

Alocação de Recursos para Redes Virtuais com Seleção de Método de Resolução via Aprendizado de Máquina

Samuel M. A. Araújo¹, Fernanda S. H. de Souza², Geraldo R. Mateus¹

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) - Belo Horizonte - MG

²Departamento de Ciência da Computação
Universidade Federal de São João del-Rei (UFSJ) - São João del-Rei - MG

smaa@dcc.ufmg.br, fsumika@ufs.j.edu.br, mateus@dcc.ufmg.br

Resumo. *A alocação de recursos é um problema NP-difícil, presente no Mapeamento de Redes Virtuais (VNE) e na Virtualização de Funções de Rede (NFV). Nesse contexto, um ponto pouco explorado na literatura é relacionado a quando se deve aplicar um método de resolução heurístico ou exato. Tanto para o VNE, quanto para o NFV, normalmente a literatura indica um tratamento heurístico devido à complexidade do problema e à elevada dimensão dos dados. Entretanto, em experimentos preliminares, percebe-se que em alguns momentos um tratamento exato pode ser aplicado em um tempo praticável. Foi notado que esses eventos não se tratam de casualidades, mas decorrem de várias condições relacionadas ao Substrato de Rede (SN) residual e às demandas de rede requisitadas. A abordagem proposta neste artigo é baseada na aplicação de técnicas de Aprendizado de Máquina (AM) para alocação de recursos em rede, e tem como objetivo prever em quais situações se pode preterir uma abordagem heurística em prol de uma exata. Para tal, foi adotado o VNE e duas abordagens da literatura. As simulações realizadas mostram que o tratamento com AM melhora a taxa de aceitação e a receita de mapeamento em relação à heurística, e reduz o tempo de processamento em relação à abordagem exata.*

Abstract. *The network resource allocation is considered an NP-hard problem, present in Virtual Network Embedding (VNE) and the Virtual Network Function (NFV). In this context, a little-explored question in the literature is related to when to apply an exact or heuristic resolution method. For both VNE and NFV, the literature usually suggests a heuristic treatment, considered the complexity and the high dimension of the data. However, in preliminary experiments, it is possible to observe that occasionally, the exact treatment can be applied in a practicable time. It is inferred that these events are not casual; they come from several conditions related to the residual Network Substrate (SN) and the network demands requested. The approach proposed in this paper is based on the application of the Machine Learning (ML) technics in order to allocate the network resources and aims to predict the situations in which the exact approach could supersede the heuristic. For this purpose, the VNE and both approaches of the literature were used. The simulations performed showed that ML treatment increases the acceptance ratio and revenue in comparison with the heuristic, and reduces the processing time in comparison with the exact approach.*

1. Introdução

O surgimento de novas demandas *web* implica em mais automação e mobilidade à sociedade, requerendo que o ser humano esteja sempre conectado [Nokia 2016]. Essas demandas podem ser voltadas às mais diferentes áreas, *e.g.*, cidades inteligentes, indústrias e mídia. Para atender a essas novas demandas, as tecnologias de virtualização de redes têm sido alvo de diferentes pesquisas nos últimos anos, se consolidando não apenas como mitigadoras dos problemas de ossificação da internet, mas como fundamentais para o desenvolvimento da computação em nuvem e das redes de comunicação de quinta geração (5G) [Fischer et al. 2013, Laghrissi and Taleb 2018].

Um dos primeiros problemas relacionados a virtualização de redes é o Mapeamento de Redes Virtuais (VNE), que consiste em alocar de forma eficiente recursos do Substrato de Rede (SN) residual, para atender as requisições virtuais demandadas pelos clientes [Fischer et al. 2013]. Posteriormente ao surgimento do VNE, um novo modelo de arquitetura de redes começou a ganhar espaço: as Redes Definidas por *Software* (SDN). Dentre as principais características do modelo SDN está empregar um controlador logicamente centralizado com uma visão global do SN, e prover uma gerência facilitada e estratégica dos recursos físicos. Características que geram benefícios inerentes a programabilidade e flexibilidade da rede [Zhao et al. 2019].

As redes SDN e o VNE emergem contiguamente como tecnologias inovadoras, ajudando a trazer flexibilidade na implantação de novas funcionalidades de rede e facilitando o gerenciamento de recursos. Neste contexto, uma nova tecnologia emerge: a Virtualização de Funções de Rede (NFV). A adoção/transição das redes antigas e legadas para a arquitetura NFV começou em 2012, e desde sua concepção tem sido foco de pesquisas em todo o mundo [Cohen et al. 2015, Laghrissi and Taleb 2018].

Um problema comum tanto para o VNE quanto para o NFV, consiste na alocação de recursos sobre o SN para atender com confiabilidade e eficiência as requisições de redes virtuais demandadas por diferentes clientes. Dada a quantidade e diversidade de requisições existentes, e a necessidade de mapeamentos bem planejados, se torna cada vez mais custosa a resolução dos problemas de alocação de recursos. Neste contexto, tanto o VNE, quanto o NFV são problemas NP-difíceis [Cohen et al. 2015, Amaldi et al. 2016].

Diversas variações dos problemas de alocação de recursos em redes podem ser encontrados na literatura. Por questões de simplicidade e clareza, neste trabalho foi adotado o VNE em uma versão básica, similar à abordada em [Chowdhury et al. 2009]. No entanto, os mesmos princípios empregados ao VNE neste artigo podem ser facilmente estendidos e aplicados a outras variações de problemas, como por exemplo, ao NFV.

Na versão do VNE adotada, o SN é composto por nós e enlaces, e são consideradas as capacidades de processamento e de largura de banda. De forma similar, as requisições de redes virtuais possuem demandas de processamento associadas a cada nó, e de largura de banda a cada enlace. Cada nó virtual de uma mesma requisição deve ser mapeado em um nó diferente do SN, e deve ser posicionado dentro de um raio de mapeamento. Cada enlace virtual de uma requisição pode ser mapeado em mais de um enlace físico e formar um caminho. Para se efetivar um mapeamento é necessário que todos os componentes alocados no SN possuam recursos suficientes para servir às demandas das requisições.

