

Servílio Souza de Assis

Otimização de Rotas em Redes  
Definidas por Software Utilizando  
Algoritmos Evolucionários

Belo Horizonte - Minas Gerais

Junho, 2021



Servílio Souza de Assis

# Otimização de Rotas em Redes Definidas por Software Utilizando Algoritmos Evolucionários

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica.

Universidade Federal de Minas Gerais - UFMG

Programa de Pós-Graduação em Engenharia Elétrica - PPGEE

Laboratório de Redes, Tecnologias e Serviços de Comunicação a Distância - LabCOM

Orientador: Prof. Luciano de Errico

Coorientador: Prof. Hani Camille Yehia

Belo Horizonte - Minas Gerais

Junho, 2021

A848o

Assis, Servílio Souza de.

Otimização de rotas em redes definidas por software utilizando algoritmos evolucionários [recurso eletrônico] / Servílio Souza de Assis. - 2021.

1 recurso online (86 f. : il., color.) : pdf.

Orientador: Luciano de Errico.

Coorientador: Hani Camille Yehia.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 85-86.

Bibliografia: f. 79-84.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Desempenho - Teses. 3. Eficiência energética - Teses. 4. Otimização multiobjetivo - Teses. I. Errico, Luciano de. II. Yehia, Hani Camille. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

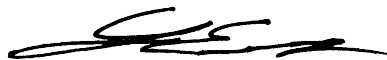
**"Otimização de Rotas em Redes Definidas por Software  
Utilizando Algoritmos Evolucionários"**

**Servílio Souza de Assis**

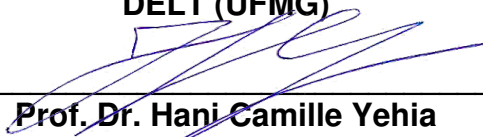
Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 28 de junho de 2021.

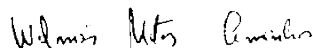
Por:



Prof. Dr. Luciano de Errico  
DELT (UFMG)



Prof. Dr. Hani Camille Yehia  
DELT (UFMG)



Prof. Dr. Walmir Matos Caminhas  
DELT (UFMG)



Prof. Dr. Adriano Vilela Barbosa  
DELT (UFMG)



Prof. Dr. Frederico Gualberto Ferreira Coelho  
DELT (UFMG)

*Dedico este trabalho primeiramente a Deus, criador de todas as coisas, por ter me dado forças e me ajudado a passar por todas as dificuldades ao longo desta dura jornada.*

*Aos meus pais, por todo incentivo, apoio e exemplo dados.*

*Por fim, dedico a todos aqueles a quem esta pesquisa possa ajudar de alguma maneira.*

# Agradecimentos

Agradeço primeiramente ao criador de todas as coisas, pois sem ele a vida e qualquer outra coisa atrelada a ela não seria possível.

Agradeço aos meus pais, irmãos e todos os meus familiares por sempre estarem comigo, me apoiando, orando e torcendo por mim durante toda a minha trajetória neste curso.

Agradeço ao meu professor orientador, Luciano de Errico, por todas as orientações, direcionamentos e aprendizados proporcionados, e ao meu professor coorientador Hani Camille Yehia por todas as contribuições dadas ao longo do desenvolvimento deste trabalho.

Agradeço aos membros do LabCOM, por todos os encontros, as apresentações, as trocas de conhecimento e os aprendizados gerados, sendo estes essenciais para o desenvolvimento deste trabalho.

Por fim, agradeço aos meus colegas discentes, aos docentes do Programa de Pós-Graduação em Engenharia Elétrica da UFMG e a todas as pessoas que, direta ou indiretamente, contribuíram para o desenvolvimento deste trabalho.

Meus sinceros agradecimentos!



*"It is hard to fail, but it is worse never to have tried to succeed".*

*(Theodore Roosevelt)*



# Resumo

A popularização de serviços e aplicações na nuvem tem mudado drasticamente os perfis de tráfego na Internet. Tais mudanças têm motivado a evolução do paradigma SDN (*Software Defined Networking*), onde são separados os planos de dados e de controle dos elementos de comutação das redes, possibilitando uma visão centralizada de gerência e uma série de novas aplicações. Nessa perspectiva, no presente trabalho foram desenvolvidos modelos de otimização, baseados em algoritmos evolucionários, para roteamento em cenários de melhor esforço e com SLA (*Service Level Agreement*), atendendo a critérios de balanceamento de carga e de eficiência energética no uso de recursos. Foram desenvolvidos e avaliados modelos de otimização para diferentes contextos, considerando a qualidade das soluções alcançadas e o desempenho dos métodos. Após diversas análises, o algoritmo NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) foi escolhido para utilização na obtenção de conjuntos de soluções ótimas. No caso de SLA, foi escolhida a técnica de agregação de objetivos em um algoritmo genético com preferências definidas a priori. Experimentos realizados através de emulação e simulação indicaram melhoria de desempenho das redes, conforme as preferências definidas por um tomador de decisões, com o algoritmo mostrando capacidade de convergir para rotas que atendam as restrições das demandas de fluxos da rede, garantindo também um uso mínimo de recursos, visando eficiência energética.

Palavras-chave: *Software Defined Networking*, Algoritmos Evolucionários, Otimização Multiobjetivo, Eficiência Energética, Desempenho.



# Abstract

The popularization of cloud services and applications has drastically changed Internet traffic profiles. Such changes have motivated the evolution of the SDN (Software Defined Networking) paradigm, where the data and control planes are separated from the switching elements of the networks, enabling a centralized management view and a series of new applications. From this perspective, optimization models were developed in this work, based on evolutionary algorithms, for routing in best effort and with SLA (Service Level Agreement) scenarios, meeting load balancing and energy efficiency criteria in the use of resources. Optimization models were developed and evaluated in different contexts, considering the quality of the achieved solutions and the performance of the methods. After several analyses, the NSGA-II (Non-dominated Sorting Genetic Algorithm II) was chosen for use in obtaining sets of optimal solutions. In the case of SLA, the objective aggregation technique with a genetic algorithm was chosen, defining a priori preferences. Experiments performed through emulation and simulation indicated an improvement in the performance of the networks, according to the preferences defined by a decision maker, with the algorithm showing the ability to converge to routes that meet the restrictions of the network's flow demands, also ensuring a minimum use of resources, aiming at energy efficiency.

Keyword: Software Defined Networking, Evolutionary Algorithms, Multiobjective Optimization, Energy Efficiency, Performance.



# Lista de Ilustrações

Figura 1 – Arquitetura de uma rede SDN. . . . .	29
Figura 2 – Componentes básicos de um <i>switch</i> OpenFlow. . . . .	31
Figura 3 – Esquema das tabelas de fluxo OpenFlow. . . . .	31
Figura 4 – Exemplo de conjunto Pareto de soluções. . . . .	33
Figura 5 – Abordagens de otimização de um problema multiobjetivo. . . . .	34
Figura 6 – Representação de uma instância do problema. . . . .	46
Figura 7 – Representação de uma solução para o problema. . . . .	47
Figura 8 – Operação genética de cruzamento. . . . .	48
Figura 9 – Operação genética de mutação. . . . .	48
Figura 10 – Representação gráfica do hipervolume em duas dimensões. . . . .	50
Figura 11 – Topologia para testes de melhor esforço na emulação. . . . .	51
Figura 12 – Arquitetura do aplicativo do controlador. . . . .	54
Figura 13 – Fluxograma de execução da simulação. . . . .	55
Figura 14 – Esquema da rede utilizada na simulação. . . . .	55
Figura 15 – Resultados obtidos para uma instância do problema de atribuição de rotas, onde em (a) têm-se uma população inicial de soluções no espaço de objetivos e em (b) a fronteira Pareto aproximada. . . . .	58
Figura 16 – <i>Boxplots</i> dos valores de hipervolume obtidos para a configuração 1. . . . .	59
Figura 17 – <i>Boxplots</i> dos valores de hipervolume obtidos para a configuração 2. . . . .	59
Figura 18 – <i>Boxplots</i> dos valores de hipervolume obtidos para a configuração 3. . . . .	59
Figura 19 – Comparação de pares de algoritmos. . . . .	63
Figura 20 – Evolução do tempo de processamento por geração. . . . .	64
Figura 21 – Tempo de execução médio dos métodos de Pareto e de agregação com variação do número de demandas. . . . .	65

Figura 22 – Tempo de execução médio dos métodos de Pareto e de agregação com variação do tamanho da topologia. . . . .	65
Figura 23 – Largura de banda média alcançada pelos fluxos admitidos na rede em função do tempo em uma execução experimental de cada cenário. . . . .	67
Figura 24 – RTT médio experimentado pelos fluxos da rede em função do tempo em cada um dos cenários reproduzidos. . . . .	68
Figura 25 – Número de enlaces utilizados na rede ao longo do tempo em cada um dos cenários reproduzidos. . . . .	69
Figura 26 – Convergência de aptidão média da população para uma execução do algoritmo. . . . .	71

# Lista de Tabelas

Tabela 1 – Mensagens OpenFlow. . . . .	32
Tabela 2 – Parâmetros estatísticos dos dados coletados de HV. . . . .	60
Tabela 3 – Resultados obtidos com ANOVA. . . . .	61
Tabela 4 – Tempo médio de processamento dos algoritmos para uma geração. . . . .	64
Tabela 5 – Resultados médios obtidos. . . . .	68
Tabela 6 – Parâmetros do otimizador. . . . .	70
Tabela 7 – Demandas de fluxos para alocação de rotas na simulação. . . . .	71
Tabela 8 – Rotas definidas no processo da simulação. . . . .	72



# Lista de Abreviaturas e Siglas

ANOVA	Analysis of Variance
ECMP	Equal Cost Multipath
HDF	Highest Demand First
HV	Hipervolume
HYPE	Hypervolume Estimation Algorithm for Multiobjective Optimization
IBEA	Indicator-Based Evolutionary Algorithm
LPF	Longest Path First
MOCELL	Cellular Genetic Algorithm for Multiobjective Optimization
MOEA	Multiobjective Evolutionary Algorithms
NMU	Next Maximum Utility
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NSGA-III	Non-dominated Sorting Genetic Algorithm III
NSP	Next Shortest Path
QoS	Quality of Service
REST	Representational State Transfer
RTT	Round Trip Time
SDF	Smallest Demand First
SDN	Software Defined Networking

SLA	Service Level Agreement
SOA	Service-Oriented Architecture
SPEA2	Strength Pareto Evolutionary Algorithm 2
SPF	Shortest Path First
STP	Spanning Tree Protocol
TE	Traffic Engineering

# Sumário

<b>1</b>	<b>Introdução</b>	<b>21</b>
1.1	Contextualização	21
1.2	Motivação e Justificativa	23
1.3	Objetivos	25
1.3.1	Geral	25
1.3.2	Específicos	25
1.4	Estrutura do Texto	26
<b>2</b>	<b>Referencial Teórico</b>	<b>27</b>
2.1	Engenharia de Tráfego	27
2.2	Redes Definidas por Software	29
2.2.1	Arquitetura	29
2.2.2	Protocolo OpenFlow	30
2.3	Otimização Multiobjetivo	33
2.3.1	Algoritmos Evolucionários	35
2.4	Problema de Otimização de Rotas	36
2.5	Trabalhos relacionados	37
2.6	Resumo	40
<b>3</b>	<b>Materiais e Métodos</b>	<b>41</b>
3.1	Definição do Problema	41
3.1.1	Otimização em Melhor Esforço	43
3.1.2	Otimização em SLA	44
3.2	Operadores Genéticos	46
3.3	Análise de Algoritmos de Pareto	49
3.4	Cenários Experimentais	51
3.5	Resumo	56
<b>4</b>	<b>Resultados e Discussão</b>	<b>57</b>
4.1	Experimento de Comparação de Algoritmos	57
4.2	Análise de Comportamento das Soluções	66
4.2.1	Política de Melhor Esforço	66

4.2.2	Política de SLA . . . . .	70
4.3	Aplicações . . . . .	73
4.4	Resumo . . . . .	74
<b>5</b>	<b>Conclusões . . . . .</b>	<b>75</b>
5.1	Limitações e Trabalhos Futuros . . . . .	77
5.2	Produção Bibliográfica . . . . .	78
	<b>Referências . . . . .</b>	<b>79</b>
	<b>Apêndice A Resultados dos Testes de Tukey . . . . .</b>	<b>85</b>

# Capítulo 1

## Introdução

Este capítulo apresenta uma contextualização sobre redes de computadores na atualidade, visando introduzir ao leitor os assuntos tratados neste trabalho. É dito sobre a caracterização do problema de otimização de rotas e sua importância, o que define a motivação e justificativa para realização deste trabalho. Por fim, são definidos os objetivos geral e específicos e a estrutura utilizada na construção do texto.

### 1.1 Contextualização

O crescente ganho de popularidade de serviços e aplicações na Internet tem contribuído com o aumento das demandas por infraestrutura e processamento em servidores dispostos em *data centers*. Esse crescimento elevado tem provocado mudanças drásticas nos padrões de tráfego de usuários, onde o volume e a diversidade de tipos de tráfego aumentam constantemente.

Nos últimos 20 anos, o número de usuários conectados à Internet tem apresentado um crescimento significativo: no início dos anos 2000, 304 milhões de pessoas eram conectadas. Ao final de 2020, o número de usuários saltou para aproximadamente 5 bilhões, o que corresponde a um aumento de mais de 16 vezes ([Internet World Stats, 2020a](#); [Internet World Stats, 2020b](#)).

Além do aumento expressivo do número de pessoas conectadas ao longo dos anos, outro ponto importante é que cada vez mais pessoas utilizam mais de um dispositivo ao mesmo tempo. Com o avanço da Internet das Coisas ([Wortmann and Flüchter, 2015](#)), e de conceitos como Indústria 4.0 ([Santos et al., 2018](#)), diferentes tipos de dispositivos pessoais,

como *Smart TVs*, *Smartphones*, *Smartwatches*, computadores portáteis, e equipamentos industriais, como sensores, atuadores e controladores, tendem também a estar cada vez mais ligados nas redes.

O aumento do número de pessoas e dispositivos conectados nas redes, e o consequente aumento do número de aplicações e serviços sendo executados nesses ambientes, geram um aumento significativo no volume de informações em trânsito. Assim, os recursos disponíveis passam a ser mais disputados para realização da comunicação. Esse fator influencia diretamente na qualidade dos serviços percebida pelos usuários e sistemas finais, a qual é função direta de parâmetros de qualidade da rede, como velocidade e latência.

Diante de cenários de volume de demandas crescentes, redes corporativas têm mudado significativamente, buscando sempre novas tecnologias para fornecer serviços com a melhor qualidade possível. Entre outras, essas questões têm motivado o desenvolvimento de soluções tecnológicas nos mais diversos tópicos de redes de telecomunicações.

Ao longo dos últimos anos, o paradigma de Redes Definidas por Software (SDN, do inglês *Software Defined Networking*) tem estado em constante evolução. SDN consiste em uma arquitetura de rede com um controlador central, que comanda e gerencia os elementos comutadores, como *switches* (Nunes et al., 2014). Para isso, o controle existente em cada um dos dispositivos da rede é transformado em um único plano de controle, logicamente centralizado, o que possibilita o desenvolvimento de uma série de novas possibilidades de gerenciamento.

Em comparação com arquiteturas tradicionais, cuja gestão é usualmente baseada em hardware, redes SDN se baseiam em software e proporcionam uma abstração da infraestrutura de redes, permitindo o provisionamento e a gestão de recursos de forma centralizada. Além disso, devido à possibilidade de utilização de um equipamento de controle dedicado, é possível realizar a alocação de mais recursos computacionais para execução de aplicações de controle.

As provedoras de serviços em redes de computadores podem trabalhar em cenários de melhor esforço e de SLA (*Service Level Agreement*) (Verma, 2004). Na política de melhor esforço, geralmente empregada em tráfego interno de *data centers* corporativos, a gerência da rede se compromete a fazer o melhor esforço possível para atender os clientes, sem acordos de parâmetros prévios. Em cenários de SLA, parâmetros de qualidade da

comunicação têm seus valores previamente definidos em um contrato, os quais o provedor é obrigado a atender para o cliente em questão.

Para controle dos parâmetros de qualidade das redes, recursos como *switches*, roteadores e enlaces de comunicação devem ser gerenciados de forma adequada. Utilizar mais recursos em uma rede pode proporcionar melhor desempenho para os usuários finais. Contudo, ao mesmo tempo, em momentos de baixa demanda, esses recursos podem passar por estados ativos em operação ociosa, gerando ineficiência no consumo de energia.

## 1.2 Motivação e Justificativa

Apesar dos constantes avanços em tecnologias de telecomunicação, ainda há muitos problemas e desafios a serem resolvidos. No contexto de redes de computadores, tais questões se tornam mais evidentes à medida que as demandas por serviços se tornam cada vez maiores.

A pandemia do coronavírus causou, no primeiro semestre do ano de 2020, um grande crescimento do uso de Internet no Brasil, segundo dados divulgados pela Agência Nacional de Telecomunicações (Anatel, 2020). O tráfego na rede sofreu um aumento entre 40% e 50%. Com o aumento do consumo, também foi verificado um aumento no número de reclamações dos consumidores, devido a problemas nas redes de comunicação, como aumento das instabilidades, aumento do tempo de resposta das requisições e diminuição das taxas de *download*. Portais de reclamações na Internet, como o Reclame Aqui<sup>1</sup>, também registraram alta nos números de manifestações (Lavado, 2020).

Adicionalmente, um dos grandes desafios em redes corporativas consiste no processo de definição de rotas eficientes para a comunicação. Arquiteturas tradicionais tratam este processo com protocolos que operam de forma distribuída nos dispositivos de comutação presentes na rede. No caso de redes SDN, dada uma requisição de tráfego, o controlador precisa tomar a decisão de qual a rota mais adequada para o fluxo (uma sequência de pacotes que possuem a mesma origem e destino) percorrer na rede até chegar em seu destino, considerando requisitos de QoS (*Quality of Service*), balanceamento de carga, utilização eficiente de recursos, dentre outros.

O balanceamento de carga nas redes visa evitar problemas como congestionamento

---

<sup>1</sup> <https://www.reclameaqui.com.br/>

e aumento expressivo de latência e requer distribuição da carga entre os recursos possíveis. Já a eficiência energética busca evitar o uso excessivo e desnecessário dos recursos disponíveis através do acionamento somente daqueles que são utilizados em um dado momento da rede.

No que concerne ao consumo de energia elétrica, *data centers* tem responsabilidade estimada de 1% do total mundial. Várias análises presentes na literatura demonstram que a energia usada nestes ambientes dobrou na última década e que esse valor triplicará ou até quadruplicará na próxima década. Assim, à medida que a demanda por serviços de telecomunicações aumenta rapidamente, o mesmo deve ocorrer com o uso global de energia (Masanet et al., 2020), o que está relacionado com questões ambientais e econômicas (Maaloul et al., 2018). Na maioria do tempo, recursos de *data centers* são subutilizados quando em comparação com sua capacidade. Na operação das redes nesses ambientes, componentes são utilizados de 30% a 40% na maioria do tempo (Staessens et al., 2011). O consumo geral de energia, entretanto, permanece quase o mesmo para quantidades variáveis de cargas de tráfego. Uma solução prática para esse problema consiste em suspender ou desligar componentes subutilizados para baixo volume de tráfego. Minimizar a quantidade de componentes ligados para acomodar um determinado tráfego em uma rede, entretanto, degrada potencialmente o desempenho da mesma (Assefa and Ozkasap, 2019a).

Questões como eficiência energética e desempenho no contexto de redes SDN têm sido tópicos amplamente pesquisados pela comunidade científica. O desempenho de uma rede é função dos parâmetros de qualidade apresentados pela mesma, como vazão de dados e tempo de espera para obtenção de resposta na comunicação. A eficiência energética é alcançada pela utilização eficiente dos recursos disponíveis nas redes, como nós de comutação (*switches* e roteadores) e enlaces, visando eliminar consumo desnecessário de energia elétrica em decorrência de equipamentos operando de forma ociosa ou desnecessária. Vários trabalhos na literatura propõem soluções para aumentar a banda e diminuir a latência das redes, bem como realizar uso eficiente de recursos, através do desenvolvimento de aplicações para redes SDN (Shirmarz and Ghaffari, 2020; Assefa and Ozkasap, 2019b).

