados nas simulações.

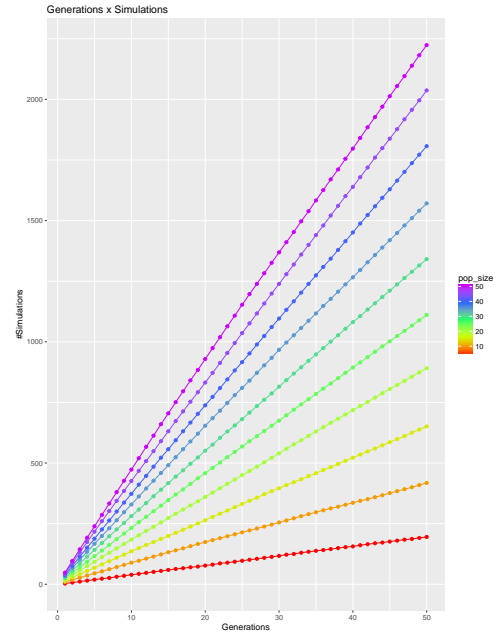


Figura 5. Número de simulações por geração.

menos de 2500 para uma população de 50 indivíduos, consistindo em um alto número de simulações para obter resultados ótimos se comparado a poucos indivíduos.

B. Comparação com exaustivo

Os resultados da simulação exaustiva para os cenários propostos neste trabalho podem ser comparados com o pareto

ótimo dos resultados obtidos com o algoritmo NSGA-II. As figuras ??, ?? e ??.

VI. CONCLUSÃO

Neste trabalho, foi apresentada uma implementação do algoritmo NSGA-II como forma de otimizar e alcançar configurações ótimas visando minimizar o número de simulações a serem executadas com o intuito de (1) minimizar a perda de pacotes, (2) minimizar o delay e (3) maximizar o



Figura 6. Número de simulações por geração.

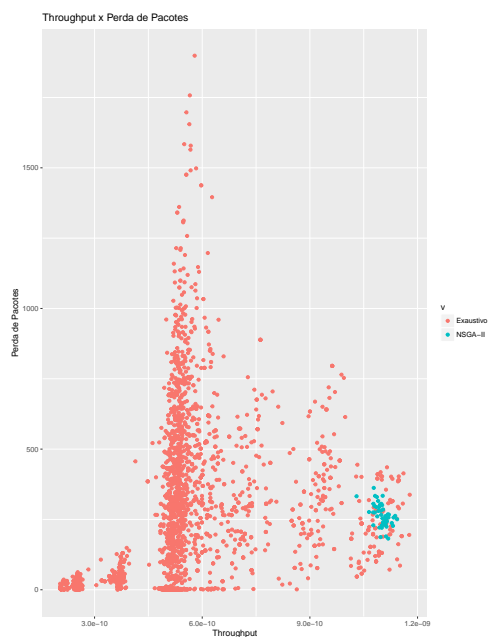


Figura 7. Número de simulações por geração.

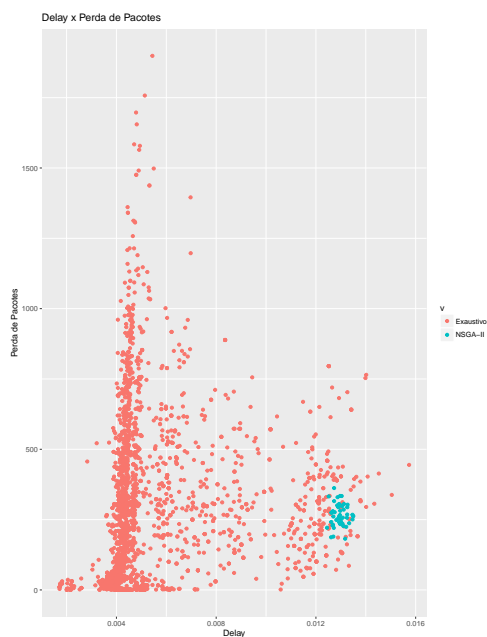


Figura 8. Número de simulações por geração.

throughput na comunicação V2V num cenário urbano real. Os resultados mostraram-se satisfatórios, com o algoritmo obtendo uma rápida convergência de soluções ótimas com poucas gerações se considerada uma população de 40 a 50 indivíduos. No entanto, é visível o alto número de simulações necessários para obter tais resultados, chegando a pouco mais de 2000 simulações.