

GARL-TE: Uma Abordagem Híbrida de Algoritmo Genético e Aprendizado por Reforço para Engenharia de Tráfego em Redes SDN

Carlos Augusto R. Soares¹, Jorge Luís M. do Amaral¹,
Alexandre Sztajnberg¹

¹ Programa de Pós-Graduação em Ciências Computacionais e Modelagem Matemática (PPG-COMPMAT) – Instituto de Matemática e Estatística (IME)
Universidade do Estado do Rio de Janeiro (UERJ)
Rua São Francisco Xavier, 524, Maracanã, Rio de Janeiro, RJ, Brasil

caugusto.rio@gmail.com, jamaral@uerj.br, alexszt@ime.uerj.br

Abstract. *Minimizing Maximum Link Utilization (MLU) is a core challenge in Traffic Engineering (TE), requiring flow allocation strategies that balance efficiency and operational cost. Traditional shortest-path protocols with static balancing (IGP+ECMP) often fail to mitigate congestion under asymmetric traffic matrices, while exact optimization models via Linear Programming (LP) face scalability issues and high computational costs. This paper introduces GARL-TE, a hybrid architecture for Software-Defined Networks (SDN) based on the principle of selective intervention. The proposal combines the global search capability of a specialized Genetic Algorithm (GA) with the responsiveness of Deep Reinforcement Learning (DRL) to optimize only a subset of critical flows. GARL-TE proposal replaces on-line LP solvers with an evolutionary engine based on adaptive Dirichlet mutation. In a controlled evaluation of the optimization module, near-optimal results and statistical robustness were observed, indicating the potential of the architecture within autonomic control loops.*

Resumo. *A minimização da máxima utilização dos enlaces (MLU) constitui um desafio central na Engenharia de Tráfego (TE), exigindo estratégias de alocação de fluxos que equilibrem eficiência e custos operacionais. Protocolos baseados em caminhos de menor custo e balanceamento estático (IGP+ECMP) frequentemente falham em mitigar congestionamentos em matrizes de tráfego assimétricas, enquanto modelos de otimização exata via Programação Linear (PL) enfrentam limitações de escalabilidade e elevado custo computacional. Este artigo apresenta o GARL-TE, uma arquitetura híbrida para Redes Definidas por Software (SDN) fundamentada no princípio de intervenção seletiva. A proposta combina a capacidade de busca global de um Algoritmo Genético (GA) especializado com a responsividade do Aprendizado por Reforço Profundo (DRL) para otimizar apenas um subconjunto de fluxos críticos. GARL-TE prevê a substituição de solucionadores de PL on-line por um motor evolutivo baseado em mutação de Dirichlet adaptativa. Em avaliação controlada, realizada sobre o módulo de otimização, observaram-se resultados próximos ao limite ótimo e indícios de robustez estatística na mitigação de gargalos, o que reforça o potencial da arquitetura em ciclos de controle autônomo.*

1. Introdução

O crescimento exponencial do volume de dados e a dinamicidade das aplicações modernas exigem que as redes de computadores operem de forma altamente eficiente. Nesse contexto, a Engenharia de Tráfego (*Traffic Engineering*, TE) desempenha um papel crítico ao direcionar fluxos de dados para evitar congestionamentos e otimizar o uso da infraestrutura. O objetivo central de grande parte das estratégias de TE é minimizar a Máxima Utilização dos Enlaces (MLU – *Maximum Link Utilization*), garantindo que nenhum caminho da rede se torne um gargalo enquanto outros permanecem ociosos [Fortz and Thorup 2002]

Considerando mecanismos usuais, o roteamento e o balanceamento de carga entre rotas alternativas dependem de protocolos tradicionais e de mecanismos como o *Equal-Cost Multi-Path* (ECMP). Contudo, o ECMP é agnóstico ao estado de congestionamento da rede e divide o tráfego apenas entre caminhos de custo idêntico, o que frequentemente resulta em valores de MLU subótimos em topologias assimétricas. Por outro lado, utilizando técnicas de otimização exata, como a Programação Linear (PL), é possível minimizar a MLU com extrema precisão; no entanto, essas abordagens esbarram em severos limites de escalabilidade computacional, tornando-se inviáveis para aplicações e ajustes online em redes dinâmicas de grande porte.

Para superar o alto custo da otimização matemática e a ineficiência de protocolos estáticos, este artigo propõe o **GARL-TE** (*Genetic Algorithm and Reinforcement Learning for Traffic Engineering*). A arquitetura híbrida combina a busca global de um Algoritmo Genético com a adaptabilidade de um agente DRL. Essa abordagem é adequada para a alta dimensionalidade de redes SDN, onde métodos de RL tabular são inviáveis para decisões em tempo real.

O sistema proposto se insere no contexto das Redes Definidas por Software (SDN), que oferece tanto a visibilidade global necessária para a coleta de dados de telemetria quanto as APIs padronizadas para atuar remotamente na infraestrutura [Kreutz et al. 2014]. Além disso, a proposta também se insere como componente de um ciclo de controle autônomo do tipo *MAPE-K* (*Monitor, Analyze, Plan, Execute, Knowledge*). Nesse contexto, o GARL-TE é concebido para atuar nos estágios de análise e planejamento, recebendo dados do monitoramento SDN para extrair conhecimento da rede e produzindo ações de reconfiguração de fluxos a serem encaminhadas ao estágio de execução via SDN.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta o problema de Engenharia de Tráfego e o papel de SDN; a Seção 3 discute a literatura correlata; a Seção 4 descreve a arquitetura proposta; a Seção 5 apresenta o desenho da avaliação inicial e detalha os cenários experimentais; a Seção 6 discute os resultados; e a Seção 7 traz as conclusões e os trabalhos futuros.

2. Engenharia de Tráfego e SDN

A Engenharia de Tráfego visa otimizar o desempenho de redes por meio da alocação eficiente de recursos. Neste trabalho, a rede é modelada como um grafo direcionado $G = (V, E)$, onde V representa o conjunto de nós e E o conjunto de enlaces. Cada enlace $e \in E$ possui uma capacidade nominal C_e . O objetivo central é acomodar uma Matriz de

Tráfego (TM), composta por um conjunto de demandas $D = \{d_1, d_2, \dots, d_K\}$, onde cada demanda d_k representa o volume de tráfego entre um par origem-destino (s_k, t_k) .