Em um sistema de Engenharia de Tráfego, balanceamento de carga e eficiência energética se caracterizam como objetivos conflitantes. Dada tal natureza entre os objetivos, melhor eficiência energética, obtida por meio do uso mínimo de recursos, pode ser alcançada em detrimento de desempenho, em termos de parâmetros de qualidade

(Fernández-Fernández et al., 2017). Pela natureza conflitante dos objetivos, tornam-se necessárias informações de preferências para a escolha de soluções no processo de otimização, que em um cenário real podem ser baseadas, por exemplo, no histórico diário de uso das redes e em diferentes momentos de bandeiras tarifárias para consumo de energia. Bandeiras tarifárias<sup>2</sup> são um mecanismo estabelecido no Brasil pela Agência Nacional de Energia Elétrica (ANEEL) para sinalizar aos consumidores as condições de geração de energia elétrica no Sistema Interligado Nacional (SIN), por meio da cobrança de valor adicional às tarifas de energia, e equalizar a parcela de custos variáveis relativa à aquisição de energia elétrica, cuja flutuação é governada por variáveis operativas do SIN.

Para contornar problemas relacionados com as questões colocadas, um *data center* pode atingir melhor desempenho, aprimorando a experiência dos usuários finais dos serviços, de forma eficiente em termos de consumo energético, por meio do desenvolvimento de aplicações de Engenharia de Tráfego para alocação de recursos no processo de roteamento.

## 1.3 Objetivos

No contexto de otimização de rotas em redes SDN, o presente trabalho propôs atender os objetivos geral e específicos, descritos nas subseções seguintes.

### 1.3.1 Geral

Em busca de uma solução para otimização de roteamento em redes SDN, o presente trabalho propôs, de forma geral, o desenvolvimento e a avaliação de modelos de otimização baseados em algoritmos evolucionários, visando obter melhoria de desempenho por meio de balanceamento de carga e melhoria de eficiência energética através do uso mínimo de recursos para acomodação do tráfego.

### 1.3.2 Específicos

Foram definidos os seguintes objetivos específicos:

- Desenvolver modelos de otimização de roteamento para redes SDN, visando obtenção de soluções para rotas com uso mínimo de recursos e balanceamento de carga, para

---

<sup>2</sup> Procedimentos de Regulação Tarifária (PRORET) - Submódulo 6.8

alcançar melhor desempenho e eficiência energética em abordagens convencionais, com operação em melhor esforço, e com SLA para serviços, em função de parâmetros de QoS;

- Definir, avaliar e comparar diferentes meta-heurísticas baseadas em algoritmos evolucionários na resolução do problema de otimização de rotas;
- Desenvolver e utilizar processos de simulação e emulação de redes para verificar os ganhos propostos com emprego das soluções desenvolvidas.

## 1.4 Estrutura do Texto

Além desta introdução, o texto da presente dissertação está estruturado em mais quatro capítulos brevemente descritos a seguir.

No Capítulo 2 é apresentado o referencial teórico, onde são introduzidos os conceitos fundamentais empregados no desenvolvimento deste trabalho e é feita uma revisão crítica do estado da arte da área.

O Capítulo 3 detalha os materiais e os métodos adotados para atingir os objetivos propostos, incluindo algoritmos, testes e ferramentas utilizadas.

O Capítulo 4 traz os resultados de otimização obtidos para as diferentes meta-heurísticas utilizadas, bem como os resultados obtidos nos processos de emulação e simulação de ambientes de redes. Também são aqui colocadas as discussões e considerações acerca desses resultados.

Por fim, no Capítulo 5, são apresentadas as conclusões obtidas com o desenvolvimento do presente trabalho. É feita uma passagem de forma geral por todo o desenvolvimento, os resultados e as contribuições geradas. Também são abordadas as limitações e as possibilidades para trabalhos futuros.

# Capítulo 2

## Referencial Teórico

O presente capítulo apresenta um referencial teórico acerca dos conceitos empregados no trabalho, e tem por objetivo fornecer uma bagagem teórica para possibilitar um melhor entendimento acerca do mesmo.

### 2.1 Engenharia de Tráfego

Engenharia de Tráfego (TE, do inglês *Traffic Engineering*) é um conceito que pode ser definido como o emprego de princípios científicos e tecnológicos para avaliar e aprimorar o desempenho de redes, através de modelagem, caracterização, medição e controle de tráfego ([Awduche et al., 1999](#)).

O aprimoramento das redes por TE pode ser realizado pelo cumprimento de uma série de objetivos através do gerenciamento da rede, que envolvem ações como planejamento de capacidade, controle de roteamento e gerenciamento de recursos. O gerenciamento de tráfego pode se dar em condicionamento de tráfego na entrada da rede e em gerenciamento de filas e controle de escalonamento nos roteadores ([Awduche et al., 2002](#)).

Dentre os objetivos que se buscam alcançar com a aplicação de TE em redes, estão:

- Garantia de qualidade de serviço necessária para diferentes tipos de aplicações, que é função de parâmetros de desempenho da rede;
- Garantia de balanceamento de carga de fluxos entre os enlaces disponíveis, evitando problemas como congestionamento e erros de colisão de dados;

- Economia e eficiência no uso de energia com acionamento somente de recursos necessários para atender as demandas em um dado momento da rede;
- Melhoria da confiabilidade da rede, entendida como tolerância a falhas e erros, como exemplo quedas de enlace.

Os aspectos de otimização da engenharia de tráfego podem ser vistos a partir de uma perspectiva de controle, podendo ser pró-ativo ou reativo. No caso pró-ativo, o sistema de controle toma medidas preventivas, com base no estado atual, para evitar estados desfavoráveis futuros previstos na rede. Também pode ser necessária uma ação para induzir um estado mais desejável no futuro. No caso reativo, o sistema de controle responde de forma adaptativa aos eventos que já ocorreram na rede ([Awduche et al., 2002](#)).

Um fluxo de dados, para entrega de um serviço pela rede, percorre um caminho composto por enlaces e nós comutadores, cujos estados implicam diretamente na qualidade percebida pelo usuário do serviço.

A qualidade de serviço pode ser entendida como um requisito de recursos de um fluxo de dados que trafega na rede. Um serviço de *streaming* de vídeo, por exemplo, pode solicitar uma largura de banda mínima e uma variação de latência máxima para garantir uma boa experiência ao usuário final. De acordo com [Chen et al. \(2004\)](#), as seguintes métricas são utilizadas como principais parâmetros para medição de qualidade de serviço:

- Tempo de resposta: tempo decorrido entre o envio da requisição de um usuário e a obtenção de uma resposta;
- Atraso (latência): tempo decorrido entre o envio do primeiro bit de um pacote de dados por um transmissor e o recebimento do mesmo bit pelo receptor na comunicação;
- *Jitter*: variação do atraso de pacotes de um mesmo fluxo de dados na comunicação;
- Vazão: a taxa de transmissão de dados em uma rede, dado em unidade de bits por segundo;
- Taxa de perda: percentual de pacotes perdidos na rede durante a comunicação.

## 2.2 Redes Definidas por Software

Redes Definidas por Software (SDN, da sigla em inglês) consiste em um paradigma de redes que propõe a utilização de um mecanismo de controle externo aos dispositivos de comutação para gerenciamento e controle das redes, o que é diferente de redes tradicionais, as quais utilizam os próprios dispositivos de comutação para implementação e execução das lógicas de controle e gerenciamento (Nunes et al., 2014).

### 2.2.1 Arquitetura

O paradigma SDN é um paradigma de redes que propõe uma arquitetura com divisão em camadas, e seu ponto principal consiste na viabilização de uma visão centralizada da rede, possibilitando que aplicações se comportem como o sistema operacional da mesma. Basicamente, a arquitetura é definida em três camadas, conforme Figura 1.

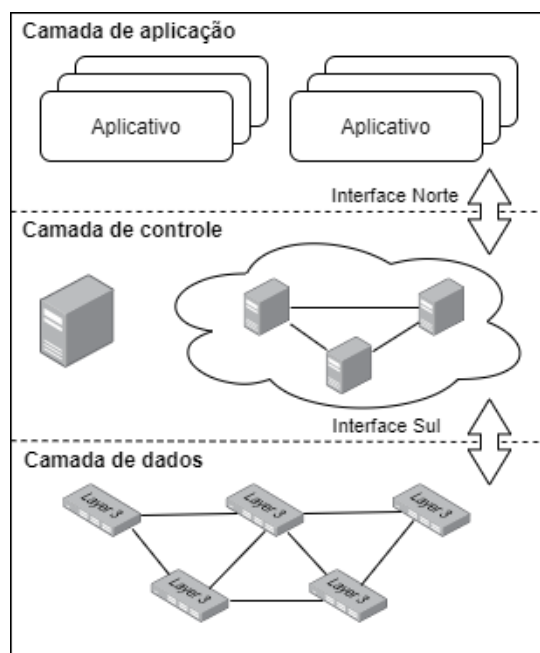


Figura 1 – Arquitetura de uma rede SDN.

A camada de dados consiste no conjunto de elementos responsáveis pelo encaminhamento de pacotes com base em comportamentos definidos remotamente. Esses comportamentos podem ser especificados pela aplicação em execução no topo do controlador e acionados de acordo com as informações extraídas no nível do pacote. A camada de controle é responsável pelo controle lógico centralizado, e é onde atua o controlador SDN. A camada de aplicação consiste em aplicativos finais que definem comportamentos para os

elementos de encaminhamento, invocando serviços na camada de controle. A comunicação entre as aplicações e controlador é feita pela interface norte, usando, por exemplo, uma interface REST (Fielding and Taylor, 2002), e a comunicação entre o controlador e os elementos de comutação se dá através de protocolos como o OpenFlow (McKeown et al., 2008).

Dentre as aplicações desenvolvidas em engenharia de tráfego utilizando redes SDN, podem-se citar diferentes questões importantes, como segurança, manutenção de qualidade de serviço oferecida, priorização de tráfego, virtualização, dentre outras vertentes.

Com o advento das arquiteturas SOA (*Service-Oriented Architecture*), a qualidade de serviços tem se tornado um aspecto importante para os consumidores. Ao apresentar uma demanda, um consumidor faz uma negociação com um fornecedor para entrar em um acordo de qualidade para prestação do serviço que, no caso de redes de computadores, estão relacionados com os parâmetros de QoS. Este acordo é chamado de SLA (*Service Level Agreement*). Os parâmetros relacionados com SLA mudam continuamente nas redes e precisam ser monitorados com objetivo de fazer cumprir o contrato, desde a admissão até o final do serviço (Keller and Ludwig, 2003).

No cenário de aplicações em redes SDN, um cliente pode fazer uma requisição para uma demanda através da interface norte de um controlador para solicitar o serviço, e este precisa, considerando o estado atual da rede e a disponibilidade dos recursos solicitados, admitir ou não o fluxo de dados da demanda solicitada, definindo uma rota para o fluxo em caso de aceite.

### 2.2.2 Protocolo OpenFlow

O protocolo OpenFlow (McKeown et al., 2008) é um protocolo padrão de comunicação entre os controladores SDN e os dispositivos presentes na camada de dados. O protocolo é padronizado e mantido pela Open Networking Foundation (ONF) visando a garantia de interoperabilidade entre dispositivos de rede e software de controle de diferentes fornecedores (Open Network Foundation, 2012b).

Um *switch* OpenFlow é um *switch* que implementa o protocolo OpenFlow em seu funcionamento e emprega técnicas de SDN para encaminhar pacotes em uma rede. O dispositivo apresenta uma ou mais tabelas de fluxo e uma tabela de grupo, que realizam

consultas nos pacotes e encaminhamento, e um canal OpenFlow para conexão com um controlador externo. A Figura 2 apresenta o esquema de um *switch* OpenFlow e seus componentes básicos.

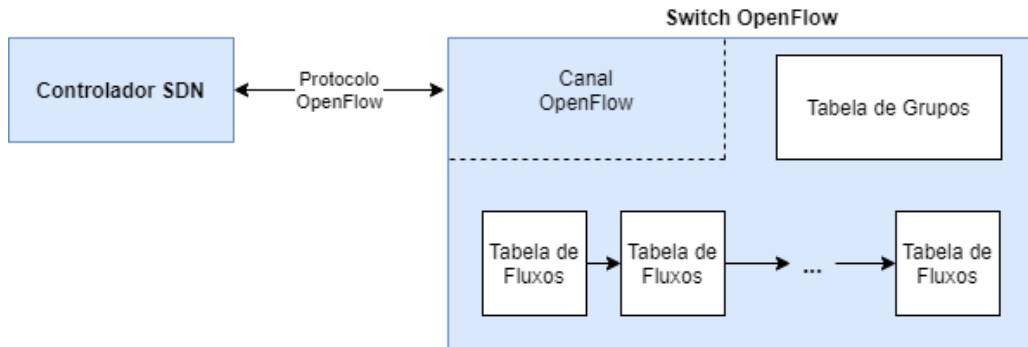


Figura 2 – Componentes básicos de um *switch* OpenFlow.

Através do protocolo OpenFlow, o controlador pode adicionar, atualizar e excluir entradas nas tabelas de fluxos. Cada tabela de fluxo possui informações para *match* com pacotes, ações, estatísticas, prioridade e tempo de expiração, como exemplificado no diagrama da Figura 3.

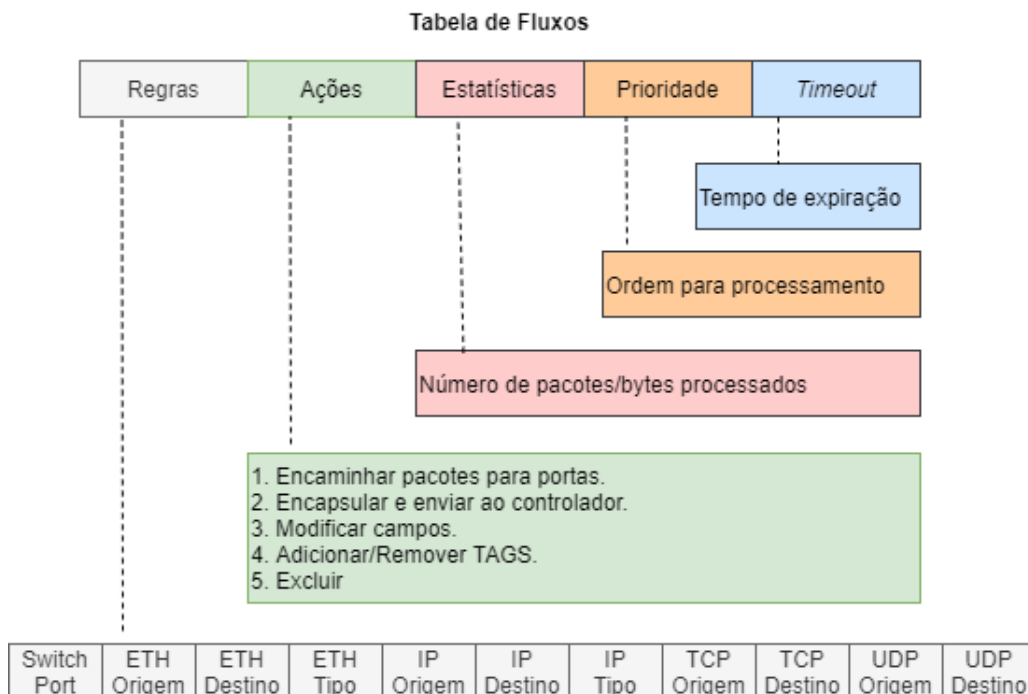


Figura 3 – Esquema das tabelas de fluxo OpenFlow.

A tabela de grupos apresenta um conjunto de grupos que permite a execução de ações específicas, possibilitando métodos adicionais de encaminhamento, como *Multicast* e *Broadcast*.

As mensagens OpenFlow se dividem basicamente em três grupos:

- Mensagens *Controller-to-Switch*: essas mensagens são iniciadas pelo controlador e utilizadas para gerenciar e/ou inspecionar os *switches*;
- Mensagens Assíncronas: são iniciadas pelo *switch* e utilizadas para atualizar o controlador sobre ocorrência de eventos da rede e mudanças no estado do *switch*;
- Mensagens Simétricas: são iniciadas tanto pelo controlador quanto pelo *switch*.

A Tabela 1 apresenta as mensagens trocadas entre controladores e *switches* e seus respectivos tipos.

Tabela 1 – Mensagens OpenFlow.

<b>Tipo</b>	<b>Mensagem</b>
Mensagens <i>Controller-to-Switch</i>	<i>Handshake</i>
	<i>Switch Configuration</i>
	<i>Flow Table Configuration</i>
	<i>Modify State Messages</i>
	<i>Multipart Messages</i>
	<i>Queue Configuration Messages</i>
	<i>Packet-Out Message</i>
	<i>Barrier Message</i>
	<i>Role Request Message</i>
Mensagens Assíncronas	<i>Set Asynchronous Configuration Message</i>
	<i>Packet-In Message</i>
	<i>Flow Removed Message</i>
Mensagens Simétricas	<i>Port Status Message</i>
	<i>Error Message</i>
	<i>Hello</i>
	<i>Echo Request</i>
	<i>Echo Reply</i>
	<i>Experimenter</i>

O protocolo OpenFlow encontra-se atualmente na versão 1.5.1. Para realização do presente trabalho, foi utilizado a versão 1.3 do protocolo ([Open Network Foundation, 2012a](#)), por ser considerada uma versão estável e mais utilizada em trabalhos publicados de Redes Definidas por Software.

## 2.3 Otimização Multiobjetivo

A otimização multiobjetivo é uma área de tomada de decisão de múltiplos critérios, onde se preocupa com problemas de otimização matemática envolvendo duas ou mais funções objetivo, simultaneamente. Em um problema de otimização multiobjetivo, não existe uma única solução com os melhores valores para cada objetivo. Geralmente, as funções objetivo são conflitantes, fazendo-se existir um número de soluções ótimas de Pareto. Uma solução é classificada como não-dominada, ótima de Pareto ou Pareto eficiente se nenhuma das funções objetivo puder ser melhorada quanto a seu valor sem degradar alguns dos outros valores dos demais objetivos. Sem informações adicionais de preferência subjetiva, todas as soluções ótimas de Pareto são consideradas igualmente boas. Em um problema de otimização multiobjetivo, o objetivo consiste em encontrar uma única solução dentro de um conjunto representativo de soluções ótimas de Pareto que expresse o melhor compromisso (*trade-off*) para satisfazer os diferentes objetivos, considerando preferências de um tomador de decisão (Abraham and Jain, 2005).

A Figura 4 apresenta um exemplo de conjunto de soluções em um espaço de objetivos bidimensional, onde busca-se minimizar o custo de cada uma das funções objetivo. Neste conjunto, é possível verificar visualmente que nenhum ponto apresenta custos menores que outro ponto para ambas funções, ou seja, as soluções são não dominadas e pertencem a um mesmo conjunto Pareto.

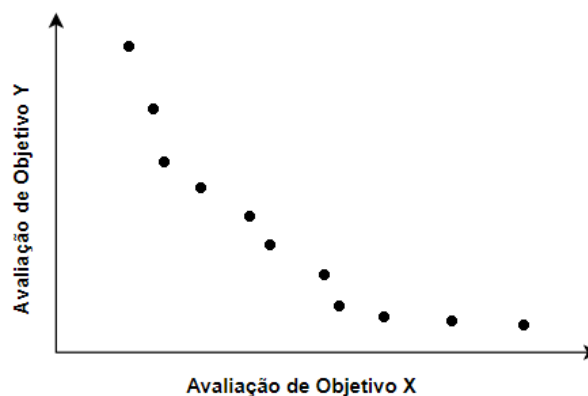


Figura 4 – Exemplo de conjunto Pareto de soluções.

Há diferentes maneiras para se tratar as preferências (ou pesos) para os objetivos em um processo de otimização multiobjetivo. O problema pode ser modelado com inserção

de preferências a posteriori, onde o algoritmo é configurado para convergir para uma fronteira Pareto aproximada, inserção a priori, onde o problema multiobjetivo é transformado em problema mono-objetivo, criando uma função que combine todas as funções objetivo atribuindo pesos de preferência, ou então através de inserções de preferências parciais, onde o problema é modificado e o resultado é uma fronteira Pareto tendenciosa. A Figura 5 apresenta um fluxograma das diferentes abordagens de algoritmos para obtenção de uma solução (Coello et al., 2007).

