

Uma Estratégia Bioinspirada para Alocação Dinâmica de SFCs em Múltiplos Domínios, Nuvens e Orquestradores NFV

Vinicius Fulber-Garcia¹, José Flauzino¹, Alexandre Huff²,
Giovanni Venâncio¹, Elias P. Duarte Jr.¹

¹ Universidade Federal do Paraná (UFPR), Curitiba - PR

² Universidade Tecnológica Federal do Paraná (UTFPR), Toledo - PR

{vinicius, jwvflauzino, giovanni, elias}@inf.ufpr.br

alexandrehuff@utfpr.edu.br

Abstract. *A Service Function Chain (SFC) defines a virtualized network service through the composition of multiple virtualized functions. The Multi-SFC, in turn, enables the construction of network services which are distributed across multiple clouds, domains, and NFV (Network Function Virtualization) orchestrators. The Multi-SFC overcomes limitations of traditional deployments, in particular the restriction of having all Virtual Network Functions (VNFs) of a given SFC allocated to a single domain or point of presence (PoP). However, allocating resources in a Multi-SFC scenario for optimized SFC deployment is a challenging task. In this work, we propose a bioinspired strategy for mapping SFCs in the Multi-SFC context. The results of convergence and execution time tests demonstrate the feasibility of the proposal, while the execution of different types of mapping requests show the solution's versatility.*

Resumo. *Uma Service Function Chain (SFC) define um serviço virtualizado de rede por meio do encadeamento de múltiplas funções virtualizadas. Nesse contexto, a solução Multi-SFC possibilita a composição de serviços de rede distribuídos em várias nuvens, domínios e orquestradores NFV (Network Function Virtualization). A Multi-SFC elimina limitações das implantações tradicionais, em particular a restrição de que todas as funções de uma SFC sejam alocadas em um único domínio ou ponto de presença (PoP). No entanto, alocar recursos no cenário da Multi-SFC para implantar SFCs de maneira otimizada é uma tarefa desafiadora. Este trabalho propõe uma estratégia bioinspirada para o mapeamento de SFCs no contexto da Multi-SFC. Os resultados dos testes de convergência e tempo de execução demonstram a viabilidade da proposta, enquanto os testes de variação de requisições de mapeamento evidenciam a versatilidade de atuação da solução.*

1. Introdução

A Virtualização de Funções de Rede (NFV - *Network Function Virtualization*) possibilita a implementação em software de funções de rede que, tradicionalmente, são implementadas em hardware, geralmente proprietário [Chiosi et al. 2012]. Cada instância de uma função virtualizada de rede é denominada de Função Virtualizada de Rede (VNF - *Virtualized Network Function*), e é executada através de tecnologias de virtualização, tais como virtualização completa, para-virtualização e containerização [Herrera and Botero 2016].

Nesse contexto, a arquitetura de referência NFV, proposta pelo *European Telecommunications Standards Institute* (ETSI), surge com o objetivo de padronizar a execução de funções e serviços NFV, além de permitir o gerenciamento e a orquestração de funções virtualizadas de rede [Fulber-Garcia et al. 2020b]. Essa arquitetura também viabiliza a composição de *Service Function Chains* (SFC) para a construção de serviços de rede complexos. Em resumo, uma SFC consiste no encadeamento de múltiplas funções de rede, possibilitando que o tráfego de rede seja processado de acordo com uma ordem pré-definida [Halpern and Pignataro 2015].

Geralmente, funções de rede de uma SFC são instanciadas em um mesmo ponto de presença (*e.g.*, nuvem, domínio). No entanto, essa abordagem é limitada por diversos fatores, como os recursos computacionais dos servidores disponíveis [Ghaznavi et al. 2017] ou funções que operam nativamente em domínios específicos.

Além disso, as plataformas existentes de gerenciamento e orquestração NFV (*e.g.*, OpenStack Tacker [OpenStack Foundation 2024], CloudStack Vines [Flauzino et al. 2021], Open Source MANO [Institute 2024], etc.) operam nativamente com módulos homogêneos de orquestração. Em outras palavras, mesmo quando essas plataformas suportam a coordenação de múltiplos orquestradores distribuídos em diferentes pontos de presença, elas não permitem a cooperação entre módulos distintos de orquestração. No entanto, com o aumento da complexidade dos serviços virtualizados e a variedade de soluções de execução, gerenciamento e orquestração NFV [Venâncio et al. 2021, Huff et al. 2018, Tavares et al. 2018], a distribuição de SFCs em múltiplos domínios e a execução das mesmas sob diferentes orquestradores em plataformas NFV heterogêneas tornam-se requisitos naturais.

Dessa forma, a Multi-SFC [Huff et al. 2020] surge como uma solução que possibilita a distribuição, execução e gerenciamento de SFCs em um contexto que envolve múltiplas nuvens, domínios e orquestradores NFV. Para atingir esse objetivo, uma SFC passa a ser composta por múltiplos segmentos, instanciados em pontos de presença diversos. Cada segmento, por sua vez, representa o bloco de construção fundamental da Multi-SFC, sendo constituído por uma subsequência de VNFs. A alocação de cada segmento de uma SFC ocorre em um domínio específico, e conexões são estabelecidas entre segmentos de diferentes domínios para compor o serviço fim-a-fim.

Um dos principais desafios apresentados pelo cenário que a Multi-SFC propõe é a implantação de SFCs em uma infraestrutura heterogênea de recursos. Especificamente, há uma demanda de um processo de otimização que leva em consideração múltiplos objetivos e restrições associadas à alocação de serviços, segmentos e funções. Esse processo faz parte da etapa conhecida como VNF-FGE (*VNF - Forwarding Graph Embedding*), a qual integra o problema mais amplo de NFV-RA (*NFV - Resource Allocation*) [Herrera and Botero 2016].