Seguindo a taxonomia apresentada por [Fischer et al. 2013], a versão do VNE

tratada nesse artigo é de natureza *online*, uma vez que as requisições não são conhecidas *a priori*, e chegam ao provedor à medida que os clientes decidam contratar tais serviços. Neste caso, a tomada de decisão sobre a aceitação ou rejeição da requisição deve ser realizada de maneira otimizada e rápida, de acordo com a disponibilidade de recursos residuais do SN no momento do mapeamento. Outra característica do problema é a não aplicação de reconfiguração, diante disso, cada requisição perdura com o mesmo mapeamento sobre o SN até ter seu tempo de vida expirado (mapeamento estático), aspectos aplicados em [Chowdhury et al. 2009, Beikta et al. 2017, Araújo et al. 2018].

Seja no problema VNE, ou no NFV, a alocação de recursos em nuvem requer uma resposta de processamento rápida para se manter níveis aceitáveis de Qualidade de Serviço (QoS) [Shi et al. 2015, Araújo et al. 2018]. Ressalta-se que o tempo de processamento máximo que uma requisição de rede pode aguardar é um parâmetro de QoS atribuído ao usuário final, *i.e.*, cada aplicação pode aguardar um tempo máximo diferente para ser processada.

1.1. Processamento Heurístico e Exato

Devido a natureza NP-difícil do VNE, a aplicação de um tratamento exato pode gerar um alto tempo de processamento e se tornar inviável. Neste caso, o uso de heurísticas é aconselhado, abrindo mão de uma solução ótima em prol de um baixo tempo de execução [Fischer et al. 2013]. No entanto, visto que cada usuário final pode aguardar um determinado período de tempo para ter seu serviço atendido, desde que não longo, fica a cargo do provedor realizar um mapeamento, dentro desse período de tempo, que seja o mais benéfico não só para o cliente, mas também para o próprio provedor. Nesse sentido, se potencialmente uma requisição for de fácil processamento, pode-se realizar um mapeamento que maximize com garantias de otimalidade os objetivos buscados, *e.g.*, lucro, desde que não extrapole o limite de tempo definido pelo usuário final. Por outro lado, essa afirmação gera um desafio, classificar em meio a um fluxo de requisições *online* quais podem ser tratadas de modo exato, ou não.

Contrapondo as afirmações anteriores, depreende-se que apesar do uso de heurísticas ser aconselhado pela literatura, em alguns cenários pode ser viável a aplicação de métodos exatos. A premissa é que a escolha do tratamento a ser aplicado está diretamente ligada às demandas e ofertas de recursos da requisição e do SN residual, aspectos que afetam diretamente o tempo de processamento de determinada requisição. Essa indagação induz à formulação da seguinte questão de pesquisa: Como identificar quando se deve aplicar um tratamento heurístico ou exato ao VNE? Nesse aspecto, juntamente aos preceitos de [Shi et al. 2015] e [Araújo et al. 2018], foi considerado que uma requisição sempre deve ser atendida com um baixo tempo de processamento.

Com a finalidade de fundamentar a questão de pesquisa deste artigo, foram realizados alguns experimentos preliminares para identificar o comportamento do modelo exato em um ambiente *online*. Para tal, um SN foi gerado pelo modelo de [Waxman 1988] com 100 nós distribuídos em um *grid* de 50×50 unidades de distância. O modelo de [Waxman 1988] foi configurado com os parâmetros $\alpha = \beta = 0.2$. Os parâmetros α e β definem a probabilidade de existir um enlace entre os pares de nós (u, v) de acordo com a equação $P(u, v) = \alpha e^{-\frac{d}{\beta L}}$, onde d é a distância euclidiana de u até v , e L é a distância máxima entre todos os nós do SN. Como resultado gerou-se um SN com

245 enlaces. Para as simulações, foram gerados dois conjuntos com 500 requisições cada: (i) com 1 requisição chegando a cada 50 unidades de tempo t (baixa densidade, cenário $S1$); (ii) com 1 requisição chegando a cada $25t$ (alta densidade, cenário $S2$). Cada roteador virtual de cada requisição possui um raio de mapeamento alto, distribuído uniformemente entre 14 e 16 unidades de distância. Tal escolha aumenta a região de mapeamento de um nó virtual, gera mais variáveis no modelo, e tende a deixá-lo mais difícil de ser resolvido.

Os demais parâmetros são os mesmos usados em [Chowdhury et al. 2009]. A abordagem exata foi desenvolvida em C++ através da API do solver *CPLEX*. A modelagem exata usada é a mesma proposta nos trabalhos de [Souza et al. 2018] e [Araújo et al. 2019], mas com o objetivo de maximizar o balanceamento de carga. Essa escolha é fundamentada em experimentos anteriores [Araújo et al. 2019], onde constatou-se que a abordagem de [Chowdhury et al. 2009] é mais lenta quando comparada ao modelo de [Souza et al. 2018]. Os testes foram realizados em um computador *Pentium G870* 3.30GHz, com 16GB de RAM DDR3 1333MHz, utilizando o sistema operacional Ubuntu 18.04.3. A Figura 1 mostra a distribuição de frequências dos tempos de processamento de cada requisição, nos cenários $S1$ e $S2$, para o SN proposto.

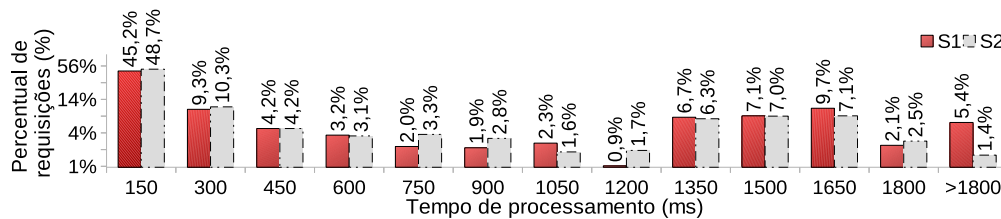


Figura 1. Histograma dos tempos de processamento de cada requisição.