A métrica-alvo de otimização é a Utilização Máxima de Enlace (*Maximum Link Utilization*, MLU), denotada por ρ_{max} . Formalmente, a utilização de um enlace e é a razão entre a carga total f_e e sua capacidade C_e . O problema de minimização da MLU pode ser expresso como:

$$\text{Minimizar } \rho_{max} = \max_{e \in E} \left(\frac{f_e}{C_e} \right) \quad (1)$$

sujeito à restrição de conservação de fluxo para cada nó $v \in V$ e cada demanda d_k :

$$\sum_{j:(i,j) \in E} x_{ij}^k - \sum_{j:(j,i) \in E} x_{ji}^k = \begin{cases} 1, & \text{se } i = s_k \\ -1, & \text{se } i = t_k \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

onde x_{ij}^k representa a fração da demanda d_k que atravessa o enlace (i, j) .

Em redes SDN, a capacidade de dividir o tráfego entre múltiplos caminhos (*flow-splitting*) permite que o controlador orquestre o encaminhamento para mitigar gargalos locais. Como estratégia, diversas soluções têm buscado aproximar o desempenho da rede do limite teórico da Programação Linear. No caso deste trabalho, o GARL-TE é proposto com esse mesmo objetivo.

3. Trabalhos Relacionados

A otimização de rotas para minimizar o MLU é um problema clássico. Abordagens baseadas em ECMP são o padrão da indústria devido à sua simplicidade, mas falham por não considerarem o uso efetivo dos enlaces. Soluções baseadas em otimização matemática, como PL, calculam o limite ótimo teórico, porém exigem alto custo computacional.

Com a maturidade das redes SDN, diversas pesquisas têm focado em como essas tecnologias podem potencializar a TE [Mendiola et al. 2017]. O uso de Algoritmos Genéticos (GA) e outras metaheurísticas, como a otimização por colônia de formigas [Hamdan et al. 2021], ganhou força por permitir a otimização multiobjetivo, embora o processo iterativo possa introduzir latência. Em outra linha, o Aprendizado por Reforço Profundo (DRL) tem sido pesquisado para TE autônomo. O *EfficientTE* [Pei et al. 2024] utiliza DRL para ajustar capacidades virtuais e mostrou-se capaz de atingir uma razão de até 84,1% do limite ótimo ideal da PL, a depender da topologia. Já o *RedTE* [Gui et al. 2024] foca na mitigação de rajadas de subsegundo, alcançando entre 91% e 95% do ótimo com latência inferior a 100 ms via arquitetura distribuída.

Abordagens recentes também tratam da redução de distúrbios na rede durante reconfigurações [Ye et al. 2022] e da migração gradual para SDN. O framework *SMART* [Tan et al. 2025] emprega DRL para definir sequências de migração, alcançando o desempenho de uma rede puramente SDN com a conversão de apenas 36% a 52% dos nós. Ao contrário de abordagens isoladas ou centradas na atuação sobre fluxos seletivos, como o CFR-RL [Zhang et al. 2020], a proposta do *GARL-TE* contempla a integração de um Algoritmo Genético (GA) para o refinamento das decisões de otimização. Em uma avaliação

independente desse módulo, conduzida no ambiente *TesteGA*, observaram-se desempenhos equivalentes a 97,63% do limite teórico na rede Abilene e a 99,74% na topologia PosA, sem a dependência de *solvers* de Programação Linear em tempo de execução.

4. Arquitetura Proposta: GARL-TE

A arquitetura do *GARL-TE* é projetada para integrar-se a fluxos de gestão autônoma, podendo ser contextualizada dentro de um ciclo de controle *MAPE-K*, conforme ilustrado na Figura 1. Nesta perspectiva, o modelo *MAPE-K* serve como uma referência ilustrativa para descrever como a proposta consome dados de monitoramento e devolve planos de reconfiguração. O foco central deste trabalho reside no núcleo decisório híbrido (RL+GA), que atua especificamente nas fases de análise e planejamento do ciclo, utilizando a visibilidade e a capacidade de atuação do plano de controle SDN para otimizar manobras de Engenharia de Tráfego [Kephart and Chess 2003, Granville et al. 2015].

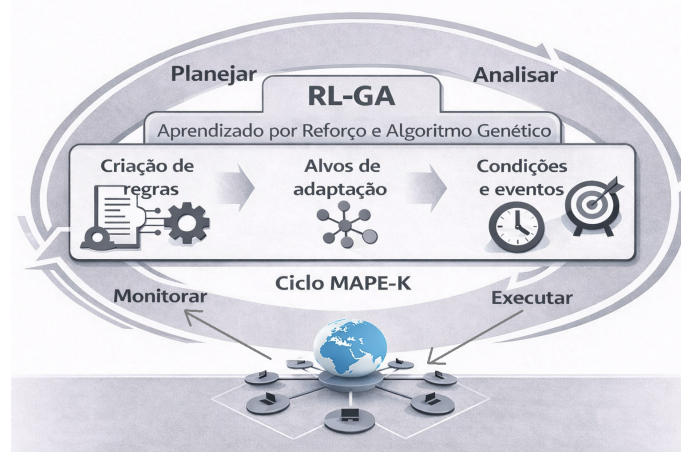


Figura 1. Integração do GARL-TE ao ciclo autônomo do MAPE-K.

A base de conhecimento (*Knowledge*) é estruturada como *Matriz de Tráfego* (TM), que sintetiza a demanda agregada de todos os pares origem-destino (s, d) da topologia. Para um intervalo de observação t , a TM é formalmente definida como:

$$TM_t = \{D_{s,d}(t) \mid s, d \in V, s \neq d\} \quad (3)$$

onde $D_{s,d}(t)$ denota o volume de tráfego a ser acomodado entre os nós. Observa-se que, em cenários operacionais, a TM não é um dado diretamente observável; sua construção exige uma etapa de síntese que processa telemetria bruta (proveniente de MIBs (*Management Information Bases*) via SNMP (*Simple Network Management Protocol*) ou de registros de fluxo via NetFlow/IPFIX) para inferir as demandas fim a fim.

