



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Alocação de Infraestruturas Virtuais utilizando Algoritmos Evolucionários Multiobjetivo

Dissertação de Mestrado

Wesley Oliveira Souza



São Cristóvão – Sergipe

2019

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Wesley Oliveira Souza

Alocação de Infraestruturas Virtuais utilizando Algoritmos Evolucionários Multiobjetivo

Proposta de Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

Orientador: Ricardo José Paiva de Britto Salgueiro

São Cristóvão – Sergipe

2019

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

Souza, Wesley Oliveira

S729a Alocação de infraestruturas virtuais utilizando algoritmos evolucionários multiobjetivo / Wesley Oliveira Souza ; orientador Ricardo José Paiva de Brito Salgueiro. - São Cristóvão, 2019.
97 f. : il.

Dissertação (mestrado em Ciência da Computação) –
Universidade Federal de Sergipe, 2019.

1. Computação. 2. Computação em nuvem. 3. Ambientes virtuais compartilhados. 4. Algoritmos computacionais. 5. Alocação de recursos. I. Salgueiro, Ricardo José Paiva de Brito orient. II. Título.

CDU 004



**UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**Ata da Sessão Solene de Defesa da Dissertação do
Curso de Mestrado em Ciência da Computação-UFS.
Candidato: Wesley Oliveira Souza**

Em 28 dias do mês de agosto do ano de dois mil e dezenove, com início às 14h00min, realizou-se na Sala de Hardware II do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **Wesley Oliveira Souza**, que desenvolveu o trabalho intitulado: "*Alocação de Infraestruturas Virtuais utilizando Algoritmos Evolucionários Multiobjetivo*", sob a orientação do Prof. Dr. **Ricardo José Paiva de Britto Salgueiro**. A Sessão foi presidida pelo Prof. Dr. **Ricardo José Paiva de Britto Salgueiro** (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. **André Britto de Carvalho** (PROCC/UFS) e, em seguida, ao Prof^ª. **Edilayne Meneses Salgueiro** (Externo ao Programa/UFS) e Prof. **Rubens de Souza Matos Júnior** (Externo à instituição/IFS). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) APROVADO "(aprovado/reprovado)". Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 25/2014/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária "Prof. José Aloísio de Campos", 28 de Agosto de 2019.


Prof. Dr. Ricardo José Paiva de Britto Salgueiro
(PROCC/UFS)
Presidente


Prof. Dr. André Britto de Carvalho
(PROCC/UFS)
Examinador Interno


Prof.ª Dr.ª Edilayne Meneses Salgueiro
(Dcomp/UFS)
Examinador Externo ao Programa


Prof. Dr. Rubens de Souza Matos Júnior
(IFS)
Examinador Externo à Instituição


Wesley Oliveira Souza
Candidato

Dedico esse trabalho aos meus pais e professores.

Agradecimentos

Esse trabalho foi realizado em um momento de grandes mudanças e desafios, não seria possível concluí-lo sem o apoio de muitas pessoas. Inicialmente, agradeço aos meus pais Ivoncio e Josefina, por todo apoio, carinho e sacrifícios que me permitiram alcançar esse momento, contem sempre comigo. As minhas irmãs Aline e Mylena. A minha esposa Karen pelo incentivo, companheirismo e paciência, no decorrer do trabalho sempre estava disposta a ajudar no que fosse necessário, acompanhou de perto a maioria das atividades deixando claro que estava me apoiando, obrigado por tudo, te amo. Vale um agradecimento especial para meu orientador Ricardo Salgueiro e a professora Edilayne Meneses Salgueiro que praticamente me trataram como um filho, obrigado por todo apoio e empenho, por me acolherem alguns anos atrás e me possibilitarem fazer parte da família do ELAN. A todos que fazem parte do ELAN, especialmente ao meu grande amigo e colega de mestrado Itauan, sempre solícito e disposto a ajudar os colegas. Não poderia deixar de citar também meus amigos do ELAN Túlio, Felipe, Jonathan, Carolina e Lucas. A todos os membros do STI da UFS, especialmente a Alexandre, Marco, Andres, Dilton, Seu Eric, Seu Thiago, Seu Cléber, Seu Valdemir, Karina e Rodrigus, por me acolher e apoiar a conclusão deste trabalho. A UFS e a CAPES pela oportunidade de ser um dos beneficiados por seu grande potencial transformador, cursando uma graduação e uma pós-graduação em Ciência da Computação e por serem agentes de transformação social. E Finalmente, a todos aqueles que lutam por políticas que possibilitam que pessoas menos favorecidas trilhem o caminho do conhecimento, talvez sem vocês eu nunca tivesse a oportunidade de realizar esse trabalho. MUITO OBRIGADO!

Cometa erros de ambição em vez de erros de acomodação. Desenvolva a força para fazer coisas ousadas, não a força para tolerar o sofrimento.

(Nicolau Maquiavel)

Resumo

Para atender os requisitos dos usuários, Provedores de Infraestrutura (InPs, do inglês *Infrastructure Providers*) começaram a oferecer Infraestruturas Virtuais (VI, do inglês *Virtual Infrastructure*) como um serviço. Dentre as tarefas necessárias para oferecer VIs como um serviço, a principal é a alocação das VIs solicitadas na infraestrutura física. O processo de alocação consiste em identificar dentro da infraestrutura um conjunto de recursos para hospedar os componentes das VIs. Porém, o processo de alocação não é trivial pois ele deve respeitar os requisitos de rede e computação pré-definidos. Além disso, para um gerenciamento efetivo da infraestrutura, o balanceamento da carga e a redução do *overhead* de alocação são essenciais. Do mesmo modo, no processo de alocação, alguns objetivos dos InPs e usuários devem ser considerados. Geralmente guiados por suas perspectivas financeiras, os InPs desejam maximizar a sua receita alocando o maior número de VIs, usando a menor infraestrutura possível. Por outro lado, os usuários desejam majoritariamente VIs eficientes e de baixo custo. Portanto, o processo de alocação é complexo e deve atender a um conjunto considerável de restrições. Para resolver esse problema, o presente trabalho apresentou um Algoritmo Evolucionário Multiobjetivo (MOEA, do inglês *Multi-Objective Evolutionary Algorithm*) para alocar VIs em uma infraestrutura física, atendendo os requisitos de computação e rede, avaliando soluções que atendam objetivos como: balanceamento de carga e baixo *overhead* de alocação. O MOEA empregou o modelo proposto para o mapeamento dos recursos virtuais das VIs nos recursos físicos da infraestrutura física. Ademais, foi desenvolvido um simulador para avaliar soluções para o problema de alocação de VIs. A avaliação experimental utilizou a técnica de simulação para avaliar o desempenho da solução proposta. Desse modo, os algoritmos foram implementados na linguagem Java, e uma análise comparativa foi realizada entre diferentes algoritmos que aplicavam o modelo de alocação proposto. Sendo assim, para avaliar o desempenho dos algoritmos foram utilizadas as métricas: tempo para atender uma requisição de VI, lucro do provedor, taxa de rejeição e eficiência do uso da infraestrutura física. Além disso, foi utilizada uma topologia baseada no modelo de três camadas da Cisco para representar a infraestrutura física. Os resultados experimentais demonstram que o MOEA baseado no Algoritmo Genético (GA, do inglês *Genetic Algorithm*) desenvolvido apresenta resultados promissores para diversos cenários, combinando rapidez e eficiência na alocação. O modelo de alocação proposto se mostrou útil para o mapeamento dos recursos virtuais nos recursos físicos. Portanto, o presente trabalho contribui com uma possível solução para o problema e abre caminho para novas propostas que podem empregar o simulador e o modelo proposto.

Palavras-chave: Computação em Nuvem, Alocação, Infraestruturas Virtuais, Algoritmos Evolucionários.

Abstract

To meet user requirements, Infrastructure Providers (InPs) began offering Virtual Infrastructure (VI) as a service. Among the tasks required to offer VIs as a service, the main one is the allocation of the requested VIs in the physical infrastructure. The allocation process consists of identifying within the infrastructure a feature set to host the components of the VIs. However, the allocation process is not trivial as it must meet predefined network and computing requirements. In addition, for effective infrastructure management, load balancing and reduction of allocation overhead is essential. Similarly, in the allocation process, some objectives of InPs and users should be considered. Generally guided by their financial perspective, InPs want to maximize their revenue by allocating as many VIs as possible using the smallest possible infrastructure. On the other hand, users mostly want efficient and low cost VIs. Therefore, the allocation process is complex and must meet a considerable set of constraints. To address this problem, the present dissertation presented an Evolutionary Multiobjective Algorithm (MOEA) to allocate VIs on a physical infrastructure, meeting computation and network requirements, evaluating solutions that meet goals such as load balancing and low allocation overhead. MOEA employed the proposed model for mapping the virtual resources of VIs into the physical resources of the physical infrastructure. In addition, a simulator was developed to evaluate solutions to the VI allocation problem. The experimental evaluation employed the simulation technique to evaluate the performance of the proposed solution. Thus, the algorithms were implemented in the Java language, and a comparative analysis was performed between different algorithms that employed the proposed allocation model. Thus, to evaluate the performance of the algorithms, the following metrics were used: time to fulfill an IV request, provider profit, rejection rate and efficiency of physical infrastructure use. In addition, a Cisco three-tier model-based topology was used to represent the physical infrastructure. The experimental results show that the developed Genetic Algorithm (GA) based MOEA presents promising results for several scenarios, combining speed and efficiency in the allocation. The proposed allocation model proved to be useful for mapping the virtual resources of VIs into the physical resources of the physical infrastructure. Therefore, the present work contributes to a possible solution to the problem and opens the way for new proposals that may employ the simulator and the proposed model.

Keywords: Cloud Computing, Allocation, Virtual Infrastructures, Evolutionary Algorithms.

Lista de ilustrações

Figura 1 – Número de estudos primários encontrados em cada base utilizando a <i>string</i> de busca.	30
Figura 2 – Número de artigos selecionados em cada base após a filtragem por meio dos critérios de seleção.	32
Figura 3 – Diagrama com a quantidade de artigos removida ou adicionada em cada fase do mapeamento.	34
Figura 4 – Gráfico que apresenta a quantidade de publicações por ano relacionados a alocação de VIs.	36
Figura 5 – Gráfico que apresenta os estudos primários por ano relacionados a VNE.	37
Figura 6 – Gráfico que apresenta os estudos primários por ano relacionados a alocação de VIs e VNE.	38
Figura 7 – Número de artigos selecionados de cada base após a etapa da leitura completa.	39
Figura 8 – Quantidade de artigos que se dedicaram a resolver o problema de alocação de VIs e VNE.	40
Figura 9 – Fluxograma do GA utilizado na abordagem proposta.	50
Figura 10 – Infraestrutura Física e Requisição de VI	52
Figura 11 – Cromossomo e codificação utilizada.	53
Figura 12 – Operação de <i>crossover</i> multiponto.	54
Figura 13 – Operação de mutação.	54
Figura 14 – Alocação de diferentes tipos de nós virtuais em físicos numa infraestrutura virtualizada.	60
Figura 15 – Exemplo de uma infraestrutura física que pode ser simulada por meio do simulador.	62
Figura 16 – Requisições de VIs simples.	63
Figura 17 – Cenário 1: tempo para alocação de uma requisição de VI.	72
Figura 18 – Cenário 1: lucro do provedor para cada algoritmo empregando 100 mil iterações.	73
Figura 19 – Cenário 1: lucro do provedor para cada algoritmo empregando 150 mil iterações.	73
Figura 20 – Cenário 1: lucro do provedor para cada algoritmo empregando 200 mil interações.	74
Figura 21 – Cenário 1: taxa de rejeição para cada algoritmo empregando 100 mil interações.	75
Figura 22 – Cenário 1: taxa de rejeição para cada algoritmo empregando 150 mil interações.	75
Figura 23 – Cenário 1: taxa de rejeição para cada algoritmo empregando 200 mil interações.	76
Figura 24 – Cenário 1: eficiência do uso da infraestrutura física dos algoritmos GA e SA.	77
Figura 25 – Cenário 2: tempo para alocação de uma requisição de VI.	78

Figura 26 – Cenário 2: lucro do Provedor para cada algoritmo empregando 100 mil interações.	79
Figura 27 – Cenário 2: lucro do provedor para cada algoritmo empregando 150 mil interações.	79
Figura 28 – Cenário 2: lucro do provedor para cada algoritmo empregando 200 mil interações.	80
Figura 29 – Cenário 2: taxa de rejeição para cada algoritmo empregando 100 mil interações.	81
Figura 30 – Cenário 2: taxa de rejeição para cada algoritmo empregando 150 mil interações.	81
Figura 31 – Cenário 2: taxa de rejeição para cada algoritmo empregando 200 mil interações.	82
Figura 32 – Cenário 2: eficiência do uso da infraestrutura física dos algoritmos GA e SA.	83
Figura 33 – Cenário 3: tempo para alocação de uma requisição de VI.	84
Figura 34 – Cenário 3: lucro do provedor para cada algoritmo empregando 100 mil interações.	85
Figura 35 – Cenário 3: lucro do provedor para cada algoritmo empregando 150 mil interações.	85
Figura 36 – Cenário 3: lucro do provedor para cada algoritmo empregando 200 mil interações.	86
Figura 37 – Cenário 3: taxa de rejeição para cada algoritmo empregando 100 mil interações.	87
Figura 38 – Cenário 3: taxa de rejeição para cada algoritmo empregando 150 mil interações.	87
Figura 39 – Cenário 3: taxa de rejeição para cada algoritmo empregando 200 mil interações.	88
Figura 40 – Cenário 3: eficiência do uso da infraestrutura física dos algoritmos GA e SA.	89

Lista de tabelas

Tabela 1 – Trabalhos relacionados e suas principais características	26
Tabela 2 – Quantidade de artigos encontrados após a aplicação da <i>string</i> de busca nas bases de pesquisa.	30
Tabela 3 – Quantidade de artigos encontrada após a filtragem por meio dos critérios de seleção.	31
Tabela 4 – Quantidade de artigos aprovados de cada base após a leitura completa. . . .	33
Tabela 6 – Referências e principais métricas utilizadas nos estudos para avaliar soluções de alocação de VIs.	41
Tabela 7 – Referências e principais métricas utilizadas nos estudos para avaliar soluções para VNE.	42
Tabela 8 – Resumo das principais características dos estudos primários relacionados a alocação de VIs.	43
Tabela 10 – Notação da formulação das funções de avaliação.	56
Tabela 11 – Fatores de estudo e níveis relacionado aos algoritmos.	68
Tabela 12 – Fatores de estudo e níveis relacionados a topologia da infraestrutura física. .	68
Tabela 13 – Fatores de estudo e níveis relacionados aos tipos de requisição de VIs. . . .	69
Tabela 14 – Cenários elaborados para a execução dos experimentos.	69
Tabela 15 – Recursos computacionais utilizados.	70
Tabela 16 – Cenário 1: tempo médio utilizado por cada algoritmo para realizar uma alocação de VI.	72
Tabela 17 – Cenário 2: tempo médio utilizado por cada algoritmo para realizar uma alocação de VI.	78
Tabela 18 – Cenário 3: tempo médio utilizado por cada algoritmo para realizar uma alocação de VI.	84