Considerando o contexto apresentado, este trabalho propõe o *Service Mapping Expedient for Networked Traffic and Environments* (SeMENTE), uma estratégia bioinspirada para a alocação dinâmica de serviços virtualizados de rede no âmbito da Multi-SFC. A solução consiste na definição de um algoritmo genético multi-objetivo que permite especificar e mapear segmentos de uma SFC, visando otimizar a alocação de recursos em diversas nuvens, domínios e orquestradores NFV. O principal objetivo da SeMENTE é

minimizar a latência inter-domínio, além dos custos financeiros associados ao uso de recursos computacionais. Ao mesmo tempo, a solução busca maximizar a disponibilidade de banda inter-domínio. Além disso, a SeMENTE considera um conjunto de restrições impostas tanto pelo serviço em processo de mapeamento quanto pelo ambiente no qual está sendo mapeado.

Em ambientes complexos de produção, como os da Multi-SFC, é natural que o estado da rede, a disponibilidade de recursos e os custos associados a esses recursos apresentem variações graduais ao longo do tempo [Luizelli et al. 2017]. Diante dessas variações, mesmo que um serviço seja implantado de maneira otimizada, seu desempenho pode ser degradado, gerando possivelmente custos superiores aos previstos em um processo de mapeamento prévio. Nesse contexto, a SeMENTE incorpora a funcionalidade de *re-mapeamento* de SFCs sob demanda, considerando um cenário envolvendo múltiplas nuvens, domínios e orquestradores NFV. Para tanto, a solução utiliza dados históricos do último mapeamento do serviço, introduzindo os melhores indivíduos encontrados em uma execução anterior como a população inicial de uma nova execução. Essa técnica visa acelerar a convergência do algoritmo e obter resultados de alta qualidade, uma vez que se espera, dada a natureza gradual das mudanças na rede, que ainda existam segmentos altamente adequados nas soluções antigas de mapeamento para um determinado serviço.

A solução proposta foi submetida a uma avaliação abrangente por meio de um conjunto de testes de mapeamento para múltiplos serviços. Inicialmente, um experimento foi conduzido para avaliar a capacidade da SeMENTE de encontrar soluções otimizadas para o problema de mapeamento, utilizando um teste de convergência. Também, foi analisado o tempo médio de execução da solução sob diferentes configurações. Por fim, foram avaliadas a capacidade do algoritmo em lidar com diversas restrições e dependências de mapeamento, assim como as vantagens potenciais da estratégia de *re-mapeamento* diante de mudanças significativas na rede ou na qualidade do serviço. Os resultados obtidos demonstram a eficiência e eficácia da SeMENTE em operar com sucesso em cenários complexos de mapeamento, produzindo soluções otimizadas para os problemas apresentados com uma progressão linear no tempo de execução.

O restante do artigo segue a seguinte organização. A Seção 2 exibe uma visão geral da solução Multi-SFC. A proposta, projeto e implementação são detalhados na Seção 3. A Seção 4 apresenta e discute os cenários de teste, bem como os resultados obtidos. Na Seção 5, são analisados os principais trabalhos relacionados. Por fim, a Seção 6 conclui o trabalho e indica direções para trabalhos futuros.

2. Multi-SFC: SFCs sobre Múltiplos Domínios, Nuvens e Orquestradores

A Multi-SFC é uma estratégia que tem por objetivo permitir a instanciação e gerência de SFCs sobre múltiplos domínios, nuvens e orquestradores. No contexto da Multi-SFC, uma SFC é constituída por múltiplos segmentos. Cada segmento é alocado em um domínio e conectado a segmentos de outros domínios. A Figura 1 ilustra diversos segmentos de um único serviço implementado como uma Multi-SFC. Assim, a Multi-SFC assume que uma SFC consiste de uma sequência pré-determinada de VNFs. Cada segmento é uma sub-sequência daquelas VNFs. Um segmento é alocado em um domínio e a comunicação entre segmentos é realizada através de tunelamento, garantindo a segurança, mesmo com a comunicação entre segmentos ocorrendo na Internet.

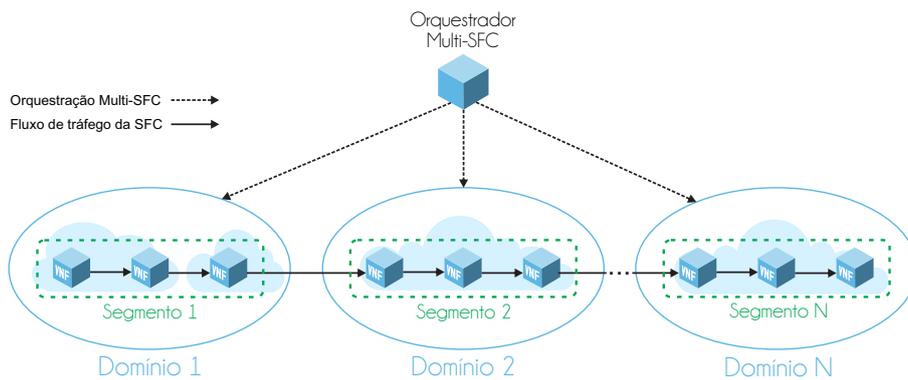


Figura 1. Multi-SFC: múltiplos segmentos de um serviço.

Os túneis fornecem uma abstração para a comunicação entre pares de segmentos. Um túnel é sempre executado através de uma VNF, por exemplo implementando VPN (*Virtual Private Network*). Na nomenclatura da IETF para SFCs, uma VNF que constitui um túnel corresponde ao SFF (*Service Function Forwarder*) inter-domínios. Qualquer SFC pode ser transformada em uma Multi-SFC definindo os segmentos que ficam em sistemas autônomos distintos e criando um túnel entre dois segmentos, que tem em cada ponta uma VNF implementando a funcionalidade do túnel. Assim há uma VNF de túnel na saída do primeiro segmento e outra VNF na entrada do segundo segmento. A Figura 2 ilustra um túnel entre dois domínios.

Essencialmente, a abordagem de tunelamento usando VNFs proposta na arquitetura Multi-SFC libera os operadores de rede de diversas configurações manuais necessárias para interconectar os diferentes domínios das Multi-SFCs. Esta abordagem também permite, por exemplo, que operadores de rede de dois domínios distintos entrem previamente em acordo sobre a tecnologia de tunelamento mais adequada a ser utilizada.