Na arquitetura proposta para o GARL-TE, os módulos *Analyze* e *Plan* são concebidos para operar de forma integrada. Nesse contexto, o agente de DRL deve identificar os fluxos críticos com o auxílio do Algoritmo Genético, que pode fornecer um sinal de recompensa inversamente proporcional à MLU (ρ_{max}). A partir dessa identificação, o Algoritmo Genético pode estimar a melhor acomodação para esses fluxos, definindo as frações de encaminhamento (flow-splitting) e produzindo ações de execução potencialmente mais robustas e próximas do limite teórico de otimização.

Diferentemente de operadores de mutação genéricos, o motor de otimização proposto foi concebido com base na mutação *dirichlet_adaptive*. Esta técnica modela as frações de encaminhamento de um par origem-destino (OD) como amostra de uma distribuição de Dirichlet, garantindo nativamente a restrição de soma unitária. A perturbação do vetor de frações atual \mathbf{x} gera um novo vetor $\mathbf{x}' \sim \text{Dirichlet}(\boldsymbol{\alpha})$, onde os parâmetros α_i evoluem com a geração atual g :

$$\alpha_i = x_i \cdot \left[C_{base} \cdot \left(1.0 + \frac{g}{G} \cdot k \right) \right] + \epsilon \quad (4)$$

Essa formulação favorece uma exploração mais abrangente nas fases iniciais do algoritmo (baixo fator de concentração) e um refinamento de maior precisão nas fases finais (alto fator de concentração). Nos experimentos conduzidos com o módulo de otimização em ambiente controlado, essa estratégia contribuiu para a convergência do GA a valores de MLU próximos ao ótimo teórico da Programação Linear.

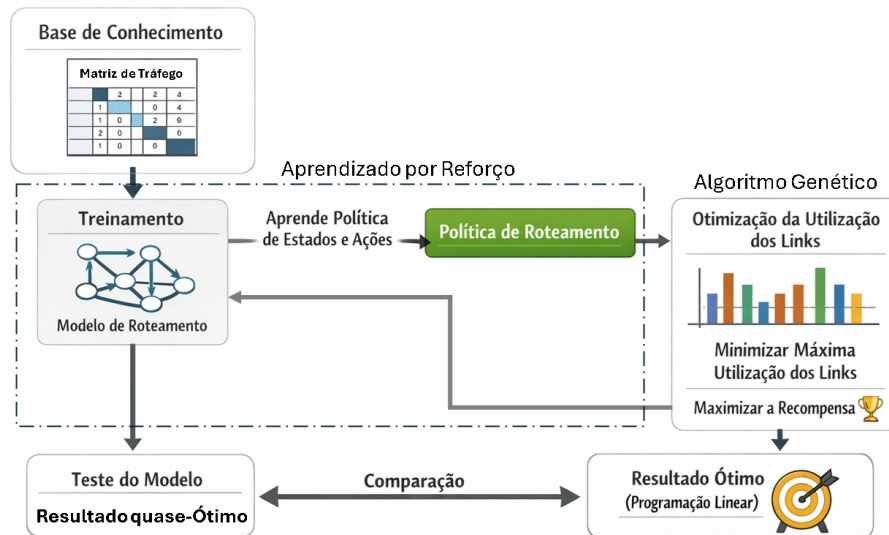


Figura 2. Arquitetura híbrida GARL-TE

A estrutura da Figura 2 apresenta a organização do processamento de dados desde a coleta de telemetria bruta até a efetiva reconfiguração do plano de dados. O fluxo inicia-se no monitor de rede, responsável pela síntese da Matriz de Tráfego instantânea que subsidia o núcleo decisório. Como já discutido, a proposta do *GARL-TE* fundamenta-se no princípio de intervenção seletiva: o agente de DRL é responsável por identificar o subconjunto de fluxos críticos que mais contribuem para o congestionamento, enquanto o Algoritmo Genético é empregado para planejar as frações de encaminhamento (*flow-splitting*) especificamente para essas demandas. Espera-se que essa cooperação reduza o espaço de busca do otimizador metaheurístico, favorecendo reconfigurações rápidas e precisas com mitigação da MLU e baixo distúrbio operacional.

5. Avaliação Inicial

Para a avaliação inicial, foi desenvolvido um ambiente de testes denominado “TesteGA”, concebido para avaliar, de forma isolada, o módulo de otimização baseado em Algoritmo Genético, futuramente passível de incorporação ao contexto do *GARL-TE*. Nesse

ambiente, incluíram-se também, como base para comparação, outros dois métodos de otimização: (i) um *solver* de Programação Linear (PL) e (ii) o protocolo *Equal-Cost Multi-Path* (ECMP). A PL fornece o limite ótimo teórico (teto de desempenho), permitindo medir o quão próximo do ótimo o motor de busca do GA proposto consegue convergir. Por sua vez, o ECMP, em conjunto com um protocolo de roteamento, são amplamente usados em redes IP, servindo como a referência de desempenho de redes convencionais. Na avaliação, utilizaram-se duas topologias de rede: uma sintética, denominada “PosA”, e a rede Abilene. Como entrada para esses experimentos, produziu-se uma matriz de tráfego específica para a topologia “PosA”, apresentada na Tabela 1. Para a Abilene, utilizou-se uma matriz extraída do ambiente de testes, proveniente de conjunto pré-armazenado [Roughan 2004].

A configuração de operadores e parâmetros do GA, validada sob o rigor das topologias de teste (detalhadas na Seção 5.1), será integrada à implementação final do GARL-TE. Esse procedimento assegura que o motor de busca global possua a precisão necessária para prover recompensas estatisticamente estáveis ao agente de DRL. Embora este trabalho foque na validação do módulo evolutivo, para a integração final no GARL-TE, preveem-se parâmetros de *Deep Reinforcement Learning* (DRL) como taxa de aprendizado $\alpha = 0,001$, fator de desconto $\gamma = 0,95$ e estratégia ϵ -greedy para exploração, utilizando redes neurais profundas para lidar com a alta dimensionalidade da topologia.