Lista de abreviaturas e siglas

ACM	<i>Association for Computing Machinery Digital Library</i>
GA	<i>Genetic Algorithm</i> (Algoritmo Genético)
IaaS	<i>Infrastructure-as-a-Service</i> (Infraestrutura como um serviço)
IEEE	<i>Institute of Electric and Electronic Engineers Xplore</i>
InPs	<i>Infrastructure Providers</i> (Provedores de Infraestrutura)
LP	<i>Linear Programming</i> (Programação Linear)
MIP	<i>Mixed Integer Programming</i> (Programação Linear Inteira Mista)
MOEA	<i>Multi-Objective Evolutionary Algorithms</i> (Algoritmos Evolucionários Multiobjetivo)
MOP	<i>Multi-Objective Optimization Problem</i> (Problema de Otimização Multiobjetivo)
NSGA-II	<i>Non dominated Sorting Genetic Algorithm II</i> (Algoritmo Genético de Ordenação Não Dominada II)
PaaS	<i>Platform-as-a-Service</i> (Plataforma como um serviço)
PSO	<i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas)
QoS	<i>Quality-of-Service</i> (Qualidade de Serviço)
SA	<i>Simulated Annealing</i> (Recozimento Simulado)
SaaS	<i>Software-as-a-Service</i> (Software como um serviço)
SD	<i>Science Direct</i>
SDN	<i>Software Defined Network</i> (Rede Definida por Software)
SLA	<i>Service Level Agreement</i> (Acordo de Nível de Serviço)
VI	<i>Virtual Infrastructure</i> (Infraestrutura Virtual)
VIE	<i>Virtual Infrastructure Embedding</i> (Incorporação de Infraestrutura Virtual)
VM	<i>Virtual Machine</i> (Máquina Virtual)
VNE	<i>Virtual Network Embedding</i> (Incorporação de Redes Virtuais)

Sumário

1	INTRODUÇÃO	16
1.1	Problemática	19
1.2	Hipótese	20
1.3	Objetivo	21
1.4	Trabalhos Relacionados	21
1.5	Organização do Trabalho	25
2	ALOCAÇÃO DE INFRAESTRUTURAS E REDES VIRTUAIS EM COMPUTAÇÃO EM NUVEM: UM MAPEAMENTO SISTEMÁTICO	27
2.1	Introdução	27
2.2	Método	28
2.2.1	Questões de Pesquisa	29
2.2.2	Estratégia de Busca e Seleção	29
2.2.3	CrITÉRIOS de Seleção	31
2.2.4	Seleção por meio da leitura completa	32
2.2.5	Considerações finais sobre o método	33
2.3	Análise dos Resultados	34
2.4	Considerações finais sobre o Mapeamento	46
3	MODELO PARA ALOCAÇÃO DE INFRAESTRUTURAS VIRTUAIS UTILIZANDO ALGORITMO EVOLUCIONÁRIO MULTIOBJETIVO	48
3.1	Otimização Multiobjetivo	48
3.2	Algoritmos Evolucionários Multiobjetivo	49
3.3	Modelo de Alocação Proposto	49
3.3.1	Representação do cromossomo	51
3.3.2	Seleção	53
3.3.3	Crossover	53
3.3.4	Mutação	53
3.3.5	Checagem de viabilidade	54
3.3.6	Função de Avaliação Multiobjetivo	55
3.3.6.1	Balanceamento de carga	55
3.3.6.2	Overhead de Alocação	58
3.4	Algoritmo	59
3.5	Simulador	59
3.5.1	Estrutura	60
3.5.2	Serviços	61

4	AVALIAÇÃO EXPERIMENTAL	65
4.1	Método de Avaliação	65
4.1.1	Definição dos objetivos e escopo do sistema	65
4.1.2	Definição dos serviços e saídas	66
4.1.3	Métricas	66
4.1.4	Definição dos Parâmetros	66
4.1.5	Fatores de estudo	67
4.1.5.1	Fatores de estudo dos algoritmos	67
4.1.5.2	Fatores de estudo da topologia da infraestrutura física	68
4.1.5.3	Fatores de estudo da topologia da requisição de VI	68
4.1.6	Técnica de avaliação	69
4.1.7	Carga de trabalho	69
4.1.8	Planejamento dos experimentos	69
4.1.9	Análise e interpretação dos resultados	71
4.2	Resultados Experimentais	71
4.2.1	Cenário 1 - Topologia com 4 racks e 4 servidores por rack	71
4.2.1.1	Tempo necessário para alocação de uma requisição de VI	71
4.2.1.2	Lucro do provedor	72
4.2.1.3	Taxa de rejeição	74
4.2.1.4	Eficiência do uso da infraestrutura física	76
4.2.2	Cenário 2 - Topologia com 4 racks e 8 servidores por rack	77
4.2.2.1	Tempo necessário para alocação de uma requisição de VI	77
4.2.2.2	Lucro do provedor	78
4.2.2.3	Taxa de Rejeição	80
4.2.2.4	Eficiência do uso da infraestrutura física	82
4.2.3	Cenário 3 - Topologia com 4 racks e 12 servidores por rack	83
4.2.3.1	Tempo necessário para alocação de uma requisição de VI	83
4.2.3.2	Lucro do provedor	84
4.2.3.3	Taxa de rejeição	86
4.2.3.4	Eficiência do uso da infraestrutura física	88
5	CONCLUSÃO	90
5.1	Considerações Finais	90
5.2	Principais Contribuições	91
5.3	Trabalhos Futuros	91
	REFERÊNCIAS	93

1

Introdução

A Computação em Nuvem instaurou uma nova era por meio do provisionamento dinâmico de recursos virtualizados e serviços conduzidos por requisitos dos usuários, ao passo que, possibilitou acesso a um conjunto compartilhado de recursos computacionais configuráveis (e.g. redes, servidores, armazenamento, aplicações e serviços) de maneira dinâmica, sob demanda e através da Internet. (MELL; GRANCE *et al.*, 2011).

Com o acesso facilitado aos recursos que necessitam para seus negócios, os usuários não precisam fazer investimentos em infraestrutura, eles podem alugar recursos que são oferecidos por Provedores de Infraestrutura (InPs, do inglês *Infrastructure Providers*) - que gerenciam sua infraestrutura de TI - com base no modelo *pay-as-you-go* (modelo de contratação onde se paga apenas pelo que se usa) (CHOWDHURY; RAHMAN; BOUTABA, 2012).

Os recursos disponibilizados pelos InPs são ofertados por meio de modelos de serviço. Os principais modelos de serviço são: Infraestrutura como serviço (IaaS, do inglês *Infrastructure-as-a-Service*), (Paas, do inglês *Platform as a Service*) e Software como serviço (SaaS, do inglês *Software-as-a-Service*) (MAIYAMA *et al.*, 2017). O presente trabalho aborda especificamente IaaS, que é a entrega de recursos computacionais virtualizados (servidores, armazenamento e rede) sob demanda (MELL; GRANCE *et al.*, 2011; BUYYA; YEO; VENUGOPAL, 2008).

Geralmente, InPs de nuvem oferecem Máquinas Virtuais (VMs, do inglês, *Virtual Machines*), baseados em serviços de hospedagem. As VMs são provisionadas por meio de um software chamado *hypervisor* (BARHAM *et al.*, 2003), que permite compartilhar os recursos de um servidor por meio de várias VMs, atendendo aos requisitos dos usuários e garantindo isolamento entre as máquinas virtuais que estão rodando no mesmo hardware físico (SAHOO; MOHAPATRA; LATH, 2010). No entanto, esses esquemas de alocação geralmente levam a problemas graves de desempenho de rede, devido à falta de otimização de rede e garantia de largura de banda (YU; XUE; ZHANG, 2015).

Por outro lado, também tem-se ofertado, de forma semelhante aos *hypervisores* de

VMs, Redes Virtuais, compostas por roteadores e enlaces virtuais. As Redes Virtuais, por sua vez, fazem uso de roteadores e enlaces na rede física, conjunto de recursos normalmente chamado de substrato da rede. Assim, permitem que várias redes virtuais heterogêneas e isoladas logicamente coexistam em uma infraestrutura física compartilhada (SHERWOOD *et al.*, 2009; CHOWDHURY; BOUTABA, 2009). Todavia, redes virtuais não oferecem recursos de computação.

Para atender as necessidades dos usuários, InPs começaram a oferecer Infraestruturas Virtuais (VI, do inglês *Virtual Infrastructure*). Basicamente, uma VI é composta por VMs interconectadas por recursos de rede, em que o número de recursos virtuais (por exemplo, VMs, comutadores, roteadores e canais de comunicação) e sua configuração (como, processamento, armazenamento e largura de banda) podem ser ajustados com base em requisitos pré-definidos (SOUZA *et al.*, 2017b). Esse novo conceito estendeu a entrega de IaaS, incluindo o provisionamento dinâmico de recursos virtuais e de comunicação (ANHALT; KOSLOVSKI; PRIMET, 2010).

Com a aplicação do conceito de VIs, VMs, roteadores e redes virtuais são oferecidos conjuntamente como um serviço, compondo dinamicamente VIs provisionadas. Essa combinação de computação em nuvem e rede de nuvem (*cloud networking*) possibilita a criação de infraestruturas inteiramente virtualizadas sobre uma rede física. (OLIVEIRA; KOSLOVSKI, 2017). Portanto, por meio da combinação da virtualização da rede de computação é possível para os InPs provisionar dentro da mesma rede física várias VIs heterogêneas com isolamento de tráfego (OLIVEIRA; KOSLOVSKI, 2017).

Utilizando VIs, InPs conseguem aproveitar todas as vantagens da virtualização (SHERWOOD *et al.*, 2009). É possível por exemplo, migrar um componente da VI (por exemplo, uma VM) para uma localização diferente em caso de falha do substrato físico (KOSLOVSKI *et al.*, 2010). Ademais, elas podem atender aos requisitos de aplicações com processamento de dados intensivo que requerem acesso às infraestruturas com alto desempenho e facilidades de movimentação de dados coordenados com recursos computacionais (ANHALT; KOSLOVSKI; PRIMET, 2010).

Por meio das VIs, InPs podem oferecer ao usuário solicitante controle total sobre a agregação de recursos de TI e de rede, mantendo os usuários a parte do gerenciamento e manutenção de recursos físicos, cabendo a ele apenas o gerenciamento dos recursos virtuais. Além disso, os usuários alugam somente os recursos que precisam. Mediante o exposto, sabendo das vantagens de utilizar VIs, usuários têm implantado VIs privadas, otimizadas para as suas aplicações para aumentar a qualidade da experiência dos usuários finais (OLIVEIRA; KOSLOVSKI, 2017).

Dentre as tarefas que os InPs precisam realizar para fornecer VIs como um serviço, a mais importante é a alocação de VIs na rede física. Esse processo consiste em basicamente, identificar dentro da infraestrutura física um conjunto de recursos para hospedar componentes da VI provisionada respeitando requisitos de computação (CPU, memória e armazenamento virtual)

e de rede (garantia de largura de banda, topologia específica e Qualidade de Serviço¹ – QoS, do inglês *Quality-of-Service*) (SOUZA *et al.*, 2017b). Em virtude disso, o processo de alocação é desafiador, e deve ser realizado da melhor maneira possível para uma boa gerência da rede física.

Para um gerenciamento efetivo da rede física, o balanceamento da carga e a redução do *overhead* da alocação são essenciais. O balanceamento de carga é fundamental para a Computação em Nuvem e impede a situação em que alguns nós ficam sobrecarregados, enquanto outros estão ociosos. Desse modo, ele pode melhorar as métricas de QoS, incluindo tempo de resposta, custo, taxa de transferência, desempenho e utilização dos recursos (GHOMI; RAHMANI; QADER, 2017). Por outro lado, o *overhead* de alocação é uma sobrecarga às vezes necessária para atender as requisições de VIs (XU; TIAN; BUYYA, 2017). Portanto, reduzir o *overhead* é essencial para diminuir os custos de provisionamento.

No processo de alocação, vários interesses ou objetivos dos InPs e usuários devem ser considerados. Geralmente guiados por suas perspectivas financeiras, os InPs desejam maximizar a sua receita alocando o maior número de VIs usando a menor infraestrutura possível (GUSTAVO; OBELHEIRO; KOSLOVSKI, 2014). Por outro lado, os usuários desejam majoritariamente VIs eficientes e com um baixo custo. Conseqüentemente, devido a multiplicidade de objetivos alguns podem entrar em conflito. Por exemplo, diminuir a fragmentação do substrato beneficia os usuários, pois geralmente diminuem a latência na comunicação entre os recursos internos de uma VI (KOSLOVSKI *et al.*, 2011). Entretanto, agrupar recursos pode dificultar a disponibilidade da VI no caso de falha de algum nó do substrato físico (GUSTAVO; OBELHEIRO; KOSLOVSKI, 2014). Além disso, outras questões relacionadas à qualidade da VI e do Acordo de Nível de Serviço² (SLA, do inglês *Service Level Agreement*) devem ser consideradas. Logo, a análise para se obter a melhor solução para a alocação deve levar em consideração os requisitos e objetivos dos InPs e usuários.

A decisão dos InPs a respeito da alocação pode ser traduzida como uma função objetivo³ e um algoritmo de alocação busca minimizar (por convenção) essa função (RUCK; OLIVEIRA; KOSLOVSKI, 2014). Entretanto, devido aos conflitos entre os requisitos e a necessidade de atender vários objetivos, não só dos InPs, mas também os dos usuários, não é adequado selecionar apenas um objetivo em detrimento de outros, é necessário adotar uma abordagem que possa considerá-los de forma simultânea.

Deste modo, a alocação pode ser tratada como um Problema de Otimização Multi-Objetivo (MOP, do inglês *Multi-Objective Optimization Problem*) que envolve a satisfação simultânea de duas ou mais funções objetivo. Nesses problemas, as funções objetivo que são otimizadas estão em conflito. Isso representa que não há somente uma melhor solução, mas um

¹ Qualidade de Serviço: capacidade de melhorar os serviços trafegados na rede sobre tecnologias de comunicação de redes de dados.