Percebe-se na Figura 1, que muitas requisições, mesmo sendo mapeadas com uma abordagem exata, possuem um tempo de processamento baixo para os dois conjuntos de requisições. Adotando-se como parâmetro de QoS o tempo máximo de processamento de $300ms$, proposto em [Araújo et al. 2018], uma grande parcela de requisições poderia ser processada de modo exato, e não extrapolaria o tempo limite. Caso o tempo máximo de processamento seja maior que $300ms$, teremos uma parcela ainda maior de requisições que poderiam ser mapeadas com garantias de otimalidade.

As variações de tempo de processamento observadas na Figura 1 não são casuais, e se devem a diversos fatores implícitos aos cenários abordados, *e.g.* se uma inviabilidade de mapeamento for detectada, o modelo exato tende a encerrar rapidamente o processamento; se a cardinalidade de roteadores demandados pela requisição for baixa, o modelo exato gerará menos variáveis e restrições, o que melhora o tempo de processamento; e se o raio de mapeamento dos roteadores da requisição for alto, o número de variáveis aumenta e o modelo exato tende a ficar mais lento. Ao considerar um cenário *online*, existe uma variabilidade de recursos demandados e ofertados ainda maior, fator que aumenta a dificuldade de se explicitar todas as situações possíveis para se definir o uso de uma abordagem heurística ou exata.

1.2. Aplicação de Técnicas de Inteligencia Artificial

Devido à complexidade de alguns problemas associados à alocação de recursos em redes, uma alternativa de tratamento adotada é o uso de Inteligencia Artificial (IA)

[Xie et al. 2019]. A IA pode potencialmente reduzir o esforço computacional relacionado ao processamento das requisições e gerar uma tomada de decisão mais assertiva em relação aos objetivos buscados [Zhao et al. 2019]. Mais estritamente, o Aprendizado de Máquina (AM) é a subárea da IA que provê ao computador a capacidade de aprender e tomar decisões sem ser explicitamente programado.

Como contribuição, este artigo apresenta um modelo baseado em técnicas de AM, que auxilia a tomada de decisão sobre quando utilizar, ou preterir, o uso de uma heurística em relação a um tratamento exato na alocação de recursos em rede. A abordagem de mapeamento proposta, chamada de híbrida, é capaz de aprender implicitamente qual é o método de mapeamento mais adequado à situação corrente. O objetivo é tratar os cenários classificados como de rápido processamento através de um método exato (aproveitando os benefícios de otimalidade da solução); e os cenários classificados como de processamento demorado através de heurísticas (gerando benefícios relativos ao baixo tempo de execução).

Neste trabalho, foram adotados os princípios de [Araújo et al. 2018] e definiu-se um mapeamento como demorado se sua resolução exata levar mais que $300ms$. Como heurística, foi adotada a proposta em [Araújo et al. 2018], e como modelo exato, aplicou-se o mesmo utilizado nos experimentos preliminares (Subseção 1.1). Apesar de ser definido previamente as abordagens, bem como o limite de $300ms$ para classificar o mapeamento de uma requisição como rápido ou demorado, essa parametrização pode ser facilmente alterada de acordo com as necessidades do provedor ou do cliente.

Este trabalho está organizado como segue. Na Seção 2 é apresentada uma breve revisão de literatura. Na Seção 3 são apresentados os modelos de classificação utilizados e seus respectivos experimentos computacionais. Na Seção 4 é apresentada a integração entre o modelo de classificação selecionado e os tratamentos exato e heurístico, seguido de seus respectivos experimentos computacionais. Por fim, na Seção 5 conclui-se o trabalho.

2. Trabalhos Relacionados

Alguns artigos quantitativos (*surveys*), correlatos ao VNE, NFV, SDN, IA e AM, e fundamentais para esta pesquisa, são sumarizados pela sua relação com esse trabalho, e mostrados na Tabela 1. Ao decorrer desta Seção também são abordados os problemas tratados sob a ótica da otimização clássica, de heurísticas e do AM.

Tabela 1. Abrangência dos artigos quantitativos correlatos a pesquisa.

Área	Contribuições relacionadas ao nosso trabalho	Autor
VNE	<ul style="list-style-type: none"> - Definições clássicas de virtualização em redes e formulações; - Estratégias de otimização exata, heurística e metaheurística; - Conceitos de abordagens <i>online</i> e estáticas; - Definições de fragmentação do SN, decomposição do VNE e métricas clássicas; - 125 trabalhos analisados. 	[Fischer et al. 2013]
NFV/ SDN	<ul style="list-style-type: none"> - Definições clássicas de NFV; - Conceitos de ciclo de vida, custo e lucro; - Estratégias de otimização exata, heurística e metaheurística; - Diferentes tipos de restrições, objetivos, métricas e aspectos de QoS; - 176 trabalhos analisados. 	[Laghrissi and Taleb 2018]
SDN	<ul style="list-style-type: none"> - Relação entre AM e redes SDN; - Modelos e exemplos de aprendizado supervisionado, não supervisionado e por reforço; - Vantagens e desvantagens dos métodos de AM; - 278 trabalhos analisados. 	[Xie et al. 2019]

Um conjunto de formulações baseadas em Programação Linear (PL) é apresentado para resolver o VNE em [Chowdhury et al. 2009]. Neste trabalho, utilizou-se as mesmas restrições de largura de banda, CPU e localização geográficas propostas em [Chowdhury et al. 2009], além de trabalhar utilizando aspectos de simulação. Em uma versão do VNE, similar à proposta em [Chowdhury et al. 2009], mas em um ambiente formado por múltiplos domínios, [Araújo et al. 2019] propõem um modelo baseado na Programação Linear Inteira (PLI) para realizar o mapeamento das requisições. No trabalho, é utilizada uma modelagem com as diferentes funções de otimização, como maximizar o lucro dos provedores e o balanceamento de carga dos enlaces do SN.