O motor de otimização evolutiva foi parametrizado especificamente para a minimização da MLU. Os passos fundamentais do processo evolutivo constam no Algoritmo 1. Inicialmente, a população é composta por vetores de frações aleatórios que respeitam a conservação de fluxo (Linha 1). O ciclo iterativo (Linhas 2–12) busca o refinamento das soluções. Na etapa de avaliação (Linhas 3–5), a função de *fitness* é definida como uma composição multiobjetivo que integra o inverso da MLU a três mecanismos de penalização: (i) **violação de capacidade**; (ii) **desbalanceamento de fluxo**; e (iii) **dispersão Top-K**, garantindo estabilidade operacional. A seleção por torneio e o *crossover* (Linha 6) promovem a recombinação de rotas promissoras. O diferencial da proposta reside na mutação de Dirichlet adaptativa (Linhas 7–10), que atua como um “salto” de exploração: o fator de concentração $C(g)$ aumenta com as gerações, permitindo uma busca global abrangente no início e um ajuste fino de alta precisão ao final. Por fim, o elitismo (Linha 11) preserva as melhores soluções encontradas para a próxima geração.

Para garantir a robustez e a reprodutibilidade dos resultados, a análise estatística baseou-se no teste não paramétrico de Wilcoxon, adotando-se um nível de significância de $\alpha = 0,05$. A partir desse parâmetro, foram construídos intervalos de confiança de 95% para as médias de MLU obtidas pelo GA e pelo ECMP.

5.1. Cenários Experimentais

As topologias selecionadas para validar o Módulo de Otimização baseado em Algoritmo Genético (GA), e os módulos de comparação, oferecem diferentes níveis de complexidade, permitindo avaliar desde a correção matemática em cenários controlados até a escalabilidade em redes de *backbone*.

Algorithm 1 Algoritmo Genético com Mutação Dirichlet Adaptativa

Entrada: Grafo G , Matriz de Tráfego TM , Conjunto de caminhos P , Máximo de Gerações T_{max}

Saída: Frações de roteamento que minimizam ρ_{max}

- 1: Inicializar População P_{pop} com N indivíduos (vetores de frações por par OD)
 - 2: **while** geração $g \leq T_{max}$ **do**
 - 3: **for all** indivíduo $i \in P_{pop}$ **do**
 - 4: Calcular fitness $F_i = 1/(\rho_{max,i} + \text{penalidades})$
 - 5: **end for**
 - 6: Selecionar pais via Torneio e aplicar Crossover
 - 7: **Mutação Dirichlet Adaptativa:**
 - 8: $C(g) = C_{base} \cdot (1.0 + \frac{g}{T_{max}} \cdot 20)$ {Concentração aumenta com g }
 - 9: $\alpha = \mathbf{x} \cdot C(g) + \epsilon$
 - 10: Amostrar novas frações $\mathbf{x}' \sim \text{Dirichlet}(\alpha)$
 - 11: Atualizar População P_{pop} com Elitismo
 - 12: $g \leftarrow g + 1$
 - 13: **end while**
 - 14: **return** Melhor indivíduo (frações de menor MLU)
-

5.1.1. Topologia PosA

A topologia *PosA*, apresentada na Figura 3, consiste em uma malha sintética estruturada como um dígrafo de 5 nós e 16 arestas direcionadas. Apesar de sua escala reduzida, esta configuração é fundamental para a validação do modelo, permitindo a emulação de fluxos de tráfego complexos e a identificação de gargalos em um ambiente em que as variáveis de rede são totalmente controláveis. Sua principal função é servir como prova de conceito para observar o comportamento do roteamento sob múltiplas demandas simultâneas.

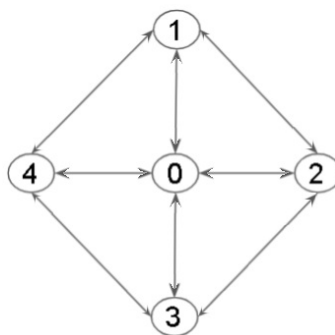


Figura 3. Topologia PosA – Estudo de caso com 5 nós.

5.1.2. Topologia Abilene

A rede *Abilene* (Figura 4) representa uma infraestrutura de *backbone* amplamente adotada como *benchmark* padrão em estudos de Engenharia de Tráfego. Composta por 12 nós e 30 enlaces bidirecionais, esta topologia emula um cenário de rede de longa distância (WAN) de escala intermediária. Sua inclusão neste trabalho é fundamental para avaliar a

escalabilidade do motor de otimização. A utilização de matrizes sintéticas densas, inspiradas nos dados estruturais desta rede [Zhang 2004], permite avaliar o comportamento do algoritmo em cenários de estresse e congestionamentos prolongados.

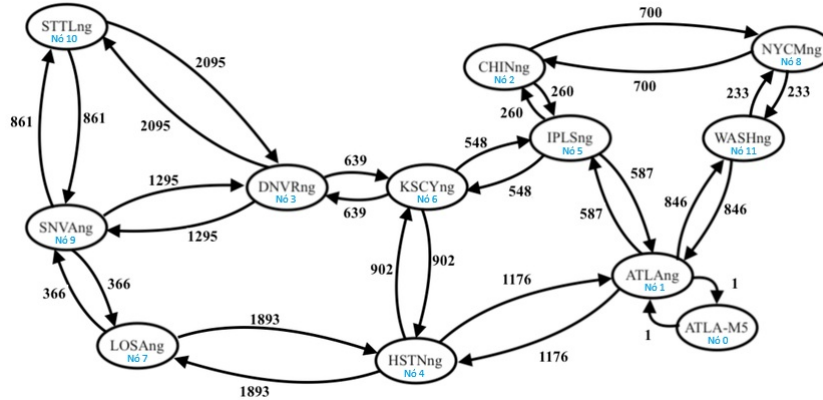


Figura 4. Topologia da rede Abilene (EUA) utilizada como cenário experimental.

6. Resultados e Análise Estatística

Esta seção apresenta os experimentos conduzidos para validar exclusivamente o Módulo de Otimização (GA). É importante reiterar que os resultados aqui expostos foram obtidos pelo programa de *benchmark* independente (ECMP, PL e GA), utilizando matrizes de tráfego sintéticas produzidas pelo TMGEN, o que serve como validação prévia deste módulo antes de sua integração na arquitetura final do GARL-TE.