² Acordo de Nível de Serviço: contrato entre duas partes: entre a entidade que pretende fornecer o serviço e o requisitante ou interessado. Esse contrato descreve o serviço, suas metas de nível, papéis e responsabilidades.

³ Função objetivo: neste caso, representa o índice de desempenho de um fator do processo de alocação.

conjunto com as melhores soluções (CARVALHO, 2013). Para obter esse conjunto de soluções pode ser utilizada a Teoria da Otimalidade de Pareto (COELLO *et al.*, 2007).

Esses problemas têm sido resolvidos através de diversos de Algoritmos Evolucionários Multiobjetivo (MOEAs, do inglês *Multi-Objective Evolutionary Algorithms*) (COELLO *et al.*, 2007). MOEAs incorporam um mecanismo de seleção, geralmente baseado nos conceitos de Otimalidade de Pareto, com o objetivo de preservar as melhores soluções encontradas durante o processo de busca enquanto continuam um progresso em direção a soluções melhores para o problema. Além disso, esses algoritmos adotam mecanismos para preservar a diversidade das soluções encontradas, a fim de evitar a convergência para uma única solução (CARVALHO, 2013).

1.1 Problemática

O gerenciamento de recursos é um dos principais desafios de IaaS. Eles podem ser alocados a qualquer momento para lidar efetivamente com as flutuações da carga de trabalho, ao mesmo tempo em que fornecem garantias de QoS aos usuários. Além disso, os recursos de computação e de rede são limitados e precisam ser compartilhados de maneira eficiente entre os usuários de maneira virtual. Para realizar um gerenciamento eficaz de recursos, precisa-se considerar os problemas, como mapeamento de recursos, provisionamento de recursos, alocação de recursos e adaptação de recursos (MANVI; SHYAM, 2014).

Atuando mais especificamente nos problemas de alocação de recursos, um dos principais desafios enfrentados pelos InPs é a alocação de VIs. Alocação que consiste em identificar, na rede física, um conjunto de recursos para hospedar os componentes da IV, respeitando-se todos os requisitos de computação e rede definidos. Com essa última restrição, a complexidade do processo de alocação tende a aumentar, visto que são criados cada vez mais critérios para garantir a qualidade da VI provisionada (SOUZA *et al.*, 2017a).

Outro fator que aumenta a complexidade do problema é o fato de que no processo de solicitação de uma VI, o usuário pode customizá-la de acordo com os requisitos de uma aplicação cujo desempenho pode ser influenciado pela configuração de rede e recursos atribuídos. Uma solução de alocação pode otimizar um simples ou múltiplo critério, tal como balanceamento de carga, decrescimento da carga da rede física, a latência da VI, redução do consumo de energia, entre outros (FISCHER *et al.*, 2013; NESI *et al.*, 2018).

Respeitando os requisitos e objetivos dos InPs e usuários, deve-se ainda considerar que uma VI pode ser alocada em qualquer lugar sobre a rede física. Não por acaso, o problema de alocação de VIs pertence ao conjunto de problemas NP-Difíceis, pois outros problemas que fazem parte desse conjunto podem ser reduzidos a ele (CHOWDHURY; BOUTABA, 2009; FISCHER *et al.*, 2013; SOUZA *et al.*, 2017b).

A Alocação de VIs pode ser vista como um mapeamento de grafos. Os vértices representam os equipamentos de computação (servidores) e de rede (*switches* e roteadores), enquanto as arestas representam os *links* ou canais de comunicação. Desse modo, o problema consiste em mapear o grafo que representa a requisição de VI no grafo que representa a infraestrutura física.

Formalmente, o grafo $G = (H, S, L, C)$ representa a infraestrutura física, onde H representa os servidores físicos, S os *switches* físicos, L os *links* físicos e C o vetor de capacidade residual de cada elemento. De maneira análoga, o grafo $G_v = (H_v, S_v, L_v, C_v)$ compõe uma requisição de IV, no qual H_v representa os nós de computação virtuais, ou VMs, S_v os *switches* virtuais, L_v os *links* virtuais e C_v o vetor de capacidades solicitadas pela requisição de VI para cada um de seus elementos. Continuando, o mapeamento das VIs em recursos físicos é dado por: $M : G_v \rightarrow (M_h, M_s, M_p)$ onde $M_h n^i \in H_v \rightarrow n \in H$ é o mapeamento das VMs para os servidores físicos, $M_s m^i \in S_v \rightarrow m \in S$ é o mapeamento dos *switches* virtuais nos *switches* físicos, sendo P o conjunto de todos os caminhos físicos da infraestrutura física, $M_p e^i \in L_v \rightarrow p \in P$ é o mapeamento de *links* virtuais sobre um caminho físico P .

Cada G_v é alocado considerando a capacidade residual do grafo G , ou seja, cada requisição de VI é alocada considerando os recursos disponíveis na infraestrutura física. Vale ressaltar que a capacidade residual de um recurso físico geralmente é maior que a capacidade requisitada para ser usada por um recurso virtual. Deste modo, um mapeamento é considerado uma solução válida se para todas as VMs, *switches* e *links* virtuais há recurso físico suficiente para provisioná-los obedecendo a configuração solicitada, ou em outras palavras, $\forall n^i \in H_v, C_v(n^i) \leq C(M_c(n^i))$, $\forall m^i \in S_v, C_v(m^i) \leq C(M_s(m^i))$ e $\forall e^i \in L_v, C_v(e^i) \leq \min(C(l) \forall l \in P)$. Ademais, todos os nós virtuais de computação e de rede que fazem parte da requisição x de VI alocados devem ser interconectados pelo caminho virtual l .

Em conclusão, diante do que foi apresentado, surgiu o seguinte problema de pesquisa: como desenvolver um mecanismo que faça o procedimento de alocação de IVs com eficiência, respeitando as restrições de computação e rede e que atenda os objetivos como, balanceamento de carga e redução do *overhead* de alocação?

1.2 Hipótese

Um mecanismo de alocação que utiliza um MOEA pode alocar VIs em uma rede física que possui topologia utilizada em *Data Centers*, respeitando os requisitos de computação (CPU, memória e armazenamento virtual) e rede (garantia de largura de banda), atendendo objetivos de interesse como balanceamento de carga e redução do *overhead* de alocação. O mecanismo deve considerar medidas de qualidade, como taxa de aceitação, tempo médio de alocação, lucro do provedor e carga dos nós físicos e *links*.

1.3 Objetivo

O objetivo principal do presente trabalho é desenvolver um Algoritmo Evolucionário Multiobjetivo (MOEA) para o processo de alocação de VIs em uma rede física que possua topologia utilizada em *Data Centers*, atendendo os requisitos de computação e rede, avaliando soluções que atendam objetivos como, balanceamento de carga e baixo *overhead* de alocação.

Para alcançar o objetivo anterior, alguns objetivos específicos precisam ser atendidos, a saber:

- Desenvolver um simulador para alocação de VIs;
- Simular redes físicas com topologias utilizadas em *Data Centers*;
- Desenvolver um modelo de alocação que utilize um MOEA para alocar VIs.
- Implementar um MOEA baseado no Algoritmo Genético (GA, do inglês *Genetic Algorithm*) para alocação de VIs;

1.4 Trabalhos Relacionados

O problema de alocação de VIs é derivado do problema de Incorporação de Redes Virtuais (VNE, do inglês *Virtual Network Embedding*) (FISCHER *et al.*, 2013; OLIVEIRA; KOSLOVSKI, 2017). Portanto, na etapa de revisão bibliográfica foi considerado o problema de alocação de VIs e o de VNE. Por meio de um mapeamento sistemático foi possível identificar que a maioria dos estudos solucionam o problema de VNE (veja o capítulo 2). Nesse tipo de alocação apenas roteadores e canais de comunicação são considerados, ignorando a existência de VMs, assim sendo, o número de recursos participantes da formulação do problema é limitado.

Geralmente, soluções para o mapeamento são realizadas em duas etapas: mapeamento dos nós e mapeamento dos canais de comunicação. Nesse sentido, Chowdhury, Rahman e Boutaba (2012) argumentaram que coordenar as fases de mapeamento de nó e *link* durante a incorporação das Redes virtuais aumenta significativamente o espaço de busca da solução, e conseqüentemente, melhora a qualidade da heurística de incorporação. Deste modo, foram propostos algoritmos para solucionar o problema de VNE utilizando técnicas de arredondamento. Extensões desses algoritmos focam no balanceamento de carga e na taxa de aceitação de requisições. Os experimentos foram realizados utilizando as métricas: taxa de aceitação, lucro gerado, custo do provisionamento, utilização média dos nós e *links*. Ademais, a topologia utilizada é gerada aleatoriamente. Segundo os resultados apresentados, os algoritmos propostos superam seus equivalentes em termos de taxa de aceitação, receita e custo de provisionamento.

As soluções para VNE e alocação de VIs necessitam atender um conjunto de objetivos dos InPs e usuários. Houidi, Louati e Zeghlache (2015) discutem sobre alguns desses objeti-

vos e afirmam que a maioria dos trabalhos anteriores sobre VNE considerou esses objetivos separadamente. Além disso, propostas de otimização multiobjetivo priorizaram ou enfatizaram algumas restrições para reduzir a complexidade do problema. Por outro lado, foi desenvolvido um algoritmo multiobjetivo para solucionar o problema de VNE levando em conta restrições como, consumo de energia, disponibilidade e balanceamento de carga. Para avaliar a solução proposta foi utilizada uma topologia aleatória e as seguintes métricas: porcentagem de requisições atendidas, porcentagem de nós ativos, tempo de resposta, custo das requisições atendidas e tempo médio de atraso. Os resultados apresentados demonstram a viabilidade da solução. Ademais, eles podem ser referência para futuros algoritmos heurísticos de VNE em ambientes de nuvem.

Finalmente, o último trabalho relacionado que se dedicou a resolver o problema de VNE foi o de [Pathak e Vidyarthi \(2017\)](#). Em seu trabalho foi aplicada a meta-heurística GA para encontrar soluções com uma alocação eficiente de redes virtuais em redes físicas aumentando o lucro de múltiplos InPs. É discutido o problema de VNE e um modelo para encontrar soluções eficientes é proposto. A análise experimental foi realizada utilizando múltiplos cenários de InPs e topologias geradas aleatoriamente. Além disso, as seguintes métricas foram utilizadas no processo de avaliação de desempenho da solução: taxa de aceitação, média do lucro, utilização dos nós e *links*. Os resultados apresentados exibem um desempenho consistente da solução em comparação com outros estudos anteriores.

O primeiro trabalho relacionado que investiu no problema de alocação de VIs foi o de [Koslovski et al. \(2011\)](#). Em seu trabalho considerou-se a localização geográfica na alocação de VIs. Foi desenvolvido uma formulação do problema de alocação e uma heurística de alocação, ambas guiadas pela localização geográfica dos componentes físicos e virtuais das VIs. O método de experimentação utilizado foi a simulação, a infraestrutura física e as requisições de VIs foram geradas utilizando uma ferramenta de geração de topologias aleatórias. Também foram consideradas métricas que refletem as perspectivas dos usuários e InPs, fragmentação do substrato físico e qualidade da alocação respectivamente. Resultados experimentais demonstraram que é possível diminuir a fragmentação e o custo do substrato. Além disso, vale destacar que os resultados evidenciam uma melhora na qualidade da alocação (perspectiva do usuário) para diferentes tamanhos de infraestruturas virtuais alocadas em um tamanho médio do substrato físico.

O estudo de [Gustavo, Obelheiro e Koslovski \(2014\)](#) foi primeiro a considerar simultaneamente a fragmentação e a sobrevivência de VIs. Foi desenvolvido uma formulação utilizando programação inteira mista que incorporasse as perspectivas dos InPs (reduzir a fragmentação) e usuários (melhorar a sobrevivência). Deste modo, desenvolveu-se um modelo de alocação ótimo para VIs com capacidade de sobrevivência. No modelo foram introduzidas métricas de falha local e global, que fornecem uma medida de sobrevivência da perspectiva dos usuários e InPs, respectivamente. Nos experimentos, foi empregada a técnica de simulação, sendo utilizada uma topologia comum em *Data Centers*, e diferentes tamanhos de substratos físicos e requisições de

VIs. Os resultados experimentais demonstraram que o modelo é efetivo em otimizar simultaneamente a fragmentação (minimizando o uso do substrato e baixando os custos do provedor) e a sobrevivência das VIs, possibilitando, assim, o atendimento de um conjunto maior de SLAs e um maior lucro ao InPs.

Yu, Xue e Zhang (2015) demonstraram algumas diferenças entre as classes de problema de VNE e alocação de VIs, denominada de Incorporação de Infraestruturas Virtuais (VIE, do inglês *Virtual Infrastructure Embedding*). Em seguida, argumentaram que a formulação em Programação Linear Inteira Mista (MIP, do inglês *Mixed Integer Programming*) proposta por Chowdhury, Rahman e Boutaba (2012) não é adequada para o problema de VIE, e a aplicação da mesma técnica de arredondamento levaria a um desempenho ruim, devido aos possíveis conflitos de alocação de recursos. Por isso, segundo os autores, em seu trabalho, foi proposta uma nova formulação do MIP para modelar o problema da VIE. Com base na formulação também foi proposta uma nova técnica de otimização denominada arredondamento sequencial, que adota um processo de avaliação iterativo para minimizar os conflitos de mapeamentos entre os componentes virtuais. Os experimentos foram realizados utilizando a técnica de simulação, a infraestrutura física e as requisições de infraestruturas virtuais foram representados como grafos aleatórios. Além disso, as métricas utilizadas na solução proposta foram: taxa de aceitação e custo médio de incorporação. Os experimentos apresentados mostram que o algoritmo proposto melhora o número e a qualidade de soluções viáveis em comparação aos algoritmos D-ViNE e R-ViNE (CHOWDHURY; RAHMAN; BOUTABA, 2012), tanto na taxa de aceitação quanto nos custos de incorporação.

Os serviços de infraestrutura em nuvem oferecem conjuntamente recursos de rede e computação, ou seja, VIs. Soares, Aparício e Sargento (2017) propõem uma forma de resolver o problema de alocação de VIs por meio de uma formulação de programação linear inteira, que apresenta diferentes estratégias de incorporação: balanceamento de carga, combinando o balanceamento de carga de recursos físicos com a redução do consumo de largura de banda; infraestrutura verde, redução do consumo de energia da infraestrutura; dinâmica, combinação entre balanceamento de carga e estratégias de economia de energia dependendo do estado do substrato. Além disso, para possibilitar otimização em tempo real, é proposta uma abordagem heurística. Uma análise comparativa do desempenho foi realizada entre as diferentes estratégias considerando dois diferentes aspectos: taxa de aceitação e consumo de energia. Na análise foi empregada a técnica de simulação para avaliar o desempenho da solução proposta. Deste modo, foram utilizadas topologias geradas aleatoriamente para a infraestrutura física e virtual. Segundo os resultados apresentados, há uma evidência clara de *trade-off* entre melhorar a taxa de aceitação e reduzir o uso dos recursos da infraestrutura física. Ademais, a heurística alcança resultados perto do ótimo em termos de aceitação de VI, enquanto reduz o consumo de energia da infraestrutura física.