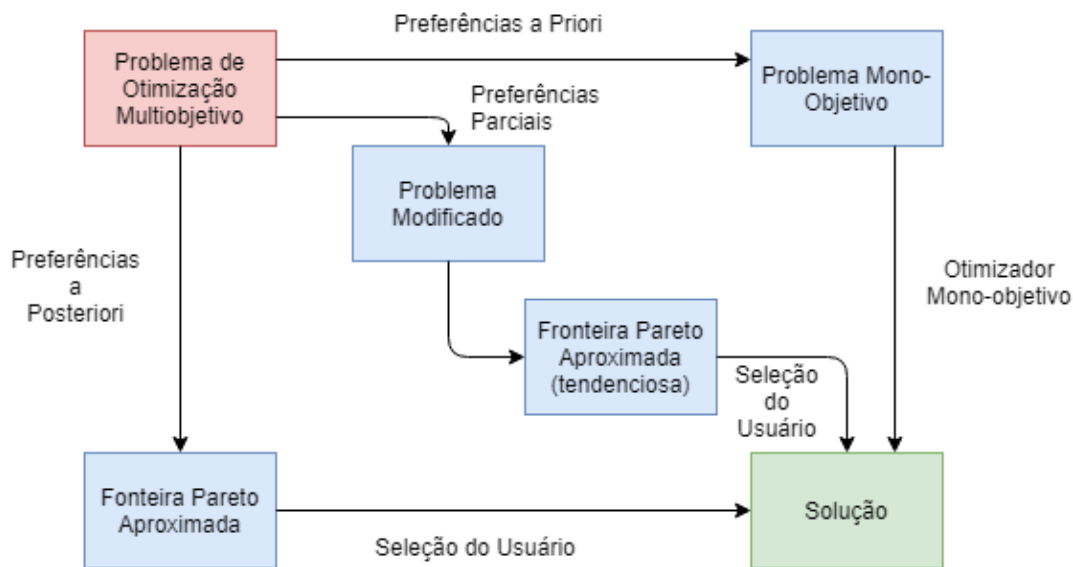


Figura 5 – Abordagens de otimização de um problema multiobjetivo.

Para obtenção de uma fronteira Pareto aproximada, o método de inserção de preferências a posteriori busca fazer a manutenção de diversidade das soluções, combinando e priorizando soluções não dominadas. Outra abordagem também utilizada na literatura consiste na execução de múltiplas instâncias do processo de otimização mono-objetivo com utilização de métodos escalares e/ou de agregação (Collette and Siarry, 2004). Alguns desses métodos são:

- Método da soma ponderada (*weighted-sum*): o problema multiobjetivo é transformado em mono-objetivo através de uma soma ponderada dos valores dos objetivos;
- Método  $\epsilon$ -restrito: um dos objetivos é escolhido para otimização e os demais são tratados como restrições de desigualdade, que com diferentes valores apresentam diferentes soluções e geram uma fronteira Pareto aproximada;

- Método  $\epsilon$ -restrito híbrido: nesta abordagem, uma soma ponderada dos objetivos é escolhida para otimização, e os objetivos são tratados como restrições de desigualdade, restringindo a região de interesse.

### 2.3.1 Algoritmos Evolucionários

Algoritmos evolucionários (EA, da sigla em inglês) são algoritmos que tentam resolver problemas de otimização imitando os processos da teoria da evolução de Darwin.

Em um EA, um conjunto inicial de soluções para o problema é criado. A partir de então, tais soluções passam a competir continuamente entre si para descobrir áreas ideais do espaço de busca. Com a execução do algoritmo, é esperado que com o tempo, as melhores soluções evoluam para alcançar uma solução ideal (Jones, 1998).

Basicamente, o processo realizado nas estratégias evolucionárias consiste na realização de uma série de iterações, denominadas gerações, onde são feitas operações de seleção, recombinação e mutação para geração de novas soluções e novas populações. A lista abaixo apresenta uma sequência genérica de passos realizados em uma estratégia evolucionária.

1. Uma população atual de  $\lambda$  indivíduos é criada aleatoriamente;
2. É feita atribuição de valor de *Fitness* para cada um dos  $\lambda$  indivíduos;
3. Uma nova geração de  $\lambda$  indivíduos é gerada por recombinação da população atual;
4. É aplicada mutação na nova geração;
5. É feita atribuição de valor de *Fitness* para cada um dos  $\lambda$  indivíduos da nova geração;
6. Uma nova população de tamanho  $\lambda$  é criada com base nas soluções da geração corrente e geração criada;
7. A nova população passa a ser a população atual;
8. Finaliza a execução caso as condições de parada forem alcançadas, ou volta para o passo do item 3 caso contrário.

## 2.4 Problema de Otimização de Rotas

No processo de definição de rotas para fluxos de dados em redes SDN, vários objetivos podem ser considerados. Os objetivos contemplados no presente trabalho foram:

- Balanceamento/distribuição de carga ao longo de um conjunto de enlaces e nós comutadores da rede disponíveis, visando aliviar a carga em cada recurso para mitigar efeitos da disputa dos fluxos pelos recursos com fim de melhorar o desempenho e consequentemente evitar problemas com congestionamento e erros de colisão de dados;
- Utilização mínima de recursos, considerando manter o mínimo de recursos consumindo potência energética em estado de funcionamento ativo, e consequentemente provendo economia de energia.

Em uma rede corporativa que utiliza uma política de melhor esforço, mais recursos podem ser alocados ao custo de um maior consumo de potência dos equipamentos em uso. Assim, dispondo de tais recursos, um tomador de decisão precisa decidir sobre o quanto de recursos alocar para atender as demandas presentes em determinado momento da rede. Para a tomada de decisão, devem ser considerados parâmetros de apoio, os quais podem ser a caracterização da bandeira tarifária de energia do horário, quantidade de usuários na rede, dentre outros.

Já em uma rede com SLA, cada demanda apresenta uma restrição de qualidade sustentada por um contrato e portanto deve ser atendida. Assim, o problema pode ser tratado como mono-objetivo, onde busca-se utilizar o mínimo de recursos necessários para atender exatamente as restrições de cada demanda.

Uma questão importante no problema de roteamento é que, dependendo da arquitetura do sistema implementado no controlador, a otimização deve ser calculada e processada em tempo real, pois as aplicações requerem a utilização da rede e precisam rapidamente fluir para o destino. Assim, um sistema de otimização de rotas em redes SDN precisa ser projetado pensando na rapidez em se alcançar uma solução plausível para o problema.

## 2.5 Trabalhos relacionados

Nesta seção, é apresentada uma revisão de literatura acerca dos temas abordados no trabalho com objetivo de definir o estado da arte. Nesse sentido, os trabalhos são categorizados em três classes, sendo elas estratégias de balanceamento de carga, estratégias de uso eficiente de recursos e otimização com múltiplos critérios.

Distribuir a carga de comunicação e processamento em redes ao longo dos recursos disponíveis é uma estratégia de gerência de ativos que pode gerar benefícios em termos de desempenho, pois a sobrecarga degrada parâmetros de qualidade.

Em (Ali et al., 2018), é proposta uma abordagem para distribuir carga entre servidores de um *data center*, utilizando o algoritmo de Dijkstra para selecionar caminhos menos sobrecarregados, evitando problemas como congestionamento e subutilização de servidores. Experimentos foram realizados em uma topologia *fat tree* e mostraram melhora nos parâmetros de vazão e latência após balanceamento de carga.

Em (Dewanto et al., 2018), é proposta uma solução em rede SDN utilizando ECMP (*Equal Cost Multipath*), com análise de largura de banda em cada enlace antes da atribuição de caminhos para os fluxos, objetivando minimizar efeitos de congestionamento e atingir melhor desempenho na rede. No trabalho, foi utilizado o algoritmo Dijkstra para cálculo dos caminhos. Para experimentos, foi utilizado o controlador Ryu. Resultados mostraram que o modelo proposto gera, em cenários de congestionamento, aumento de vazão e diminuição de perdas de pacote e atraso. Em cenários sem congestionamento, o processamento com Dijkstra gera uma pequena perda em relação aos parâmetros de vazão e atraso.

Em (Chakravarthy and Amutha, 2019), é proposto um modelo de busca entre os melhores caminhos entre origem e destino na rede. Em uma primeira etapa, é utilizado o algoritmo de Floyd para identificar os melhores caminhos. Em seguida, é utilizado um algoritmo para determinar os melhores caminhos, dentre aqueles obtidos na etapa anterior, considerando parâmetros de qualidade coletados. Nos experimentos, foram utilizados o Mininet (Handigol et al., 2012) em conjunto com o controlador ONOS (Berde et al., 2014). Os resultados mostraram que o modelo consegue determinar os melhores caminhos de forma dinâmica, e comparando com o modelo de menor caminho, há uma diminuição significativa dos RTTs (*Round Trip Time*) observados.

Em (Rupani et al., 2020), é proposto um modelo de predição baseado em redes neurais para estimar e escolher caminhos entre origem e destino, com menor carga para fluxos admitidos na rede. O modelo de rede neural utilizado tem como entrada a banda, latência, número de saltos e perda de pacotes para prever sobre a sobrecarga. No trabalho, foram feitos testes com redes contendo 8 e 16 hosts. Os resultados mostraram melhoras nos parâmetros de vazão e latência, onde os ganhos se tornam mais significativos em ambientes com mais nós, e conseqüentemente com mais tráfego.

A eficiência energética em um *data center* pode ser obtida por meio da implementação de estratégias para minimizar a potência consumida em equipamentos, o que pode ser alcançado com a alocação eficiente de recursos nas redes. Por ser um eixo relevante de otimização, trabalhos na literatura têm despendido esforços neste contexto.

Markiewicz et al. (2014) propõe uma solução de economia de energia para habilitar o mínimo de recursos possíveis em uma rede SDN, baseada nas estratégias SPF (*Shortest Path First*), LPF (*Longest Path First*), SDF (*Smallest Demand First*) e HDF (*Highest Demand First*). No trabalho, foi feita uma modelagem para uma abordagem em programação linear inteira e, em seguida, foi apresentada uma heurística com as diferentes abordagens. Com os experimentos realizados, foi possível verificar que em momentos de pouca demanda, como à noite, o consumo de energia pode ser reduzido de maneira significativa, sendo de até 45%.

No trabalho de Al-Musawi and Al-Khatib (2019), é proposta uma heurística para monitorar a rede com fim de alocar recursos de forma eficiente, garantindo ativação somente de elementos necessários para atender o estado da rede e deixando os elementos ociosos desativados. No desenvolvimento do trabalho, foram realizados experimentos em uma rede com máquinas virtuais, cujos resultados mostraram diminuição significativa de consumo de energia, em comparação com outras técnicas na literatura.

Assefa and Ozkasap (2019a) propõem em seu trabalho uma métrica de eficiência energética aplicável ao contexto de redes SDN e um método de programação inteira, buscando maximizar a eficiência energética da rede. No trabalho, experimentos foram executados, cujos resultados mostraram que o modelo proposto pode gerar economia de energia de até 35%, em comparação com os métodos SPF, NSP (*Next Shortest Path*), e NMU (*Next Maximum Utility*).

Em um ambiente de *data center*, um gestor de ativos busca sempre fazer a melhor gestão possível. Melhor desempenho e eficiência energética são, dentre outras questões, objetivos importantes e, muitas das vezes, conflitantes entre si. Nesse sentido, a literatura também apresenta trabalhos com construção de modelos de otimização multiobjetivo, visando alcançar soluções de operação em redes que apresentem bom desempenho e sejam, ao mesmo tempo, energeticamente eficientes.

Fernández-Fernández et al. (2017) apresentam em seu trabalho uma estratégia de roteamento baseada no SPEA2, com fim de otimizar a eficiência energética da rede considerando também as restrições de QoS (*Quality of Service*). O modelo desenvolvido no trabalho consiste em minimizar a utilização de enlaces na rede e o custo relacionado com os caminhos escolhidos para as demandas, o que está relacionado com a qualidade requerida e disponível. No trabalho, o modelo foi testado em uma rede de topologia Abilene, utilizando SNDLib (Orlowski et al., 2010).

Em (Jaglarz et al., 2020), é proposto um método de otimização de múltiplos critérios com foco em escalabilidade, utilizando decomposição de Lagrange aprimorado por sequências ergódicas, onde soluções são obtidas com base em pesos atribuídos para os critérios. Testes de otimização com diferentes topologias oriundas do SNDLib mostraram a eficiência e escalabilidade do modelo proposto.

No presente trabalho foi proposta uma modelagem de dois problemas de otimização. Em um cenário, é considerado contextos com SLA por definição de parâmetros de qualidade mínima e, em outro, é considerado política de melhor esforço, onde os recursos são explorados sob demanda. Neste caso, um conjunto Pareto aproximado de soluções é obtido para que regras de tomada de decisão possam escolher ou permutar soluções com base em critérios previamente definidos.

Os assuntos discutidos e os resultados apresentados no presente trabalho expandem uma visão para uma série de aplicações, com alocação dinâmica de recursos na rede com base no estado de uso, gerando ganho nos dois universos de análises, sendo eles a sustentabilidade por meio do consumo de energia reduzido devido alocação eficiente de recursos, bem como a melhoria da experiência dos usuários, por meio da exploração dos recursos necessários para obtenção de bom desempenho. É realizada uma análise dos processos de otimização relacionados com roteamento e suas implicações em cenários reais de rede, evidenciando

os potenciais ganhos em cada um dos casos trabalhados.

## 2.6 Resumo

Este capítulo descreveu os conceitos fundamentais e as características do problema de interesse para realização do entendimento do trabalho. Além disso, também foi feita uma análise de trabalhos relacionados. A otimização de rotas visando uso eficiente de recursos e melhoria de desempenho é entendida como uma questão tratada em engenharia de tráfego. Redes SDN permitem a implementação de uma rede com um controlador centralizado, que através do protocolo OpenFlow, faz a gestão de tráfego e recursos na mesma. No contexto e otimização multiobjetivo, dada a natureza conflitante entre minimização da utilização de recursos e balanceamento de carga, algoritmos evolucionários são uma alternativa e foram utilizados para solução dos problemas tratados no presente trabalho.

# Capítulo 3

## Materiais e Métodos

Este capítulo apresenta os materiais e métodos utilizados para a realização do trabalho. O desenvolvimento se deu em três etapas principais. A primeira foi a modelagem do problema para ambos os casos trabalhados, melhor esforço e SLA. A segunda consistiu no estudo para avaliação da eficiência de diferentes técnicas computacionais para a solução do problema, tanto em termos de qualidade das soluções quanto de desempenho, sendo este segundo ponto compreendendo uma comparação entre tempos de execução e a utilização de métodos escalares e de Pareto. Por fim, foram feitas simulações computacionais e emulações de cenários reais de redes, para avaliar as premissas levantadas nos modelos desenvolvidos.

### 3.1 Definição do Problema

O problema consiste em realizar a otimização do processo de definição de rotas em ambiente SDN, visando obter soluções com o melhor desempenho possível, através da prevenção de sobrecarga de fluxos em recursos específicos da rede, e, ao mesmo tempo, com uso mínimo necessário de recursos, visando diminuir o consumo de energia relacionado com a ativação desnecessária de nós comutadores e enlaces.

Dada uma topologia de rede, com  $m$  comutadores (*switches*),  $n$  hosts (terminais conectados na rede) e  $k$  demandas de fluxos a serem admitidas, define-se os conjuntos  $S$  de comutadores e  $H$  de hosts, conforme denotado nas equações (3.1) e (3.2), respectivamente.

$$S = \{ s_i \mid 1 \leq i \leq m \} \quad (3.1)$$

$$H = \{ h_i \mid 1 \leq i \leq n \} \quad (3.2)$$

Dispondo dos nós clientes e comutadores na rede, é necessário adotar uma representação para os enlaces entre comutadores na topologia. Para tal, é definida uma matriz  $T$ , definida em (3.3), onde  $t_{ij}$  deve ser definido como 1 se  $s_i$  conecta  $s_j$  ou  $i$  é igual a  $j$ , e  $-1$  caso contrário.

$$T [ t_{ij} ]_{m \times m} \quad (3.3)$$

Na matriz  $T$ , não são considerados os enlaces entre hosts e comutadores. Para tal, considera-se que cada host  $h_i \in H$  está conectado em um comutador  $s_j \in S$ , sendo esta conexão obtida através da relação  $c$  (3.4), que é sustentada por uma matriz constante  $m \times n$ .

$$h \longrightarrow c(h), \quad c : H \longrightarrow S \quad (3.4)$$

Dada uma demanda por tráfego entre dois hosts  $h_i$  e  $h_j$ , faz-se necessário determinar um caminho entre os comutadores  $c(h_i)$  e  $c(h_j)$ . Assim, define-se um conjunto  $F$  de pares de comutadores de origem e destino determinados pelas demandas a serem admitidas na rede, conforme (3.5).

$$F [ f_i ]_k, \quad f_i \in S^2 \quad (3.5)$$

Para uma instância do problema, uma solução  $Z$  consiste em um conjunto de listas de nós intermediários para cada fluxo, sendo estas um subconjunto do complemento de nós comutadores em relação aos nós de origem e destino, conforme (3.6).

$$Z [ z_i ]_k, \quad z_i \subseteq S - f_i \quad (3.6)$$

Para uma solução  $Z$ , é definida uma matriz  $U$  (3.7), que determina a quantidade de fluxos que passam em cada enlace entre comutadores na rede, onde os índices representam

os nós de origem e destino.

$$U = [u_{ij}]_{m \times m} \quad (3.7)$$

As estruturas até aqui definem uma instância para o problema de roteamento em uma rede e uma solução. No problema de otimização, é necessária a definição dos objetivos, os quais variam com o contexto abordado, sendo estes a adoção da política de melhor esforço na rede e a adoção de SLA com base em parâmetros de QoS.

### 3.1.1 Otimização em Melhor Esforço

No contexto de operação de melhor esforço, não é possível se dizer que, visando uma solução que seja energeticamente eficiente e apresente bom desempenho, haja uma solução ótima. Naturalmente, os objetivos são conflitantes, uma vez que ao utilizar mais recursos para atender as demandas, o desempenho tende a ser melhor, pois a concorrência por recursos é menor e os efeitos de congestionamento são reduzidos. Por outro lado, utilizar mais recursos significa aumentar o consumo energético.

Neste caso, as funções objetivo do problema têm como base de cálculo a matriz  $U$  definida anteriormente em (3.7). O primeiro objetivo consiste na maximização do balanceamento de carga. Para mensurar o nível de balanceamento de carga, foi utilizada a dispersão dos valores de utilização de cada enlace por cada fluxo de comunicação admitido na rede, que pode ser obtida pela variância da matriz. Testes realizados mostraram que a variância da matriz de utilização de enlaces para um determinado conjunto de fluxos admitidos na rede tende a diminuir à medida que esses fluxos se espalham, pois a média permanece e os desvios são diminuídos. O valor da variância é igual a zero quando a rede apresenta um estado de equidade de utilização de todos os enlaces e tende a aumentar à medida que os fluxos passam a se concentrar em enlaces específicos. Considerando este comportamento, é esperado que o valor da variância da matriz  $U$  seja minimizado. O segundo objetivo foi definido como minimização do número de enlaces utilizados, visando economia de energia através do uso de poucos recursos. Assim, para mensurar o nível de uso dos recursos, foi utilizado a quantidade de enlaces utilizados na rede. Nesse sentido, os

objetivos do processo de otimização são apresentados em (3.8) e (3.9).

$$\arg \min \text{VAR}(U) \quad (3.8)$$

$$\arg \min \sum_{i=1}^m \sum_{j=i+1}^m \begin{cases} 1 & u_{ij} > 0 \\ 0 & \text{outro caso} \end{cases} \quad (3.9)$$

Nesta abordagem, realizar o processo de otimização tem por objetivo gerar um conjunto Pareto aproximado de soluções. Em operação, torna-se necessário definir a prioridade dos objetivos, o que pode variar em mudanças de cenários e implicar no intercâmbio de soluções obtidas.

### 3.1.2 Otimização em SLA

Para abordagem com SLA, cada demanda deve estar associada com um conjunto de parâmetros de qualidade de serviço. Conforme discutido no Capítulo 2, existem vários parâmetros que influenciam na QoS. No contexto das restrições, alguns desses parâmetros são caracterizados como limites inferiores, como o mínimo de banda necessária para determinado serviço. Já outros parâmetros são caracterizados como limites superiores, como os valores máximos de *jitter* e atraso admitidos para a aplicação.