6.1. Validação Controlada: Topologia PosA

A topologia *PosA* foi adotada como estudo de caso inicial para validar a convergência do algoritmo em um cenário controlado de 5 nós e 12 enlaces. O cenário experimental configurou uma população de 1200 indivíduos ao longo de 250 gerações. Os operadores genéticos foram definidos com taxa de *crossover* de 0,8, seleção por torneio com tamanho $k = 3$ e a mutação de Dirichlet adaptativa já descrita. Esses valores foram escolhidos para equilibrar a exploração do espaço de busca com a velocidade de convergência. Para este teste, as demandas de tráfego foram distribuídas de forma homogênea entre todos os pares origem-destino, conforme detalhado na Tabela 1. Nesta matriz, cada célula representa o volume de tráfego agregado (em unidades de fluxo) a ser transportado de um nó de origem (linha) para um nó de destino (coluna). A escolha de uma carga uniforme (valor 2 para todos os pares $s \neq d$) permite avaliar a precisão matemática do motor de otimização em um cenário de simetria perfeita, onde o algoritmo deve ser capaz de distribuir o tráfego de forma equitativa sem privilegiar rotas específicas, servindo como uma validação fundamental da lógica de realimentação.

Os resultados de convergência de MLU para a *PosA* indicam que o Algoritmo Genético (GA) alcançou uma MLU média de 3,0081, com desvio-padrão igual a 0,0041. O comportamento do algoritmo frente aos *baselines* e sua evolução ao longo das gerações são ilustrados nas Figuras 5 e 6, respectivamente. Observa-se na Figura 5 que o GA estabiliza em um patamar de utilização próximo ao limite teórico (PL), superando o ECMP.

Tabela 1. Matriz de demandas homogêneas para a topologia PosA.

Origem \ Destino	Nó 0	Nó 1	Nó 2	Nó 3	Nó 4
Nó 0	0	2	2	2	2
Nó 1	2	0	2	2	2
Nó 2	2	2	0	2	2
Nó 3	2	2	2	0	2
Nó 4	2	2	2	2	0

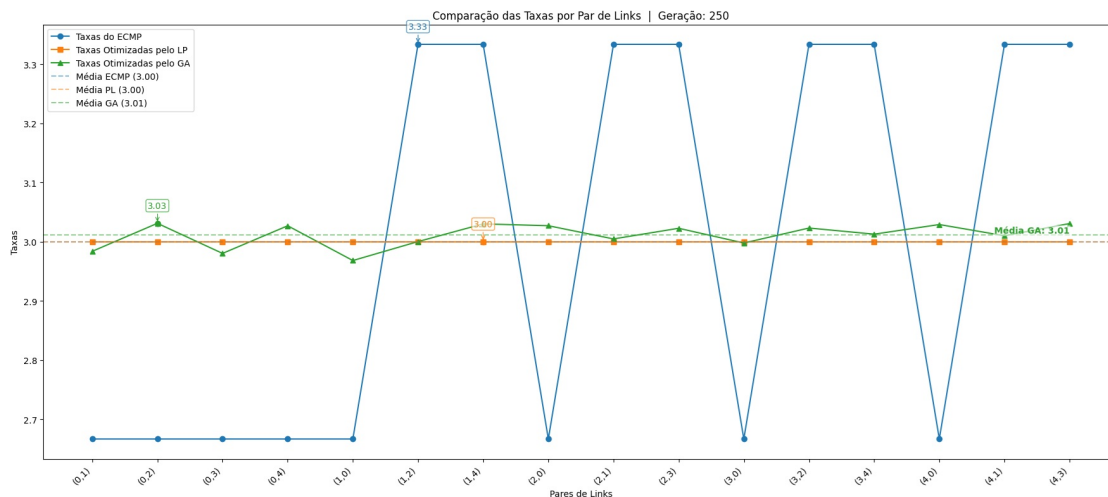


Figura 5. PosA – comparativo de MLU entre ECMP, PL e o método proposto (GA).

A Figura 6 detalha a evolução do *fitness*, evidenciando a descida acentuada nas gerações iniciais e a estabilização em torno da geração 150, o que demonstra a eficiência do fator de concentração da mutação de Dirichlet para realizar o ajuste fino local.

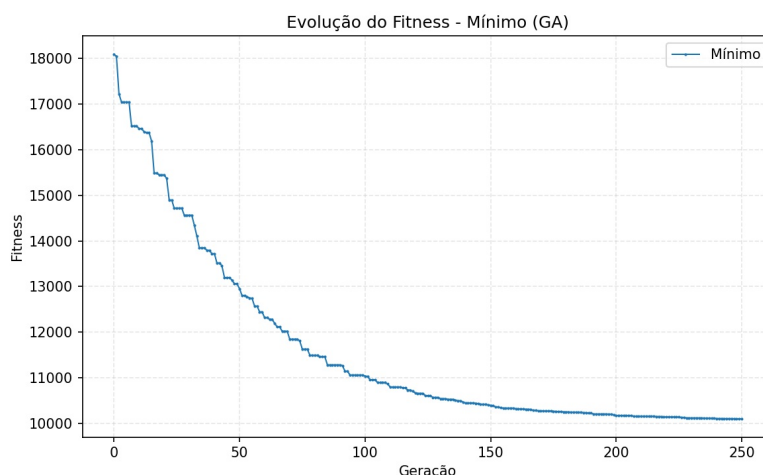


Figura 6. PosA – evolução do *fitness* ao longo das gerações.

Análise de Significância da MLU. Para validar a robustez estatística do componente de otimização, realizou-se uma análise com uma amostra de $n = 30$ rodadas independentes no cenário PosA. Os dados, apresentados na Figura 7, foram submetidos ao teste

de Shapiro–Wilk, que apontou desvio da normalidade ($p = 0,011$). Diante da impossibilidade de assumir uma distribuição gaussiana, aplicou-se o teste não paramétrico de Wilcoxon para comparar a mediana das execuções do GA ao limite teórico ótimo. O teste confirmou a convergência do algoritmo para o valor-alvo de 3,00, apresentando média de 3,0081 e intervalo de confiança estreito [3,0065; 3,0096]. A estabilidade operacional é reforçada pelo baixo nível de dispersão estatística validado pelo teste qui-quadrado ($p = 0,930$), indicando que o motor evolutivo mantém variância controlada e resultados previsíveis, com diferença relativa de apenas 0,26% em relação ao ótimo teórico.