Novos paradigmas como as Redes Definidas por Software (SDN, do inglês *Software-*

Defined Networking) podem simplificar o provisionamento de VIs. [Souza et al. \(2017b\)](#) propuseram uma abordagem de alocação de VIs considerando os principais desafios de gerenciamento introduzidos por SDN. Deste modo, a alocação de IVs foi formulada utilizando a técnica de MIP. Em seguida, as restrições do MIP são relaxadas para obter um Programa Linear (LP, do inglês *Linear Program*) e técnicas de arredondamento são aplicadas para uma proposta de solução aceitável. Assim, é proposta uma heurística que inova por selecionar candidatos para hospedar IVs baseando-se nas particularidades do SDN, além das restrições das VMs. Os experimentos foram realizados utilizando a topologia *fat-tree*, comum em *Data Centers* de nuvem. Além disso, foram utilizadas as métricas: lucro do provedor, fragmentação do *Data Center*, tempo Médio pra alocação de uma requisição, taxa de aceitação e latência média de uma VI, possibilitando, assim, avaliar o desempenho do mecanismo pela perspectiva do provedor e dos inquilinos. Os resultados apresentados indicam que em *Data Centers* baseados em SDN é possível realizar a alocação de VIs com garantias de QoS sem afetar as métricas do provedor, além de reduzir a latência percebida pelos usuários.

Ainda no tema de alocação conjunta de redes e máquinas virtuais. O trabalho de [Oliveira e Koslovski \(2017\)](#) propôs um algoritmo para realizar a alocação de VIs. Modelando o problema como um mapeamento de grafos, a abordagem desenvolvida parte da premissa que ambos os grafos, o da infraestrutura física e o da requisição de VI são grafos do tipo árvore. A abordagem de alocação é dividida em três etapas. Na primeira é realizada uma conversão dos grafos das VIs e da rede física em árvores. Na segunda é executado um procedimento para agrupar as informações das subárvores, reduzindo assim o espaço de busca. E finalmente, no terceiro é executado o algoritmo para a alocação das VIs. Na avaliação experimental foi utilizada a técnica de simulação, o algoritmo foi implementado em C++ e as métricas utilizadas para avaliar o desempenho da solução foram: fragmentação do *Data Center*, lucro do provedor, tempo médio para alocação de uma VI, taxa de aceitação e carga dos nós físicos e *links*. Além disso, foi utilizada a topologia baseada no modelo de 3 camadas da Cisco para representar a infraestrutura física e topologias hierárquicas para representar as requisições de IVs. Os resultados indicam que o algoritmo proposto mantém uma baixa fragmentação e carga no *Data Center* combinado com uma alta taxa de aceitação.

Semelhante ao trabalho de [Koslovski et al. \(2011\)](#) e [Gustavo, Obelheiro e Koslovski \(2014\)](#), [Raugust et al. \(2018\)](#) apresentam uma alternativa para melhorar a confiabilidade e a capacidade de sobrevivência de VIs baseada em réplicas de componentes críticos. Para isso, a alocação de VIs foi formulada como um problema de Programação Linear Mista. Por conseguinte, as restrições da formulação são relaxadas obtendo, assim, um modelo em Programa linear. Deste modo, o mecanismo de alocação proposto é composto do modelo de Programação Linear e o Algoritmo de Recozimento Simulado (SA, do inglês *Simulated Annealing*). O modelo de programação linear é usado como entrada para o SA, compondo, assim, um mecanismo com requisitos de confiabilidade e capacidade de sobrevivência. Na avaliação da abordagem foi utilizada a técnica de simulação, foram selecionados provedores públicos de nuvem e foi coletada

a probabilidade de falha em cada zona. Além disso, foi utilizado um conjunto de métricas que representam as perspectivas dos usuários. Os resultados mostraram que a solução é efetiva em termos de confiabilidade e sobrevivência, sem inflar o preço do provisionamento.

Considerando os trabalhos relacionados apresentados, algumas observações quanto às limitações das soluções devem ser apontadas. A primeira, segundo Luizelli *et al.* (2013), Oliveira e Koslovski (2017) é que a busca por soluções ótimas tem aplicabilidade limitada em cenários reais. Nesse tipo de cenário os usuários buscam respostas imediatas para as suas requisições, o que torna a aplicação de soluções ótimas impraticável (OLIVEIRA; KOSLOVSKI, 2017). A segunda, é a topologia da rede física na qual o algoritmo vai tomar as decisões de alocação. Oliveira e Koslovski (2017), afirmam que propostas que não utilizam topologias que representam cenários reais podem influenciar na taxa de aceitação das requisições. Finalmente, a terceira está relacionada ao custo do pré-processamento necessário para o algoritmo entrar em ação. As propostas de Oliveira e Koslovski (2017) e Raugust *et al.* (2018) precisam de vários passos de pré-processamento, aumentando o tempo para a obtenção das soluções.

Na Tabela 1, tem-se todos os trabalhos relacionados. Para cada estudo é apresentado a referência, o tipo de alocação (se o trabalho se dedicou a resolver o problema de VNE ou alocação de VIs), a estratégia de otimização utilizada, a topologia empregada nos experimentos e a principal contribuição de cada trabalhos. Desse modo, a Tabela 1, fornece uma visão geral das principais características dos trabalhos relacionados. Ademais, no último item também é apresentado o presente trabalho.

Considerando os avanços realizados por outros trabalhos, esse trabalho visa contribuir com um MOEA para alocação de VIs baseado no GA. Além disso, superar algumas limitações de trabalhos anteriores como a utilização de topologias aleatórias na avaliação da solução e a necessidade de um pré-processamento para a execução do algoritmo de alocação.

1.5 Organização do Trabalho

Assim, essa proposta está dividida em demais capítulos de tal modo que: o Capítulo 2 apresenta o mapeamento sistemático, método utilizado para realizar a revisão bibliográfica do presente trabalho; enquanto o Capítulo 3 descreve o modelo de alocação e o MOEA proposto para o problema de alocação de VIs. O Capítulo 4 exhibe o método empregado para a avaliação da solução e discute os resultados obtidos. Finalmente, o Capítulo 5 conclui o trabalho dissertando sobre as considerações finais, principais contribuições e trabalhos futuros.

Tabela 1 – Trabalhos relacionados e suas principais características

Referência	Alocação	Otimização	Topologia	Principal Contribuição
(CHOWDHURY; RAHMAN; BOUTABA, 2012)	VNE	Heurística	Aleatória	Alocação conjunta de elementos de computação e comunicação.
(HOUIDI; LOUATI; ZEGHLACHE, 2015)	VNE	Ótima	Aleatória	Algoritmo para VNE multiobjetivo.
(PATHAK; VIDYARTHI, 2017)	VNE	Meta-Heurística	Aleatória	Um modelo de alocação que utiliza o GA para resolver o problema de VNE.
(KOSLOVSKI <i>et al.</i> , 2011)	VI	Heurística	Aleatória	Heurística de alocação guiada pela localização geográfica dos componentes físicos e virtuais das VIs.
(GUSTAVO; OBELHEIRO; KOSLOVSKI, 2014)	VI	Ótima	<i>Fat-Tree</i>	Modelo de alocação ótimo que considera a fragmentação do substrato físico da rede e a sobrevivência das VIs.
(YU; XUE; ZHANG, 2015)	VI	Heurística	Aleatória	Técnica que minimiza os conflitos de mapeamento ao reavaliar iterativamente o programa com base em decisões anteriores.
(SOARES; APARÍCIO; SARGENTO, 2017)	VI	Ótima e Heurística	Aleatória	Heurística que tem uma estratégia dinâmica de incorporação de VIs combinando balanceamento de carga e estratégias de economia de energia.
(SOUZA <i>et al.</i> , 2017b)	VI	Ótima	<i>Fat-Tree</i>	Mecanismo de alocação que aloca VIs em <i>Data Centers</i> implementados com SDN respeitando requisitos de QoS.
(OLIVEIRA; KOSLOVSKI, 2017)	VI	Heurística	<i>Fat-Tree</i>	Um algoritmo para alocação de VIs online baseado em árvores.
(RAUGUST <i>et al.</i> , 2018)	VI	Heurística	<i>Full-mesh</i>	Um mecanismo de alocação que considera requisitos de sobrevivência e a confiabilidade de VIs.
Esse Estudo	VI	Meta-Heurística	<i>Fat-Tree</i>	MOEA baseado no GA para alocação de VIs.

Fonte: elaboração própria

2

Alocação de Infraestruturas e Redes Virtuais em Computação em Nuvem: Um Mapeamento Sistemático

Neste capítulo é apresentado o mapeamento sistemático sobre os estudos que se dedicaram a resolver o problema de alocação de VIs e o problema de VNE e as suas principais características.

2.1 Introdução

A computação em nuvem possibilitou aos usuários o acesso a um conjunto compartilhado de recursos computacionais configuráveis (e.g. redes, processamento, armazenamento, aplicações e etc) de maneira dinâmica, sob demanda em uma base de pagamento pelo uso (MELL; GRANCE *et al.*, 2011). Esses recursos são oferecidos por meio de modelos de serviço. O presente trabalho aborda temas relacionados ao modelo de serviço IaaS.

InPs de IaaS têm oferecido Redes Virtuais e VIs como um serviço para seus usuários. Uma rede virtual é uma combinação de elementos de rede ativos e passivos (nós de rede e *links* de rede) sobre uma rede física (FISCHER *et al.*, 2013). Já uma VI é uma infraestrutura totalmente virtualizada que parece ser física, mas na realidade compartilha o substrato subjacente com outras VIs durante um determinado período de tempo (OLIVEIRA; KOSLOVSKI, 2017).

Para oferecer Redes Virtuais e VIs os InPs têm de lidar com os problemas relacionados a alocação dos recursos necessários para provisionar esses serviços. Formalmente, o problema de alocar VIs sobre infraestruturas físicas pode ser descrito como uma extensão do problema VNE (OLIVEIRA; KOSLOVSKI, 2017). Assim sendo, o processo de alocação de VIs ou Redes Virtuais é basicamente identificar dentro do *Data Center* um conjunto de recursos para hospedar componentes da VI provisionada respeitando requisitos de computação (CPU, memória e armazenamento virtual) e de rede.

Uma solução para um problema de VNE pode ser adaptada para solucionar problemas de alocação de VIs. O problema de VNE considera em seu processo de alocação computadores, roteadores e canais de comunicação. De forma similar o problema de alocação de VIs considera a alocação de recursos relacionados a servidores, somando esse aos considerados pelo problema de VNE no processo de alocação.

Esse processo de alocação é um desafio para os InPs. O problema de alocação de VIs e o problema de VNE são classificados como NP-Difícil. Ademais, o processo de alocação não pode ser realizado de maneira aleatória pois há requisitos que devem ser respeitados e objetivos a serem alcançados. Logo, a complexidade desses tipos de problemas deve ser levada em consideração no desenvolvimento de novas soluções.

Portanto, é necessário considerar a proximidade dos problemas e a necessidade dos InPs de IaaS de soluções para identificar de modo eficiente recursos físicos para hospedar VIs e redes virtuais. Diante disso, esse mapeamento sistemático tem como objetivo identificar e mapear os estudos que contemplam soluções para o problema de alocação de VIs e o de VNE no ambiente de Computação em Nuvem. Além disso, demonstrar as principais características dos estudos, facilitando, assim, o desenvolvimento de novas soluções e a comparação dessas com alguma já desenvolvida.

Assim, esse mapeamento está dividido da seguinte forma: a seção 2.1 refere-se a esta introdução; a seção 2.2.4 apresenta o método adotado nesse mapeamento; a seção 2.3 apresenta a análise dos resultados; e finalmente, a seção 2.4 apresenta as conclusões.

2.2 Método

Com o propósito de estudar e mapear o estado da arte a cerca dos trabalhos que se dedicaram a resolver o problema de alocação de VIs e ao problema de VNE e as suas principais características, foi adotado o método de mapeamento sistemático, que de acordo com a abordagem de Petersen *et al.* (2008) e Petersen, Vakkalanka e Kuzniarz (2015), consiste em um protocolo sistemático para a busca e seleção de estudos relevantes com o objetivo de extrair dados e mapear os resultados para um problema de pesquisa específico.

Desse modo, para seguir o método, nesta seção, é descrito como o processo de busca e seleção dos estudos primários (publicações encontradas nas bases de dados da área de computação aprovadas pelo processo de revisão por pares) foi realizado. Para isso, foi necessário definir as questões de pesquisa, estratégias de busca e definições dos critérios de seleção. Esse procedimento pode ser resumido nas três etapas abaixo:

- Etapa 1: busca nas bases utilizando a *string* de busca e ferramentas de busca avançada disponíveis.
- Etapa 2: filtragem dos artigos encontrados na etapa 1 utilizando os critérios de seleção.

- Etapa 3: leitura completa dos artigos e triagem dos estudos primários.

O procedimento de busca nas bases (etapa 1) foi realizado dia 11 de Setembro de 2018. Logo, trabalhos publicados após essa data não são considerados nesse mapeamento. Dessa forma, o procedimento de busca e seleção dos estudos primário é descrito com detalhes nas próximas seções desse mapeamento.

2.2.1 Questões de Pesquisa

Para alcançar o objetivo proposto por esse mapeamento foram elaboradas as questões de pesquisa abaixo:

- QP1: Em quais anos há publicações sobre alocação de Infraestruturas e Redes Virtuais?
- QP2: Qual a base que mais contribuiu com estudos primários após os procedimentos de seleção?
- QP3: Como estão divididos quantitativamente os artigos que solucionam a alocação de Redes e Infraestruturas Virtuais?
- QP4: Quais as principais métricas utilizadas para avaliar o desempenho das abordagens propostas para resolver os problemas de VNE e alocação de VIs?
- QP5: Qual o tipo de otimização, técnica de avaliação e topologia e principal contribuição dos trabalhos que se dedicaram a resolver o problema de VNE e alocação de VIs?
- QP6: Dentre os artigos que utilizaram a técnica de simulação, algum propôs um simulador para auxiliar na avaliação de propostas para resolver o problema de VNE e alocação de VIs?