Associado a um conjunto  $F$  que determina as demandas entre os hosts, conforme a Equação (3.5), são definidas duas matrizes,  $Q_{inf}$  e  $Q_{sup}$ , contendo respectivamente os valores dos  $a$  parâmetros considerados limites inferiores (3.10) e os  $b$  valores considerados limites superiores (3.11), para as  $k$  demandas a serem admitidas na rede.

$$Q_{inf} = [q_{ij}]_{k \times a} \quad (3.10)$$

$$Q_{sup} = [q_{ij}]_{k \times b} \quad (3.11)$$

Como as demandas apresentam restrições de qualidade para admissão na rede, é necessário dispor de uma matriz tridimensional  $W$  que apresenta os valores correntes para

cada um dos parâmetros de QoS avaliados em determinado momento da rede para cada enlace entre comutadores disponíveis na topologia, conforme (3.12).

$$W = [w_{ijk}]_{m \times m \times (a+b)} \quad (3.12)$$

Neste contexto, o problema em questão passa a ser caracterizado como mono-objetivo, onde busca-se minimizar o uso de recursos na rede, considerando as restrições de QoS. Conforme discutido em (Carvalho and Errico, 2013), uma possível abordagem consiste em utilizar um algoritmo genético, definindo como objetivo uma função de agregação entre o objetivo e as restrições, atribuindo pesos para as mesmas. Assim, as violações de restrição passam a implicar em uma penalidade na aptidão das soluções.

Dispondo de uma solução  $Z$  (3.6) e uma matriz  $U$  (3.7) relacionada, é possível então determinar o objetivo e as restrições no cenário de SLA. Dada uma solução  $s$ , são definidas três funções:  $R_u(s)$ , que determina o percentual de utilização de enlaces, conforme (3.9),  $R_{Qinf}(s)$ , que determina o percentual de extrapolamento para os parâmetros de QoS de valores caracterizados como limites inferiores para a solução  $s$ , e  $R_{Qsup}$ , para o percentual de extrapolamento dos parâmetros de QoS caracterizados como limites superiores para a solução  $s$ . Para cálculo de extrapolamento, é necessário dispor da matriz  $W$ . Numa implementação, essa matriz deve ser gerada com medições periódicas dos parâmetros de QoS pelos controladores da rede.

Desta maneira, uma solução para o problema pode ser entendida como uma solução que utilize o mínimo de recursos possíveis, atendendo às restrições de qualidade. A função objetivo agregada passa a ser apresentada como uma função de custo, definida por (3.13), onde então busca-se a minimização do uso de recursos e da violação de restrições de qualidade, para obter uma única solução final para o problema. Os valores de  $X_i$ , com  $i$  variando de 1 a 3, são os pesos atribuídos para cada uma das parcelas na função de custo, e seus valores expressam a preferência subjetiva da contribuição de cada parcela no custo total e são definidos como parâmetros no processo de otimização.

$$\arg \min f(s) = X_1 R_u(s) + X_2 R_{Qinf}(s) + X_3 R_{Qsup}(s) \quad (3.13)$$

Como resultado do processo de otimização, é esperado que os valores de  $R_{Qinf}(s)$  e  $R_{Qsup}(s)$  sejam zerados, e  $R_u(s)$  apresente o menor valor possível. Contudo, o algoritmo

pode convergir para soluções que não zerem os extrapolamentos dos limites de QoS pré-estabelecidos, o que pode ocorrer em decorrência de ausência de recursos necessários para atender determinada demanda. Ao aumentar os valores de  $X_2$  e  $X_3$ , a pressão de seleção tende a eliminar mais rapidamente soluções ineficazes do espaço de objetivos.

## 3.2 Operadores Genéticos

Na literatura, existem vários algoritmos evolucionários que podem ser utilizados para realização do processo de otimização de rotas. Na modelagem do problema para utilização de tais algoritmos, é necessário definir uma representação para as soluções e os operadores genéticos, o que foi feito conforme descrições a seguir.

A definição de uma instância do problema consiste em uma matriz de topologia e um conjunto de pares de demandas. A Figura 6 apresenta um exemplo de instância do problema.

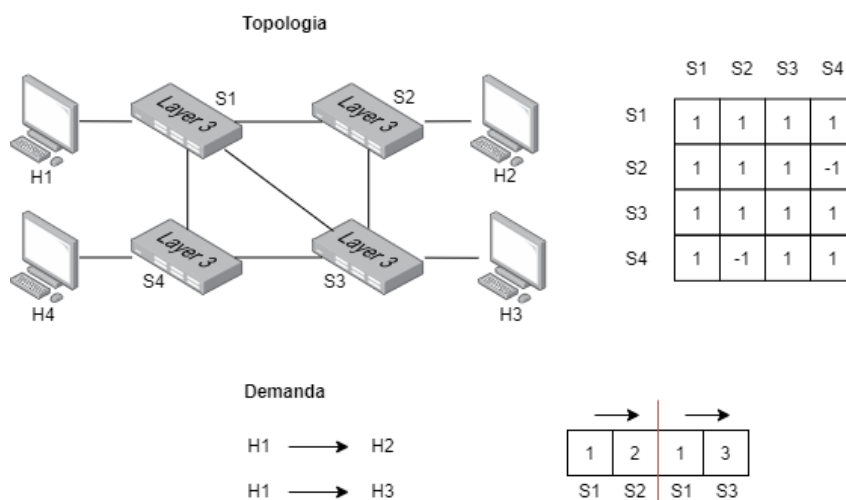


Figura 6 – Representação de uma instância do problema.

Para representação de uma solução para o problema, é criada uma lista de tamanho  $k$ , com vetores de tamanho  $m - 2$  para compor os possíveis nós intermediários entre a origem e o destino. Como os vetores são definidos com tamanho estático, o valor  $-1$  é utilizado para representar a ausência de nó intermediário. Assim, a solução para uma demanda deve ser um subconjunto de todos os outros *switches* e valores unitários negativos. A Figura 7 apresenta um exemplo de representação.

A representação não aloca variáveis para demandas em que os hosts de origem e

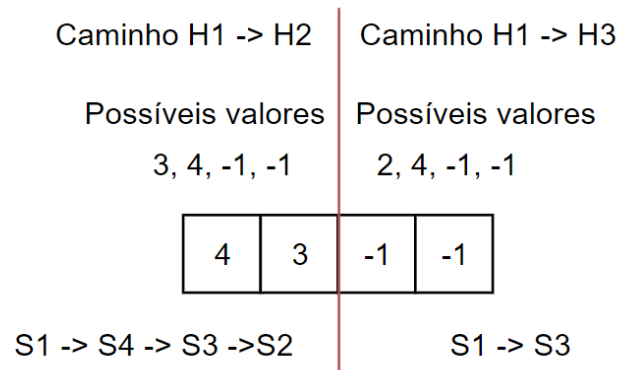


Figura 7 – Representação de uma solução para o problema.

destino se encontram conectados em um mesmo *switch*. No processo de inicialização da população, as soluções são criadas selecionando um subconjunto aleatório do conjunto de valores possíveis para cada uma das demandas.

Para o operador de cruzamento, foi utilizada uma estratégia de definir um ponto de corte para cada caminho presente na representação da solução: o filho recebe os primeiros valores de um pai até tal ponto para cada caminho e, em seguida, recebe os valores do outro pai, ignorando valores se forem nós já adicionados. Ao final, valores negativos são adicionados se necessário para completar a solução. Este cenário pode ocorrer, por exemplo, quando os valores do segundo pai, após o ponto de corte, apresentam valores de nós intermediários já presentes no primeiro pai, antes do ponto de corte. O processo gera dois filhos, alterando a ordem dos pais para cada um. A Figura 8 apresenta um exemplo do processo de cruzamento.

Para operação de mutação, foi definido o seguinte processo: seleciona um caminho dentro de uma solução e em seguida seleciona uma posição aleatória. A diferença é calculada entre os valores possíveis para aquele caminho e os valores já presentes nele, gerando um conjunto complemento. Em seguida, é feita uma troca de valores entre uma posição aleatória do caminho e um valor aleatório do conjunto complemento. A Figura 9 apresenta um exemplo visual da operação de mutação.

Ao aplicar os operadores genéticos, soluções inefectíveis podem ser geradas, sugerindo caminhos por enlaces que não existem, o que seria o caso, no exemplo da Figura 7, se a primeira solução estivesse com os valores 3 e 4, sugerindo que a demanda de *H1* para *H2* passe pelos *switches* *S1*, *S3*, *S4* e *S2* em sequência, sendo que o salto de *S4* para *S2*

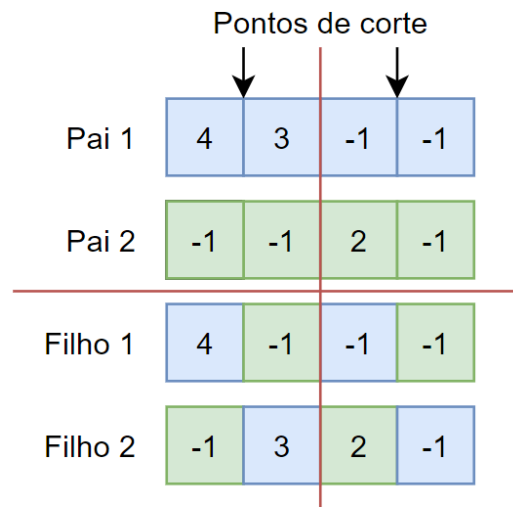


Figura 8 – Operação genética de cruzamento.

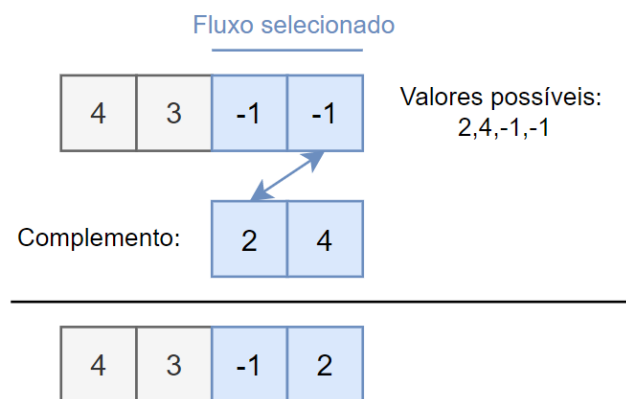


Figura 9 – Operação genética de mutação.

não é topologicamente possível. Esses casos são tratados com penalizações nos valores das funções de avaliação, fazendo-se com que a própria pressão de seleção elimine, ao longo das gerações do algoritmo, tais soluções.

No caso do algoritmo para solução do problema em cenário de melhor esforço, o cálculo dos valores das funções objetivo é realizado através de uma heurística que processa a solução em função da topologia, gerando a matriz  $U$  e computando os valores de dispersão e utilização de enlaces. No caso do cenário de SLA, a matriz  $W$  também é utilizada para se computar o valor do objetivo. Assim, conforme discutido na modelagem, o problema é tratado de forma mono-objetivo, onde a função de avaliação consiste na agregação do objetivo com as restrições com atribuição de pesos. Havendo um único valor de avaliação,

a seleção de soluções candidatas para compor as próximas gerações no processo utilizam uma ordenação linear das soluções pelas avaliações obtidas.

Na realização do processo de otimização, os algoritmos evolucionários têm os parâmetros de taxas de cruzamento e mutação, tamanho de população e número de gerações pré definidos, e a cada geração realizam seleção de pais por torneio, operações genéticas, e substituição. O critério de parada foi definido como o número máximo de avaliações de solução.

### 3.3 Análise de Algoritmos de Pareto

A análise de diferentes técnicas evolucionárias aplicadas na solução dos problemas de otimização de rotas foi realizada considerando modelagem dos operadores genéticos para aplicação de otimização no problema de roteamento. Assim, o próximo passo do trabalho consiste na definição dos algoritmos a serem avaliados no processo. A literatura apresenta uma série de algoritmos evolucionários para otimização multiobjetivo. Dentro desta gama, foram escolhidos os seguinte algoritmos:

- HypE: O algoritmo usa o valor de hipervolume como base para avaliação das soluções no processo de otimização. Tal indicador permite a otimização da dominância no contexto de fronteira de Pareto (Bader and Zitzler, 2011);
- IBEA: utiliza indicadores para comparar as soluções, como  $\epsilon$ , ou o hipervolume, comparando as soluções com base no subespaço dominado (Zitzler and Künzli, 2004);
- MOCELL: a principal característica deste algoritmo é que cada solução só pode se combinar com outras soluções em uma vizinhança de tamanho definido. Assim, a ideia é realizar exploração melhor do espaço de busca (Nebro et al., 2009);
- NSGA-II: utiliza o conceito de dominância para seleção das melhores soluções e o *crowding distance* para manutenção da diversidade na análise de soluções em uma mesma fronteira Pareto (Deb et al., 2002);
- NSGA-III: uma adaptação do NSGA-II, esse algoritmo substitui o operador de *crowding distance* por um operador de clusterização que é auxiliado por um conjunto de pontos de referência distribuídos no espaço de objetivos (Deb and Jain, 2013);

- SPEA-2: utiliza a combinação de contagem e ranqueamento de dominância e uma métrica para manutenção de diversidade baseada no KNN (Zitzler et al., 2001).

Todos os algoritmos foram implementados com adaptação da modelagem dos operadores genéticos proposta, com parâmetros equivalentes e um número fixo de iterações.

Para avaliação dos modelos, foram considerados dois parâmetros, o hipervolume das fronteiras Pareto aproximadas e o tempo de execução das gerações, que se relaciona diretamente com o esforço computacional demandado na técnica.

O hipervolume (Zitzler and Thiele 1999) é um indicador que mede o volume coberto por uma fronteira Pareto, utilizando um ponto de referência. O valor da medida mede a qualidade do Pareto, dizendo sobre a convergência e diversidade das soluções.

Para o cálculo do hipervolume, é construído um hipercubo para cada ponto do Pareto, juntamente com o ponto de referência. O valor final consiste na união dos volumes. A Figura 10 apresenta um exemplo gráfico da área do hipervolume para um problema com dois objetivos.

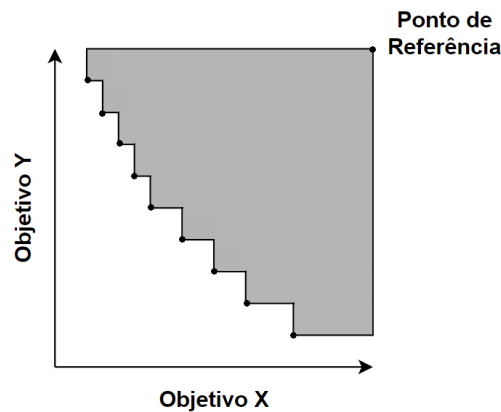


Figura 10 – Representação gráfica do hipervolume em duas dimensões.

A implementação dos algoritmos foi realizada com base no *jMetalPython* (Benítez-Hidalgo et al., 2019), um framework escrito em Python para otimização multiobjetivo. O framework apresenta a implementação base de uma série de algoritmos, além de problemas para teste de *benchmark* e indicadores de qualidade. Os algoritmos foram implementados e adaptados para utilização dos operadores descritos e propostos anteriormente.

As execuções foram realizadas em um computador com as seguintes configurações:

- Processador Intel Core i7-7500U 2.7GHz;

- Memória 8GB DDR4;
- Armazenamento Sandisk SSD PLUS 480 GB;
- Sistema operacional Ubuntu 18.04.3 LTS.

### 3.4 Cenários Experimentais

Após a construção e avaliação dos modelos de otimização, os algoritmos indicados foram utilizados para avaliar os resultados obtidos em cenários reais de redes. Os experimentos são descritos a seguir.

No contexto de otimização em cenário de melhor esforço, foi desenvolvido e executado um ambiente emulado de rede. A topologia de testes utilizada para realização do experimento foi definida como uma topologia mesh, conforme Figura 11, com valor de 1 Mbps de vazão para cada enlace. Uma vez que o algoritmo proposto tem a capacidade de explorar diversos caminhos buscando soluções para as demandas, a topologia totalmente conectada foi adotada por proporcionar maior espaço de busca e recursos a serem explorados.

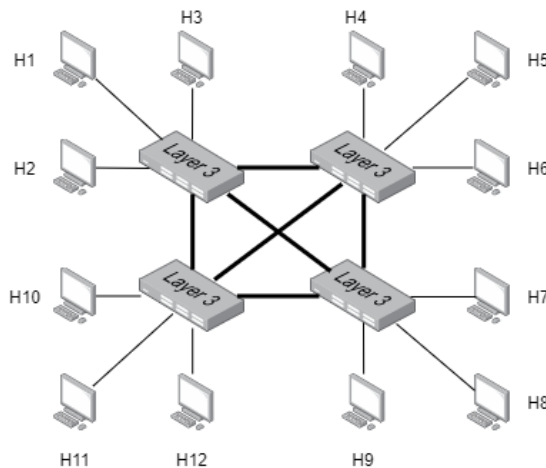


Figura 11 – Topologia para testes de melhor esforço na emulação.

A emulação foi feita através do Mininet ([Handigol et al., 2012](#)), na versão 2.2.2, definindo uma rotina de experimentos através de um script específico em Python. Mininet é uma plataforma de emulação de redes, com suporte para SDN, que permite o teste de ambientes utilizando o protocolo OpenFlow. É uma das ferramentas mais populares no

contexto de SDN na comunidade de pesquisadores da área, devido a sua simplicidade, disponibilidade e flexibilidade (Lantz et al., 2010).

O ambiente foi criado para executar um cenário de rede por um período de 120 segundos, adicionando uma nova demanda por fluxo de dados TCP a cada 20 segundos, gerado com IPERF<sup>1</sup>. O IPERF é uma ferramenta frequentemente usada em trabalhos relacionados com o paradigma SDN (Farhady et al., 2015; Huang et al., 2016), que reúne em uma única aplicação o relatório da análise de várias métricas, como a vazão, a capacidade máxima fim-a-fim a nível de transporte, o jitter e a perda de pacotes. O script de experimentos foi programado para coletar dados de utilização de enlace, banda e RTT (*Round Trip Time*).

Na realização do experimento, foram utilizados *Open vSwitches* (OvS) (Pfaff et al., 2015), na versão 2.9.8, para a execução do processo de *forwarding*, que é definido pelo encaminhamento de pacotes que chegam em um nó comutador para o próximo nó, sendo este próximo componente do caminho a percorrer até o destino final da comunicação. O OvS é uma implementação de um *switch* ethernet virtual, cujas portas são conectadas em outros componentes de rede, como outros *switches* ou máquinas virtuais. A implementação consiste em um projeto de código aberto apoiado pela comunidade e possui suporte para OpenFlow.

O experimento foi realizado em três diferentes cenários. No primeiro cenário, foi utilizado o modo *standalone* nos *switches*. Esse modo permite o uso do STP (*Spanning-tree Protocol*) (IEEE, 2004), que possibilita a utilização de enlaces redundantes na topologia, gerando uma árvore de dispersão e mantendo outros enlaces de backup. Nos outros dois cenários, foi utilizado o modo OpenFlow, onde o controlador é consultado pelos *switches* para atribuição de regras de encaminhamento, utilizando o algoritmo de otimização. Em cada caso, para seleção da solução final dentro dos Pareto aproximados, um dos objetivos (balanceamento ou minimização de uso de enlaces) foi priorizado, atribuindo-se ao objetivo priorizado o peso 0,9 e ao outro o peso 0,1. Assim, o algoritmo tem a capacidade de ter uma pequena característica de minimização de enlaces nas soluções balanceadas, e de forma análoga, uma pequena tendência de balanceamento quando priorizando a minimização de enlaces, de forma não trabalhar uma otimização mono-objetivo em ambos cenários.

A literatura apresenta vários controladores SDN. O Ryu (Morita et al., 2012) é

---

<sup>1</sup> <https://iperf.fr/>

um controlador desenvolvido em Python, com arquitetura centralizada, a qual é mais utilizada em redes de baixa escala, onde o espaço físico e o tamanho das topologias são menores, como LANs (*Local Area Networks*). O controlador apresenta suporte para as versões do OpenFlow 1.0 a 1.5, sendo uma alternativa indicada para aplicações de pesquisa (Islam et al., 2020). Trabalhos comparativos têm mostrado que o Ryu apresenta boas avaliações em características como suporte a *switches* virtuais, modularidade, documentação, compatibilidade de linguagem de programação e disponibilidade de interface de usuário. Além disso, em termos de desempenho, o controlador apresenta consistência em seu comportamento (Zhu et al., 2019). Desta forma, para o desenvolvimento do aplicativo de otimização de rotas foi utilizado o controlador Ryu, na versão 4.32.