A modelagem em PLI proposta em [Araújo et al. 2019] é similar a proposta em [Beikta et al. 2017] e [Souza et al. 2018], e utilizada neste trabalho para realizar os mapeamentos de requisições de forma exata. Abordando a mesma variação do VNE tratada em [Chowdhury et al. 2009], mas de maneira heurística, o trabalho de [Araújo et al. 2018] propõe uma abordagem de rápida execução chamada de GRASPF-RVNS que executa um mapeamento em até $300ms$. Neste trabalho foi adotada a abordagem GRASPF-RVNS para realizar os mapeamentos das requisições de forma heurística, mas sua condição de parada foi alterada de $300ms$ para 10 iterações do GRASP e 20 iterações da busca local (RVNS) sem melhora.

Para prover o posicionamento de funções virtuais de rede em um modelo de comunicação com endereçamento a um único destino, [Cohen et al. 2015] propõem uma resolução exata através de PLI em um ambiente estático. O objetivo de abordagem de [Cohen et al. 2015] é minimizar a distância entre os usuários finais e as funções virtuais de redes alocadas. Em um contexto similar, [Luizelli et al. 2015] apresentam uma modelagem em PLI e uma heurística para resolver o posicionamento e encadeamento das funções de redes. A heurística proposta por [Luizelli et al. 2015] faz uso da própria PLI para realizar as iterações, e cada iteração é feita com um limite de tempo. Ao final de cada iteração a melhor solução é guardada e o número de funções instanciadas passa a ser uma restrição à iteração seguinte.

Muitos pesquisadores estudam sobre o potencial da aplicação de AM em redes [Zhao et al. 2019, Xie et al. 2019]. O trabalho de [Cheng and Chang 1997] utiliza AM supervisionado por meio do Perceptron Multicamadas (MLP) para realizar classificações e auxiliar a tomada de decisões no problema de controle de admissão em redes do tipo *Asynchronous Transfer Mode*. Segundo [Cheng and Chang 1997], o tratamento desse problema é dificultado por diversas características de tráfego e fortes requisitos de QoS. O trabalho de [Cheng and Chang 1997] considera o status de congestionamento da rede local para classificar uma requisição como rejeitada. Similarmente a [Cheng and Chang 1997], nesta proposta é considerado o estado do SN residual, mas para classificar o tempo de processamento da requisição, e não sua rejeição ou aceitação.

Em uma aplicação de AM no VNE, [Blenk et al. 2016] propõem o uso de Redes Neurais Recorrentes (RNN) para um controle de admissão de requisições virtuais. Para gerar o aprendizado, 18 RNNs diferentes foram treinadas usando um conjunto de dados gerado por simulações. Foi demonstrado que as RNNs atingem precisões acima de 89%, aprendendo com êxito quais requisições podem ser aceitas e processadas; e quais devem ser rejeitadas por alguma inviabilidade. Segundo [Blenk et al. 2016], as requisições não processadas geram um ganho computacional de processamento evitado pelo provedor.

Em uma vertente aplicada as redes SDN, [Amaral et al. 2016] apresentam um modelo para a classificação de tráfego resultante. A classificação de tráfego é benéfica para os operadores de rede alocarem os recursos mais eficientemente. Na mesma linha de [Amaral et al. 2016], esta proposta atua no campo da classificação, mas para auxiliar no processamento de uma requisição virtual e alocação de recursos.

3. Aplicação do Aprendizado de Máquina

O modelo de classificação utilizado nesse trabalho aplica o AM supervisionado para classificar qual seria o tempo de processamento gasto por uma requisição, se ela fosse mapeada por uma abordagem exata. Como se trata de uma proposta original, não foram encontrados trabalhos similares ou *datasets* na literatura para serem usados como referência. Optou-se por experimentar e analisar todo um conjunto de classificadores conhecidos na literatura, para após definir qual o mais indicado para o caso em estudo.

3.1. Classificadores e Caracterização dos *Datasets*

A tarefa dos algoritmos de classificação é prever a qual classe os dados de entrada pertencem [Zhao et al. 2019]. Neste trabalho considerou-se duas classes: se uma requisição vai demandar um alto tempo de computação através de um método de resolução exato (mais de $300ms$, classe 0), ou não (classe 1). Os classificadores selecionados para os experimentos foram o *Naive Bayes* (NBC), *Random Forest* (RF), *Gradient Boosted Trees* (GBT), *AdaBoost* (ADA) e o MLP, sendo todos importados do *scikit-learn toolbox*.

O *dataset*¹ utilizado para os treinamentos foi gerado previamente a partir de simulações usando o *SN*, a abordagem exata, e os mesmos parâmetros de requisições, e configurações de *hardware* propostos nos experimentos preliminares (Subseção 1.1). Como o *dataset* foi obtido a partir de simulação, não existem valores faltantes. O conjunto de *features* (colunas) de cada entrada do *dataset* (linhas) é formado por valores que correspondem ao residual de CPU ofertado por cada nó físico, residual de banda ofertado por cada enlace físico, demanda de CPU requisitada por cada nó virtual, demanda de banda requisitada por cada enlace virtual, raio de mapeamento de cada requisição, localização geográfica de cada componente virtual, e o tempo de processamento (objetivo). A geração das requisições para serem mapeadas possuem uma aleatoriedade referente à quantidade de enlaces e roteadores virtuais demandados. Para se contornar esse problema, foi definido que cada entrada do *dataset* possui até 12 roteadores e 50 enlaces virtuais, e caso estes não sejam demandados em sua totalidade, são substituídos com o valor 1 nas respectivas colunas e linhas. Desse modo, cada entrada do *Dataset* possui exatamente 531 *features*, geradas por 39.825 requisições previamente processadas. Por fim, o *dataset* possui 59.8% de requisições rotuladas como pertencentes à classe 1, e os outros 40,2% como pertencem à classe 0.

