

Improving QoS in Vehicular ad-hoc Networks using a Multi-Objective Genetic Algorithm

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—As rápidas mudanças de topologia em Redes Ad-Hoc Veiculares (VANETs) são comuns devido a grande mobilidade dos nós. Para obter uma transmissão de pacotes eficiente, é preciso avaliar os parâmetros de rede, e encontrar um valor ótimo de acordo com a rede e condições de tráfego. Neste artigo, é apresentado um esquema de otimização Multiobjetiva utilizando o algoritmo *Fast Non-Dominated Sorting Algorithm* (NSGA-II) para melhorar a Qualidade de Serviço (QoS) em redes VANET.

Palavras Chave: VANETs, QoS, NSGA-II, Perda de Pacotes, Delay, Throughput.

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, [5] destaca o cenário dinâmico e a necessidade de protocolos específicos para redes veiculares que sejam menos afetados pelas frequentes mudanças na rede. [6] 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. [6] também destaca a importância de uma janela de contenção dinâmica, que seja otimizada de acordo com o congestionamento no canal. [9] mostra que a interferência é um dos principais fatores da degradação de transmissão em VANETs, a qualidade do sinal pode ser calculada através da SINR, que é considerado um bom indicador das condições de rede. Por isso, SINR será utilizado como parâmetro juntamente com a potência de transmissão e o tamanho da janela de contenção.

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. O cenário utilizado para a simulação foi 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.

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 [10]. 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. A função *SortByRankAndDistance* ordena a população em uma hierarquia de Pareto *fronts* não dominados, já a *CrowdingDistanceAssignment* calcula a distância média entre membros de cada *front*. Já a função *CrossoverAndMutation* executa as operações de mutação e cruzamento do GA. Já as funções *SelectParentsByRankAndDistance* e *SortByRankAndDistance* fazem a seleção dos membros da população em termos de precedência do *front* a qual a solução pertence, e a distância dele para o *front*, calculado anteriormente pelo *CrowdingDistanceAssignment*.

IV. AVALIAÇÃO DO DESEMPENHO

Neste trabalho, os cenários descritos na seção III foram simulados utilizando a ferramenta *Vehicles in Network Simulation* (VEINS), que é composto de dois simuladores, o

Algoritmo 1: Pseudocode for NSGAII.

Input: $Population_{size}$, ProblemSize, $P_{crossover}$, $P_{mutation}$
Output: Children

```

1 Population  $\leftarrow$  InitializePopulation( $Population_{size}$ , ProblemSize);
2 EvaluateAgainstObjectiveFunctions(Population);
3 FastNondominatedSort(Population);
4 Selected  $\leftarrow$  SelectParentsByRank(Population,  $Population_{size}$ );
5 Children  $\leftarrow$  CrossoverAndMutation(Selected,  $P_{crossover}$ ,  $P_{mutation}$ );
6 while  $\neg$ StopCondition() do
7   EvaluateAgainstObjectiveFunctions(Children);
8   Union  $\leftarrow$  Merge(Population, Children);
9   Fronts  $\leftarrow$  FastNondominatedSort(Union);
10  Parents  $\leftarrow$   $\emptyset$ ;
11   $Front_L \leftarrow \emptyset$ ;
12  foreach  $Front_i \in$  Fronts do
13    CrowdingDistanceAssignment( $Front_i$ );
14    if Size(Parents)+Size( $Front_i$ ) >  $Population_{size}$  then
15       $Front_L \leftarrow i$ ;
16      Break();
17    else
18      Parents  $\leftarrow$  Merge(Parents,  $Front_i$ );
19    end
20  end
21  if Size(Parents) <  $Population_{size}$  then
22     $Front_L \leftarrow$  SortByRankAndDistance( $Front_L$ );
23    for  $P_1$  to  $P_{Population_{size}-Size(Front_L)}$  do
24      Parents  $\leftarrow P_i$ ;
25    end
26  end
27  Selected  $\leftarrow$  SelectParentsByRankAndDistance(Parents,  $Population_{size}$ );
28  Population  $\leftarrow$  Children;
29  Children  $\leftarrow$  CrossoverAndMutation(Selected,  $P_{crossover}$ ,  $P_{mutation}$ );
30 end
31 return Children;
```

OMNeT++, que é um simulador de redes baseado em eventos e o *Simulation of Urban Mobility* (SUMO) para simulação de tráfego em estradas, que pode extrair rotas do mundo real através do banco de dados do OpenStreetMap [12]. Ambos os simuladores são gratuitos e dispõe de tutoriais para os interessados em aprender a utilizá-los. A simulação no VEINS foi realizada de forma exaustiva, dados os parâmetros de configuração que serão descritos a seguir, e foi gerado 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 [12], 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 [10], adaptando a mesma para o problema e proposito deste trabalho.