Na aplicação desenvolvida, o controlador realiza revisão periódica do estado topológico da rede utilizando protocolo LLDP (*Link Layer Discovery Protocol*) (IEEE, 2015) e define operações com base em dois eventos principais: conexão de *switch* e entrada de pacotes. Na conexão, é feito primeiramente o *handshaking* OpenFlow, onde o controlador e o *switch* trocam mensagens "Hello", o controlador emite uma mensagem "Feature Request", e o *switch* então responde com uma mensagem "Feature Reply", contendo algumas informações, como tamanho de buffer para pacotes e número de tabelas suportado. Em seguida, é instalada uma regra de encaminhamento de pacote para o controlador, com prioridade mínima. Dessa maneira, um pacote é encaminhado ao controlador quando não há *match* das informações do pacote com alguma regra de encaminhamento. Quando o controlador recebe um pacote, este então adiciona uma regra definida para o fluxo em questão em cada um dos *switches* no caminho e retorna o pacote. A Figura 12 apresenta os fluxos de processamento de levantamento de topologia e reação aos eventos trabalhados.

As regras de encaminhamento fazem *match* com endereços MAC de origem e destino dos pacotes e são adicionadas ao *switch* com validade de quatro segundos. Assim, periodicamente, a tabela é atualizada. Ao receber um pacote de entrada, o controlador verifica se existe uma rota determinada para o fluxo entre os hosts envolvidos. Caso não exista, um novo processo de otimização é executado para definir um conjunto de rotas para o novo fluxo, alterando as rotas dos fluxos correntes se necessário, e atualizar as regras para o fluxo relacionado ao pacote recebido. Antes de realizar o processo de definição de rotas, é verificado se o pacote analisado apresenta endereço de destino inalcançável, ou se é uma mensagem broadcast MAC, como mensagem ARP, e faz tratamento específico,

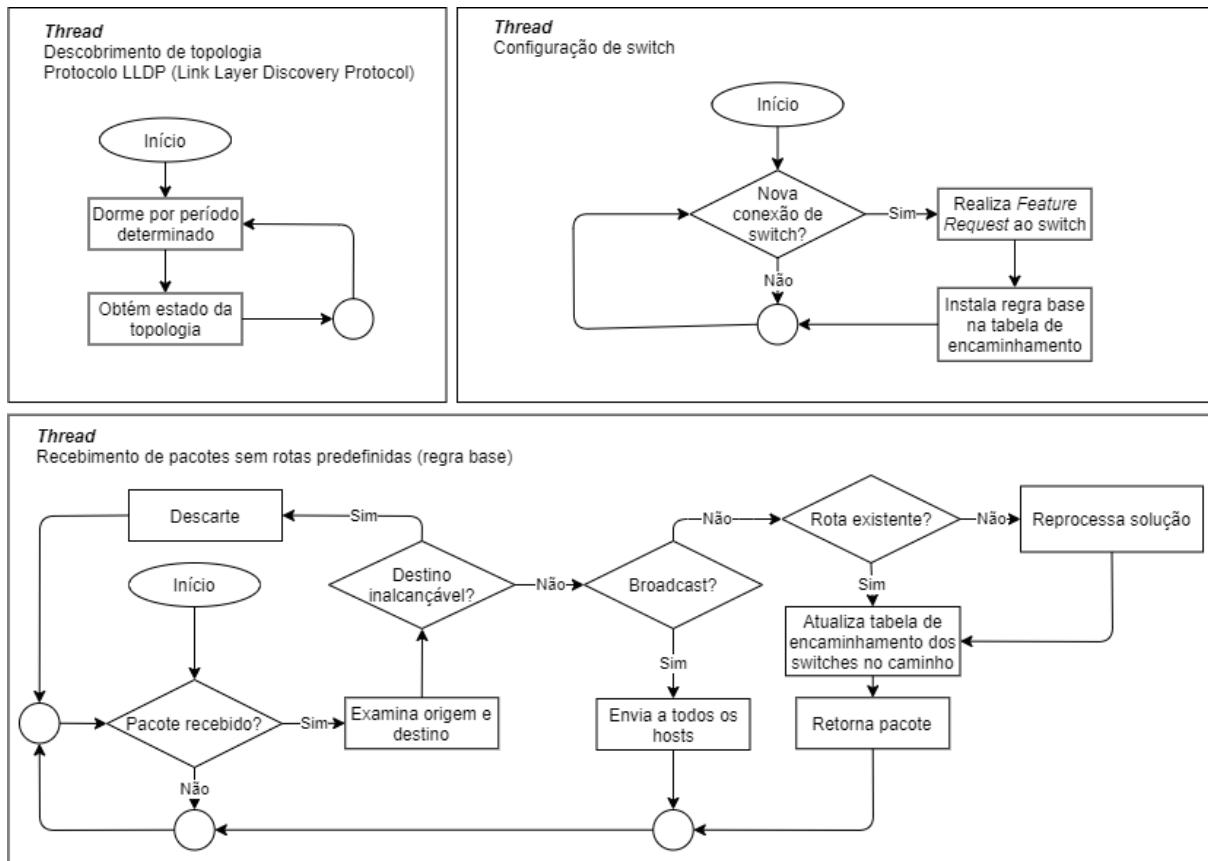


Figura 12 – Arquitetura do aplicativo do controlador.

conforme mostrado no diagrama.

No processo de avaliação do algoritmo genético no cenário de SLA, foi desenvolvido um processo de simulação computacional de cenários de redes. Tal processo consiste em uma implementação em Python (versão 3.6.5) onde se define uma topologia inicial, em seguida uma lista de demandas a serem admitidas em tal topologia, com suas respectivas restrições de QoS. O otimizador então é chamado sequencialmente para cada demanda, admitindo-a e atualizando o estado da rede, sendo esta atualização uma diminuição dos recursos disponíveis em intensidade equivalente aos valores requeridos para banda e um valor de  $1ms$  adicional de atraso para representar o tempo incrementado pela disputa do recurso. A Figura 13 apresenta um diagrama do processo de simulação utilizado para execução do referido experimento.

Para validar o processo de otimização, foi definido um cenário de rede hipotético com políticas de SLA com enlaces apresentando valores iniciais de atraso de  $1ms$  e banda de  $100Mbps$ , conforme apresentado na Figura 14.

Utilizando o modelo e simulando variações de valores de banda e atraso para os

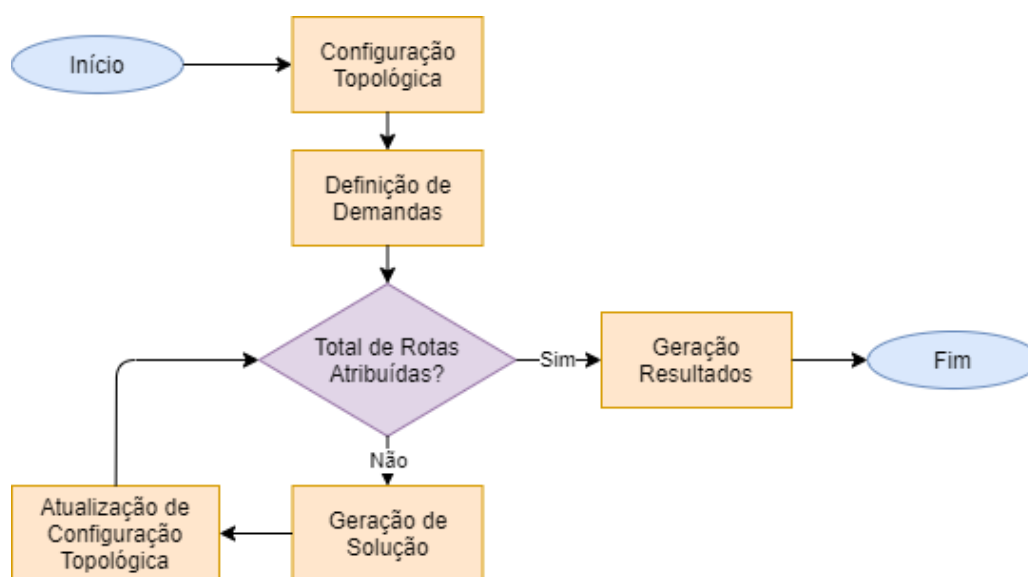


Figura 13 – Fluxograma de execução da simulação.

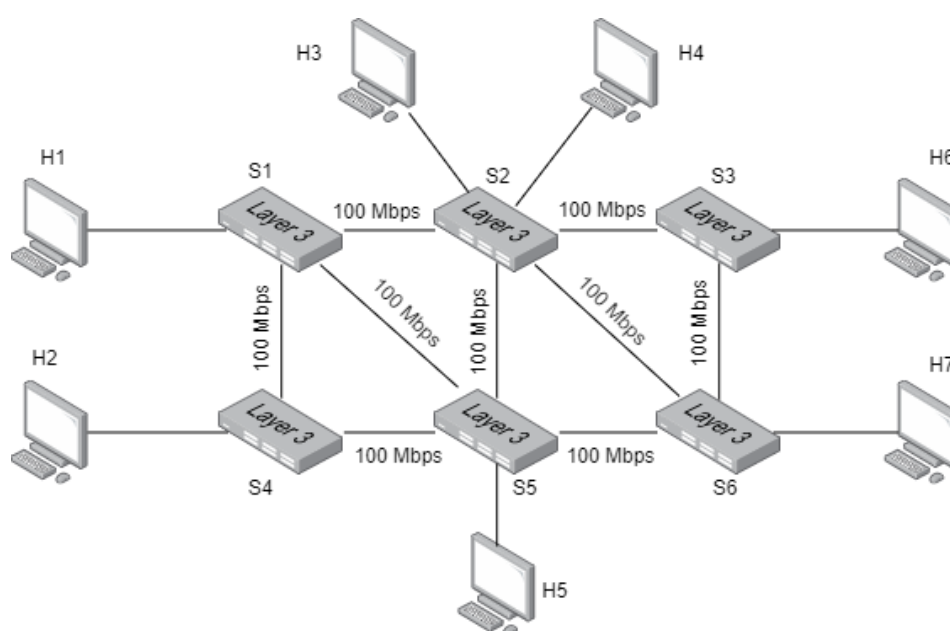


Figura 14 – Esquema da rede utilizada na simulação.

enlaces após admissão de fluxos, o algoritmo foi aplicado para escolha das soluções tendo em vista os objetivos trabalhados. Assim, tal experimento teve por objetivo verificar se, dadas as restrições e os objetivos colocados e ainda considerando o estado atual da rede, o algoritmo consegue realizar atribuição coerente de rotas para os fluxos.

O critério de parada adotado para o cenário de simulação foi definido como o número máximo de avaliação de funções, que é dado pelo produto entre o tamanho da população e a quantidade de épocas. Conforme discutido em (Kim, 2013), existem outras técnicas de critério de parada que podem ser consideradas visando melhor identificação

de cenários com ausência de soluções que não atendem os requisitos de QoS, como por exemplo a ausência de melhoria significativa nas soluções nas últimas iterações, sendo esta quantidade um parâmetro a ser definido.

### **3.5 Resumo**

Neste capítulo foram apresentadas a metodologia de desenvolvimento do presente trabalho. Primeiramente, foi feita uma definição matemática do problema. Em seguida, foi feita a definição dos operadores genéticos para solução do problema. Assim, foi feito um planejamento experimental para comparação dos diferentes algoritmos levantados para solução do problema multiobjetivo e avaliação experimental do impacto do uso de soluções otimizadas nas redes por meio de emulação e simulação nos cenários de política de melhor esforço e SLA, respectivamente.

# Capítulo 4

## Resultados e Discussão

O presente capítulo apresenta os resultados obtidos para cada um dos experimentos executados, sendo estes para os cenários de otimização para abordagens de melhor esforço e SLA. As seções seguintes abordam, separadamente, os resultados obtidos para cada um dos casos.

### 4.1 Experimento de Comparação de Algoritmos

No processo de otimização com múltiplos critérios, inicialmente várias instâncias do problema foram analisadas, variando a topologia da rede e a quantidade de fluxos a serem adotados. A Figura 15 apresenta o conjunto inicial e a aproximação do Pareto para uma instância determinada por topologia de 15 *switches* totalmente ligados entre si e 44 fluxos a serem alocados. Para esta instância, foram escolhidos parâmetros com melhores resultados de convergência e obtenção da fronteira Pareto aproximada de forma empírica, ficando o tamanho de população de 80, taxa de mutação de 0,25, taxa de cruzamento de 0,75 e um total de 250 gerações.

Os testes sugeriram fronteiras Pareto convexas para o problema, sendo o Pareto, em instâncias para topologias com ausência de enlaces, definido como subconjunto do Pareto de soluções para topologia totalmente conectada. O espaço de busca cresce significativamente com o aumento do número de vértices comutadores e da quantidade de demandas a serem atendidas, fazendo a convergência ficar mais lenta.

Os experimentos de comparação de algoritmos foram preliminares para subsidiar a escolha de um algoritmo para emprego no processo de otimização de roteamento no

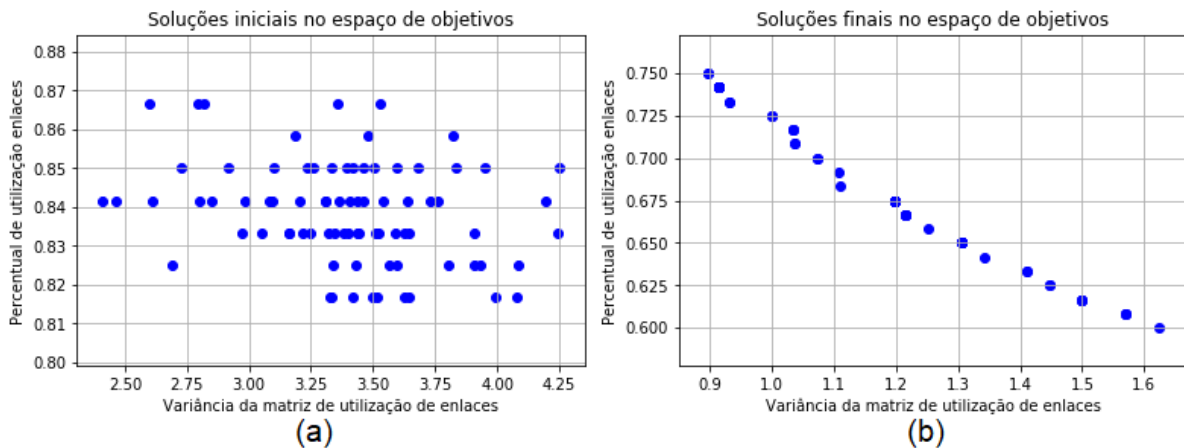


Figura 15 – Resultados obtidos para uma instância do problema de atribuição de rotas, onde em (a) têm-se uma população inicial de soluções no espaço de objetivos e em (b) a fronteira Pareto aproximada.

controlador SDN utilizado no experimento posterior.

No processo de comparação dos algoritmos empregados na otimização de rotas para o problema de interesse, foram realizadas 30 execuções para cada um dos algoritmos, considerando três configurações distintas.

As configurações utilizadas foram:

- Configuração 1: Topologia com 15 nós e 44 demandas;
- Configuração 2: Topologia com 10 nós e 33 demandas;
- Configuração 3: Topologia com 5 nós e 22 demandas.

Cada uma dessas configurações apresentam um potencial distinto de exploração no espaço de busca, onde quanto maior a topologia e o número de demandas, maior é o número de soluções possíveis para o problema. As diferentes configurações foram definidas para se investigar as diferenças de comportamentos e resultados obtidos em função do tamanho dos problemas.

Os algoritmos tiveram seus parâmetros configurados de forma equivalente, definindo como critério de parada o número de avaliação de funções, equivalente a 2500, com tamanho populacional de 100 soluções. Os gráficos presentes nas figuras 16, 17 e 18 apresentam, respectivamente, as distribuições dos valores de hipervolume obtidos nos cenários especificados nas configurações 1, 2 e 3.

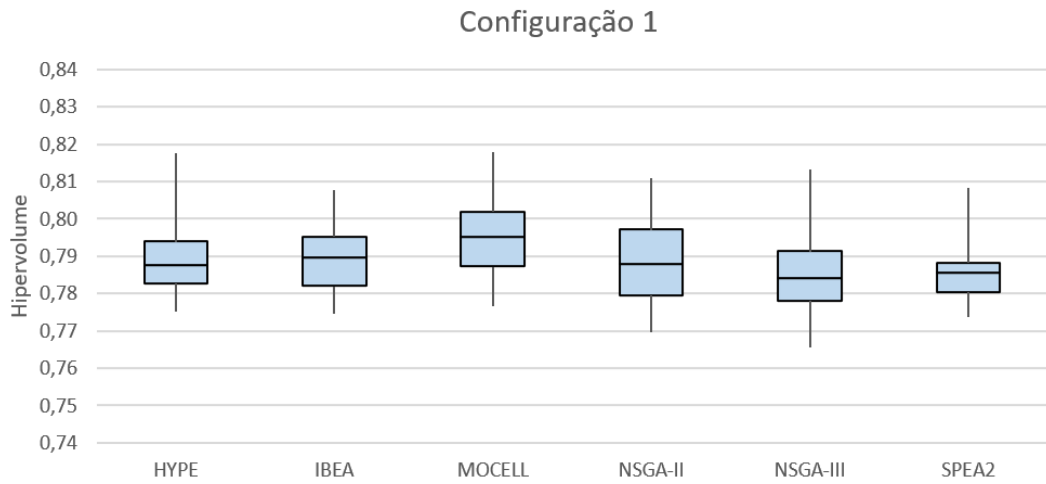


Figura 16 – *Boxplots* dos valores de hipervolume obtidos para a configuração 1.

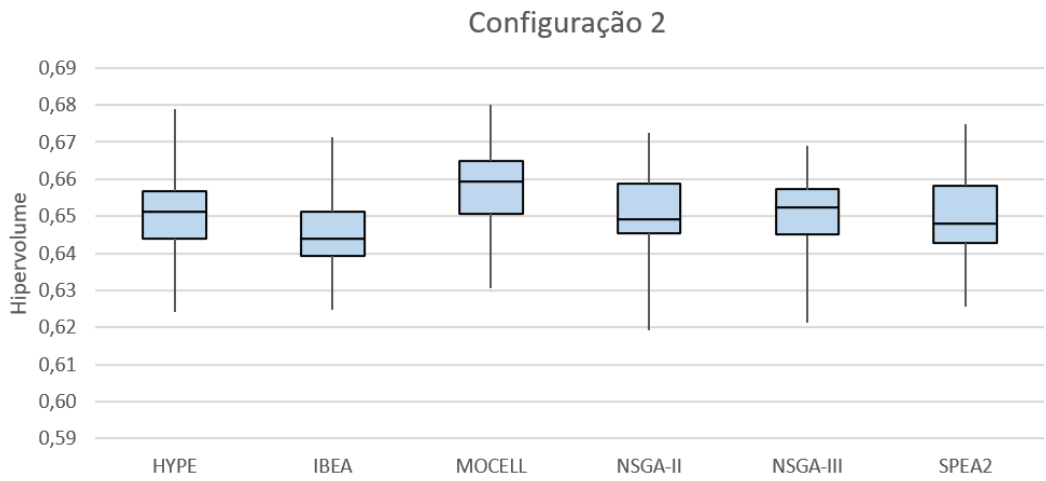


Figura 17 – *Boxplots* dos valores de hipervolume obtidos para a configuração 2.

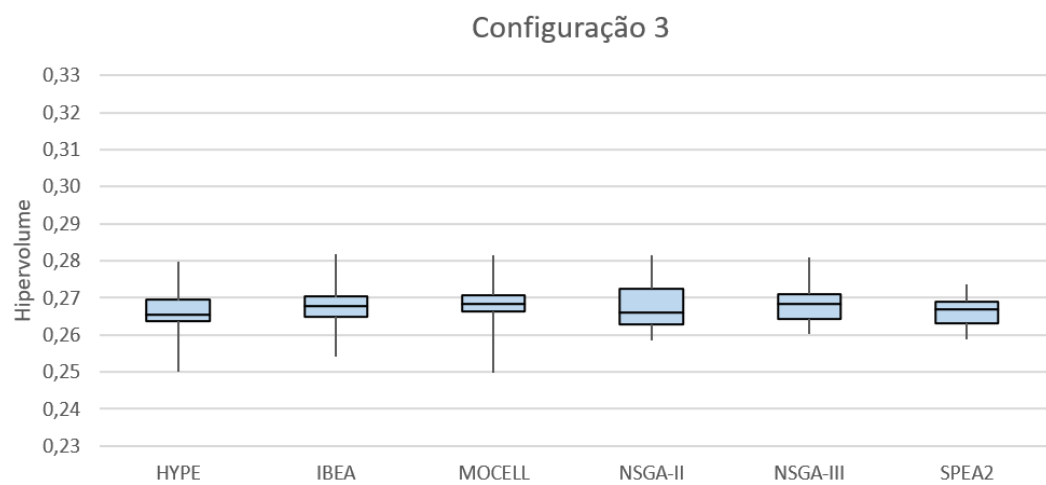


Figura 18 – *Boxplots* dos valores de hipervolume obtidos para a configuração 3.