As respostas dessas questões são essenciais para se obter uma visão geral sobre o tema proposto e demonstrar tendências e lacunas existentes. Além disso, traz dados importantes para a elaboração de novos trabalhos, como por exemplo, métricas utilizadas, estratégias para analisar soluções e as principais abordagens já desenvolvidas na área.

2.2.2 Estratégia de Busca e Seleção

A finalidade da estratégia de busca e seleção é selecionar os estudos primários (estudos completos disponíveis em bases de pesquisa e aprovados pela revisão por pares) relacionados com o tema do mapeamento. No presente trabalho a estratégia de busca e seleção foi dividida nas três etapas abaixo:

Para a execução da busca, foram selecionadas as bases de dados da área de computação: *IEEE Xplore* (IEEE), *ACM Digital Library* (ACM), *Science Direct* (SD) *Wiley Digital Library*

(WDL) e a *Springer Link* (SL). Para uso sem restrições de download, foi utilizado o portal de periódicos da CAPES. Nas bases selecionadas foram aplicadas ferramentas de filtragem (busca avançada), visando reduzir o número de trabalhos fora do escopo.

Na prospecção dos estudos primários foi empregada a seguinte *string* de busca em todas as bases: **((("virtual infrastructures") AND allocation) AND "cloud computing")**. Com a aplicação dessa *string* foi possível encontrar um total de 470 artigos, conforme a Tabela 2, em quem é possível observar a quantidade de artigos prospectados em cada base.

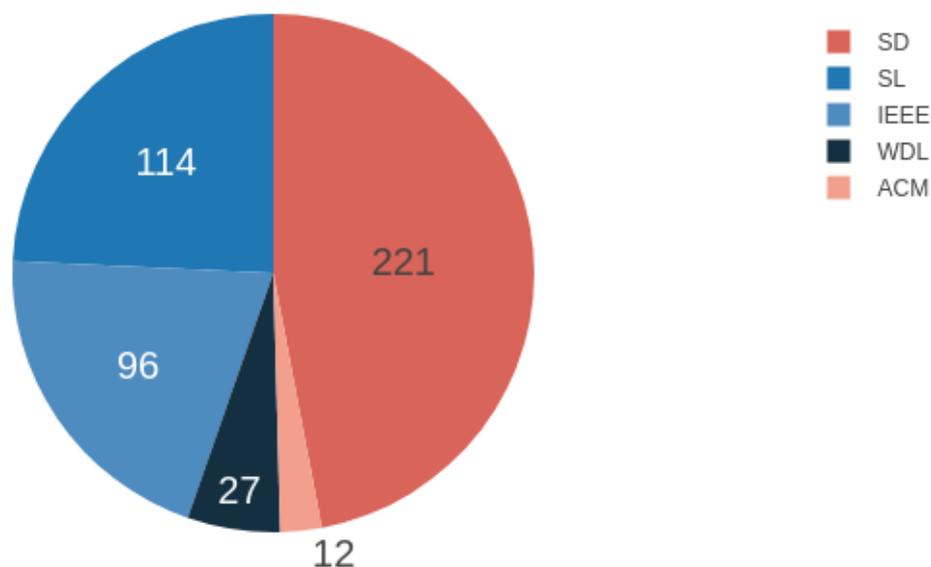
Tabela 2 – Quantidade de artigos encontrados após a aplicação da *string* de busca nas bases de pesquisa.

Base	Número de artigos
ACM	12
IEEE	96
SD	221
SL	114
WDL	27

Fonte: elaboração própria

Na Figura 1, tem-se exibido o número de artigos encontrados nas bases com a aplicação da *string* de busca. A Base SD foi a que mais retornou trabalhos, 221 (47%); seguida pela SL com 114 (24,3%); IEEE com 96 (20,4%); WDL com 27 (5,7%) e a ACM com 12 (2,6%).

Figura 1 – Número de estudos primários encontrados em cada base utilizando a *string* de busca.



Fonte: elaboração própria

Ao fim da busca deu-se início ao processo de filtragem dos artigos por meio dos critérios de seleção.

2.2.3 Critérios de Seleção

Com o propósito de selecionar artigos relevantes para o objetivo desse mapeamento sistemático foram definidos critérios para a inclusão e exclusão. O estudo contou com os seguintes critérios de inclusão:

1. Estudos primários com o foco em alocação de Infraestruturas Virtuais no Ambiente de Computação em Nuvem;
2. Estudos primários focados na alocação de Redes Virtuais no Ambiente de Computação em Nuvem.

A confirmação dos critérios de inclusão foi dada a partir da análise do resumo de cada um dos artigos encontrados. Em paralelo a verificação dos critérios de inclusão, foi realizada a análise dos artigos quanto aos critérios de exclusão explicitados abaixo:

1. Artigos duplicados;
2. Artigos que não apresentam o seu conteúdo na íntegra;
3. Artigos que não apresentam foco na alocação de VIs e VNE em Computação em Nuvem.

Encerrada a filtragem dos artigos por meio aplicação dos critérios de inclusão e exclusão, dos 470 artigos encontrados aplicando a *string* de busca, foram selecionados 47 para compor os estudos primários. Cabe apontar que foram encontrados 2 artigos duplicados.

Na Tabela 3 são apresentados os resultados das buscas nas bases de dados de computação utilizando a *string* de busca e após a aplicação dos critérios de inclusão. As bases SD e SL foram as que mais tiveram artigos reprovados pelos critérios de seleção.

Tabela 3 – Quantidade de artigos encontrada após a filtragem por meio dos critérios de seleção.

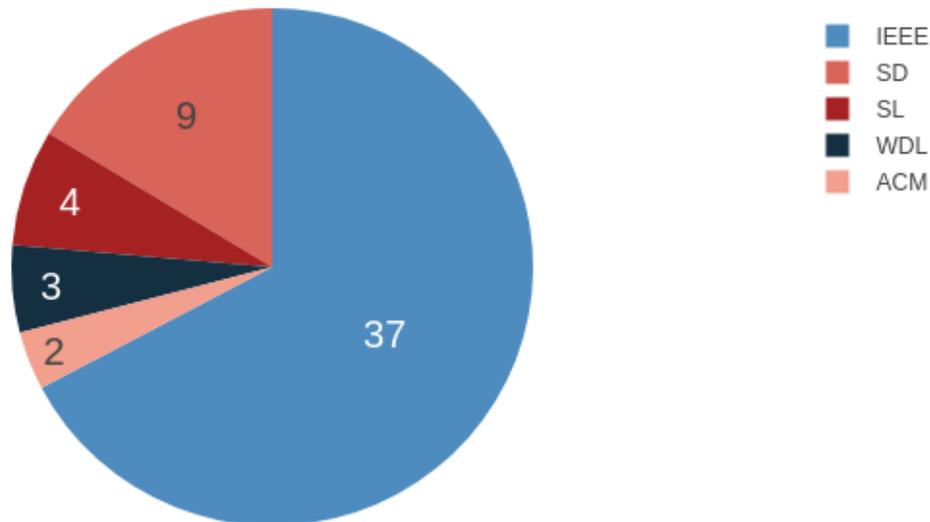
Base	Resultado das buscas	Após aplicação dos critérios de seleção
ACM	12	2
IEEE	96	37
SD	221	9
SL	114	4
WDL	27	3

Fonte: elaboração própria

Na Figura 2, tem-se a participação de cada base no conjunto dos trabalhos selecionados após a filtragem. Chama a atenção a quantidade de artigos aprovados pela filtragem da IEEE comparando com as outras bases. Após a aplicação dos critérios de seleção, a base IEEE teve 37

(67,3%) trabalhos aprovados para a próxima etapa, SD 9 artigos (16,4 %), SL 4 artigos (7,3%), WDL 3 artigos (5,7%) e ACM 2 (2,6%) artigos.

Figura 2 – Número de artigos selecionados em cada base após a filtragem por meio dos critérios de seleção.



Fonte: elaboração própria

Após a filtragem por meio dos critérios de seleção os artigos são encaminhados para a leitura completa.

2.2.4 Seleção por meio da leitura completa

Na etapa de aplicação dos critérios de seleção foram verificados apenas os resumos dos artigos para realizar o processo de filtragem. Na presente etapa foi realizada a leitura completa para se obter uma maior precisão na seleção dos estudos primários que vão ser utilizados para responder as questões de pesquisa. Além disso, dois artigos foram adicionados na etapa final de leitura pois são trabalhos relevantes para a classe de problemas VNE.

Na leitura completa foi possível analisar com mais profundidade se os trabalhos atendiam os critérios de seleção definidos. Além disso, foi possível estudar se as questões de pesquisa eram adequadas para o tema proposto. Na Tabela 4, tem-se a quantidade de artigos das bases aprovados e classificados como estudos primários.

Tabela 4 – Quantidade de artigos aprovados de cada base após a leitura completa.

Base	Número de artigos
ACM	1
IEEE	27
SD	1
SL	0
WDL	2

Fonte: elaboração própria

Por meio do processo de busca e seleção, e em seguida do de leitura, foi possível obter a participação de cada base na totalidade dos trabalhos selecionados. A base IEEE obteve grande maioria com 27 de 31 (87,1%) dos estudos primários, a base WDL veio em seguida com 2 de 31 (6,5%), ACM e SD obtiveram 1 de 31 (3,2%) cada e a base SL não teve nenhum trabalho selecionado.

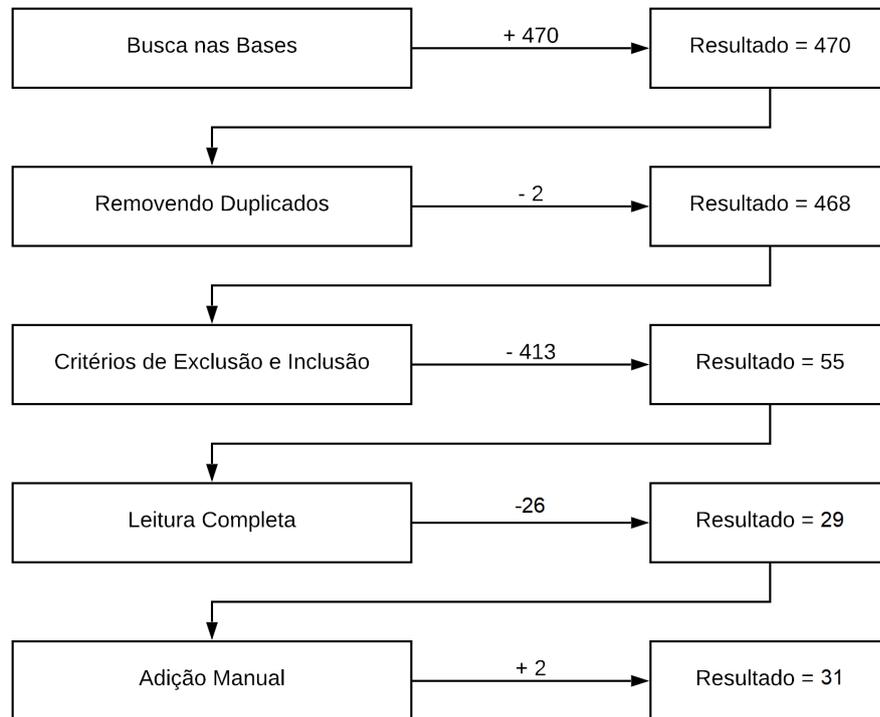
Finalmente, de posse dos trabalhos selecionados (estudos primários) foi possível responder as questões de pesquisa definidas na seção 2.2.1 desse mapeamento.

2.2.5 Considerações finais sobre o método

O método desse mapeamento foi elaborado com base em trabalhos da literatura que tinham como objetivo realizar o mapeamento sistemático de trabalhos de seus temas. Ele pode ser dividido nas etapas: elaboração das questões de pesquisa, estratégias de busca e seleção e a aplicação dos critérios de seleção em conjunto com a leitura completa.

A cada etapa do processo de busca e seleção e aplicação dos critérios de seleção vários artigos foram desclassificados. O objetivo era eliminar trabalhos que estavam fora do escopo desse mapeamento. Na Figura 3 tem-se o resumo de todos os processos realizados para selecionar os estudos primários.

Figura 3 – Diagrama com a quantidade de artigos removida ou adicionada em cada fase do mapeamento.



Fonte: elaboração própria

Assim, pode-se concluir que após todo esse processo de filtragem obteve-se um conjunto relevante de estudos primários dentro do escopo proposto. Tornando assim possível com estes responder as questões de pesquisa definidas na seção 2.2.1 desse mapeamento.

2.3 Análise dos Resultados

Nesta seção, são apresentados os resultados da análise dos estudos primários, de modo que responda as questões de pesquisa delineadas na seção 2.2.1 desse mapeamento. Os estudos primários selecionados podem ser vistos na Tabela 5.

Tabela 5 – Estudos primários selecionados.

Título	Referência	Base
<i>Optimal resource allocation for survivable virtual infrastructures</i>	(GUSTAVO; OBELHEIRO; KOSLOVSKI, 2014)	IEEE
<i>Dynamic strategies for the optimal embedding of Virtual Infrastructures</i>	(SOARES; APARÍCIO; SARGENTO, 2017)	IEEE
<i>GPU-Accelerated Algorithms for Allocating Virtual Infrastructure in Cloud Data Centers</i>	(NESI <i>et al.</i> , 2018)	IEEE

Continua na próxima página

Tabela 5 – Estudos primários selecionados.