3.2. Métricas de Avaliação em AM

Para o treinamento dos classificadores foi utilizada a técnica de validação cruzada *k-fold*, com $k = 5$. As métricas são calculadas em função do número de instâncias classificadas

¹O *dataset*, o código em *Python*, e os classificadores treinados estão disponíveis em homepages.dcc.ufmg.br/~smaal/

como sendo da classe 1, sendo que elas pertencem à classe 1 (verdadeiro positivo, VP); classificadas como sendo da classe 0, sendo que elas pertencem à classe 0 (verdadeiro negativo, VN); classificadas como sendo da classe 1, mas que pertencem à classe 0 (falso positivo, FP); e as classificadas como sendo da classe 0, mas que pertencem à classe 1 (falso negativo, FN) [Powers and Ailab 2011]. Com esses valores se define uma matriz de confusão para cada classificador, e calcula-se as seguintes métricas: acurácia $A = \frac{VP+VN}{VP+FP+VN+FN}$, que representa o quão o modelo está correto; e precisão $P = \frac{VP}{VP+FP}$, que retrata quanto das amostras classificadas como corretas, realmente estavam corretas.

3.3. Resultados e Análises dos Classificadores

As Tabelas 2 e 3 apresentam respectivamente as matrizes de confusão e as métricas aferidas para cada classificador treinado. A maioria dos métodos avaliados foi capaz de detectar uma quantidade expressiva de requisições com alto tempo de execução (VP). Inicialmente, percebe-se que o NBC foi extremamente eficiente em classificar corretamente os VP (Tabela 2), com um alto valor P (Tabela 3) e baixo FP, mas em contrapartida gerou uma alta taxa de FN, o que prejudicou a acurácia.

Tabela 2. Matrizes de confusão obtidas nas classificações das requisições.

		1	0	1	0	1	0	1	0	1	0
Classe predita	1	VP	FP	22534	1293	22155	1672	22214	1613	21641	2186
	0	FN	VN	4111	11883	1963	14034	1870	14124	1813	14181
Exemplo		NBC		RF		GBT		ADA		MLP	

Tabela 3. Métricas obtidas nas classificações das requisições.

Classificadores	NBC	RF	GBT	ADA	MLP
Acurácia (%)	0.865	0.909	0.913	0.90	0.904
Precisão (%)	0.902	0.894	0.898	0.866	0.929

De modo geral, todos os classificadores treinados, obtiveram resultados satisfatórios, exceto o NBC, demonstrando uma boa capacidade de aprendizagem, com a acurácia chegando a 91,3% para o GBT. Em linhas gerais, o algoritmo que se mostrou mais promissor para o cenário de classificação, apresentando altos valores de VP e VN foi o GBT, sendo este adotado para o restante do trabalho.

4. Solução Proposta

A abordagem híbrida, apresentada neste trabalho, tem o objetivo de se valer das benesses dos métodos exatos e heurísticos. Caso classificado que o tempo de processamento de uma requisição seja baixo, a escolha mais adequada é aplicar um tratamento exato, aproveitando a otimalidade da solução. Caso o tempo de processamento seja classificado com alto, maior que um limite de viável definido pelo cliente ou provedor, utiliza-se um tratamento heurístico, aproveitando os benefícios do baixo tempo de execução.

4.1. Abordagem híbrida

A Figura 2 mostra uma abstração do funcionamento da abordagem híbrida, na qual existe um fluxo *online* de chegada de requisições para serem mapeadas conforme a ordem de chegada. Nesse modelo, foi assumido que o classificador já está treinado e possui um nível de aprendizado satisfatório para uma boa classificação. O funcionamento da abordagem híbrida mostrada na Figura 2 é descrito a seguir:

- i) A requisição 1 chega ao provedor no tempo $t = 1$ para ser processada em um SN residual. Neste caso o mapeamento é classificado como rápido, e resolvido de modo exato com garantias de otimalidade da solução;
- ii) A requisição 2 chega ao provedor no tempo $t=2$ para ser processada em um SN residual com os recursos parcialmente alocados para servir à requisição 1. Suponha que exista uma disputa de recursos entre as requisições 1 e 2, e que existam poucos recursos disponíveis para atender à requisição 2. Neste caso o mapeamento é classificado como demorado, e resolvido de modo heurístico em um tempo de processamento viável;
- iii) A requisição 3 chega ao provedor no tempo $t=3$, mas antes de ser processada, a requisição 1 é terminada e devolve os recursos antes alocados para o SN. Neste caso o SN residual possui recursos parcialmente alocados para servir à requisição 2. Suponha que não exista uma disputa de recursos entre as requisições 2 e 3, e que existam muitos recursos disponíveis. Neste caso o mapeamento é classificado como rápido, e resolvido de modo exato com garantias de otimalidade da solução;
- iv) A requisição 4 chega ao provedor no tempo $t=4$ para ser processada em um SN residual com os recursos parcialmente alocados para servir às requisições 2 e 3. Cabe ao classificador definir em tempo polinomial se a requisição 4 pode ser mapeada rapidamente com um tratamento exato, se não, aplica-se um tratamento heurístico.

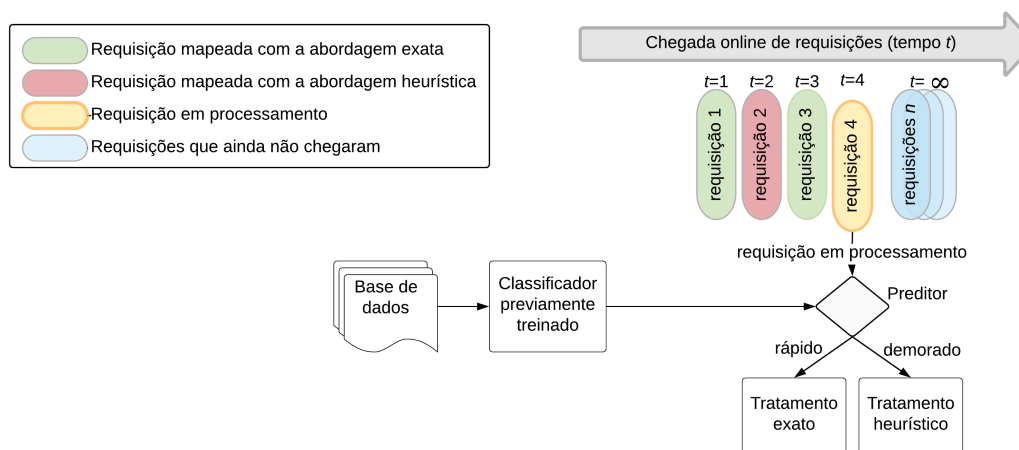


Figura 2. Fluxograma do processo de classificação e processamento de uma sequência de requisições tratadas pela abordagem híbrida.