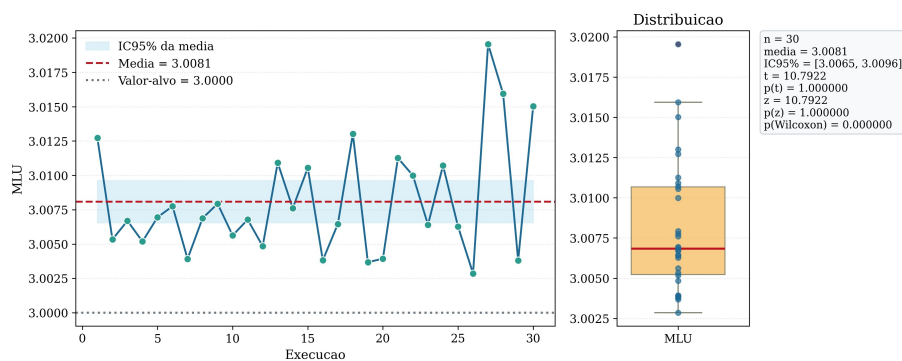


Figura 7. Significância estatística da MLU para a topologia PosA ($n = 30$).

Análise Estatística de Convergência (*Best Fitness*). A estabilidade do componente de otimização na topologia PosA foi corroborada pela análise do *best fitness*, detalhada na Figura 8. A partir de $n = 30$ execuções independentes, observou-se convergência para a média de 55.040,96. Devido ao desvio da normalidade indicado pelo teste de Shapiro–Wilk ($p = 0,010$), utilizou-se o teste de Wilcoxon para validar a repetibilidade dos resultados. O nível de significância ($p = 2,56 \times 10^{-6}$) e o estreito intervalo de confiança [55.033,29; 55.048,63] comprovam que o motor evolutivo mantém a integridade da busca e a qualidade das soluções de forma consistente. O teste qui-quadrado ($p = 0,930$) reforça o controle sobre a dispersão, validando a precisão do algoritmo em convergir para soluções de alta fidelidade no cenário avaliado.

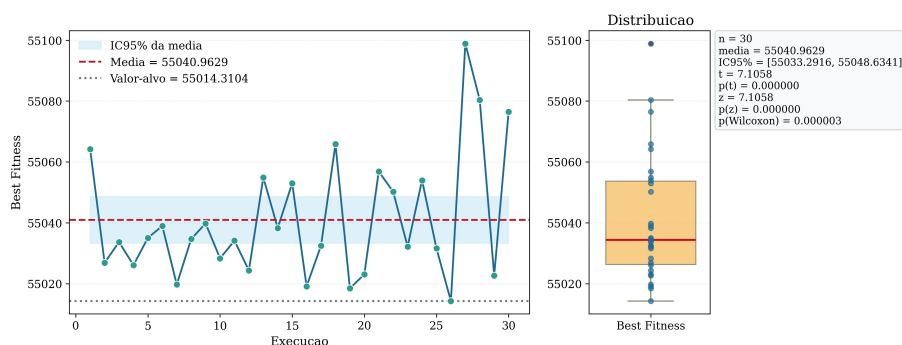


Figura 8. Significância estatística do *best fitness* para a topologia PosA ($n = 30$).

Análise do Custo Computacional (*Runtime*). Por fim, avaliou-se o custo computacional do Módulo de Otimização (GA) por meio do tempo de execução (*runtime*), cujos resultados estão consolidados na Figura 9. Os dados indicam um tempo médio de execução

de aproximadamente 102,6 segundos, com intervalo de confiança para a média situando-se entre 95,77 s e 109,43 s. Conforme ilustrado pela distribuição no gráfico, o tempo de execução apresentou um comportamento estatístico compatível com a normalidade ($p = 0,583$ no teste de Shapiro–Wilk). O teste qui-quadrado para a variância ($p = 0,930$) reforça a baixa dispersão observada. Essa estabilidade sugere que o Algoritmo Genético mantém um desempenho temporal previsível, fator crítico para garantir que a latência de decisão do plano de controle SDN permaneça dentro de limites operacionais aceitáveis.

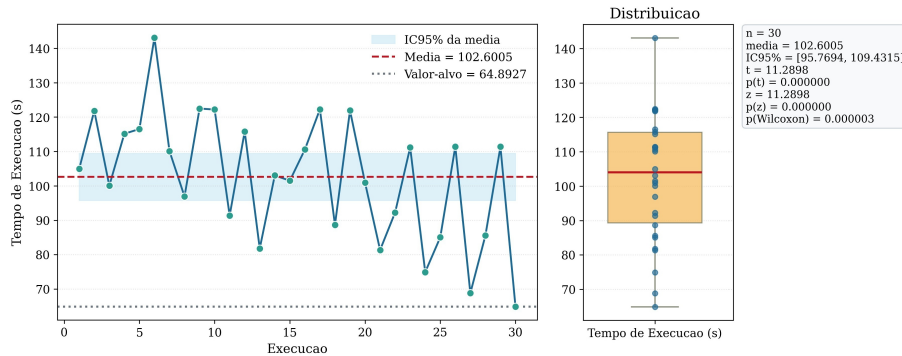


Figura 9. Significância estatística do *runtime* da topologia PosA ($n = 30$).

6.2. Testes de Escalabilidade: Topologia Abilene

O segundo conjunto de experimentos avaliou a rede *Abilene* sob condições de tráfego heterogêneo. A dinâmica de convergência e a comparação de desempenho são apresentadas nas Figuras 10 e 11.

A rede *Abilene* (Figura 10), exige um esforço maior de exploração devido ao maior número de enlaces, mas o GA mantém a curva de aprendizado estável até a convergência. O gráfico comparativo na Figura 11 demonstra que o GA consegue achatar os picos de utilização presentes no ECMP, aproximando-se da curva ótima.