A. Parâmetros de Configuração

Os parâmetros de configuração definidos foram a Potência de Transmissão ($txpower$), a SINR ($slotlength$), e o tamanho da janela de contenção (CW_{min} e CW_{max}). Para avaliar o desempenho do NSGA-II em relação ao exaustivo, foi definido um conjunto de valores que compõe o espaço de

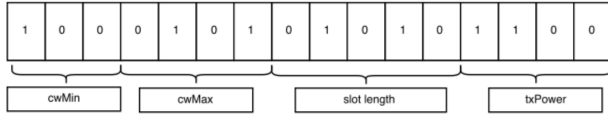


Figura 1. Indivíduo utilizado no algoritmo representando uma possível configuração para simulação.

busca, e ele está disposto da seguinte forma: $txpower = [1, 4, 6, 10, 12, 14, 17, 20, 24, 27, 29, 31, 32] \text{ dBm}$; $CW_{min} = [16, 32, 64, 128, 256, 512] \text{ bits}$; $CW_{max} = [1024, 2048, 4096] \text{ bits}$; $slotlength = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100] \mu S$. Assim, o espaço de busca é de 4680 possibilidades ou simulações, considerando que o tempo para executar uma simulação seja de 600 segundos, o tempo necessário para simular todo o espaço é de 780 horas, pouco mais de um mês. Logo, a utilização de uma técnica de otimização pode reduzir o tempo de simulação e alcançar um resultado próximo do ótimo mais rapidamente.

B. 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*.

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 $throughput^{-1}$ em virtude dos valores simulados do *throughput* serem muito baixos.

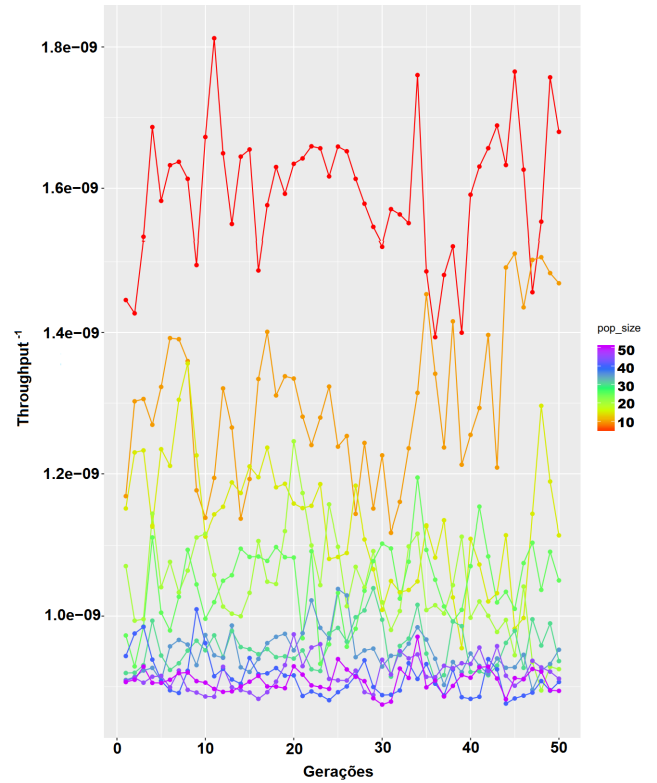


Figura 2. Resultado em termos de gerações por média de $1/Throughput$ alcançados nas simulações.

Em termos de perda de pacotes, os resultados convergem 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 5. Observa-se que o número de simulações chega a pouco 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 6, 7 e 8.