No tratamento *online*, de modo similar a [Chowdhury et al. 2009], as requisições são enfileiradas e mapeadas seguindo a ordem de chegada, como mostrado na Figura 2. Sempre que uma requisição chega, ela é classificada de acordo com sua própria demanda de recursos, e com os recursos residuais do SN. É importante ressaltar que o classificador já foi treinado previamente, com várias configurações de requisições e recursos residuais, para efetuar uma classificação o mais assertiva possível (Subseção 3.1). Mas, em alguns casos, pode ocorrer uma classificação errada. Se uma requisição VP for rotulada como VN (*i.e.*, um FN), implicaria abrir mão da otimalidade, visto que ela poderia ser processada de modo exato, mas não foi. Do contrário, se uma requisição VN rotulada for como VP (*i.e.*, um FP), implicaria aplicar um tratamento exato em uma requisição difícil, o que pode gerar uma rejeição, se não for resolvido dentro do limite de tempo de estabelecido de $300ms$.

4.2. Métricas de Avaliação em Redes

As métricas utilizadas neste trabalho são as mesmas estabelecidas em [Chowdhury et al. 2009], são elas:

- i)* Receita de mapeamento, fornece um indicador de quanto um provedor receberá ao realizar o mapeamento de uma requisição. Expresso em função dos recursos virtuais demandados por cada requisição.
- ii)* Custo de mapeamento, fornece um indicador de quanto um provedor gastará ao realizar o mapeamento de uma requisição. Expresso em função dos recursos físicos gastos para atender cada requisição.
- iii)* Taxa de aceitação, expresso pela razão do número de requisições aceitas, e o número de requisições que chegaram até o fim do tempo de simulação;

4.3. Abordagens Avaliadas e Cenários de Simulação

Os experimentos computacionais desta seção foram realizados no mesmo ambiente, com o mesmo SN e cenários de avaliação ($S1$ e $S2$) propostos nos experimentos preliminares (Subseção 1.1). A classificação de uma requisição como rápida ou demorada foi feita com o classificador já treinado GBT. Com o intuito de avaliar o desempenho da solução proposta, diferentes abordagens que maximizam o balanceamento de carga são utilizadas:

- i)* Abordagem exata, utiliza a abordagem PLI_lb , proposta em [Araújo et al. 2019]. Para o processamento de cada requisição foi utilizado um tempo limite de 180 segundos;
- ii)* Abordagem heurística, utiliza a abordagem $GRASPF-RVNS$, proposta em [Araújo et al. 2018]. Como condição de parada foi adotado 10 iterações do $GRASP$ e 20 do $RVNS$ sem melhora, uma vez que valores superiores não apresentaram melhoras;
- iii)* Abordagem híbrida, realiza a integração entre o classificador GBT, e as abordagens heurística e exata.

4.4. Resultados e Análises da Abordagem híbrida

A Figura 3 apresenta a taxa de aceitação, o tempo computacional, a receita gerada e o custo para o processamento de cada cenário. Em cada gráfico apresentado é possível observar a métrica em questão no eixo y , e as abordagens exata, heurística e híbrida no eixo x . Devido à abordagem híbrida realizar o mapeamento ora de modo exato, ora heurístico, sua respectiva coluna é representada com o percentual que cada tratamento colaborou com a solução final. As abordagens híbrida e heurística são não determinísticas, foram executadas 10 vezes e são mostradas com um intervalo de 95% de confiança.

A abordagem heurística adotada ($GRASPF-RVNS$) obteve um bom desempenho em todos os experimentos realizados (Figura 3), gerando um espaço de otimização pequeno para melhorias em relação ao resultado da abordagem exata. Mesmo assim, a abordagem híbrida conseguiu elevar a taxa de aceitação e a receita de mapeamento em relação a abordagem heurística, e manteve um custo de mapeamento baixo, além de não infringir o limite de tempo de processamento, por requisição, definido pelo usuário ($300ms$). Ressalta-se que caso seja utilizada alguma outra heurística, que potencialmente gere resultados melhores que a heurística adotada, essa melhora incidirá positivamente no método híbrido, melhorando o desempenho da abordagem proposta.

Observando a abordagem híbrida nas Figuras 3(a) e 3(c), constata-se que apesar de 56,8% dos mapeamentos aceitos (284 ± 2 requisições) terem sido realizados de modo exato, essa proporção é responsável por somente 27,9s, *i.e.*, uma média aproximada de 100ms por requisição, bem menor que o limite aceitável de 300ms. Em contrapartida, os 28,0% processados pelo tratamento híbrido de maneira heurística, se tivessem sido processados de modo exato, teriam elevado o tempo computacional aferido de 36,9s para 979,0s. Essa mesma relação se repete de maneira similar nas Figuras 3(b) e 3(d). Essa análise evidencia que o classificador proposto conseguiu separar com êxito as requisições entre de processamento rápido ou demorado. Essas constatações, juntamente com os testes preliminares, corroboram com a afirmação que apesar do problema ser NP-difícil, em alguns momentos existem condições implícitas ao cenário que permitem a aplicação de um tratamento exato em um tempo viável.