Em uma primeira visão, é verificado que os valores de HV (Hipervolume) alcançados pelas execuções diminuem com a diminuição do problema. Isso é esperado, pois conforme colocado, problemas menores apresentam espaços de busca menores e consequentemente menos soluções ótimas. Tal quantidade influencia no cálculo do indicador.

Apesar de haver discrepâncias dentro dos grupos, tais diferenças não são visualmente expressivas nos gráficos. A Tabela 2 apresenta os principais parâmetros estatísticos acerca dos grupos de dados analisados.

Tabela 2 – Parâmetros estatísticos dos dados coletados de HV.

		Contagem	Soma	Média	Mediana	Variância
Configuração 1	HYPE	30	23,6779	0,7893	0,7876	9,49E-05
	IBEA	30	23,6739	0,7891	0,7896	7,94E-05
	MOCELL	30	23,8606	0,7954	0,7951	1,10E-04
	NSGA-II	30	23,6480	0,7883	0,7880	1,31E-04
	NSGA-III	30	23,5503	0,7850	0,7843	1,13E-04
	SPEA2	30	23,5653	0,7855	0,7855	6,18E-05
Configuração 2	HYPE	30	19,4991	0,6500	0,6513	1,17E-04
	IBEA	30	19,3770	0,6459	0,6440	1,06E-04
	MOCELL	30	19,7574	0,6586	0,6595	1,17E-04
	NSGA-II	30	19,5409	0,6514	0,6492	1,43E-04
	NSGA-III	30	19,5092	0,6503	0,6525	1,11E-04
	SPEA2	30	19,4953	0,6498	0,6479	1,28E-04
Configuração 3	HYPE	30	7,9904	0,2663	0,2654	3,90E-05
	IBEA	30	8,0278	0,2676	0,2676	2,48E-05
	MOCELL	30	8,0263	0,2675	0,2682	3,57E-05
	NSGA-II	30	8,0413	0,2680	0,2660	4,38E-05
	NSGA-III	30	8,0547	0,2685	0,2684	2,84E-05
	SPEA2	30	7,9839	0,2661	0,2669	1,39E-05

Dispondo dos dados dos experimentos, o próximo passo foi realizar uma análise estatística para dizer se as médias obtidas pelos algoritmos apresentam diferenças relevantes em cada um dos três casos trabalhados, e se podemos assumir, com um nível definido de confiabilidade, diferenças entre as médias populacionais dos grupos investigados. Para isso, foi feito uma análise de ANOVA (*Analysis of Variance*) de uma via para os resultados obtidos em cada uma das configurações.

O ANOVA de uma via é um método estatístico que permite avaliar se existe diferença significativa entre diferentes grupos de dados com uma variável independente com base em uma estatística F (Montgomery and Runger, 2010). No teste, duas hipóteses são

avaliadas sobre as médias populacionais dos conjuntos de dados avaliados. Tais hipóteses são definidas conforme (4.1).

$$\begin{cases} H_0 : \mu_1 = \mu_2 = \mu_3 \dots = \mu_k \\ H_1 : \text{Médias não são todas iguais} \end{cases} \quad (4.1)$$

O teste foi executado a um nível de significância  $\alpha$  de 0,05 para cada configuração. Os resultados são apresentados na Tabela 3 e discutidos em seguida.

Tabela 3 – Resultados obtidos com ANOVA.

Cenário	Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Config, 1	Entre grupos	0,0021	5	0,000412	4,191	0,0013	2,266
	Dentro dos grupos	0,0171	174	9,84E-05			
	Total	0,0192	179				
Config, 2	Entre grupos	0,0026	5	0,000519	4,315	0,0010	2,266
	Dentro dos grupos	0,0209	174	0,000120			
	Total	0,0235	179				
Config, 3	Entre grupos	0,0001	5	2,63E-05	0,850	0,5164	2,266
	Dentro dos grupos	0,0054	174	3,09E-05			
	Total	0,0055	179				

Nas configurações 1 e 2, o *valor-P* é inferior ao nível de significância  $\alpha$ , e isso significa que há evidências estatísticas de que há pelo menos um par dentre os conjuntos analisados que apresenta médias populacionais diferentes entre si. Já no caso da configuração 3, onde o *valor-P* é maior que  $\alpha$ , a hipótese  $H_0$  é aceita. A mesma análise pode ser feita ao comparar os valores de  $F$  com  $F$  crítico, onde nos casos 1 e 2 o valor é superior, já no caso 3 o valor é inferior.

Com tais resultados, há evidências de que todos os algoritmos apresentam comportamentos similares quando sujeitos a instâncias de problemas de roteamento com topologias menores e número de demandas menores, onde os resultados são mais concentrados e equivalentes. Por haver um espaço de busca reduzido, a busca tende a encontrar soluções ótimas em uma velocidade maior, independente da técnica utilizada. À medida que o problema aumenta de tamanho, a exploração de espaço de soluções passa a ser mais

complexa e então diferentes abordagens de exploração passam a apresentar diferenças mais significativas.

O teste ANOVA requer atendimento de três premissas:

- Independência e aleatoriedade das observações nos conjuntos analisados, o que é atendida pela ausência de qualquer viés nos cenários experimentais de coleta de dados;
- Normalidade dos dados, que foi validada através do teste de Shapiro-Wilk com um nível de significância  $\alpha$  de 0.05;
- Homocedasticidade, definida pela homogeneidade das variâncias dentro dos tratamentos realizados, que foi validado pelo teste de Levene.

O teste ANOVA diz somente sobre a igualdade das médias populacionais entre os grupos de dados analisados, mas não explicita, no caso da rejeição da hipótese nula, quais pares de grupos apresentam diferenças significativas nas médias populacionais, e o tipo de relação, se é de superioridade ou inferioridade. Assim, utilizando somente a análise de variância, não é possível fazer uma recomendação de um fator específico para um determinado resultado esperado em um grupo de valores.

Nesse sentido, foi também realizado um teste de Tukey da diferença honestamente significativa (HSD) (Montgomery and Runger, 2010), que consiste em comparar todos os pares de médias dos grupos, considerando uma diferença mínima significativa (DMS). O teste foi executado para as configurações 1 e 2, a um nível de significância  $\alpha$  de 0.05, onde foram detectadas diferenças significativas entre médias pelo ANOVA, visando identificar objetivamente os pares de grupos que apresentam diferenças entre si e posteriormente fazer recomendações de uso das técnicas. A Figura 19 apresenta os principais resultados e evidencia os pares de algoritmos cujas médias obtidas são significativamente diferentes entre si. Os algoritmos estão ranqueados conforme suas médias e as colunas com marcações representam a detecção de diferenças significativas entre os métodos marcados. Os resultados completos para o testes realizados par a par estão disponíveis no apêndice A.

Em ambas as configurações, a maior média foi obtida para o algoritmo MOCELL, estando ranqueado em primeiro lugar. Na configuração 1, os resultados obtidos com teste de Tukey apontaram diferença significativa entre as médias do MOCELL com os algoritmos

Configuração 1						Configuração 2													
Rank	Algoritmo	Média	Diferença						Rank	Algoritmo	Média	Diferença							
			1	2	3	4	5	6				1	2	3	4	5	6		
1	MOCELL	0.7954					o	o	1	MOCELL	0.6586					o	o	o	o
2	HYPE	0.7893							2	NSGA-II	0.6514								
3	IBEA	0.7891							3	NSGA-III	0.6503	o							
4	NSGA-II	0.7883							4	HYPE	0.6500	o							
5	SPEA2	0.7855	o						5	SPEA2	0.6498	o							
6	NSGA-III	0.7850	o						6	IBEA	0.6459	o							

Figura 19 – Comparação de pares de algoritmos.

SPEA2 e NSGA-III, respectivamente ranqueados nas posições 5 e 6. Assim, considerando tal configuração, não há evidência estatística que indique a utilização de um entre os quatro primeiros algoritmos ranqueados.

Na configuração 2, o teste de Tukey apontou diferenças significativas do MOCELL para com os 4 últimos algoritmos, sendo respectivamente NSGA-III, HYPE, SPEA2 e IBEA. Neste caso, não há evidências para indicar uma técnica entre MOCELL e NSGA-II.

Considerando os cenários adotados para os experimentos sob ótica das técnicas de análise de variâncias e teste de tukey ao nível de significância de 0.05, portanto, é recomendado a utilização de um dentre os algoritmos NSGA-II e MOCELL para obtenção de melhores valores de hipervolume dos Paretos aproximados.

No contexto de otimização de rotas em redes SDN, uma das possíveis abordagens para aplicação das técnicas consiste na otimização de demandas ao receber requisições na rede. Neste caso, o tempo de execução passa também a se tornar um parâmetro importante.

O tempo de execução é função da complexidade computacional das técnicas e do tamanho de suas entradas. Nos experimentos realizados, foram levantados os tempos para execução das gerações. Considerando este valor, é possível determinar o tempo de execução do algoritmo com base no número de gerações. A Figura 20 apresenta um gráfico do aumento de tempo para processamento dos três cenários, enquanto a Tabela 4 apresenta as médias de tempos. Os dados consideram a execução de uma geração do processo de otimização com um tamanho populacional de 100.

O algoritmo com custo computacional mais caro é o HYPE. Isso é esperado, pois a técnica faz uso de hipervolume para avaliação das soluções ao longo das gerações, e tal indicador apresenta uma complexidade computacional considerável. Apesar do IBEA

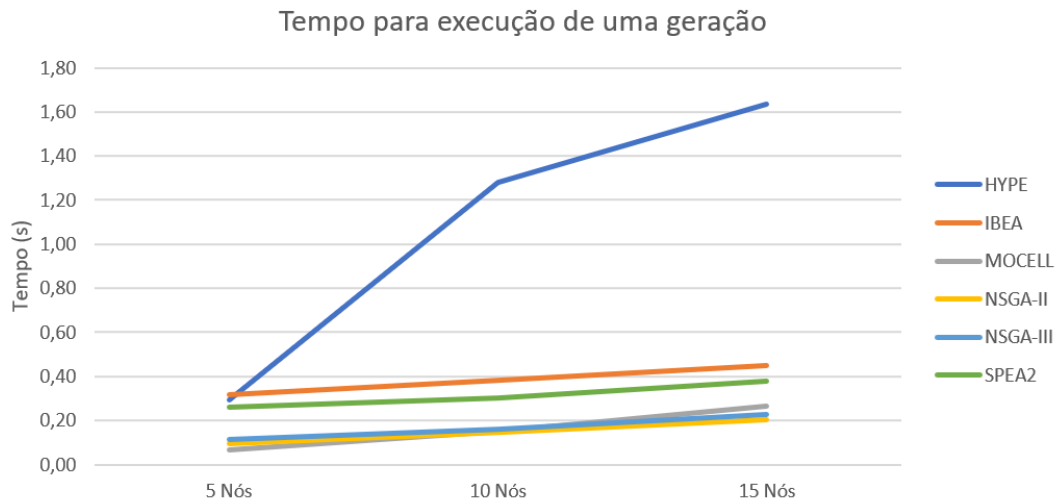


Figura 20 – Evolução do tempo de processamento por geração.

Tabela 4 – Tempo médio de processamento dos algoritmos para uma geração.

Valores médios		
1	NSGA-II	148 ms
2	MOCELL	160 ms
3	NSGA-III	166 ms
4	SPEA2	312 ms
5	IBEA	382 ms
6	HYPE	1069 ms

poder ser utilizado também com o indicador de hipervolume, ele foi configurado para utilizar o indicador  $\epsilon$ .

Ambos os algoritmos que apresentaram melhores resultados, pelos testes estatísticos, também apresentam os menores tempos de execução por geração. Assim, utilizando o tempo médio como critério de desempate, é indicado finalmente a utilização do NSGA-II.

No cenário de otimização em tempo real, uma importante questão consiste no tempo médio de processamento das técnicas. O processo de otimização baseado em Pareto, de forma geral, utiliza recursos para manter a diversidade e convergência das soluções. Ao final do processo de otimização, tem-se como resultado um conjunto Pareto aproximado de soluções, onde em seguida deve ser aplicada uma técnica de tomada de decisão para escolha da solução final.

Uma alternativa para o processo consiste na utilização de um método linear, como a agregação dos objetivos e utilização de um algoritmo genético simples. Com o propósito

de entender tal diferença, foi feita uma comparação entre o NSGA-II e um GA básico para otimização mono-objetivo, utilizando agregação de funções com pesos específicos para cada um dos objetivos e variando número de nós na topologia e número de demandas. As figuras 21 e 22 apresentam os gráficos de tempo médio para processamento de uma geração, avaliando respectivamente o tempo de processamento em função da variação de número de demandas e em função do tamanho da topologia.

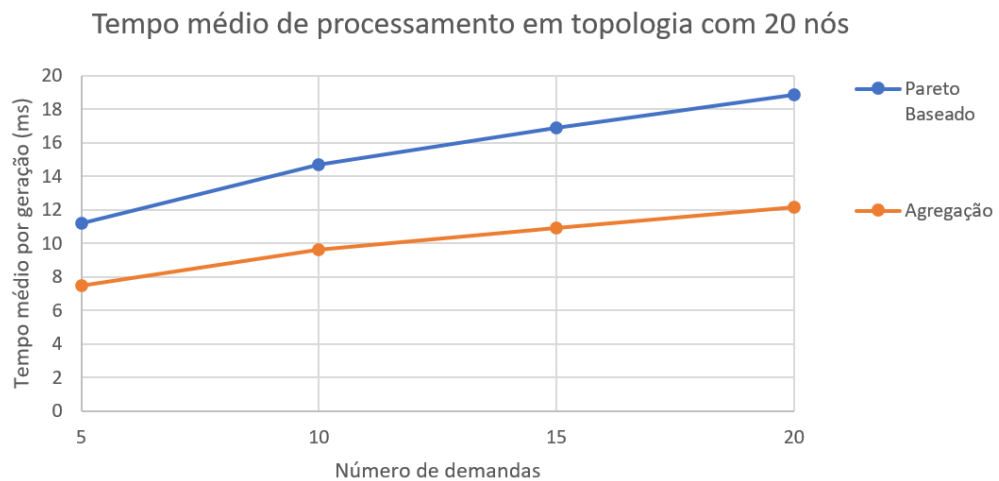


Figura 21 – Tempo de execução médio dos métodos de Pareto e de agregação com variação do número de demandas.

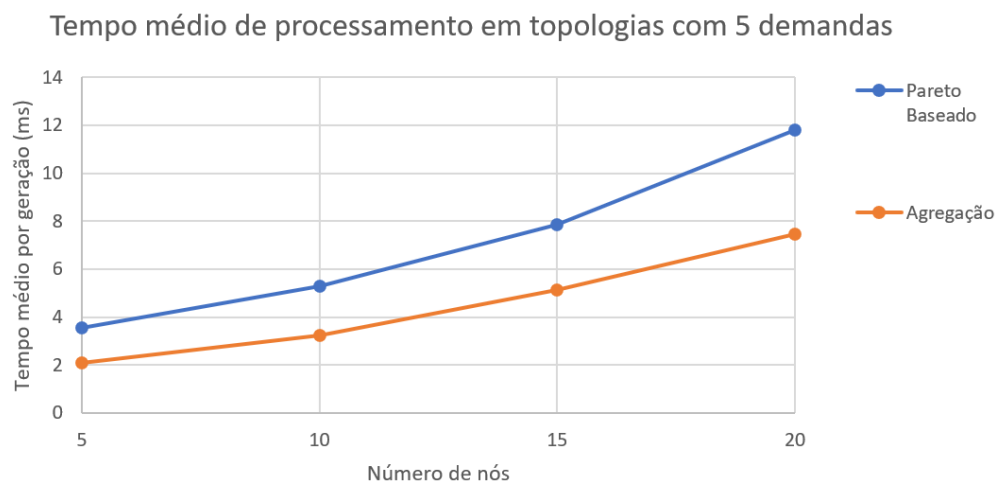


Figura 22 – Tempo de execução médio dos métodos de Pareto e de agregação com variação do tamanho da topologia.

Os experimentos mostraram um aumento expressivo no esforço computacional percebido com o aumento da complexidade topológica, gerando um aumento médio de 245% no tempo de processamento entre os cenários extremos do gráfico. Já no cenário de

mudança do número de demandas, foi percebido um aumento médio de 65% no tempo de processamento, incremento menor em comparação com a variação topológica. Ainda, como esperado, o processo utilizando agregação e atribuindo pesos requer menor esforço para execução das gerações.

Assim, dependendo da aplicação, caso as prioridades sejam definidas a priori, pode ser utilizado uma técnica de agregação para obtenção de melhores resultados em termos de tempo de processamento. Contudo, caso o processo requeira uma análise do conjunto Pareto para analisar as soluções, a solução com NSGA-II é adequada.

A criticidade do tempo de execução das técnicas depende da arquitetura da aplicação de controle utilizada. No caso de otimização em tempo real, de acordo com demandas nas redes, o tempo é um fator crítico. Para minimizar este efeito em grandes redes, podem ser consideradas a utilização de controladores SDN de arquitetura distribuída, ou mecanismos de otimização paralela com atualização incremental de rotas. Já em um cenário onde o processamento de otimização é executado a priori, mantendo soluções a serem utilizadas durante a operação da rede, o tempo de execução passa a não apresentar tal criticidade.

## 4.2 Análise de Comportamento das Soluções

Dado os modelos de otimização e as análises propostas, foram realizados experimentos visando verificar o comportamento das soluções em cenários reais de rede, seja por emulação ou simulação. As seções a seguir tratam os experimentos realizados nesse contexto.

### 4.2.1 Política de Melhor Esforço

Para emulação da rede, os parâmetros do algoritmo foram definidos de forma empírica, ficando 10 para tamanho de população, 25 gerações e taxas de cruzamento e mutação de 0,85 e 0,15, respectivamente. Após a execução de cada instância do processo de otimização, é feita a atribuição de pesos aos objetivos, onde as soluções candidatas da fronteira Pareto aproximada são reavaliadas por agregação dos valores e então ordenadas para escolha de uma solução final.

Foi executada uma emulação para cada um dos três cenários trabalhados. Os fluxos de comunicação, criados com o IPERF, tiveram seus parâmetros de qualidade medidos durante todo o experimento. Ao final, tais valores foram agregados por média para visualização do estado de qualidade rede ao longo do tempo. A Figura 23 apresenta as médias das vazões obtidas pelos fluxos ao longo da execução dos experimentos para cada um dos cenários analisados e a Figura 24 apresenta os valores médios de RTT experimentados pelos fluxos admitidos ao longo dos experimentos, em uma execução de cada um dos três cenários. Nestes gráficos, a cada 20 segundos passados nos experimentos, uma nova comunicação entre 2 hosts é adicionada na rede.