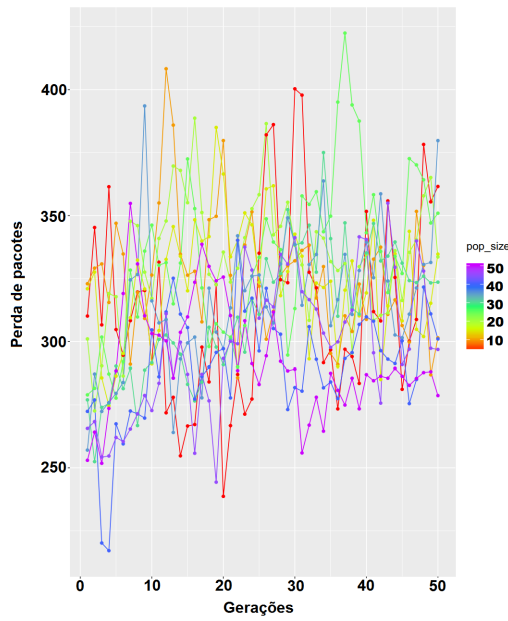


Figura 3. Resultado em termos de gerações por média de perda de pacotes alcançados nas simulações.

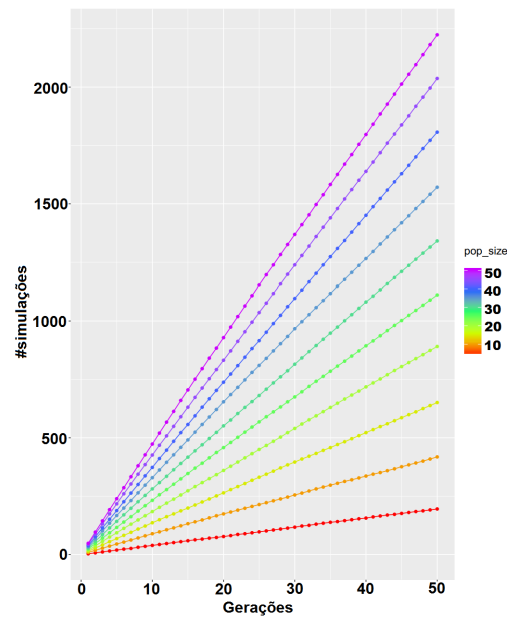


Figura 5. Número de simulações por geração.



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

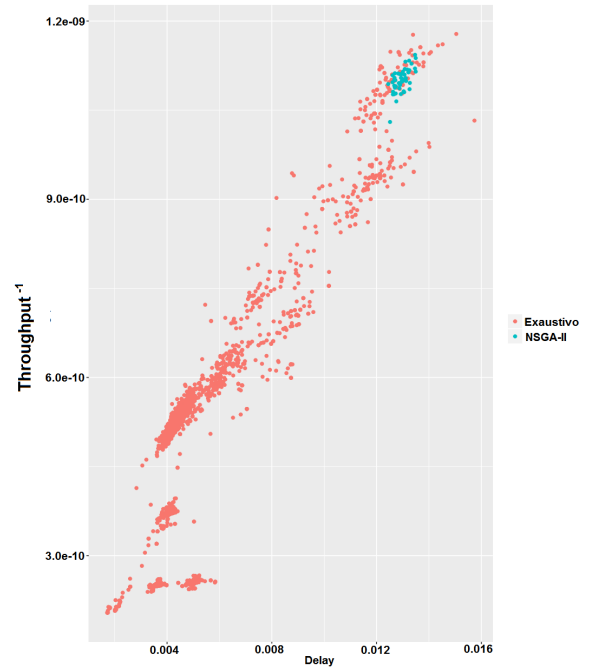


Figura 6. Comparação entre a simulação exaustiva e o NSGA-II em termos de Delay X Throughput.

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 throughput na comunicação V2V num cenário urbano real. Os resultados mostraram que é possível obter 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 chegar a tal conclusão, chegando a pouco mais de 2000 simulações. Para trabalhos futuros, sugere-se a implementação de outras técnicas multiobjetivas com o intuito de avaliar a mais eficiente para este caso.