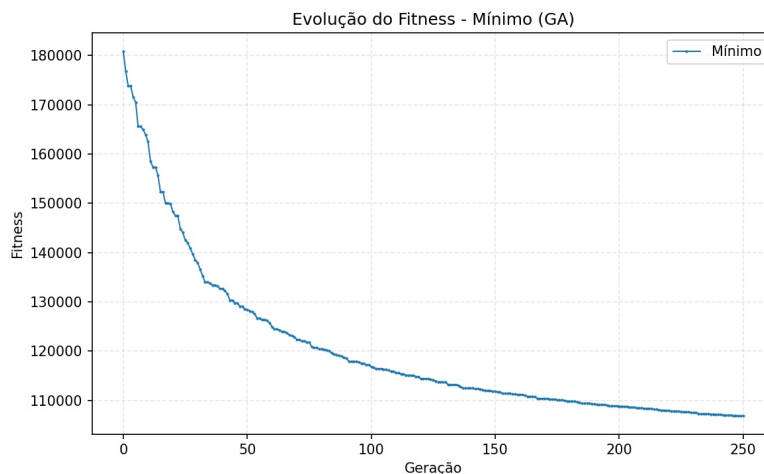


Figura 10. ABILENE – evolução do *fitness* ao longo das gerações.

Análise de Significância da MLU. Para validar a eficácia do módulo de otimização proposto para o *GARL-TE* em maior escala, realizou-se uma análise de significância da MLU

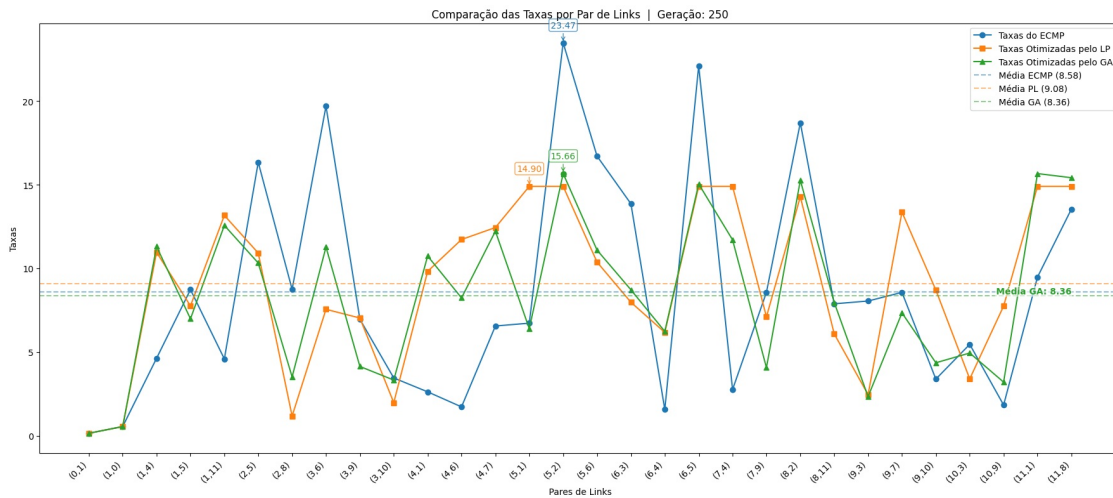


Figura 11. ABILENE, comparando MLU do ECMP, LP e o método proposto (GA).

em 30 rodadas independentes na topologia *Abilene*. A Figura 12 apresenta a distribuição dos resultados e o comportamento estatístico em relação ao alvo de otimização. O teste de Shapiro–Wilk apontou desvio da normalidade ($p = 2,65 \times 10^{-5}$), impedindo o uso de testes paramétricos. Assim, aplicou-se o teste não paramétrico de Wilcoxon para avaliar a convergência da mediana do GA ao alvo de otimização (16,10). O teste confirmou a estabilidade estatística ($p = 2,56 \times 10^{-6}$), com média de 16,49 e intervalo de confiança [16,35; 16,63]. Esses dados mostram que, mesmo diante da complexidade e da densidade de tráfego da rede *Abilene*, o Algoritmo Genético converge de forma robusta e mantém variância controlada ($p = 0,930$ no teste qui-quadrado), validando a precisão do motor de busca em maior escala.

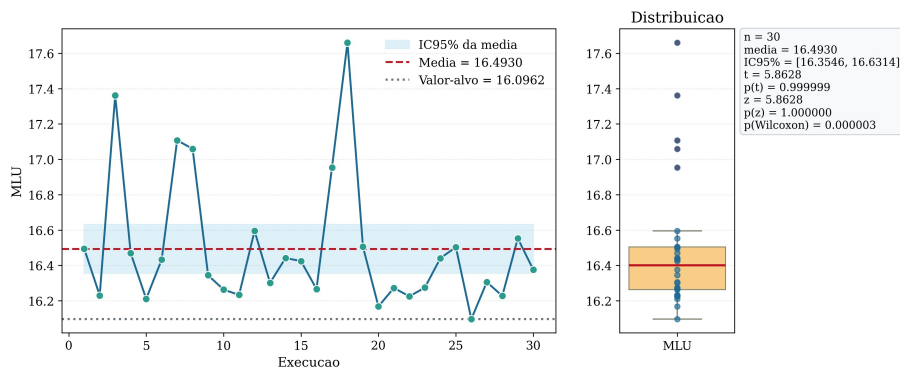


Figura 12. Significância estatística da MLU para a topologia *Abilene* ($n = 30$).

Análise Estatística de Convergência (*Best Fitness*). A estabilidade do componente de otimização na topologia *Abilene* foi avaliada por meio da análise de significância do *best fitness*. A precisão do algoritmo é corroborada pela análise de significância do *best fitness*, detalhada na Figura 13. A partir de $n = 30$ execuções, observou-se convergência para o valor médio de 777.889. Devido à ausência de normalidade confirmada pelo teste de Shapiro–Wilk ($p = 2,00 \times 10^{-6}$), utilizou-se o teste de Wilcoxon para validar a reprodutibilidade dos resultados. O elevado nível de significância ($p = 2,56 \times 10^{-6}$) e o estreito intervalo de confiança [773.434; 782.343] comprovam que o motor evolutivo mantém

a integridade da busca e a qualidade das soluções de forma consistente, mesmo sob as restrições mais severas da topologia Abilene.