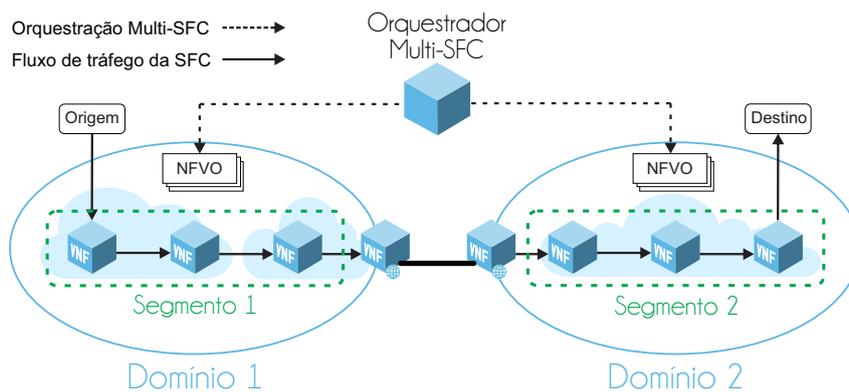


Figura 2. Multi-SFC sobre um par de domínios.

O principal módulo da arquitetura Multi-SFC é o Orquestrador Multi-SFC, também ilustrado na Figura 2. O Orquestrador Multi-SFC foi proposto alinhado com as definições do *framework* NFV-MANO da ETSI, garantindo interoperabilidade entre diferentes soluções NFV. O Orquestrador Multi-SFC implementa um conjunto de funcionalidades que permitem realizar a composição e orquestração de SFCs em cada domínio. As funcionalidades incluem desde o gerenciamento dos descritores de VNFs e SFCs, até a instanciação e destruição das mesmas. O módulo realiza ainda a tradução de operações

genéricas da Multi-SFC para as respectivas operações de cada orquestrador NFV, bem como configura as SFCs entre diferentes plataformas NFV, incluindo o encaminhamento automático de tráfego inter-domínios.

A Multi-SFC também fornece uma API (*Application Programming Interface*) para facilitar a integração de diversas aplicações cliente. As principais operações disponibilizadas pela API incluem as seguintes. A operação *compose_SFP* permite o encadeamento das VNFs (*i.e.*, *Service Function Path* - SFP) em uma SFC a partir de informações dos identificadores dos *VNF Packages* armazenados no repositório da Multi-SFC. A operação *get_policies* retorna a lista de políticas e restrições que podem ser utilizadas pelo classificador da SFC. A operação *create_SFC* recebe o descritor da SFC construído a partir das operações anteriores e tem a responsabilidade de instanciar as VNFs e os segmentos da SFC nos diferentes domínios e orquestradores NFV de maneira autônoma.

As funcionalidades abstraídas pela API da Multi-SFC referem-se basicamente às operações de instanciação, consulta e destruição de VNFs e SFCs nas diferentes plataformas NFV, bem como as configurações necessárias para realizar a interconexão dos segmentos da SFC distribuídos em diferentes domínios.

3. Mapeamento Multi-domínio Bioinspirado no Contexto da Multi-SFC

A solução proposta neste trabalho, denominada *Service Mapping Engine for Networked Traffic and Environments* (SeMENTE), é fundamentada nas características operacionais da Multi-SFC e objetiva realizar o mapeamento de uma SFC em múltiplos domínios de maneira otimizada. Além do mapeamento das VNFs que compõem a SFC originalmente, também considera-se a alocação e mapeamento das funções de rede para tunelamento inter-domínio, quando necessário. Além disso, a estratégia de mapeamento proposta contempla a possível existência das seguintes dependências, definidas para cada uma das funções que compõem um serviço: (i) domínio específico; (ii) de tipos de domínio (*e.g.*, nuvem, nevoeiro, borda); (iii) e de orquestrador (*e.g.*, OpenStack Tacker [OpenStack Foundation 2024], CloudStack Vines [Flauzino et al. 2021], Open Source MANO [Institute 2024]).

A solução SeMENTE adota um modelo de otimização baseado em três métricas principais: custo de implantação, latência inter-domínio e largura de banda inter-domínio. O custo de implantação refere-se aos encargos financeiros associados à alocação e manutenção de funções de rede virtualizadas em domínios específicos ou pontos de presença. A latência inter-domínio corresponde ao tempo de ida e volta (*i.e.*, *Round Trip Time*) observado em enlaces específicos de comunicação entre diferentes domínios ou pontos na rede. Por fim, a largura de banda inter-domínio refere-se à capacidade de transmissão de dados nos enlaces mencionados anteriormente. Por sua vez, o objetivo da otimização é minimizar a soma do custo e da latência inter-domínio, ao mesmo tempo em que se busca maximizar a soma da largura de banda entre os domínios. O modelo de análise adotado é o das fronteiras de Pareto [Ngatchou et al. 2005].

O modelo de documento de requisição de serviço adotado pela SeMENTE emprega uma estrutura YAML (*YAML Ain't Markup Language* [YAML Organization 2024]) dividida em três segmentos principais: `DOMAINS`, `REQUIREMENTS`, e `TOPOLOGY`. O segmento `DOMAINS` define cada ponto de presença disponível para uma função de rede, especificando características como custo, tipo, orquestradores disponíveis e transições

possíveis entre domínios, onde são estabelecidos valores de largura de banda disponíveis e latência. No segmento REQUIREMENTS, são indicadas expressões relacionais opcionais que determinam a aceitação ou rejeição de um mapeamento com base em limites definidos para as métricas de otimização. Por fim, o segmento TOPOLOGY utiliza uma modificação do modelo SCAG [Fulber-Garcia et al. 2020a] para definir cadeias de serviço, incorporando marcadores para indicar dependências de tipo de domínio e orquestradores.

A solução proposta foi implementada¹ utilizando a linguagem de programação Python 3 e uma versão adaptada da biblioteca de otimização multi-objetivo Platypus [Platypus Organization 2024]. A heurística genética escolhida foi a NSGAI [Seshadri 2006], devido às suas características de elitismo (*i.e.*, a persistência dos melhores indivíduos), a qual se mostra adequada para o problema de mapeamento abordado neste trabalho.