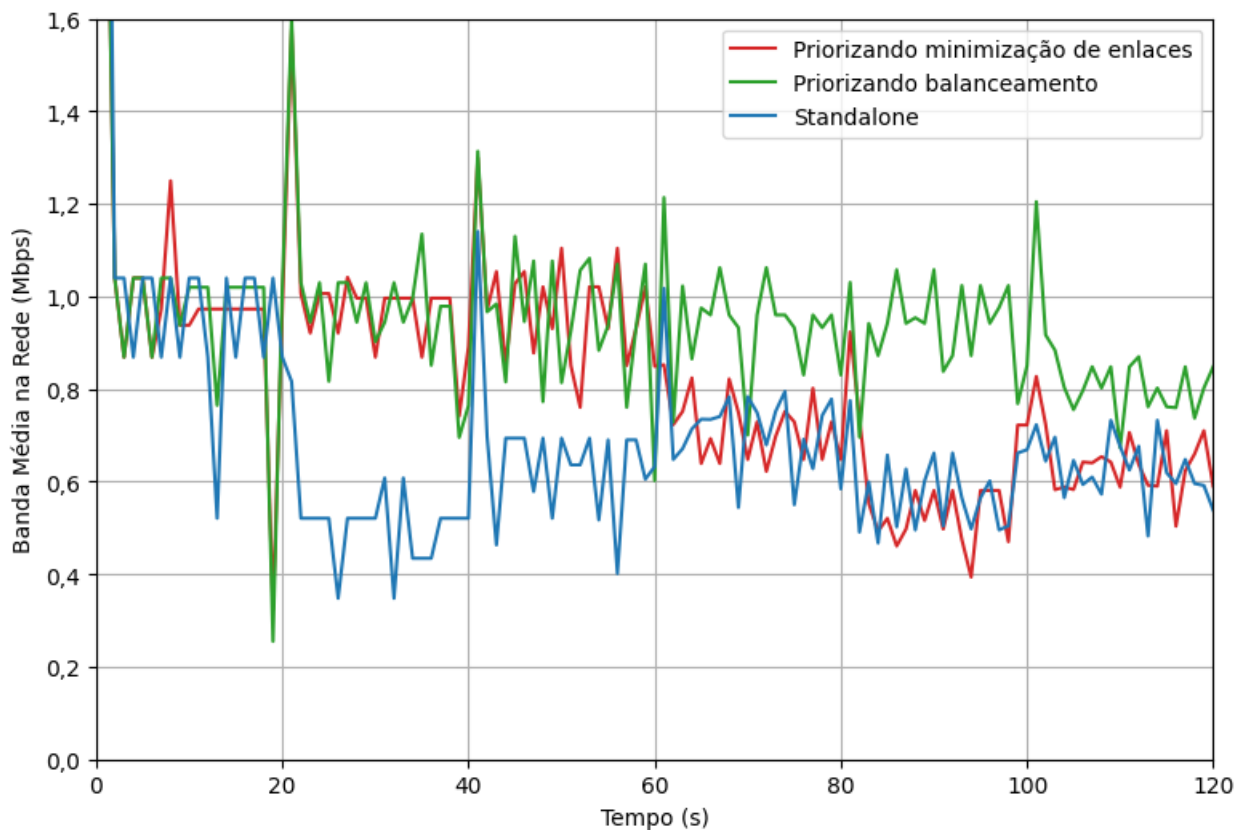


Figura 23 – Largura de banda média alcançada pelos fluxos admitidos na rede em função do tempo em uma execução experimental de cada cenário.

Nos momentos iniciais da emulação, a diferença é pequena, uma vez que o espaço de busca é pequeno e a fronteira Pareto aproximada é concentrada em um padrão de rotas. Além disso, há apenas uma demanda e ainda não há disputa por recursos. Com o aumento do número de demandas na rede, as soluções começam a divergir, se observando maiores valores de vazão e menores valores de RTT para soluções que priorizam balanceamento, onde a utilização de recursos é maior, diminuindo os efeitos da concorrência pelos mesmos.

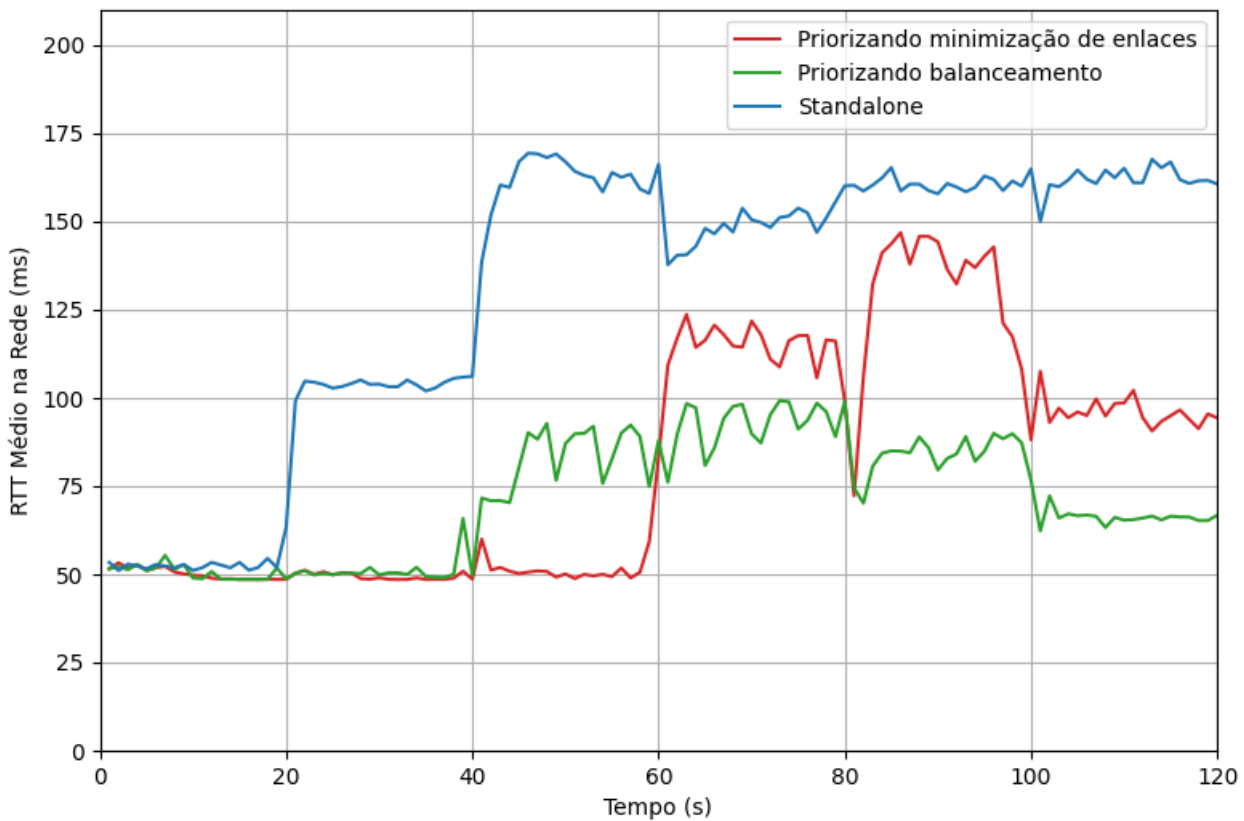


Figura 24 – RTT médio experimentalado pelos fluxos da rede em função do tempo em cada um dos cenários reproduzidos.

Além da vazão, o número de enlaces entre comutadores utilizados na rede também foi capturado para avaliação, conforme apresentado na Figura 25, onde é possível verificar que soluções mais balanceadas proporcionam maior vazão, ao custo de alocar mais recursos para atender o conjunto de demandas na rede, à medida que a capacidade demandada aumenta. Os valores médios para as variáveis analisadas são apresentados na Tabela 5.

Tabela 5 – Resultados médios obtidos.

Cenário	Vazão	RTT	Utilização de enlaces
Balanceamento	0,94 Mbps	71,09 ms	3,28
Eficiência Energética	0,81 Mbps	81,93 ms	2,80
Standalone	0,68 Mbps	131,62 ms	2,83

Os resultados obtidos para uma rede em modo *standalone* tendem a ser próximos de soluções no algoritmo genético que priorizam mais o objetivo de minimização de utilização de enlaces, à medida que mais fluxos são admitidos na rede.

Com a minimização de utilização de enlaces, menores caminhos são adotados

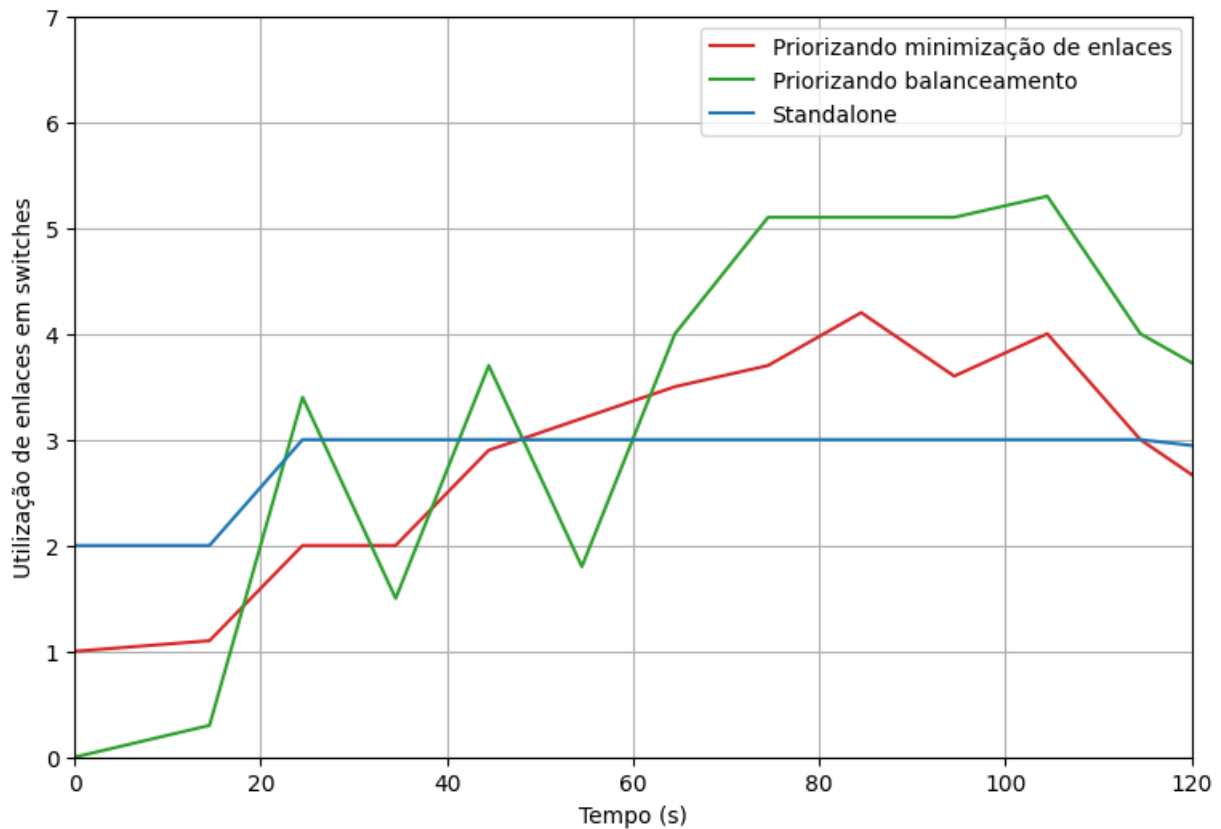


Figura 25 – Número de enlaces utilizados na rede ao longo do tempo em cada um dos cenários reproduzidos.

para atender as demandas, priorizando também a utilização de enlaces e *switches* já utilizados. Assim, o uso mínimo de enlaces e *switches* diminui a energia consumida pelos equipamentos, que é dado pela soma dos produtos entre as potências dos equipamentos em modo de operação e o tempo em que permanecem ativados.

No controlador, a periodicidade de 4 segundos para atualização da tabela de fluxos pode causar uma elevação indesejada na carga de trabalho, em cenários de alto número de demandas, prejudicando a escalabilidade. Uma alternativa para proporcionar escalabilidade seria retirar o tempo de duração da regra, e enviar comandos de exclusão de entradas nas tabelas de encaminhamento dos *switches* ao detectar mudança de rotas para demandas na rede, antes de atribuir novas regras.

O processo de otimização realizado experimentalmente pôde priorizar desempenho e eficiência energética de acordo com pesos de importância, os quais podem variar de acordo com caracterização da demanda na rede e por mudanças de bandeiras tarifárias em horários de pico de consumo energético.

### 4.2.2 Política de SLA

Para realização do processo de otimização na simulação do cenário de SLA, os parâmetros do algoritmo genético foram adaptados para a instância utilizada na simulação e definidos conforme apresentado na Tabela 6.

Tabela 6 – Parâmetros do otimizador.

Parâmetro	Valor
Tamanho da População	30
Quantidade de Épocas	30
Taxa de Cruzamento	75%
Taxa de Mutação	25%
$X_1$	0,2
$X_2$	0,4
$X_3$	0,4

Utilizando o algoritmo genético com a modelagem proposta para o cenário de simulação, foi possível gerar rotas considerando restrições de QoS e minimizando o uso de recursos na rede. Os parâmetros utilizados possibilitaram uma boa convergência para cada um dos casos, o que pode ser visto com base na média populacional de avaliação das soluções ao longo das iterações do algoritmo. Como a função de avaliação consiste em uma combinação dos valores de percentual de utilização de recursos e percentual de extrapolação das restrições de QoS, é esperado que, ao longo das gerações, os valores de avaliação diminuam, tendendo a zero se possível. A Figura 26 apresenta a evolução do *fitness* populacional médio, dado pela média de valores de função objetivo das soluções, ao longo das gerações de uma execução do algoritmo.

No início de uma execução, com a criação de rotas aleatórias dentro do espaço de busca, soluções infactíveis podem ser criadas, em decorrência da topologia não ser totalmente conectada. Os recursos de atraso infinito e banda zero para caminhos não existentes fazem com que essas soluções sejam rapidamente eliminadas da população, devido à pressão de seleção do algoritmo. Devido aos parâmetros de peso dos objetivos, o algoritmo dá primeiramente mais ênfase para as restrições de QoS em relação à minimização de utilização de enlaces.

Inicialmente, foi feita a configuração topológica, composta pelas estruturas que definem o estado inicial da rede, sendo a matriz com valores de banda e atraso, topologia

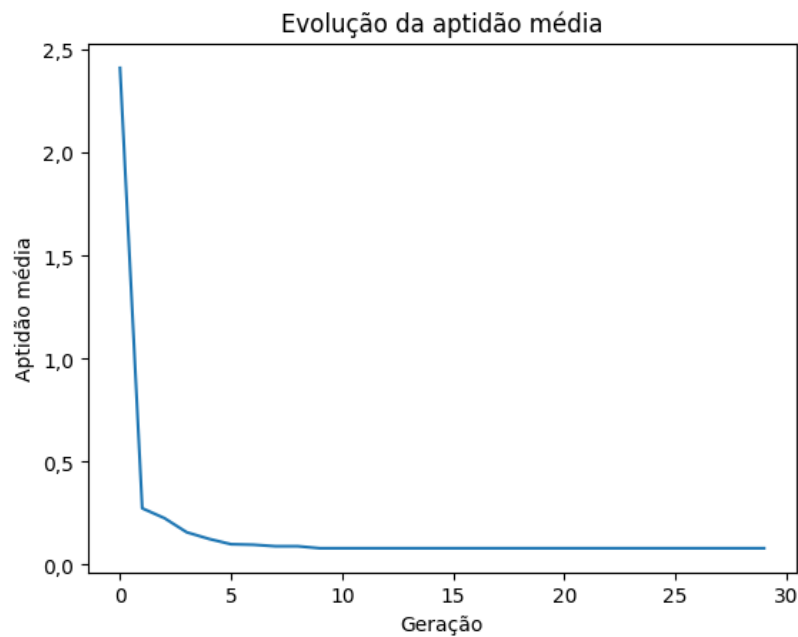


Figura 26 – Convergência de aptidão média da população para uma execução do algoritmo.

e utilização de enlaces. Em seguida, são definidas as demandas que serão alocadas em sequência na rede. Iterativamente, o otimizador é chamado para gerar uma rota para uma demanda na rede, então as informações topológicas são atualizadas, realizando diminuição dos recursos disponíveis nos caminhos já alocados para fluxos admitidos anteriormente. Ao final, são gerados os resultados, compostos pelo conjunto de rotas atribuídas à lista de demandas.

No processo de simulação implementado em Python (conforme descrito no Capítulo 3), foram definidas três demandas para serem inseridas na rede mostrada na Figura 14. As demandas são definidas por um par de hosts de origem e destino, que indicam a direção do fluxo de dados entre os mesmos, além das restrições de qualidade. A Tabela 7 mostra as demandas e seus requisitos, em ordem de atribuição.

Tabela 7 – Demandas de fluxos para alocação de rotas na simulação.

<b>Demanda</b>	<b>Origem</b>	<b>Destino</b>	<b>Banda mínima</b>	<b>Atraso máximo</b>
1	H1	H7	40 Mbps	3 ms
2	H3	H7	40 Mbps	3 ms
3	H5	H4	80 Mbps	5 ms

Os estados dos enlaces são alterados após a alocação das rotas. Cada enlace utilizado em uma rota para um fluxo com determinados requisitos tem como valor de vazão

final a subtração do valor corrente pelo valor utilizado, onde por exemplo, um enlace com capacidade de 100 *Mbps* passa a fornecer 60 *Mbps* após ter 40 *Mbps* alocados para uma demanda. Os valores de atraso foram acrescidos 1 *ms* para cada fluxo admitido, buscando considerar tempos adicionais de fila de encaminhamento para fluxos concorrentes.

O processo de simulação foi executado para gerar rotas para as três demandas, seguindo a ordem definida. A Tabela 8 apresenta as rotas alocadas para cada uma das demandas após execução da simulação.

Tabela 8 – Rotas definidas no processo da simulação.

<b>Demanda</b>	<b>Rota</b>
1	S1 – > S5 – > S6
2	S2 – > S5 – > S6
3	S5 – > S4 – > S1 – > S2

Para a primeira demanda atendida, é possível observar no diagrama da rede (Figura 14), que o menor número de saltos entre origem e destino é três. Como no estado em que o fluxo foi solicitado não havia nenhum fluxo alocado, o algoritmo de otimização retornou um caminho com o menor número de saltos. Para tais situações, o algoritmo sempre tenderá a convergir para um menor caminho, devido à restrição de minimização do número de enlaces utilizados.

Após admissão de um fluxo de dados, com as matrizes de utilização de enlaces e de recursos disponíveis alteradas, os resultados dos processos de otimização realizados pelo algoritmo passam a ser influenciados por tais variáveis, não necessariamente retornando os menores caminhos.

No processamento da segunda demanda, o menor caminho entre origem e destino é composto por dois saltos em dois *switches*, contudo, o algoritmo convergiu para um caminho composto por três saltos, e com um enlace já utilizado pela rota admitida anteriormente. É possível observar que a solução mantém o mínimo necessário de utilização de enlaces na rede como um todo e atende às restrições da demanda.

Ao admitir uma rota para a última demanda, o algoritmo convergiu para uma rota composta por quatro saltos. Considerando a topologia utilizada na simulação, existem outros caminhos com menores saltos, mas que não atendem às restrições. Os demais caminhos, apesar de menores, não possuem banda para atender as restrições.

A estratégia de otimização utilizada, com emprego de um algoritmo genético, se mostrou eficiente no que concerne a definição de rotas para fluxos, o que pôde ser demonstrado na simulação realizada com a geração rotas que atendam as restrições de QoS, e também minimizando a utilização de enlaces na medida do possível. Um ponto importante é que, com a modelagem realizada do otimizador, é possível adicionar facilmente mais parâmetros de QoS, como por exemplo o *jitter*, onde cada novo parâmetro deve ser adicionado nas matrizes de limites e de medidas para os mesmos nos caminhos da rede.

Em um ambiente real de rede, é de responsabilidade do controlador SDN manter as estruturas que definem o estado da rede, tanto de topologia quanto de parâmetros de desempenho, além de manter as informações de fluxos já admitidos para serem considerados na admissão de novos fluxos. Uma solução de controlador pode implementar o modelo de otimização desenvolvido para admissão de fluxos, e precisa também implementar mecanismos para inspecionar os fluxos ao longo de sua existência nas redes, verificando o uso correto dos recursos disponibilizados.

### 4.3 Aplicações

No cenário de gestão de recursos e tráfego dentro de uma rede SDN, várias abordagens podem ser utilizadas para empregar os processos de otimização propostos, o que depende da arquitetura das aplicações desenvolvidas para os controladores.

Dada uma malha complexa de rede e um conjunto definido de regras, o processo de otimização pode ser executado a priori, gerando diversas soluções prévias para diferentes cenários. Na aplicação de otimização em tempo real, os algoritmos também podem ser utilizados. Nesse caso, deve-se definir o critério de parada e os recursos disponíveis para otimização, ainda considerando o tamanho dos problemas, visando a entrega de soluções e rotas em um tempo específico.

Através dos experimentos realizados, foi possível observar a capacidade de exploração dos algoritmos evolucionários nos espaços de busca dos problemas, entregando potencialmente soluções ótimas de rotas para fluxos de comunicação nas redes.