Título	Referência	Base
<i>Locating Virtual Infrastructures: Users and InP perspectives</i>	(KOSLOVSKI <i>et al.</i> , 2011)	IEEE
<i>Towards Min-Cost Virtual Infrastructure Embedding</i>	(YU; XUE; ZHANG, 2015)	IEEE
<i>Allocation of Virtual Infrastructures on Multiple IaaS Providers with Survivability and Reliability Requirements</i>	(RAUGUST <i>et al.</i> , 2018)	IEEE
<i>QoS-Aware Virtual Infrastructures Allocation on SDN-Based Clouds</i>	(SOUZA <i>et al.</i> , 2017b)	IEEE
<i>A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements</i>	(OLIVEIRA; KOSLOVSKI, 2017)	WDL
<i>A Cloud-Oriented Algorithm for Virtual Network Embedding over Multi-Domain</i>	(LI; SAIDI; CHEN, 2016)	IEEE
<i>VNE-Greedy: Virtual Network Embedding Algorithm Based on OpenStack Cloud Computing Platform</i>	(KARMOSHI <i>et al.</i> , 2016)	IEEE
<i>A Markov Reward Model Based Greedy Heuristic for the Virtual Network Embedding Problem</i>	(BIANCHI; PRESTI, 2016)	IEEE
<i>A batch approach for a survivable virtual network embedding based on Monte-Carlo Tree Search</i>	(SOUALAH <i>et al.</i> , 2015)	IEEE
<i>A reliable Network embedding algorithm based on game theory within cloud's backbone</i>	(SOUALAH <i>et al.</i> , 2014)	IEEE
<i>Distributed Virtual Network Embedding with Bloom Filter</i>	(QING; FENG; JIN, 2015)	IEEE
<i>Exact Multi-Objective Virtual Network Embedding in Cloud Environments</i>	(HOUIDI; LOUATI; ZEGH-LACHE, 2015)	IEEE
<i>Ant colony optimization based energy efficient virtual network embedding</i>	(GUAN <i>et al.</i> , 2015)	IEEE
<i>Exact Adaptive Virtual Network Embedding in Cloud Environments</i>	(HOUIDI; ZEGHLACHE, 2013)	IEEE
<i>Breaking the atomicity of virtual network embedding</i>	(WEN; QIAN; LU, 2013)	IEEE
<i>Virtual network embedding through topology-aware node ranking</i>	(CHENG <i>et al.</i> , 2011)	ACM
<i>Virtual network embedding through topology awareness and optimization</i>	(CHENG <i>et al.</i> , 2012)	SD
<i>Link mapping-oriented ant colony system for virtual network embedding</i>	(ZHENG <i>et al.</i> , 2017)	IEEE
<i>A novel virtual network embedding scheme based on Gomory-Hu tree within cloud's backbone</i>	(SOUALAH <i>et al.</i> , 2016)	IEEE
<i>Availability-Aware Virtual Network Embedding for Multi-tier Applications in Cloud Networks</i>	(SHEN <i>et al.</i> , 2015)	IEEE
<i>Sharing Based Virtual Network Embedding Algorithm With Dynamic Resource Block Generation</i>	(MAO <i>et al.</i> , 2015)	IEEE
<i>PR-VNE: Preventive reliable virtual network embedding algorithm in cloud's network</i>	(SOUALAH <i>et al.</i> , 2013)	IEEE
<i>Efficient Embedding of Virtual Networks to Distributed Clouds Via Exploring Periodic Resource Demands</i>	(XU; LIANG; XIA, 2016)	IEEE
<i>Effective and efficient AI-based approaches to cloud resource provisioning</i>	(YANG <i>et al.</i> , 2013)	IEEE
<i>MINTED: Multicast Virtual Network Embedding in Cloud Data Centers With Delay Constraints</i>	(AYOUBI <i>et al.</i> , 2015)	IEEE

Continua na próxima página

Tabela 5 – Estudos primários selecionados.

Título	Referência	Base
<i>Virtual network embedding with pre-transformation and incentive convergence mechanism</i>	(WANG <i>et al.</i> , 2017)	WDL
<i>ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping</i>	(CHOWDHURY; RAHMAN; BOUTABA, 2012)	IEEE
<i>A model for virtual network embedding across multiple infrastructure providers using genetic algorithm</i>	(PATHAK; VIDYARTHI, 2017)	IEEE

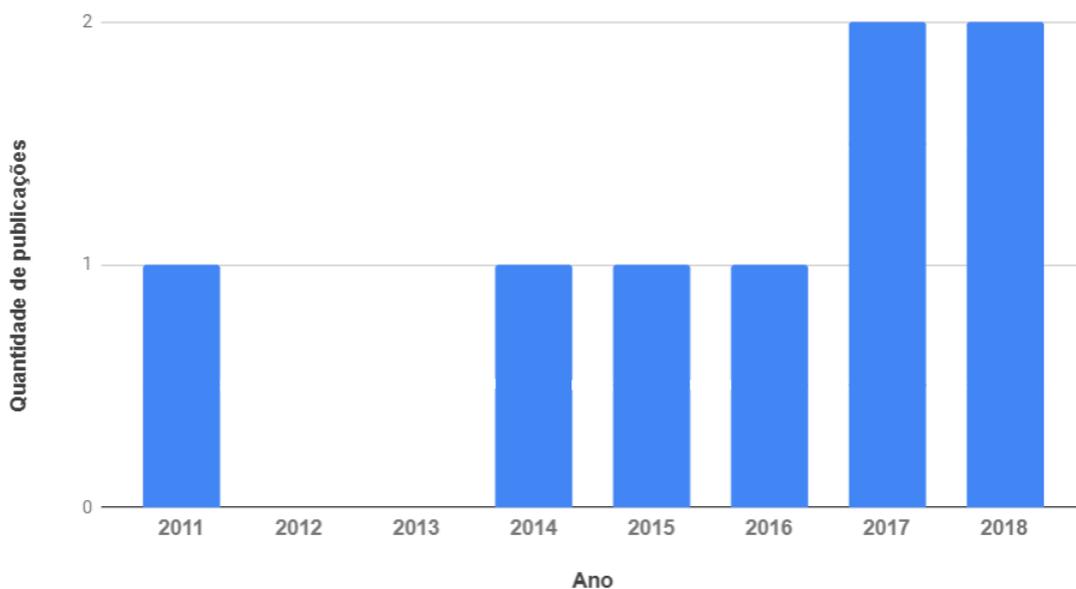
Fonte: elaboração própria

Após selecionar os estudos primários foram respondidas as questões de pesquisa. Vale frisar que o objetivo das questões é responder indagações que surgiram em contato com o tema do mapeamento. Ademais, conhecer as abordagens utilizadas e métodos para analisar o desempenho das soluções dos principais trabalhos. Dessa forma, a seguir são apresentadas as questões de pesquisa acompanhadas de suas respostas.

QP1: Em quais anos há publicações sobre alocação de Infraestruturas e Redes Virtuais?

Essa questão objetiva exibir o número de publicações voltadas para alocação de Infraestruturas e Redes virtuais (VNE). A Figura 4 exibe o número de artigos que abordam o problema de alocação de Infraestruturas Virtuais ao longo dos anos.

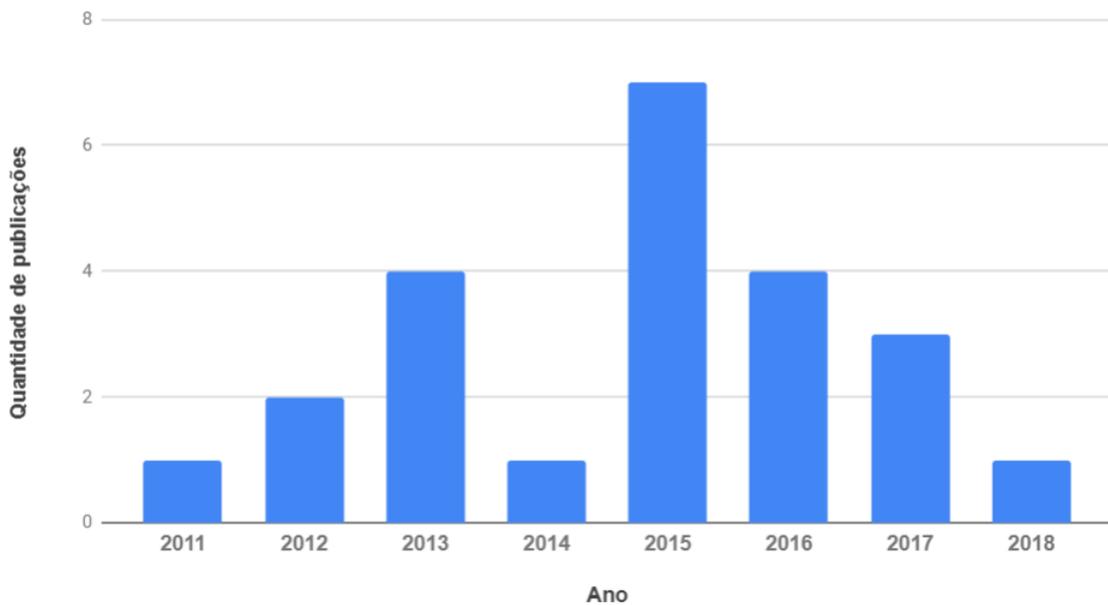
Figura 4 – Gráfico que apresenta a quantidade de publicações por ano relacionados a alocação de VIs.



Fonte: elaboração própria

O número de trabalhos que abordam VNE é exposto na Figura 5. É possível observar um grande número de artigos publicados no ano de 2016 comparado com os outros anos.

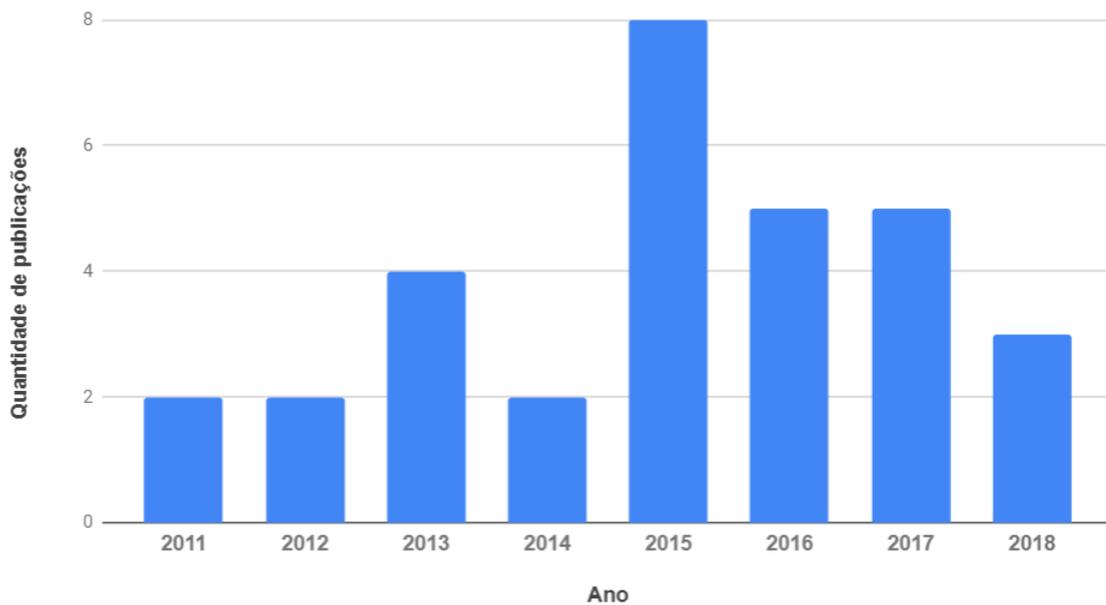
Figura 5 – Gráfico que apresenta os estudos primários por ano relacionados a VNE.



Fonte: elaboração própria

Na Figura 6 é apresentada a soma do número de artigos que aborda o problema de alocação de VIs com VNE. Analisando a Figura é possível observar que há uma quantidade considerável de artigos que abordam os temas alocação de VIs e VNE nos últimos 3 anos, uma evidência de que os temas continuam relevantes.

Figura 6 – Gráfico que apresenta os estudos primários por ano relacionados a alocação de VIs e VNE.



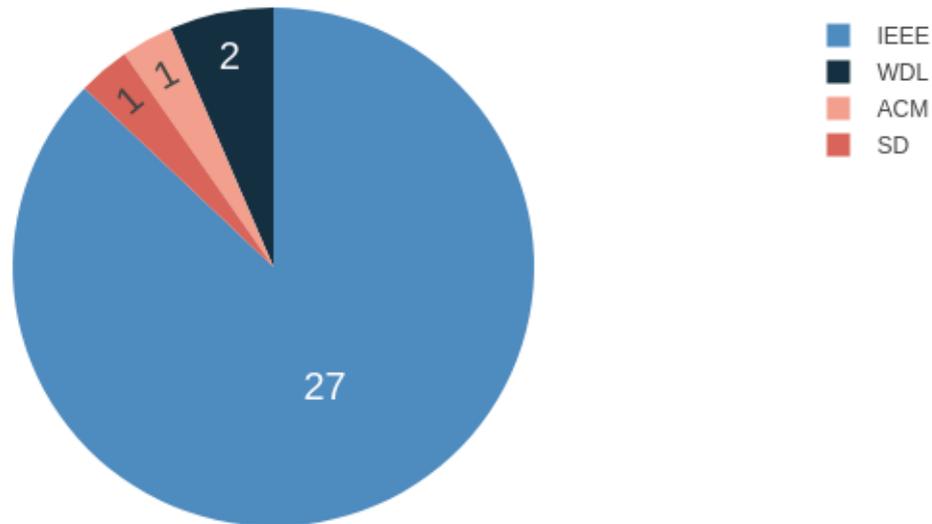
Fonte: elaboração própria

Vale ressaltar, que os resultados apresentados nessa seção foram inferidos a partir da leitura aprofundada dos estudos primários selecionados. Portanto, os artigos publicados que não foram contabilizados nesse estudo não se encaixam em nenhum dos critérios de inclusão descritos na seção 2.2.3.

QP2: Qual a base que mais contribuiu com estudos primários após os procedimentos de seleção?

Analisando a Figura 7, fica claro que a base que mais contribuiu com estudos primários foi a IEEE com 27 de 31 (87,1%) dos estudos.

Figura 7 – Número de artigos selecionados de cada base após a etapa da leitura completa.



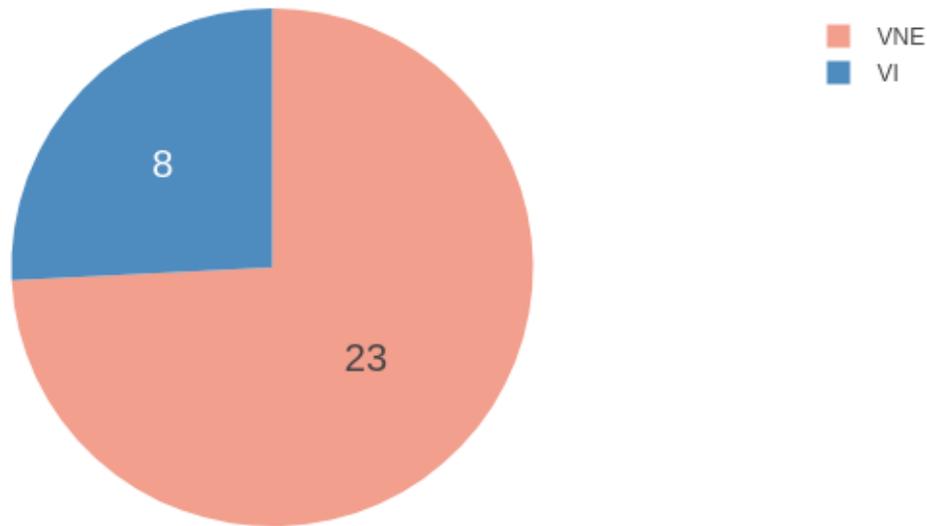
Fonte: elaboração própria

Esse resultado pode ser uma evidência de que fazem parte da IEEE as conferências e revistas que mais aceitam artigos sobre os temas.