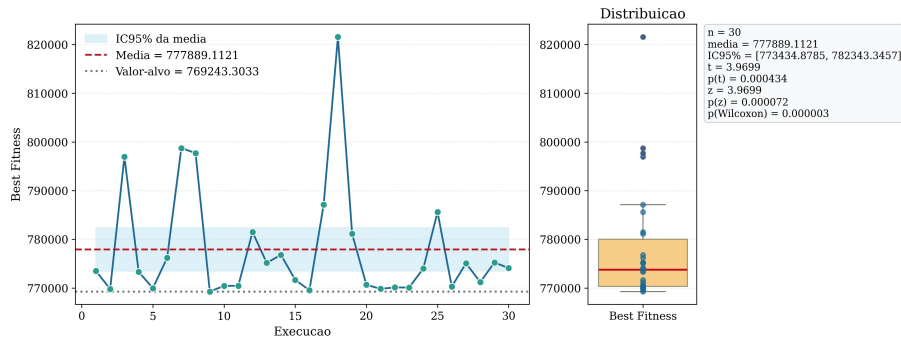


Figura 13. Significância estatística do *best fitness* da topologia Abilene ($n = 30$).

Análise do Custo Computacional (*Runtime*). Por fim, avaliou-se o custo computacional do componente de otimização na topologia de maior porte. A Figura 14 apresenta a distribuição do tempo de execução para as 30 rodadas experimentais conduzidas pelo programa de comparação.

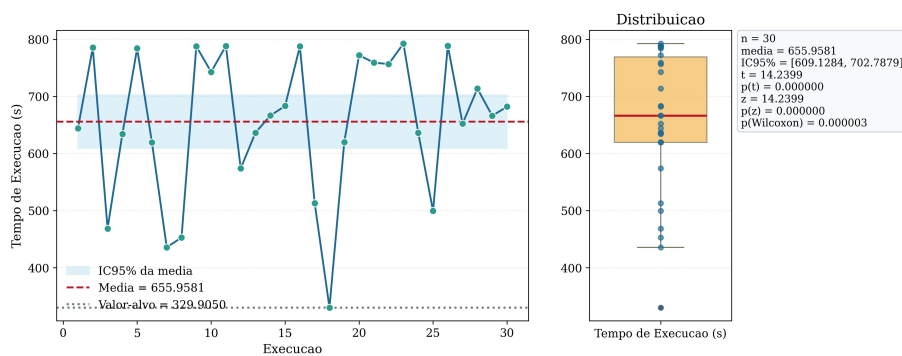


Figura 14. Significância estatística do *runtime* da topologia Abilene ($n = 30$).

7. Conclusões e Trabalhos Futuros

Este trabalho propôs o *GARL-TE*, arquitetura híbrida que integra Algoritmo Genético e DRL para superar a rigidez do ECMP e a baixa escalabilidade da Programação Linear. Integrado ao ciclo *MAPE-K*, o *GARL-TE* demonstra potencial para gestão autônoma e otimização do plano de controle de redes SDN.

Os resultados foram obtidos no ambiente de testes, *TesteGA*, desenvolvido para avaliar o módulo de otimização baseado em Algoritmo Genético. O motor evolutivo do GA, fundamentado na mutação de Dirichlet adaptativa, mostrou-se robusto no planejamento das frações de encaminhamento, alcançando resultados próximos ao limite ótimo com baixa latência. Os experimentos conduzidos nas topologias *PosA* e *Abilene*, no *TesteGA*, validaram a hipótese de que a intervenção seletiva em fluxos críticos basta para aliviar enlaces congestionados, preservando a estabilidade da rede.

No futuro, prevê-se incorporar otimizador ao *GARL-TE*, integrando-o à decisão com DRL. Planeja-se também evoluir o *GARL-TE* para Engenharia de Tráfego baseada em intenções (*Intent-Based Traffic Engineering*), permitindo que operadores definam

metas de redução de carga nos enlaces por interface de alto nível. Pretende-se investigar otimização multiobjetivo para equilibrar a minimização da MLU com eficiência energética, validar o modelo em *Mininet* com telemetria em tempo real e expandir a avaliação para topologias como a rede *RNP* brasileira.

Referências

- Fortz, B. and Thorup, M. (2002). Optimizing OSPF weights with arbitrary traffic matrices. *IEEE Journal on Selected Areas in Communications*, 20(4):150–162.
- Granville, L. Z. et al. (2015). Autonomic management of SDN networks. In *IFIP/IEEE International Symposium on Integrated Network Management*.
- Gui, F., Wang, S., Li, D., Chen, L., Gao, K., Min, C., and Wang, Y. (2024). Redte: Mitigating subsecond traffic bursts with real-time and distributed traffic engineering. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 71–85.
- Hamdan, M. et al. (2021). DPLBAnt: Improved load balancing technique based on detection and rerouting of elephant flows in software-defined networks. *Computer Communications*, 180:315–327.
- Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50.
- Kreutz, D., Ramos, M., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.
- Mendiola, A. et al. (2017). A survey on the contributions of software-defined networking to traffic engineering. *IEEE Communications Surveys & Tutorials*, 19(2):918–953.
- Pei, X. et al. (2024). Enabling efficient routing for traffic engineering in SDN with deep reinforcement learning. *Computer Networks*, 241:110220.
- Roughan, M. (2004). A practical guide to measured of traffic matrices. In *Proc. of ACM SIGCOMM on Network Troubleshooting*, pages 261–266.
- Tan, K. Y., Tan, S. C., and Chuah, T. C. (2025). A multi-phase drl-driven sdn migration framework addressing budget, legacy service compatibility, and dynamic traffic. *IEEE Access*, 13:33202–33219.
- Ye, M. et al. (2022). Flexdate: Flexible and disturbance-aware traffic engineering with reinforcement learning in software-defined networks. *IEEE/ACM Transactions on Networking*, 31(4):1433–1448.
- Zhang, J., Ye, M., Guo, Z., Hong, C. Y., Hui, J. W., Guan, H., and Wang, X. (2020). CFR-RL: Traffic engineering with reinforcement learning in SDN. *IEEE Journal on Selected Areas in Communications*, 38(10):2249–2259.
- Zhang, Y. (2004). Abilene traffic matrix dataset. Disponível em: <http://www.cs.utexas.edu/~yzhan/research/AbileneTM/>.