As principais características da solução, no contexto de algoritmos genéticos, são detalhadas a seguir:

- **Indivíduo:** um indivíduo define uma alternativa de mapeamento para um serviço de rede em um ambiente específico. Cada indivíduo possui um genótipo, representado por um vetor, refletindo a cadeia de serviço requisitada. Cada gene (posição no vetor) no genótipo representa uma função de rede específica do serviço, sendo que o alelo (valor em uma posição do vetor) de um gene indica o ponto de presença no qual a função correspondente deve ser mapeada;
- **População:** uma população consiste no conjunto de indivíduos em um determinado momento no tempo, denominado geração. Esses indivíduos passam por processos de reprodução para gerar a próxima geração. Vale destacar que, devido à utilização de um algoritmo elitista, os melhores indivíduos da população, determinados pela fronteira de Pareto relativa, são preservados na geração subsequente;
- **Inicialização:** a população inicial (primeira geração) é originada por meio de um operador de geração. No contexto da SeMENTE, o gerador selecionado emprega uma estratégia de geração aleatória, assegurando, contudo, todas as dependências de domínio e a presença de enlaces de comunicação válidos entre os domínios, caso ocorram transições. O gerador pode ser reaplicado durante a execução do algoritmo genético, contribuindo para aprimorar a variabilidade genética de uma geração e a capacidade de exploração do espaço de busca do algoritmo;
- **Seleção:** a etapa de seleção compreende a escolha dos indivíduos que serão submetidos ao processo de cruzamento. A solução proposta adota a seleção por torneio binário, em que dois indivíduos da população são selecionados aleatoriamente, e aquele que apresenta os melhores resultados para as métricas avaliadas é escolhido para o cruzamento;
- **Cruzamento:** o operador de cruzamento desempenha um papel central na geração de indivíduos da geração G a partir da geração $G - 1$ por meio da mesclagem de genes. Optando por um modo de operação que não gera efeitos colaterais na manutenção de potenciais dependências de domínio, o algoritmo de cruzamento SBX [Chacón and Segura 2018] foi empregado na SeMENTE;
- **Mutação:** o operador de mutação realiza alterações em alelos de genes específicos por meio da técnica de troca controlada de bits, na qual bits aleatórios do valor do

¹Disponível em <https://github.com/ViniGarcia/SeMENTE>

alelo (um índice que indica um ponto de presença ou domínio específico) são modificados para gerar um novo índice válido, indicando assim um novo ponto de presença. É importante notar que o operador de mutação não atua sobre genes com dependências de domínio.

A solução proposta permite a configuração de vários parâmetros por parte do usuário, influenciando em sua execução e, conseqüentemente, nos resultados gerados. Entre os principais parâmetros, destacam-se: critério de parada do algoritmo por tempo máximo ou por número máximo de gerações; tamanho da população e; taxa de cruzamento e taxa de mutação. O modelo de execução da SeMENTE é organizado em um ciclo de sete (7) etapas, conforme ilustrado na Figura 3.

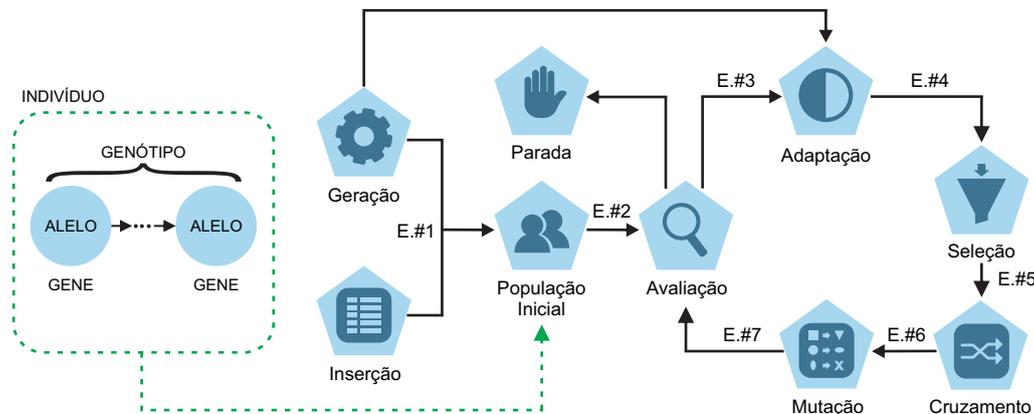


Figura 3. Modelo de execução da solução SeMENTE.

Na primeira etapa (E. #1), a população inicial é formada por meio de um mecanismo de geração, previamente apresentado, ou de inserção. Ao contrário do mecanismo de geração, o mecanismo de inserção não gera indivíduos aleatórios para compor a população inicial; em vez disso, ele introduz indivíduos predefinidos em uma lista. Esse método pode ser empregado para reutilizar indivíduos altamente adequados ao mapeamento requisitado em um momento X , embora, em um momento $X + Y$, tais indivíduos podem não representar completamente uma solução viável devido a mudanças ocorridas na rede. Contudo, é importante notar que as mudanças na rede geralmente são graduais e não catastróficas. Ou seja, certos genes dos melhores indivíduos do momento X ainda podem ser altamente adequados ao momento $X + Y$, e a utilização desses genes na população inicial tende a acelerar a convergência do algoritmo e melhorar a qualidade dos novos indivíduos gerados.

A segunda etapa (E. #2) corresponde à avaliação da população inicial com base na função objetivo, que visa minimizar o somatório dos custos financeiros e o total da latência inter-domínio, enquanto maximiza o somatório da largura de banda inter-domínio. Essa avaliação quantifica a adequação de um indivíduo para a solução do problema de mapeamento requisitado. A terceira etapa (E. #3) envolve a realização de ajustes eventuais na população visando aprimorar sua variabilidade, equilibrando os processos de exploração (*exploring*) e aprofundamento (*exploiting*) da solução. Nessa etapa, a cada décima geração, ocorre um processo de limpeza, no qual N indivíduos na fronteira de Pareto corrente com o mesmo valor de *fitness* (i.e., o mesmo resultado de avaliação para todas as métricas consideradas) são reduzidos a 1. O número total de indivíduos

eliminados durante o processo de limpeza é compensado na população pela geração de novos indivíduos aleatórios, que são imediatamente avaliados.