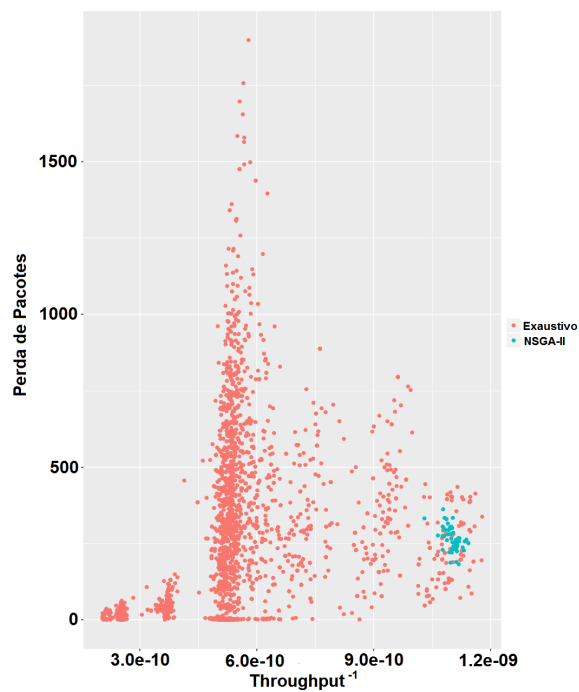


Figura 7. Comparação entre a simulação exaustiva e o NSGA-II em termos de Throughput X Perda de Pacotes.

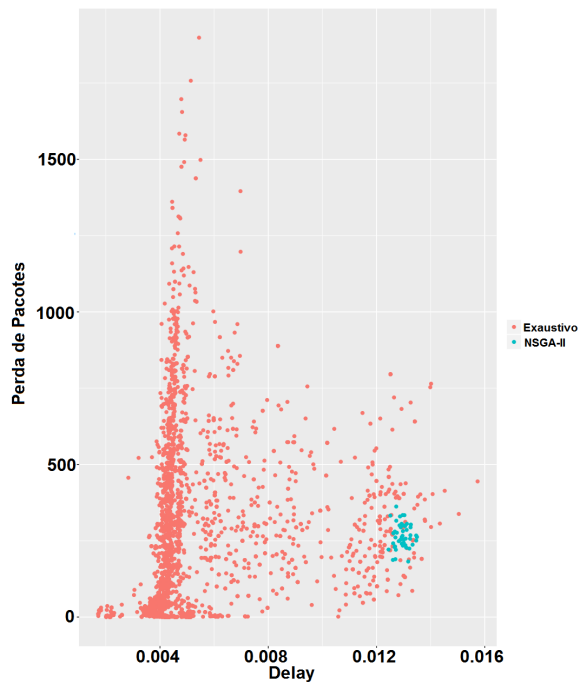


Figura 8. Comparação entre a simulação exaustiva e o NSGA-II em termos de Delay X Perda de Pacotes.

REFERÊNCIAS

- [1] L. Vanajakshi, G. Ramadurai, and A. S. Anand, "Intelligent transportation systems," 2010.
- [2] S. Anwer and C. Guy, "A survey of vanet technologies," 2016.
- [3] D. Jiang and L. Delgrossi, "Ieee 802.11p: Towards an international

standard for wireless access in vehicular environments," in *VTC Spring*, 2008.

- [4] P. Dai, K. Liu, L. Feng, Q. Zhuge, V. C. S. Lee, and S. H. Son, "Towards real-time and temporal information services in vehicular networks via multi-objective optimization," *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pp. 671–679, 2016.
- [5] F. D. da Cunha, L. A. Villas, A. Boukerche, G. Maia, A. C. Viana, R. A. F. Mini, A. A. F. Loureiro, and F. Cunha, "Data communication in vanets: Survey, applications and challenges," 2017.
- [6] J. M.-Y. Lim, Y. C. Chang, J. Loo, and M. Y. Alias, "Improving vanet performance with heuristic and adaptive fuzzy logic scheme," *Wireless Personal Communications*, vol. 83, pp. 1779–1800, 2015.
- [7] A. A. Almohammed, N. K. Noordin, and S. Saeed, "Evaluating the impact of transmission range on the performance of vanet," 2016.
- [8] D. B. Rawat, D. C. Popescu, G. Yan, and S. Olariu, "Enhancing vanet performance by joint adaptation of transmission power and contention window size," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 1528–1535, 2011.
- [9] R. K. Schmidt, T. Kollmer, T. Leinmller, B. Bddeker, and G. Schafer, "Degradation of transmission range in vanets caused by interference," *Praxis der Informationsverarbeitung und Kommunikation*, vol. 32, pp. 224–234, 2009.
- [10] J. Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes*, 1st ed. Lulu.com, 2011.
- [11] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. Evolutionary Computation*, vol. 6, pp. 182–197, 2002.
- [12] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.