Devido à dinamicidade das redes, a utilização de diferentes configurações em diferentes cenários podem propiciar melhor experiência para o usuário, por meio do desempenho melhorado, bem como a economia de energia, gerando *data centers* mais

sustentáveis com baixo consumo de energia, e contribuindo conseqüentemente para a diminuição da emissão de dióxido de carbono no ambiente.

Adicionar recursos na rede é uma boa estratégia, considerando uma gestão otimizada, onde em cenários de alta demanda, mais estrutura pode ser ativada e, em cenários de demanda reduzida, os recursos podem ser devidamente desabilitados, através de heurísticas que permitam tal ação, como discutido por (Al-Musawi and Al-Khatib, 2019), visando reduzir o consumo de energia nos *data centers*.

## 4.4 Resumo

O presente capítulo apresentou os resultados experimentais obtidos no trabalho e uma série de discussões e conclusões acerca dos mesmos. Primeiramente, foi feita uma avaliação de um conjunto de algoritmos empregados para solução do problema multiobjetivo de otimização de rotas, visando balanceamento de carga e uso eficiente de recursos nas redes. Os algoritmos foram avaliados sob perspectiva da qualidade das soluções geradas e do tempo de processamento gasto. Em seguida, foram feitos alguns experimentos de cenários reais de rede, com simulação e emulação, onde foi possível verificar os efeitos do processo de otimização. Por fim, foi feita uma análise de possíveis abordagens para aplicações das técnicas propostas no presente trabalho.

# Capítulo 5

## Conclusões

No âmbito de redes SDN, balanceamento de carga e eficiência energética são questões relevantes a serem consideradas no processo de estabelecimento de rotas para demandas de tráfego. Em um cenário de rede SDN, um controlador recebe uma demanda de um cliente e então precisa tomar uma decisão sobre qual rota determinar para o fluxo em questão. No caso de SLA, restrições de qualidade são consideradas e, havendo recursos disponíveis, as demandas são admitidas, ou em caso negativo, rejeitadas. Assim, é responsabilidade do provedor realizar contrato de fornecimento de serviços com SLA definido somente considerando recursos disponíveis em sua infraestrutura. No caso de contextos em melhor esforço, é necessário que se realize uma priorização adequada de acordo com o estado atual da rede. O processo de determinação das rotas pode ser visto como um problema de otimização, onde são buscadas soluções de rotas que gerem bom desempenho para os serviços e com uso eficiente de recursos.

Neste segmento, o presente trabalho teve por objetivo modelar, implementar e avaliar modelos de otimização do processo de determinação de rotas solicitadas por um cliente de uma rede SDN, considerando contextos de melhor esforço e SLA, buscando neste último atender restrições de QoS solicitadas utilizando o mínimo de recursos necessários. Para tal, os modelos foram baseados em algoritmos evolucionários, objetivando a geração de soluções de rotas tendo em vista os objetivos de balanceamento de carga, evitando problemas com congestionamento, e utilização mínima de enlaces, visando economia de energia.

Para o cenário de SLA, a abordagem utilizada consistiu em transformar em objetivos as restrições de QoS juntamente com o uso eficiente de recursos, sendo esta

última por meio da minimização da utilização de enlaces ao definir uma rota para uma demanda. Já para o caso de melhor esforço, a estratégia utilizada foi a obtenção de uma fronteira Pareto aproximada de soluções, com objetivos de balanceamento de carga e minimização de enlaces. Para emprego dos algoritmos evolucionários, também foram definidos uma representação de solução, além dos operadores genéticos.

Nas comparações dos algoritmos baseados em obtenção de Paretos, foram utilizadas diferentes instâncias de problemas. Os algoritmos foram executados várias vezes em cada cenário para permitir uma análise estatística sobre seus resultados. Os parâmetros utilizados para as comparações foram os valores de hipervolume das fronteiras obtidas e o tempo médio de execução para as gerações dos algoritmos. Após as análises, o NSGA-II foi escolhido para utilização para obtenção de Paretos. No caso em que é possível trabalhar com definição de preferências a priori, foi utilizada a combinação dos objetivos de modo permitir a realização do processo de otimização mono-objetivo, visando menor tempo de execução do otimizador.

Para validar o funcionamento do otimizador em um cenário de política de melhor esforço, foi feita uma emulação de um ambiente real de rede com Mininet, atribuindo várias demandas em sequência na rede e avaliando os valores médios de banda alcançada pelos fluxos, RTT e percentual de enlaces utilizados na rede em três diferentes cenários: modo *standalone* com STP, priorizando soluções balanceadas e priorizando minimização de enlaces. Nesse processo, foi possível verificar melhoras nos parâmetros de qualidade trabalhados, de acordo com as preferências subjetivas adotadas por um tomador de decisões na escolha das soluções. A utilização de soluções balanceadas proporcionaram um aumento de 38% na banda e uma diminuição de 46% no RTT, ao custo de utilizar 16% a mais de enlaces, quando comparado com o STP. Já a otimização minimizando a utilização de enlaces, comparando com o STP, ainda gerou um ganho de 19% da banda e uma diminuição de 38% do RTT, com utilização equivalente de recursos, se este resultado em decorrência da dinamicidade do processo de otimização na geração das rotas.

O processo de avaliação em um cenário com SLA requer um controlador com uma arquitetura diferente do primeiro experimento. Nesse sentido, visando a validação do algoritmo desenvolvido em um cenário de sucessivas requisições de fluxos em uma rede, foi implementado e executado um processo de simulação em Python. Os resultados apresentados mostraram que o algoritmo tem capacidade de convergir para rotas que

atendam as restrições das demandas de fluxos da rede, garantindo também um uso mínimo necessário de recursos.

Conforme foi percebido na revisão de literatura realizada, trabalhos têm proposto soluções para os problemas de balanceamento de carga e eficiência energética, utilizando diferentes técnicas e produzindo resultados na mesma direção. No que concerne à otimização, os algoritmos evolucionários se mostraram uma boa alternativa, apresentando boas características de convergência e se mostrando robustos na supressão de soluções ineficazes ou com violação de restrições.

Em uma rede SDN, uma estratégia para mitigar eventuais problemas de desempenho consiste em manter um conjunto redundante de recursos e ativá-los conforme necessidade, por meio de um processo de otimização com definição dinâmica de objetivos e restrições, conforme abordado e mostrado nos experimentos realizados.

## 5.1 Limitações e Trabalhos Futuros

O presente trabalho objetivou realizar uma análise dos algoritmos evolucionários empregados no processo de otimização de rotas em redes SDN, verificando a qualidade das soluções geradas, o tempo necessário para execução e as implicações em cenários reais de redes. Para tal, alguns processos de emulação e simulação foram desenvolvidos.

Contudo, o trabalho não apresentou uma arquitetura final para implementação e suas contribuições se limitam às investigações realizadas ao horizonte do processo de otimização utilizando algoritmos evolucionários.

Como trabalhos futuros, poderão ser adicionados na modelagem mais objetivos e restrições, considerando outras possíveis abordagens definidas por um gestor de ativos de uma rede. Além disso, outra questão importante é aprimorar a escalabilidade do processo de otimização para cenários de topologias grandes e com grandes quantidades de demandas, considerando mudanças na arquitetura final da aplicação e utilização de controladores distribuídos. No caso da abordagem com SLA, pretende-se ainda desenvolver um aplicativo de controlador SDN para uma rede e então utilizar o modelo de otimização desenvolvido para determinação das rotas das demandas admitidas. Outro ponto importante é realizar comparações dos algoritmos evolucionários com outras diferentes técnicas de otimização, como heurísticas de otimização em grafos e programação linear.

## 5.2 Produção Bibliográfica

Este trabalho gerou, até o momento, uma publicação em congresso referente aos resultados obtidos para avaliação do impacto de soluções otimizadas no cenário de melhor esforço, onde foi utilizado o Mininet para a emulação da rede e o Ryu como controlador SDN.

O trabalho intitulado "Uma abordagem multiobjetivo para definição de rotas em Redes Definidas por Software (SDN)" ([Assis and Errico, 2020](#)), foi publicado no XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2020).

Em sequência, pretende-se elaborar um artigo contemplando todos os resultados finais do presente trabalho, para publicação em periódico da área.

# Referências

- Abraham, A. and Jain, L. (2005). Evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, pages 1–6. Springer.
- Al-Musawi, R. and Al-Khatib, O. (2019). Optimized energy aware resource allocation algorithm using software defined network technology. *Journal of Telecommunications and Information Technology*, (2):83–91.
- Ali, T. E., Morad, A., and Abdala, M. (2018). Load balance in data center sdn networks. *International Journal of Electrical and Computer Engineering*, 8(5):3086.
- Anatel (2020). Relatório analítico do impacto da pandemia de covid-19 no setor de telecomunicações do brasil. URL: <https://www.gov.br/anatel/pt-br/dados/acompanhamento/relatorios-de-acompanhamento/2020>.
- Assefa, B. G. and Ozkasap, O. (2019a). A novel utility based metric and routing for energy efficiency in software defined networking. In *IEEE International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–4.
- Assefa, B. G. and Ozkasap, O. (2019b). A survey of energy efficiency in sdn: Software-based methods and optimization models. *Journal of Network and Computer Applications*, 137:127–143.
- Assis, S. S. and Errico, L. (2020). Uma abordagem multiobjetivo para definição de rotas em redes definidas por software (sdn). pages 1–5. XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBRT 2020. DOI: 10.14209/SBRT.2020.1570645099.
- Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and Xiao, X. (2002). Overview and principles of internet traffic engineering. Technical report, RFC 3272, may.

- Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and McManus, J. (1999). Requirements for traffic engineering over mpls.
- Bader, J. and Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76.
- Benítez-Hidalgo, A., Nebro, A. J., García-Nieto, J., Oregi, I., and Ser, J. D. (2019). jmetalpy: A python framework for multi-objective optimization with metaheuristics. *Swarm and Evolutionary Computation*, page 100598.
- Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., et al. (2014). Onos: towards an open, distributed sdn os. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 1–6.
- Carvalho, P. D. O. and Errico, L. (2013). Um sistema de engenharia de tráfego autogerenciável para redes ip. Master's thesis, Universidade Federal de Minas Gerais.
- Chakravarthy, V. D. and Amutha, B. (2019). Path based load balancing for data center networks using sdn. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(4):3279–3285.
- Chen, Y., Farley, T., and Ye, N. (2004). Qos requirements of network applications on the internet. *Information Knowledge Systems Management*, 4(1):55–76.
- Coello, C. A. C., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.
- Collette, Y. and Siarry, P. (2004). *Multiobjective optimization: principles and case studies*. Springer Science & Business Media.
- Deb, K. and Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

- Dewanto, R., Munadi, R., and Negara, R. M. (2018). Improved load balancing on software defined network-based equal cost multipath routing in data center network. *JURNAL INFOTEL*, 10(3):157–162.
- Farhady, H., Lee, H., and Nakao, A. (2015). Software-defined networking: A survey. *Computer Networks*, 81:79–95.
- Fernández-Fernández, A., Cervelló-Pastor, C., and Ochoa-Aday, L. (2017). A multi-objective routing strategy for qos and energy awareness in software-defined networks. *IEEE Communications Letters*, 21(11):2416–2419.
- Fielding, R. T. and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2):115–150.
- Handigol, N., Heller, B., Jeyakumar, V., Lantz, B., and McKeown, N. (2012). Reproducible network experiments using container-based emulation. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 253–264.
- Huang, T., Yu, F. R., Zhang, C., Liu, J., Zhang, J., and Liu, Y. (2016). A survey on large-scale software defined networking (sdn) testbeds: Approaches and challenges. *IEEE Communications Surveys & Tutorials*, 19(2):891–917.
- IEEE (2004). Ieee standard for local and metropolitan area networks: Media access control (mac) bridges. *IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998)*, pages 1–281.
- IEEE (2015). Ieee standard for local and metropolitan area networks– station and media access control connectivity discovery corrigendum 2: Technical and editorial corrections. *IEEE Std 802.1AB-2009/Cor 2-2015 (Corrigendum to IEEE Std 802.1AB-2009)*, pages 1–68.
- Internet World Stats (2020a). Internet growth statistics. URL: <https://www.internetworldstats.com/emarketing.htm>.
- Internet World Stats (2020b). World internet usage and population statistics. URL: <https://www.internetworldstats.com/stats.htm>.

- Islam, M. T., Islam, N., and Al Refat, M. (2020). Node to node performance evaluation through ryu sdn controller. *Wireless Personal Communications*, 112:555–570.
- Jaglarz, P., Boryło, P., Szymański, A., and Chołda, P. (2020). Enhanced lagrange decomposition for multi-objective scalable te in sdn. *Computer Networks*, 167:106992.
- Jones, G. (1998). Genetic and evolutionary algorithms. *Encyclopedia of Computational Chemistry*, 2:1127–1136.
- Keller, A. and Ludwig, H. (2003). The wsla framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management*, 11(1):57–81.
- Kim, J. L. (2013). Genetic algorithm stopping criteria for optimization of construction resource scheduling problems. *Construction Management and Economics*, 31(1):3–19.
- Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 1–6.
- Lavado, T. (2020). Com maior uso da internet durante pandemia, número de reclamações aumenta; especialistas apontam problemas mais comuns. URL: <https://g1.globo.com/economia/tecnologia/noticia/2020/06/11/com-maior-uso-da-internet-durante-pandemia-numero-de-reclamacoes-aumenta-especialistas-apontam-problemas-mais-comuns.ghtml>.
- Maaloul, R., Chaari, L., and Cousin, B. (2018). Energy saving in carrier-grade networks: A survey. *Computer Standards & Interfaces*, 55:8–26.
- Markiewicz, A., Tran, P. N., and Timm-Giel, A. (2014). Energy consumption optimization for software defined networks considering dynamic traffic. In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pages 155–160.
- Masanet, E., Shehabi, A., Lei, N., Smith, S., and Koomey, J. (2020). Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74.

- Montgomery, D. C. and Runger, G. C. (2010). *Applied statistics and probability for engineers*. John Wiley & Sons.
- Morita, K., Yamahata, I., and Linux, V. (2012). Ryu: Network operating system. In *OpenStack Design Summit & Conference*.
- Nebro, A. J., Durillo, J. J., Luna, F., Dorronsoro, B., and Alba, E. (2009). Mocell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems*, 24(7):726–746.
- Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., and Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, 16(3):1617–1634.
- Open Network Foundation (2012a). Openflow v1.3.0. URL: <https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf>.
- Open Network Foundation (2012b). Software-defined networking: The new norm for networks. *ONF white paper*, pages 1–12.
- Orlowski, S., Wessälly, R., Pióro, M., and Tomaszewski, A. (2010). Sndlib 1.0—survivable network design library. *Networks: An International Journal*, 55(3):276–286.
- Pfaff, B., Pettit, J., Koponen, T., Jackson, E., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., Amidon, K., and Casado, M. (2015). The design and implementation of open vswitch. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 117–130, Oakland, CA. USENIX Association.
- Rupani, K., Punjabi, N., Shamdasani, M., and Chaudhari, S. (2020). Dynamic load balancing in software-defined networks using machine learning. In *Proceeding of International Conference on Computational Science and Applications*, pages 283–292. Springer.
- Santos, B. P., Alberto, A., Lima, T. D. F. M., and Charrua-Santos, F. M. B. (2018). Indústria 4.0: desafios e oportunidades. *Revista Produção e Desenvolvimento*, 4(1):111–124.
- Shirmarz, A. and Ghaffari, A. (2020). Performance issues and solutions in sdn-based data center: A survey. *The Journal of Supercomputing*, 76:7545–7593.

- Staessens, D., Sharma, S., Colle, D., Pickavet, M., and Demeester, P. (2011). Software defined networking: Meeting carrier grade requirements. In *2011 18th IEEE workshop on local & metropolitan area networks (LANMAN)*, pages 1–6. IEEE.
- Verma, D. C. (2004). Service level agreements on ip networks. *Proceedings of the IEEE*, 92(9):1382–1388.
- Wortmann, F. and Flüchter, K. (2015). Internet of things. *Business & Information Systems Engineering*, 57(3):221–224.
- Zhu, L., Karim, M. M., Sharif, K., Li, F., Du, X., and Guizani, M. (2019). Sdn controllers: Benchmarking & performance evaluation. *CoRR abs/1902.04491*. <http://arxiv.org/abs/1902.04491>.
- Zitzler, E. and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *International conference on parallel problem solving from nature*, pages 832–842. Springer.
- Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103.
- Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271.

## Apêndice A

### Resultados dos Testes de Tukey

## Resultados teste de Tukey - Configuração com 15 nós

Par	Diferença	SE	Q	CI Inferior	CI Superior	Média Crítica	p-valor
HYPE-IBEA	1,4E-04	1,8E-03	0,07	-0,0072	0,0075	0,0073810	1,0000
HYPE-MOCELL	6,1E-03	1,8E-03	3,36	-0,0013	0,0135	0,0073810	0,1697
HYPE-NSGA-II	1,0E-03	1,8E-03	0,55	-0,0064	0,0084	0,0073810	0,9988
HYPE-NSGA-III	4,3E-03	1,8E-03	2,35	-0,0031	0,0116	0,0073810	0,5600
HYPE-SPEA2	3,8E-03	1,8E-03	2,07	-0,0036	0,0111	0,0073810	0,6870
IBEA-MOCELL	6,2E-03	1,8E-03	3,44	-0,0012	0,0136	0,0073810	0,1515
IBEA-NSGA-II	8,6E-04	1,8E-03	0,48	-0,0065	0,0082	0,0073810	0,9994
IBEA-NSGA-III	4,1E-03	1,8E-03	2,27	-0,0033	0,0115	0,0073810	0,5948
IBEA-SPEA2	3,6E-03	1,8E-03	2,00	-0,0038	0,0110	0,0073810	0,7196
MOCELL-NSGA-II	7,1E-03	1,8E-03	3,91	-0,0003	0,0145	0,0073810	0,0678
MOCELL-NSGA-III	1,0E-02	1,8E-03	5,71	0,0030	0,0177	0,0073810	0,0011
MOCELL-SPEA2	9,8E-03	1,8E-03	5,43	0,0025	0,0172	0,0073810	0,0023
NSGA-II-NSGA-III	3,3E-03	1,8E-03	1,80	-0,0041	0,0106	0,0073810	0,8003
NSGA-II-SPEA2	2,8E-03	1,8E-03	1,52	-0,0046	0,0101	0,0073810	0,8903
NSGA-III-SPEA2	5,0E-04	1,8E-03	0,28	-0,0069	0,0079	0,0073810	1,0000

## Resultados teste de Tukey - Configuração com 10 nós

Par	Diferença	SE	Q	CI Inferior	CI Superior	Média Crítica	p-valor
HYPE-IBEA	4,1E-03	2,0E-03	2,03	-0,0041	0,0122	0,00815963	0,7042
HYPE-MOCELL	8,6E-03	2,0E-03	4,30	0,0005	0,0168	0,00815963	0,0320
HYPE-NSGA-II	1,4E-03	2,0E-03	0,70	-0,0068	0,0096	0,00815963	0,9964
HYPE-NSGA-III	3,4E-04	2,0E-03	0,17	-0,0078	0,0085	0,00815963	1,0000
HYPE-SPEA2	1,3E-04	2,0E-03	0,06	-0,0080	0,0083	0,00815963	1,0000
IBEA-MOCELL	1,3E-02	2,0E-03	6,33	0,0045	0,0208	0,00815963	0,0002
IBEA-NSGA-II	5,5E-03	2,0E-03	2,73	-0,0027	0,0136	0,00815963	0,3876
IBEA-NSGA-III	4,4E-03	2,0E-03	2,20	-0,0038	0,0126	0,00815963	0,6278
IBEA-SPEA2	3,9E-03	2,0E-03	1,97	-0,0042	0,0121	0,00815963	0,7311
MOCELL-NSGA-II	7,2E-03	2,0E-03	3,60	-0,0009	0,0154	0,00815963	0,1159
MOCELL-NSGA-III	8,3E-03	2,0E-03	4,13	0,0001	0,0164	0,00815963	0,0449
MOCELL-SPEA2	8,7E-03	2,0E-03	4,36	0,0006	0,0169	0,00815963	0,0282
NSGA-II-NSGA-III	1,1E-03	2,0E-03	0,53	-0,0071	0,0092	0,00815963	0,9991
NSGA-II-SPEA2	1,5E-03	2,0E-03	0,76	-0,0066	0,0097	0,00815963	0,9946
NSGA-III-SPEA2	4,6E-04	2,0E-03	0,23	-0,0077	0,0086	0,00815963	1,0000