As etapas E. #4, E. #5 e E. #6 desempenham a função de formar a próxima geração de indivíduos, aplicando, sequencialmente, os operadores de seleção, cruzamento e mutação à população atual. Por fim, a sétima etapa (E. #7) consiste na avaliação da nova geração formada e marca o início de um novo ciclo, que percorre as etapas de três (3) a sete (7). O ciclo de execução se repete até que, após um processo de avaliação, o critério de parada seja alcançado. A solução exibe a fronteira de Pareto atual, correspondente ao momento em que o critério de parada foi alcançado, como resultado da execução.

4. Avaliação Experimental e Resultados

A solução SeMENTE foi validada por meio de um conjunto de experimentos visando determinar a eficiência e eficácia da estratégia de mapeamento de serviços em ambientes distribuídos e heterogêneos². O serviço de teste compreende uma SFC com topologia linear contendo entre 5 a 10 VNFs. O ambiente distribuído disponível para a instanciação das SFCs é representado por um grafo completo com 150 vértices, onde os vértices indicam pontos de presença e as arestas representam as conexões entre esses pontos. Um ponto de presença pode estar localizado na nuvem, no nevoeiro ou na borda, e pode suportar de 1 a 4 plataformas de gerenciamento e orquestração NFV, incluindo OpenStack Tacker, Open Source MANO, CloudStack Vines e Open Baton. O grafo e os valores para as métricas de otimização relacionadas aos vértices (custo financeiro) e arestas (latência e largura de banda) foram gerados aleatoriamente em uma faixa de valores pré-definida. Apesar de que redes reais geralmente não são organizadas como grafos completos, a escolha desse tipo de grafo visa aumentar o espaço de busca, permitindo avaliar a solução em um cenário desafiador de otimização.

Para a realização dos experimentos foram adotadas as seguintes configurações genéticas: tamanho da população igual a 100, taxa de cruzamento igual a 100%, e taxa de mutação igual a 30%. Essas configurações foram estabelecidas com base em testes empíricos prévios, os quais indicaram que o problema de mapeamento considerado se beneficia de populações com alta variabilidade, evitando que o algoritmo estagne em ótimos locais durante a execução. O critério de parada da solução, no entanto, é uma configuração determinada com base nas características específicas de cada experimento. A máquina utilizada para a execução dos experimentos possui um processador Intel Core i5-3330 (3.0 GHz), 8 GB de RAM (DDR3, 1333 MHz) e sistema operacional Ubuntu 20.04. Cada experimento foi repetido 30 vezes neste ambiente.

Os resultados obtidos são analisados considerando a técnica de avaliação para otimizações multi-objetivo de fronteira de Pareto [Lotov and Miettinen 2008], especialmente as fronteiras relativas de Pareto. Nessa técnica, os indivíduos presentes na fronteira mais externa (a fronteira 0 ou fronteira de Pareto) são intitulados indivíduos não dominados, pois sempre apresentam valores em pelo menos uma de suas métricas de otimização melhores do que aqueles que coexistem na mesma fronteira. Essa dinâmica também se aplica às fronteiras mais internas (1, 2, 3, ...), com a adição de que existe ao menos um indivíduo melhor em todos os aspectos avaliados (dominante), presente em todas as fronteiras mais externas. Por exemplo, para cada indivíduo na fronteira 1, há pelo menos um

²Disponíveis em <https://github.com/ViniGarcia/SeMENTE/tree/SBRC2024>

indivíduo na fronteira 0 que o domina. A fronteira absoluta de Pareto é determinada ao ter acesso a todas as possibilidades de solução de um dado problema. Por outro lado, a fronteira relativa de Pareto é calculada considerando todas as possibilidades de solução encontradas durante um processo ou passo de otimização, independentemente de representarem o conjunto completo ou não.

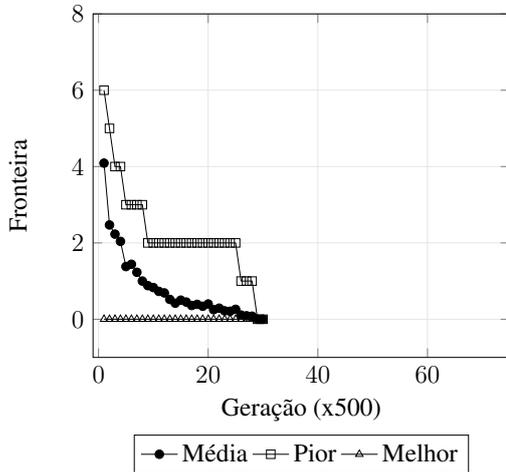


Figura 4. Convergência (T.5)

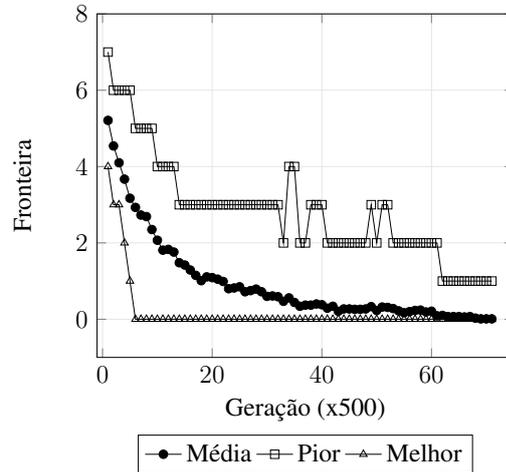


Figura 5. Convergência (T.10)

O primeiro experimento teve como objetivo verificar a capacidade de convergência do algoritmo em direção a uma fronteira de Pareto relativa. A verificação de convergência é realizada após a execução de 500 gerações (um passo de execução), e o critério de parada é definido pela ausência de modificações na fronteira de Pareto entre um passo de execução e outro. As Figuras 4 (5 funções) e 5 (10 funções) exibem os resultados obtidos passo a passo, indicando a melhor, a pior e a média das fronteiras onde estão presentes os indivíduos da fronteira de Pareto em um determinado momento da execução. Os resultados revelam a capacidade de convergência do algoritmo a uma fronteira de Pareto relativa ao longo do tempo. No entanto, o número de gerações necessário para atingir essa convergência é diretamente proporcional ao tamanho do espaço de busca. Mesmo que o ambiente distribuído seja o mesmo para ambos os casos de teste, o tamanho da cadeia influencia diretamente no número de combinações possíveis para a formação de resultados válidos de mapeamento. Portanto, para uma SFC com 5 funções, são necessários 30 passos de execução para alcançar a convergência. Por outro lado, para uma SFC com 10 funções, 71 passos foram executados até a convergência ser detectada.