QP3: Como estão divididos quantitativamente os artigos que solucionam a alocação de Redes e Infraestruturas Virtuais?

A finalidade dessa questão é saber qual problema, alocação de VIs ou VNE, obteve mais atenção dos estudos primários selecionados. Verificando a Figura 8, é possível observar que a maioria dos estudos 23 dos 31 (74,2%) se dedicaram a resolver o problema de VNE, e 8 dos 31 (25,8%) ao problema de Alocação de VIs.

Figura 8 – Quantidade de artigos que se dedicaram a resolver o problema de alocação de VIs e VNE.



Fonte: elaboração própria

Esse resultado mostra que o problema de alocação de VIs está mais aberto a novas soluções que o de VNE, pois é um problema mais novo, derivado do VNE. Além disso, várias soluções já desenvolvidas para VNE podem ser aplicadas para alocação de VIs.

QP4: Quais as principais métricas utilizadas para avaliar o desempenho das abordagens propostas para resolver os problemas de VNE e alocação de VIs?

Métricas são critérios definidos para comparar o desempenho de uma solução, sistema e etc. Devem ser definidas considerando os possíveis resultados em sua execução, tanto para uma execução bem sucedida, quanto para o caso de erros eventuais ou falhas. Essa questão tem como finalidade responder quais as principais métricas utilizadas na avaliação de desempenho de soluções para alocação de VIs e para VNE.

Na Tabela 6, tem-se as principais métricas utilizadas para avaliar soluções para alocação de VIs e os respectivos trabalhos que as utilizaram.

Tabela 6 – Referências e principais métricas utilizadas nos estudos para avaliar soluções de alocação de VIs.

Referência	Taxa de aceitação	Lucro do Provedor	Fragmentação do <i>Data Center</i>
(CHOWDHURY; RAHMAN; BOUTABA, 2012)	✓		
(SOARES; APARÍCIO; SARGENTO, 2017)	✓		
(SOUZA <i>et al.</i> , 2017b)	✓	✓	✓
(OLIVEIRA; KOSLOVSKI, 2017)	✓	✓	✓
(NESI <i>et al.</i> , 2018)	✓		✓

Fonte: elaboração própria

Foram selecionadas as três métricas mais frequentes nos estudos primários, conforme a Tabela 6, as quais são: taxa de aceitação, lucro do provedor e fragmentação do *Data Center*. A taxa de aceitação, segundo Oliveira e Koslovski (2017), quantifica a porcentagem de sucesso de VIs alocadas. Dado um intervalo, ela é calculada dividindo o número de VIs alocadas pelo número total de requisições. Lucro do provedor é dado pela quantidade de requisições de VIs que foram atendidas pelo provedor (SOUZA *et al.*, 2017b). A Fragmentação do *Data Center* indica a porcentagem de recursos ativos que hospedam VIs, calculados dividindo o número de recursos ativos pelo número total de recursos disponíveis (OLIVEIRA; KOSLOVSKI, 2017).

De maneira semelhante, para o problema de VNE, foram selecionadas as três métricas mais presente nos trabalhos. A Tabela 7 exibe as métricas de taxa de aceitação, lucro do provedor e taxa de rejeição.

Tabela 7 – Referências e principais métricas utilizadas nos estudos para avaliar soluções para VNE.

Referência	Taxa de aceitação	Lucro do Provedor	Taxa de rejeição
(LI; SAIDI; CHEN, 2016)	✓	✓	
(QING; FENG; JIN, 2015)	✓		
(HOUIDI; LOUATI; ZEGHLACHE, 2015)	✓		
(HOUIDI; ZEGHLACHE, 2013)	✓		
(WEN; QIAN; LU, 2013)	✓		
(CHENG <i>et al.</i> , 2011)	✓		
(CHENG <i>et al.</i> , 2012)	✓		
(ZHENG <i>et al.</i> , 2017)	✓		
(YANG <i>et al.</i> , 2013)	✓		
(MAO <i>et al.</i> , 2015)	✓		
(CHOWDHURY; RAHMAN; BOUTABA, 2012)	✓	✓	
(PATHAK; VIDYARTHI, 2017)	✓		
(BIANCHI; PRESTI, 2016)		✓	✓
(SOUALAH <i>et al.</i> , 2015)		✓	✓
(SOUALAH <i>et al.</i> , 2016)		✓	
(SHEN <i>et al.</i> , 2015)		✓	✓
(AYOUBI <i>et al.</i> , 2015)		✓	
(KARMOSHI <i>et al.</i> , 2016)			✓
(SOUALAH <i>et al.</i> , 2014)			✓
(SOUALAH <i>et al.</i> , 2013)			✓

Fonte: elaboração própria

Estudando as métricas utilizadas nos trabalhos de Alocação de VIs e VNE foi possível concluir que as métricas de taxa de aceitação e lucro do provedor são definidas de forma análoga. Provavelmente, a métrica já era utilizada em VNE e foi aplicada para Alocação de VIs. Vale frisar que nem sempre uma métrica utilizada nos trabalhos de VNE é também empregada nos de Alocação de VIs. Um exemplo disso é a métrica taxa de rejeição, muito utilizada nos estudos primários de VNE selecionados por esse mapeamento, que não foi utilizada em nenhum de Alocação de VIs. Essa métrica pode ser calculada dividindo o número de requisições de VIs rejeitadas pelo número total de requisições.

Concluindo, essas métricas podem ser utilizadas em trabalhos futuros que visem analisar o desempenho de soluções para os dois problemas já citados. São métricas amplamente utilizadas e que vão facilitar a comparação do desempenho da solução proposta com outra já existente.

QP5: Qual o tipo de otimização, técnica de avaliação, topologia e principal contribuição dos trabalhos que se dedicaram a resolver o problema de alocação de VIs e VNE?

Essa questão visa mostrar uma visão geral das principais características de cada trabalho, mostrando elementos essenciais para o desenvolvimento e comparação de soluções. Na Tabela 8,

tem-se as principais características dos trabalhos que se dedicaram a alocação de VIs.

Tabela 8 – Resumo das principais características dos estudos primários relacionados a alocação de VIs.

Referência	Otimização	Topologia	Principal contribuição
(CHOWDHURY; RAHMAN; BOUTABA, 2012)	Ótima	<i>Fat-Tree</i>	Modelo de alocação ótimo que considera a fragmentação do substrato físico da rede e a sobrevivência das VIs.
(SOARES; APARÍCIO; SARGENTO, 2017)	Ótima e Heurística	Aleatória	Heurística que utiliza uma estratégia dinâmica de incorporação de VIs combinando balanceamento de carga e estratégias de economia de energia.
(NESI <i>et al.</i> , 2018)	Heurística	Aleatória	Refatoração de algoritmos para alocação de VIs para executarem em GPUs.
(KOSLOVSKI <i>et al.</i> , 2011)	Heurística	Aleatória	Heurística de alocação guiada pela localização geográfica dos componentes físicos e virtuais das VIs.
(YU; XUE; ZHANG, 2015)	Heurística	<i>Full-mesh</i>	Aplicação de técnica de otimização denominada arredondamento sequencial para alocação de VIs.
(RAUGUST <i>et al.</i> , 2018)	Ótima	<i>Fat-Tree</i>	Mecanismo de alocação que considera requisitos de sobrevivência e a confiabilidade de VIs.
(SOUZA <i>et al.</i> , 2017b)	Heurística	<i>Fat-Tree</i>	Mecanismo de alocação que aloca VIs em <i>Data Centers</i> implementados com SDN respeitando requisitos de QoS.
(OLIVEIRA; KOSLOVSKI, 2017)	Heurística	<i>Fat-Tree</i>	Algoritmo para alocação de VIs online baseado em árvores.

Fonte: elaboração própria

Semelhante a Tabela 8, a Tabela 9, traz os estudos primários de VNE e suas principais características.

Tabela 9 – Resumo das principais características dos estudos primários relacionados a VNE.

Referência	Otimização	Topologia	Principal contribuição
(LI; SAIDI; CHEN, 2016)	Ótima	Aleatória	Estrutura de mapeamento de <i>links</i> de vários domínios que combina mapeamento de <i>links</i> de como a otimizar a utilização geral dos recursos.
(KARMOSHI <i>et al.</i> , 2016)	Ótima	Aleatória	Implementação de um algoritmo para VNE na plataforma de nuvem do OpenStack.
(BIANCHI; PRESTI, 2016)	Ótima	Aleatória	Algoritmo de alocação que utiliza uma abordagem de classificação dos nós baseado nos processos de recompensa de Markov.
(SOUALAH <i>et al.</i> , 2015)	Heurística	Aleatória	Estratégia de incorporação de redes virtuais baseada no algoritmo de otimização <i>Monte Carlo Tree Search</i> .
(SOUALAH <i>et al.</i> , 2014)	Heurística	Aleatória	Estratégia de incorporação de redes virtuais preventiva baseado na teoria dos jogos.
(QING; FENG; JIN, 2015)	Heurística	Aleatória	Algoritmo para alocação de VN distribuído que utiliza técnicas de aprendizagem e inferência.

Continua na próxima página

Tabela 9 – Resumo das principais características dos estudos primários relacionados a VNE.

Referência	Otimização	Topologia	Principal contribuição
(HOUIDI; LOUATI; ZEGHLACHE, 2015)	Ótima	Aleatória	Algoritmo de alocação de VN multiobjetivo baseado numa formulação em programação linear mista.
(GUAN <i>et al.</i> , 2015)	Meta-heurística	Aleatória	Algoritmo baseado na meta-heurística de colônia de formigas.
(HOUIDI; ZEGHLACHE, 2013)	Meta-heurística	Aleatória	Modelo de previsão para prever as demandas periódicas de recursos.
(WEN; QIAN; LU, 2013)	Ótima	Aleatória	Uma nova abordagem que quebra a atomicidade do pedido de uma VNE.
(CHENG <i>et al.</i> , 2011)	Heurística	Aleatória	Classificação dos nós utilizando a caminhada aleatória de Markov.
(CHENG <i>et al.</i> , 2012)	Meta-heurística	Aleatória	Aplicação da caminhada aleatória de Markov com o PSO no VN.
(ZHENG <i>et al.</i> , 2017)	Meta-heurística	Aleatória	Aplicação da meta-heurística do sistema de colônia de formigas no VNE.
(SOUALAH <i>et al.</i> , 2016)	Heurística	Aleatória	Técnica com base na transformação da árvore <i>Gomory-Hu Tree</i> e a otimização linear inteira para VNE.
(SHEN <i>et al.</i> , 2015)	Heurística	<i>Fat-Tree</i>	Alocação de VNE considerando conjuntamente os custos agregados de largura de banda e disponibilidade.
(MAO <i>et al.</i> , 2015)	Ótima e Heurística	Aleatória	Algoritmo para VNE que combina o algoritmo de caminho mais curto com a técnica proposta RBS-VNE.
(SOUALAH <i>et al.</i> , 2013)	Meta-Heurística	Aleatória	Aplicação da meta-heurística do sistema de colônia de formigas no VNE com confiabilidade e preventiva quanto a falhas.
(XU; LIANG; XIA, 2016)	Ótima e Heurística	Aleatória	Modelo de previsão de demandas periódicas de recursos com base em suas demandas históricas.
(YANG <i>et al.</i> , 2013)	Meta-Heurística	Aleatória	Aplicação da técnica de evolução diferencial no VNE.
(AYOUBI <i>et al.</i> , 2015)	Meta-Heurística	<i>Fat-Tree</i>	Aplicação da meta-heurística de busca tabu no VNE.
(WANG <i>et al.</i> , 2017)	Meta-Heurística	<i>Fat-Tree</i>	Aplicação da meta-heurística PSO na resolução do VNE.
(CHOWDHURY; RAHMAN; BOUTABA, 2012)	Ótima e Heurística	Aleatória	Primeiro trabalho a propor alocação conjunta de nós e <i>links</i> por meio de algoritmos baseados em arredondamento.
(PATHAK; VIDYARTHI, 2017)	Meta-Heurística	Aleatória	Aplicação do GA para problemas de VNE gerenciando múltiplas redes físicas.

Fonte: elaboração própria

Analisando as Tabelas 8 e 9, é possível observar que de cada estudo foi coletado o ano, a referência, a estratégia de otimização utilizada pela abordagem, a topologia e a principal

contribuição.

Dentre as possíveis estratégias de otimização, nos estudos desse mapeamento foram encontrado três tipos: ótima, heurística e meta-heurística. As estratégias ótimas tentam encontrar um valor ótimo para a função objetivo definida pelo provedor, e o tempo para encontrar esse valor geralmente é exponencial. Uma estratégia heurística permite encontrar um valor aproximado do ótimo a favor de um tempo linear para encontrar esse valor. Uma estratégia meta-heurística é um método heurístico para resolver de forma genérica problemas empregando algum grau de aleatoriedade para encontrar soluções ótimas (ou tão próximas da ótima quanto possível) (LUKE, 2013).

O emprego de determinadas topologias de rede física contribui significativamente para a redução das taxas de rejeição e, portanto, para um melhor aproveitamento dos recursos (LUIZELLI *et al.*, 2013). Logo, a topologia escolhida é um fator importante para analisar o desempenho de uma solução.

Dentre as topologias utilizadas pelos estudos primários vale citar a aleatória e a *fat-tree*. A topologia aleatória ou gerada aleatoriamente é a mais utilizada tanto nos trabalhos que se dedicam aos problemas de alocação de VIs como os de VNE. A maioria dos trabalhos que utilizaram topologias aleatórias as empregaram usando a ferramenta *GT-ITM tool* (ZEGURA; CALVERT; BHATTACHARJEE, 1996). A topologia *fat-tree* é comumente utilizada em *Data Centers* e é aplicada no projeto de *Data Centers* da Cisco (CISCO, 2013).

As contribuições de cada trabalho também estão presentes de forma resumida nas Tabelas 8 e 9. Dessa maneira, a resposta para essa questão forneceu uma visão geral de algumas das principais características dos trabalhos e a sua contribuição para a literatura.

QP6: Dentre os artigos que utilizaram a técnica de simulação, algum propôs um simulador para auxiliar na avaliação de propostas para resolver o problema de alocação de VIs e VNE?