Comparando as Figuras 3(a) e 3(b), foi percebido que ao se aumentar a frequência de chegada das requisições, a taxa de mapeamentos realizada pela abordagem híbrida com um tratamento heurístico tende a diminuir, sendo respectivamente de 28% e 18%. Essa queda pode ser explicada pela dificuldade de realizar mapeamentos heurísticos em momentos de uma alta chegada de requisições, poucos recursos disponíveis, e consequente maior disputa de recursos. Em contrapartida, a taxa de mapeamentos realizada pela abordagem híbrida com um tratamento exato se manteve estável, variando entre 55% e 56,8%, aspecto indica que em cenários com mais disputa de recursos, e potenciais rejeições, o método exato possui um desempenho mais regular.

Em todos os experimentos, mas mais nitidamente no cenário *S2* (Figuras 3(b), 3(f) e 3(h)), a abordagem híbrida conseguiu uma taxa de mapeamento e uma receita maior que a heurística, mas com um custo de mapeamento relativamente igual. Esse aspecto reflete a melhora no balanceamento de carga gerada pelo tratamento híbrido, visto que ele consegue aceitar mais requisições sem um aumento significativo no custo. Esse aspecto é benéfico tanto para o cliente, que vai possuir uma conexão mais balanceada, quanto para o provedor, que vai gerar uma receita maior com um custo constante.

No cenário *S1* (Figura 3(a)) todas as variações conseguiram uma taxa de aceitação relativamente alta, uma vez que as requisições chegaram com um alto intervalo de tempo e não sobrecarregaram os recursos disponíveis no *SN*. Mas a medida que a frequência de chegada das requisições aumenta, passando para o cenário *S2* (Figura 3(b)), a taxa de aceitação tende a diminuir acentuadamente. No experimento da Figura 3(b), considerando que a abordagem *PLI_lb* é ótima, pode-se afirmar que o *SN* não consegue servir todo o conjunto de requisições entrantes com um tratamento estático.

A Figura 4 mostra em intervalos de tempo de $0t$ a $10.000t$, a receita que cada abordagem gerou em relação aos valores recebidos por requisição ativa no *SN*, para tal foi utilizada uma execução aleatória de cada método. É nítido o aumento de receita gerado pela abordagem híbrida em relação à heurística. Essa situação é proveniente não só do aumento do número de requisições aceitas e ativas, o que gera uma receita maior (Figura 3(b)); mas também em função da receita ser determinada pelos recursos virtuais demandados por requisição, nesse sentido a abordagem híbrida conseguiu mapear uma parcela maior de requisições com uma alta cardinalidade equipamentos, que provavelmente a abordagem heurística rejeitou.

Um ponto a se observar é que a abordagem híbrida além de melhorar a taxa de aceitação em relação à heurística, reduziu eficientemente o tempo de execução em relação ao tratamento exato (Figura 3(c) e 3(d)). Esse aspecto demonstra a efetividade da aplicação de AM na alocação de recursos. Especificamente na Figura 3(b), o aumento na taxa de aceitação da abordagem híbrida em relação à heurística é consequência de 55,0% das requisições aceitas terem sido mapeadas de modo exato, o que potencialmente gerou um uso mais conciso dos recursos do *SN*. Por outro lado, a redução do tempo de processamento da abordagem híbrida em relação ao tratamento exato, se dá pelo fato das requisições mais difíceis de serem mapeadas terem sido tratadas de modo heurístico.

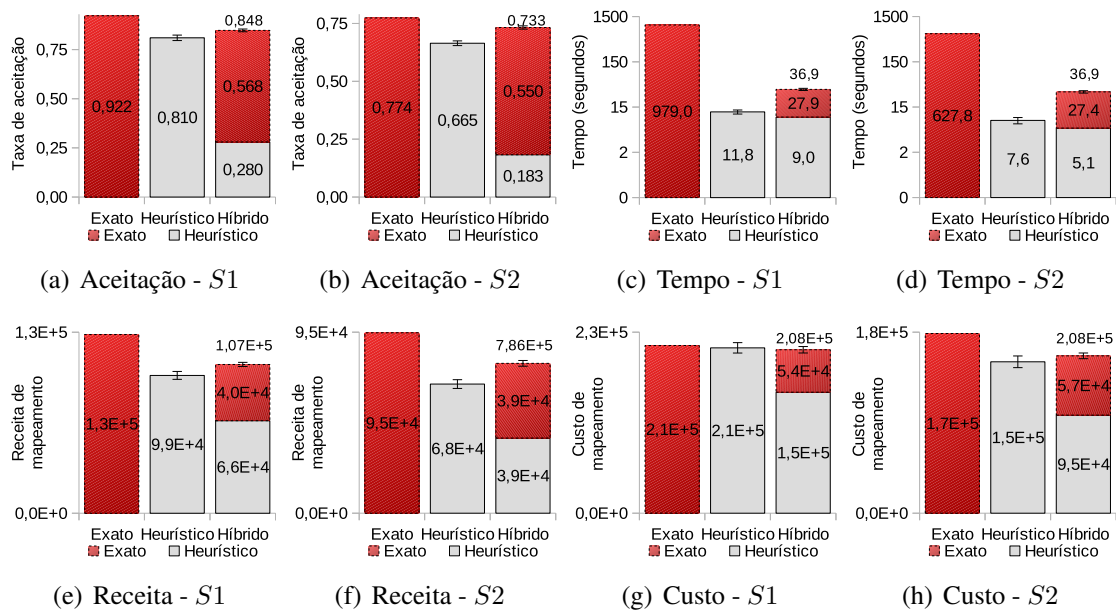


Figura 3. Taxa de aceitação final, tempo total, receita e custo de mapeamento.

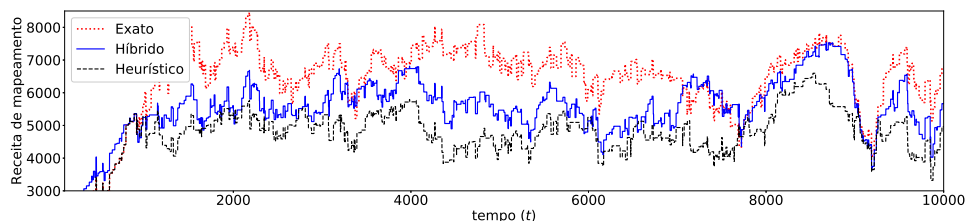


Figura 4. Receita média dos mapeamentos realizados no cenário S2.