O segundo experimento apresenta resultados relacionados ao tempo de execução da SeMENTE à medida que o número de gerações executadas varia. As Figuras 6 e 7 apresentam a média e o desvio padrão do tempo de execução para uma SFC contendo, respectivamente, 5 e 10 VNFs. É possível observar uma progressão linear do tempo de execução, indicando que ao dobrar o número de gerações a serem executadas, o tempo de execução também dobra, aproximadamente. Outro ponto relevante é que não há alterações significativas no tempo de execução para um mesmo número de gerações, independentemente do tamanho da cadeia de serviço. Esse fenômeno ocorre porque a avaliação da cadeia consiste em processos aritméticos com baixo custo computacional quando comparados às rotinas genéticas e à análise das fronteiras de Pareto. Consequentemente, o tamanho da cadeia não é determinante para o tempo de execução quando estabelecido um

número de gerações; no entanto, SFCs maiores requerem a execução de um maior número de gerações para convergência, resultando, tipicamente, em um maior tempo de execução para alcançar resultados globalmente ótimos.

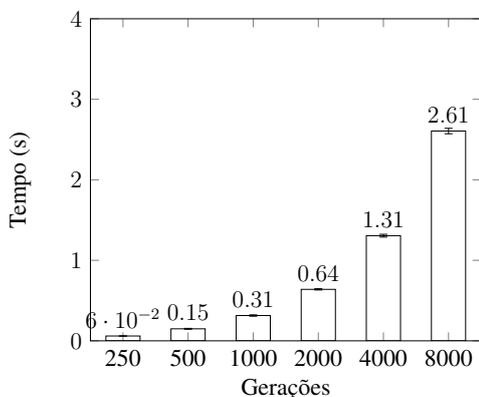


Figura 6. Tempo (T.5)

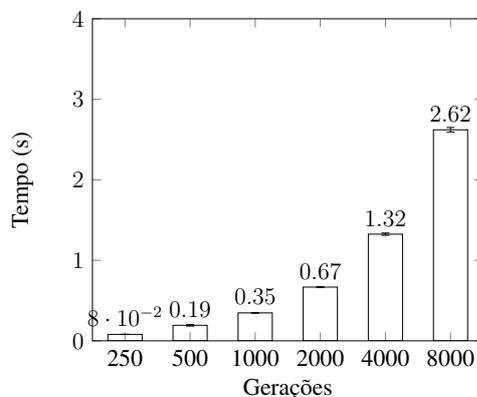


Figura 7. Tempo (T.10)

O terceiro experimento analisa diferentes casos de mapeamento das cadeias de serviço. Os resultados apresentados na Tabela 1 destacam os melhores candidatos para o mapeamento (indivíduos na fronteira de Pareto relativa), considerando cada métrica de otimização. Por exemplo, na linha relacionada ao custo, é apresentado o candidato com o menor custo encontrado na fronteira de Pareto relativa resultante da execução da solução SeMENTE, e entre parênteses, o valor do conjunto de métricas do mesmo candidato no formato (Custo, Latência, Largura de Banda). Neste teste, o critério de parada adotado foi um limite de tempo, permitindo a execução do algoritmo genético por 60 segundos, e três casos diferentes foram considerados: (i) *Livre*, onde nenhuma restrição ou dependência foi definida para o mapeamento das SFCs de 5 e 10 funções; (ii) *Orquestrador Definido*, onde todas as funções dependem da disponibilidade do orquestrador OpenStack Tacker nos pontos de presença selecionados; e (iii) *Latência Zero*, onde uma restrição de latência zero é aplicada ao mapeamento.

	T.5			T.10		
	<i>Livre</i>	<i>Orquestrador Definido</i>	<i>Latência Zero</i>	<i>Livre</i>	<i>Orquestrador Definido</i>	<i>Latência Zero</i>
Custo (\$)	569 (569, 237, 46524)	611 (611, 158, 9716)	-	1914 (1914, 883, 199424)	3607 (3607, 580, 224508)	-
Latência (ms)	30 (921, 30, 44601)	17 (1127, 17, 17151)	-	168 (14319, 168, 183098)	221 (7032, 221, 141849)	-
Largura de Banda (Mbps)	158094 (3714, 215, 158094)	152012 (4747, 330, 152012)	-	344252 (11820, 780, 344252)	340813 (8220, 962, 340813)	-

Tabela 1. Melhores resultados em diferentes cenários de mapeamento.

No experimento, o caso *Livre* obteve resultados otimizados para todas as métricas. Esse caso representa um *baseline*, pois trabalha com o maior espaço de busca, uma vez que todos os pontos de presença estão disponíveis para uso na procura por mapeamentos candidatos para as SFCs requisitadas. Já o caso *Orquestrador Definido* restringe o espaço de busca, uma vez que apenas pontos de presença com suporte ao OpenStack Tacker podem ser empregados no mapeamento. Essa dependência aplicada a todas as funções limita as opções de pontos de presença de 150 para 70. Atender a essa

dependência de domínio gera dois efeitos imediatos: (i) a eliminação de bons candidatos ao mapeamento, dado que estes utilizam domínios sem a presença do orquestrador requisitado e; (ii) facilitar que as melhores alternativas de mapeamento relacionadas aos domínios com o orquestrador requisitado sejam encontradas mais rapidamente, visto que apenas um subespaço de busca de indivíduos, quando comparado ao espaço de busca do caso *Livre*, é explorado e aprofundado ao longo da execução da solução.

O último caso do experimento, *Latência Zero*, representa um cenário no qual a *SeMENTE* não é capaz de executar com sucesso. Nesse caso, as funções do serviço de rede não apresentam nenhuma dependência, mas uma restrição é imposta ao processo de otimização: a **latência inter-domínio** deve ser igual a zero. Essa restrição implica que todas as funções de rede do serviço devem ser mapeadas em um mesmo ponto de presença (domínio), o que restringe o número de possibilidades de soluções válidas de 150^n (onde n é o número de funções de rede na SFC) para 150. Assim, o processo de geração de novos indivíduos, através de cruzamento e mutação, onde se aprimora a variabilidade da população pela mistura e alteração de suas características genéticas, em conjunto com um espaço de busca limitado ao extremo, não permite que alguma solução válida seja identificada para esse caso. Porém, cabe ressaltar que, para casos como o *Latência Zero*, processos de busca exaustiva podem ser aplicados com sucesso.

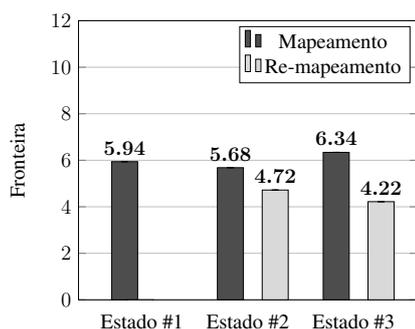


Figura 8. Variação (T.5)

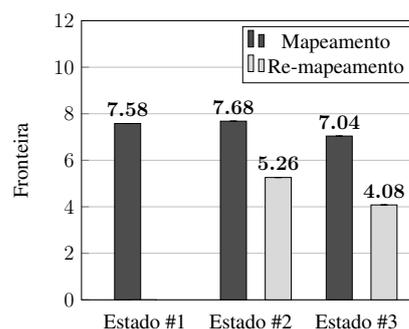


Figura 9. Variação (T.10)

Por fim, o quarto caso verifica os ganhos relacionados ao re-mapeamento de serviços quando mudanças graduais mais significativas ocorrem na rede. Nesse caso, o estado da rede foi modificado em dois momentos em uma linha temporal contínua, gerando assim os seguintes estados: Estado #1, estado inicial da rede onde um serviço é mapeado; Estado #2, estado onde 20% das conexões (por escolha aleatória) apresentam uma redução na latência inter-domínio, disparando um mapeamento/re-mapeamento do serviço ; e Estado #3, estado onde 20% das conexões (por escolha aleatória) apresentam um aumento na latência inter-domínio, disparando um novo mapeamento/re-mapeamento. O re-mapeamento utiliza o mecanismo de inserção para injetar a última fronteira de Pareto encontrada como população inicial, enquanto que o mapeamento simplesmente emprega o mecanismo de geração para criar novos indivíduos aleatórios.

As Figuras 8 (5 funções) e 9 (10 funções) exibem os resultados obtidos, sendo que cada barra indica a média da fronteira de Pareto relativa dos indivíduos retornados como resultado (mapeamentos candidatos) após 8000 gerações executadas. Como é possível observar, a reutilização de indivíduos previamente otimizados para um estado específico da rede como população inicial (re-mapeamento) apresenta um efeito positivo na quali-

dade dos resultados gerados pela solução. Ou seja, diante de mudanças graduais na rede e/ou no serviço, dados históricos podem servir como uma base sólida para definir pontos de partida na exploração e aprofundamento do espaço de busca, uma vez que uma solução otimizada para um estado anterior pode permanecer inteiramente ou parcialmente viável. No entanto, é importante ressaltar que tal comportamento não é esperado em casos de mudanças catastróficas na rede ou no serviço, sendo vantajoso descartar dados históricos e realizar um novo processo de mapeamento.

Os experimentos conduzidos e os resultados apresentados ao longo desta seção demonstram a viabilidade do uso da solução proposta para o mapeamento de serviços de rede em ambientes habilitados pela Multi-SFC. Essa conclusão fundamenta-se na capacidade do algoritmo de convergir e otimizar progressivamente os resultados, bem como na sua escalabilidade, evidenciada pela progressão linear do tempo de execução em relação às gerações. Por fim, a habilidade de lidar com dependências, particularmente aquelas relacionadas à Multi-SFC (como dependências de tipo de domínio e orquestrador), juntamente com a capacidade de realizar processos de re-mapeamento, define o escopo de atuação da solução SeMENTE e sugere oportunidades para sua oferta como um serviço.

5. Trabalhos Relacionados

Recentemente, várias soluções têm sido propostas para abordar o problema VNF-FGE. Em particular, trabalhos recentes têm destacado a aplicabilidade de algoritmos genéticos na resolução do problema de mapeamento de cadeias de serviço de rede em ambientes distribuídos [Fulber-Garcia et al. 2023], como aqueles habilitados pela Multi-SFC. Embora altamente eficientes e eficazes, poucas soluções baseadas em algoritmos genéticos foram propostas no contexto mencionado, sendo as duas mais proeminentes denominadas GA+LCB [Rodis and Papadimitriou 2021] e GeSeMa [Fulber-Garcia et al. 2023].

A solução GeSeMa utiliza uma estratégia genética com função objetivo personalizável baseada na heurística SPEA2 para o mapear serviços virtualizados em ambientes multi-datacenters. Por outro lado, a solução GA+LCB emprega uma estratégia genética mono-objetivo própria, visando otimizar um indexador criado pela maximização da disponibilidade de enlaces, largura de banda inter-domínio, fator de disponibilidade de domínio, além da minimização do atraso inter-domínio. Apesar das semelhanças operacionais com a solução SeMENTE, a aplicação do GeSeMa ou do GA+LCB no contexto da Multi-SFC é limitada por dois motivos principais: *i*) a ausência de suporte para a definição de dependências de tipo de domínio e de orquestrador e *ii*) a falta de alocação e cálculo de custos relacionados às funções de tunelamento durante transições entre domínios. Além disso, destaca-se que a função objetivo de otimização da solução GA+LCB não é personalizável, sendo incompatível com aquelas adotadas pela SeMENTE.

Além das soluções de mapeamento fundamentadas em estratégias genéticas, é importante destacar aquelas que se dedicam à re-implantação de serviços virtualizados, seja nas tarefas de incorporação por mapeamento ou não. Dentre essas abordagens, está o NFV-PEAR [Miotto et al. 2019], uma solução baseada em uma modelagem ILP do problema e voltada para a contínua re-composição e re-incorporação de SFCs. Essa estratégia visa minimizar a quantidade de recursos consumidos e as modificações nos mapeamentos dos serviços, preservando assim padrões de desempenho, dependências e restrições.