Contrastando uma mudança na taxa de chegada das requisições, de um cenário mais denso (*S2*), para um menos denso (*S1*), percebe-se que o tempo de execução tende a aumentar no tratamento exato, como verificado comparando as Figuras 3(d) e 3(c). Esse fato decorre de uma alta abundância de recursos disponíveis em *S1* em relação a *S2*, o que gera mais restrições ativas no modelo, e como consequência mais soluções viáveis a serem analisadas no espaço de soluções, dificultando o processo de otimização exato.

5. Conclusões

Neste trabalho foi proposta uma abordagem baseada em AM que é capaz de auxiliar na tomada de decisão sobre em quais momentos se pode preterir uma abordagem

heurística em relação a uma exata no problema de alocação de recursos em redes. A abordagem proposta, chamada de híbrida, trabalha dentro de um intervalo de tempo viável para o cliente, para produzir um mapeamento que seja o mais otimizado possível em termos de balanceamento de carga. Neste sentido a abordagem híbrida conseguiu se beneficiar da otimalidade dos mapeamentos exatos em alguns casos, e do baixo tempo de processamento das heurísticas em outros.

Foi escolhido o VNE como ambiente para avaliar a proposta. Nos experimentos computacionais notou-se que diversos fatores podem influenciar o tempo de mapeamento de uma requisição, e que explicitar todos esses fatores pode ser custoso. Neste contexto, o AM foi aplicado para gerar um classificador que de maneira implícita classifique se uma requisição é de rápido processamento, nesse caso se aplica um tratamento exato; ou de processamento demorado, neste caso se aplica um tratamento heurístico.

Foram treinados vários classificadores conhecidos da literatura, dentre eles, o GBT foi o que melhor se comportou, com uma acurácia de 91,3%. Fazendo uso do GBT, a abordagem híbrida reduziu o tempo de processamento em relação ao modelo exato, além de ter aumentado a taxa de aceitação e a receita em relação à heurística. Em contra partida, a abordagem híbrida gerou um tempo computacional um pouco mais alto que a heurística, mas em momento algum descumpriu o limite de tempo de QoS estabelecido.

A abordagem híbrida se mostrou promissora quando aplicada no VNE. Em trabalhos futuros, pretende-se estender esse arcabouço teórico às redes NFV, tratar requisições com demanda elástica, mapeamentos dinâmicos e prioridades de requisições.

Agradecimentos

Os autores agradecem ao CNPq, CAPES e FAPEMIG.

Referências

- Amaldi, E., Coniglio, S., Koster, A., and Tieves, M. (2016). On the computational complexity of the virtual network embedding problem. *Electronic Notes in Discrete Mathematics*, 52:213–220.
- Amaral, P., Dinis, J., Pinto, P., Bernardo, L., Tavares, J., and Mamede, H. S. (2016). Machine learning in software defined networks: Data collection and traffic classification. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–5.
- Araújo, S., Souza, F. S., and Mateus, G. R. (2019). Abordagens exata e heurística para o mapeamento de redes virtuais. In *Anais Estendidos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 153–160, Porto Alegre, RS, Brasil. SBC.
- Araújo, S. M. A., de Souza, F. S. H., and Mateus, G. R. (2018). Virtual network embedding in multi-domain environments with energy efficiency concepts. In *2018 International Conference on Information Networking (ICOIN)*, pages 205–210.
- Beikta, C., Gzpek, D., Ula, A., and Lokman, E. (2017). Secure virtual network embedding with flexible bandwidth-based revenue maximization. *Comput. Netw.*, 121(C):89–99.

- Blenk, A., Kalmbach, P., van der Smagt, P., and Kellerer, W. (2016). Boost online virtual network embedding: Using neural networks for admission control. In *2016 12th International Conference on Network and Service Management (CNSM)*, pages 10–18.
- Cheng, R. . and Chang, C. . (1997). Neural-network connection-admission control for atm networks. *IEE Proceedings - Communications*, 144(2):93–98.
- Chowdhury, N. M. M. K., Rahman, M. R., and Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *IEEE INFOCOM 2009*, pages 783–791.
- Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D. (2015). Near optimal placement of virtual network functions. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1346–1354.
- Fischer, A., Botero, J. F., Beck, M. T., de Meer, H., and Hesselbach, X. (2013). Virtual Network Embedding: A Survey. *IEEE Communications Surveys Tutorials*, 15(4):1888–1906.
- Laghrissi, A. and Taleb, T. (2018). A Survey on the Placement of Virtual Resources and Virtual Network Functions. *IEEE Communications Surveys Tutorials*, pages 1–1.
- Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P., and Gaspary, L. P. (2015). Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 98–106.
- Nokia (2016). White paper: Dynamic end-to-end network slicing for 5g. [online] available at <https://resources.ext.nokia.com/asset/200339>.
- Powers, D. and Ailab (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *J. Mach. Learn. Technol*, 2:2229–3981.
- Shi, R., Zhang, J., Chu, W., Bao, Q., Jin, X., Gong, C., Zhu, Q., Yu, C., and Rosenberg, S. (2015). Mdp and machine learning-based cost-optimization of dynamic resource allocation for network function virtualization. In *2015 IEEE International Conference on Services Computing*, pages 65–73.
- Souza, B. A., Mateus, G. R., and de Souza, F. S. (2018). Compact and extended formulations for the virtual network embedding problem. *Electronic Notes in Discrete Mathematics*, 64:205 – 214. 8th International Network Optimization Conference - INOC 2017.
- Waxman, B. M. (1988). Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622.
- Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., and Liu, Y. (2019). A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. *IEEE Communications Surveys Tutorials*, 21(1):393–430.
- Zhao, Y., Li, Y., Zhang, X., Geng, G., Zhang, W., and Sun, Y. (2019). A survey of networking applications applying the software defined networking concept based on machine learning. *IEEE Access*, 7:95397–95417.