Já em [Chen et al. 2018] é apresentado um algoritmo genético dedicado à seleção

e compartilhamento de VNFs para re-incorporar cadeias de serviço em ambientes distribuídos de virtualização. Seu objetivo é otimizar a utilização de recursos computacionais e o atraso fim-a-fim do serviço. Contudo, é importante salientar que as mesmas limitações previamente discutidas para as soluções de mapeamento no contexto da Multi-SFC aplicam-se às soluções de re-composição e re-incorporação de serviços citadas. Também, evidencia-se que a heurística genética abordada em [Chen et al. 2018] não inclui nenhum mecanismo para aproveitar dados históricos durante a execução da re-incorporação de serviços, ao contrário do que é realizado pela solução SeMENTE.

6. Conclusão

O crescente aumento da popularidade e complexidade dos serviços virtualizados de rede, combinado com o contínuo avanço das soluções NFV, torna inevitável a viabilidade da distribuição de SFCs em múltiplos domínios. Nesse cenário, a abordagem Multi-SFC possibilita a distribuição, execução e gerenciamento de SFCs em diversas nuvens, domínios e orquestradores NFV. Contudo, para atender a determinadas restrições de custo e desempenho, é essencial contar com um processo otimizado de mapeamento de cadeias de serviços nos pontos de presença disponibilizados pela Multi-SFC.

Nesse contexto, o presente trabalho introduziu a solução SeMENTE, uma estratégia bioinspirada para o mapeamento dinâmico de SFCs em múltiplos domínios, nuvens e orquestradores. Utilizando um algoritmo genético multi-objetivo, a solução visa otimizar o mapeamento de recursos considerando diversas métricas de otimização, como o custo financeiro de implantação, a latência inter-domínio e a largura de banda inter-domínio. Adicionalmente, a solução oferece a capacidade de re-mapeamento sob demanda de SFCs em resposta a possíveis alterações no estado da rede. A avaliação experimental destacou a viabilidade, eficiência e eficácia da solução SeMENTE, evidenciando seu potencial para aprimorar o processo de implantação e gerenciamento de SFCs distribuídas em diversos cenários.

Trabalhos futuros englobam a expansão das capacidades da solução SeMENTE e sua integração total com a Multi-SFC por meio da configuração automática do serviço no ambiente de rede. Em outras palavras, além de indicar os pontos de presença nos quais cada VNF de uma SFC deve ser alocada, a SeMENTE será responsável por configurar a Multi-SFC para que esta implemente a alocação otimizada do serviço na rede. Além disso, objetivamos comparar a solução com outras que, mesmo com compatibilidade limitada à SeMENTE, apresentem um características semelhantes e comparáveis à ela.

Referências

- Chacón, J. and Segura, C. (2018). Analysis and enhancement of simulated binary crossover. In *Congress on Evolutionary Computation*.
- Chen, Z. et al. (2018). Delay optimization oriented service function chain migration and re-deployment in operator network. *ACTA ELECTONICA SINICA*, 46(9).
- Chiosi, M. et al. (2012). Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action. In *SDN and OpenFlow World Congress*.
- Flauzino, J. et al. (2021). Gerência e orquestração de funções e serviços de rede virtualizados em nuvem cloudstack. In *Workshop de Gerência e Operação de Redes e Serviços*.

- Fulber-Garcia, V. et al. (2020a). Cusco: a customizable solution for nfv composition. In *International Conference on Advanced Information Networking and Applications*.
- Fulber-Garcia, V. et al. (2020b). Network service topology: Formalization, taxonomy and the custom specification model. *Computer Networks*, 178.
- Fulber-Garcia, V. et al. (2023). Customizable mapping of virtualized network services in multi-datacenter environments based on genetic metaheuristics. *Journal of Network and Systems Management*, 31(4).
- Ghaznavi, M. et al. (2017). Distributed service function chaining. *Journal on Selected Areas in Communications*, 35(11).
- Halpern, J. and Pignataro, C. (2015). Service Function Chaining (SFC) Architecture. Technical Report 7665, IETF.
- Herrera, J. G. and Botero, J. F. (2016). Resource allocation in nfv: A comprehensive survey. *Transactions on Network and Service Management*, 13(3).
- Huff, A. et al. (2018). A holistic approach to define service chains using click-on-osv on different nfv platforms. In *Global Communications Conference*.
- Huff, A. et al. (2020). Building multi-domain service function chains based on multiple nfv orchestrators. In *Conference on Network Function Virtualization and Software Defined Networks*.
- Institute, E. T. S. (2024). Osm: Open source mano.
- Lotov, A. V. and Miettinen, K. (2008). Visualizing the pareto frontier. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*.
- Luizelli, M. C. et al. (2017). The actual cost of software switching for nfv chaining. In *Symposium on Integrated Network and Service Management*.
- Miotto, G. et al. (2019). Adaptive placement & chaining of virtual network functions with nfv-pear. *Journal of Internet Services and Applications*, 10.
- Ngatchou, P. et al. (2005). Pareto multi objective optimization. In *International Conference on Intelligent Systems Application to Power Systems*.
- OpenStack Foundation (2024). Tacker: Openstack nfv orchestration.
- Platypus Organization (2024). Platypus - multiobjective optimization in python.
- Rodis, P. and Papadimitriou, P. (2021). Intelligent network service embedding using genetic algorithms. In *Symposium on Computers and Communications*.
- Seshadri, A. (2006). A fast elitist multiobjective genetic algorithm: Nsga-ii. *MATLAB Central*, 182.
- Tavares, T. N. et al. (2018). Niep: Nfv infrastructure emulation platform. In *International Conference on Advanced Information Networking and Applications*.
- Venâncio, G. et al. (2021). Beyond vnf: Filling the gaps of the etsi vnf manager to fully support vnf life cycle operations. *International Journal of Network Management*, 31(5):e2068.
- YAML Organization (2024). Yaml: Yaml ain't markup language.