Todos os trabalhos selecionados por esse mapeamento utilizaram a técnica de simulação para avaliar o desempenho de suas soluções propostas. Isso deve-se ao fato de que simulações podem incorporar mais detalhes do problema e oferecem uma maior facilidade para combinação de parâmetros. Além disso, oferece mais flexibilidade para alteração das configurações (JAIN, 1991).

Nenhum estudo primário selecionado nesse mapeamento que se dedicou a resolver o problema de alocação de VIs propôs um simulador que possa ser utilizado para avaliar novas soluções. Por outro lado, nos estudos que investiram no problema de VNE, foi possível encontrar trabalhos que propõem simuladores ou utilizam simuladores propostos por terceiros de trabalhos que não fazem parte desse mapeamento.

Dentre os estudos que propõem simuladores tem-se o de Cheng *et al.* (2011) que propôs

um simulador chamado *VNE-RW Simulator*. Bianchi e Presti (2016), Chowdhury, Rahman e Boutaba (2012), Soualah *et al.* (2014) e Cheng *et al.* (2012) afirmam ter desenvolvido simuladores mas não deram nomes e nem disponibilizaram para o uso de terceiros.

No conjunto dos trabalhos que utilizaram simuladores de terceiros tem-se o de Qing, Feng e Jin (2015) que utilizou um simulador denominado *VNE-SIM* (HAERI; TRAJKOVIĆ, 2016). O estudo de Wang *et al.* (2017) empregou o simulador *CloudSim* (CALHEIROS *et al.*, 2011) e o trabalho de (PATHAK; VIDYARTHI, 2017) utilizou a ferramenta MATLAB (MATLAB, 2019) para executar as suas simulações.

Desse modo, conclui-se que a técnica de simulação para avaliar o desempenho de soluções para problemas de alocação de VIs e VNE é amplamente utilizada.

2.4 Considerações finais sobre o Mapeamento

Neste trabalho, realizou-se um mapeamento sistemático que objetivou identificar e mapear os estudos que contemplam a alocação de VIs e VNE no ambiente de Computação em Nuvem. O processo de mapeamento sistemático foi conduzido usando um protocolo de busca e seleção de estudos explanado na seção 2.2.4 desse trabalho. Do método especificado, foram extraídos e analisados dados de 31 trabalhos relacionados a problemas de alocação de VIs e VNE. Dos estudos analisados foram coletadas as principais características com o objetivo de facilitar o desenvolvimento de novas soluções e a comparação dessas com alguma já desenvolvida.

A partir dos resultados obtidos foi possível concluir que os temas de VNE e Alocação de VIs continuam relevantes, pois há uma concentração relevante de trabalhos aprovados em conferências e revistas nos últimos anos. Ademais, as conferências e revistas que mais aceitaram os artigos relacionados aos temas analisados indexam artigos na base da IEEE.

Acrescenta-se também que dentre os estudos primários selecionados 74,2 % se dedicam a resolver o problema de VNE, evidenciando que o problema de alocação de VIs oferece mais oportunidades para novos trabalhos. Além disso, várias soluções já desenvolvidas para VNE podem ser aplicadas para alocação de VIs.

Ademais, as principais métricas para alocação de VIs são: taxa de aceitação, lucro do provedor e fragmentação do *Data Center*. Para VNE é taxa de aceitação, lucro do provedor e taxa de rejeição. Essas métricas podem ser utilizadas em trabalhos futuros que tenham como objetivo resolver um dos problemas, facilitando assim, a comparação da solução proposta com alguma já desenvolvida.

A propósito, quanto as principais estratégias de otimização, vale destacar as estratégias heurísticas e meta-heurísticas. Devido ao fato de o problema de alocação de VIs e VNE ser NP-Difícil, essas estratégias são empregadas para encontrar soluções ótimas ou tão próximas da ótima quanto possível.

Por fim, observando as análises experimentais dos estudos foi possível identificar que todos os trabalhos utilizam a técnica de simulação para avaliar as suas soluções. Essa técnica oferece uma maior capacidade de incorporar mais detalhes do problema e uma maior facilidade para a combinação de parâmetros. Além disso, o emprego de determinadas topologias pode influenciar diretamente no desempenho da solução. Dentre os trabalhos selecionados as duas topologias mais utilizadas foram a aleatória, geralmente gerada com o auxílio de ferramentas e a *Fat-Tree*, que é amplamente utilizada em *Data Centers* de alta resiliência.

É possível destacar como principal contribuição desse mapeamento a sumarização dos trabalhos de alocação de VIs e VNE. Ademais, as métricas, estratégias de otimização e demais características dos estudos primários apresentados nesse mapeamento fornecem uma visão geral das áreas que pode facilitar a elaboração de novos trabalhos.

3

Modelo para Alocação de Infraestruturas Virtuais utilizando Algoritmo Evolucionário Multiobjetivo

No presente capítulo é apresentado o modelo para alocação de Infraestruturas Virtuais utilizando um MOEA para resolver o problema descrito na seção 1.1. Sendo assim, são explicados os motivos pelos quais essa abordagem foi escolhida para resolver o problema e como ela funciona. Resumidamente, o objetivo do presente trabalho é alocar VIs em redes físicas atendendo os requisitos de computação e rede, avaliando soluções que atendam objetivos como balanceamento de carga e baixo *overhead* de alocação. Isto posto, para alcançá-lo é necessário empregar uma abordagem multiobjetivo.

3.1 Otimização Multiobjetivo

Formalmente, um problema de Otimização Multiobjetivo (MOP, do inglês *Multi-Objective Problem*) envolve a satisfação simultânea de duas ou mais funções objetivo (CARVALHO, 2013). Nesse tipo de problema, as funções objetivo que são otimizadas às vezes podem estar em conflito, ou seja, se o valor de uma função objetivo melhora, em geral, o valor de outra função piora (LUKE, 2013).

No presente trabalho, há dois objetivos de interesse a se otimizar: balanceamento de carga e *overhead* de alocação. Sendo assim, esses objetivos precisam ser descritos matematicamente por meio de funções de qualidade. Além disso, é fundamental a aplicação de uma técnica que possibilite a satisfação simultânea de ambas as funções. Logo, sabendo que estamos lidando com um problema multiobjetivo, é indispensável a aplicação de algoritmos adequados para a obtenção de uma solução.

3.2 Algoritmos Evolucionários Multiobjetivo

Os algoritmos evolucionários são adequados para resolver problemas de otimização multiobjetivo, porque lidam simultaneamente com um conjunto de possíveis soluções (comumente chamado de população). Deste modo, MOPs têm sido resolvidos através desses algoritmos que, quando aplicados a MOPs, são chamados de MOEA (COELLO *et al.*, 2007).

Há vários algoritmos evolucionários disponíveis na literatura, o Algoritmo Genético de Triagem não Dominado II (NSGA-II, do inglês *Non dominated Sorting Genetic Algorithm II*) (DEB *et al.*, 2000), o Algoritmo de Otimização por Enxame de partículas (PSO, do inglês: *Particle Swarm Optimization*) (SHI *et al.*, 2001) e o GA (WHITLEY, 1994), dentre outros. Dentre eles foi selecionado o GA para essa proposta devido a sua facilidade de implementação e pela experiência do autor em já tê-lo aplicado em outros trabalhos.

3.3 Modelo de Alocação Proposto

Nessa seção, é apresentado o modelo baseado no GA para alocação de VIs. O GA é um algoritmo evolucionário bio-inspirado largamente utilizado em problemas de otimização combinatória (PATHAK; VIDYARTHI, 2017). Ele emprega de forma adaptada princípios da natureza, como seleção e mutação, na construção de algoritmos que buscam uma melhor solução para um determinado problema, tudo isso por meio da evolução de populações de soluções codificadas em cromossomos artificiais. Uma visão geral do GA pode ser encontrada em Whitley (1994).

O primeiro passo do GA para solucionar um problema é a geração aleatória de um conjunto de soluções potenciais ou cromossomos, esse conjunto é comumente chamado de população. As soluções são geradas dentro de um espaço de busca, onde, no presente trabalho, foram consideradas as possibilidades de solução para o problema de alocação de VIs. Desse modo, o espaço de busca contém todas as possíveis alocações para uma determinada requisição de VI em uma rede física selecionada.

O GA explora o espaço de busca utilizando a técnica de sobrevivência do mais apto (WHITLEY, 1994). A cada nova geração as soluções consideradas "ruins" são descartadas e as classificadas como "boas" são mantidas e utilizadas para a geração de novas soluções. Ademais, as soluções são avaliadas por meio de funções de avaliação.

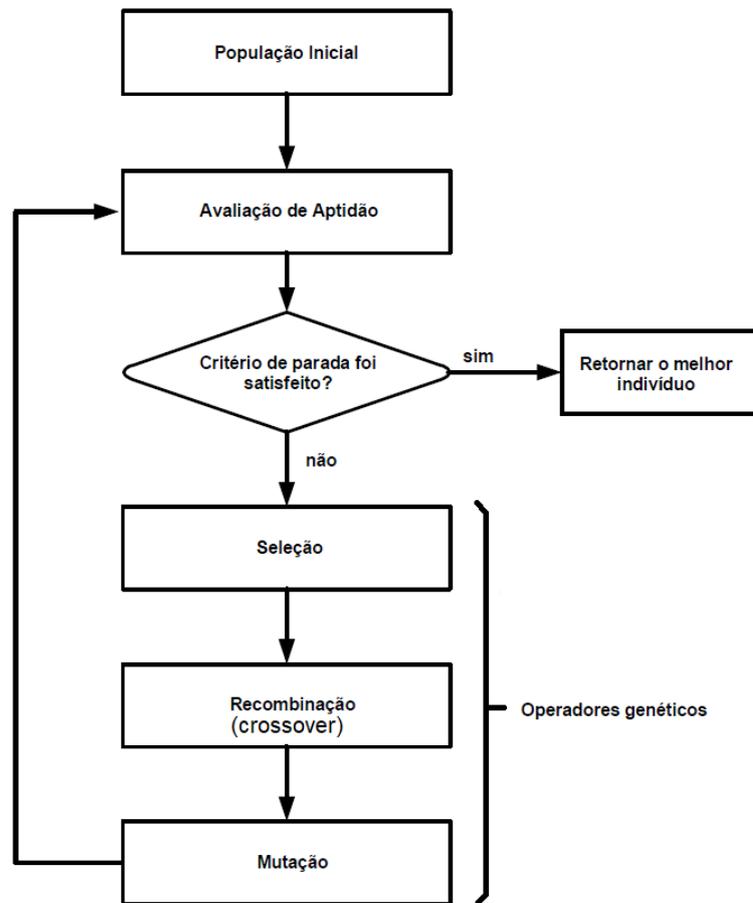
O GA empregado no presente trabalho consiste em uma população inicial, seguida dos operadores de seleção, *crossover* e mutação:

- **Seleção:** escolha de boas soluções dentro da população por meio de uma função de adequabilidade ou viabilidade.

- **Crossover ou recombinação:** troca de valores de genes entre um par de potenciais cromossomos para gerar novas soluções potenciais.
- **Mutação:** alteração aleatória dos valores dos genes de soluções potenciais.

Esses operadores são empregados de forma sequencial com o objetivo de se obter soluções melhores a cada geração. Na Figura 9, tem-se o fluxograma do GA proposto nesse trabalho.

Figura 9 – Fluxograma do GA utilizado na abordagem proposta.



Fonte: elaboração própria

No fluxograma da Figura 9, é possível observar que após gerar a população inicial aleatoriamente, o GA realiza um conjunto de procedimentos em sequência na população. Esses procedimentos são realizados sequencialmente em forma de passos. Desse modo, no primeiro passo é realizada a avaliação de aptidão; no segundo, é feita uma análise da população. Finalmente, no último e terceiro passo são aplicados os operadores do GA.

Na avaliação de aptidão, todos os indivíduos da população são avaliados por meio das funções de qualidade. Após essa avaliação, é analisado se algum membro da população, ou seja, se alguma solução ou cromossomo atende os critérios de parada definidos. Em caso positivo, a

execução pára e a solução que atende os requisitos é selecionada. Caso contrário, os operadores do GA são aplicados.

Os operadores do GA, Seleção, *Crossover* e Mutação, respectivamente, são aplicados para gerar novas soluções que podem ser melhores que as anteriores. Depois da aplicação dos operadores é completada uma geração e a execução volta para o passo de avaliação de aptidão. Além disso, as soluções consideradas "ruins" são descartadas. Sendo assim, as características que tornam soluções boas são preservadas e utilizadas na geração de novas soluções. Desse modo, o algoritmo caminha para melhores soluções até encontrar uma que atenda a condição de parada.

3.3.1 Representação do cromossomo

Um cromossomo ou individuo da população é uma solução potencial, ou seja, representa uma solução completa para um determinado problema. Geralmente, um cromossomo consiste em um conjunto de genes que por meio de uma codificação podem representar uma solução. Assim sendo, um cromossomo pode representar o mapeamento de recursos virtuais de uma VI para recursos físicos de uma infraestrutura, ou seja, cada cromossomo pode corresponder a uma decisão de alocação.

A estrutura do cromossomo utilizada no presente trabalho foi adaptada de [Pathak e Vidyarthi \(2017\)](#) para o problema de alocação de VIs. Sendo assim, o cromossomo é um vetor que possui como comprimento a soma da quantidade de elementos da infraestrutura física (servidores, *switches* e *links*). Além disso, empregou-se a codificação inteira para representar o espaço de busca do problema a ser resolvido, ou seja, o mapeamento dos recursos virtuais das VIs para recursos físicos da infraestrutura.

O processo de construção do cromossomo é baseado na infraestrutura física, para cada infraestrutura física, um tipo de cromossomo é construído considerando as seguintes premissas: (1) o tamanho do cromossomo ou número de *locus* é igual a soma de elementos da infraestrutura física; (2) cada *locus* do cromossomo corresponde a um recurso físico; (3) as informações que os *locus* do cromossomo recebem são ordenados pelo tipo de recurso, servidores, *switches* e links respectivamente; (4) os *locus* recebem um número ou *allele* que corresponde ao identificador do recurso virtual da VI, ou seja, quando no cromossomo existe um *locus* não vazio, isso quer dizer que há um mapeamento de um recurso virtual da VI para um recurso físico da infraestrutura.

Dado o processo de estruturação e codificação do cromossomo, na Figura 10a, tem-se o exemplo de uma infraestrutura física que vai alocar a VI definida anteriormente, ela possui os seguintes recursos: oito servidores, três *switches* e dez *links*. Por outro lado, na Figura 10b, é exibida uma possível VI com os seguintes recursos: dois servidores, um *switch* e dois *links*. Nessas condições o procedimento de estruturação do cromossomo e codificação se dá pelas seguintes etapas:

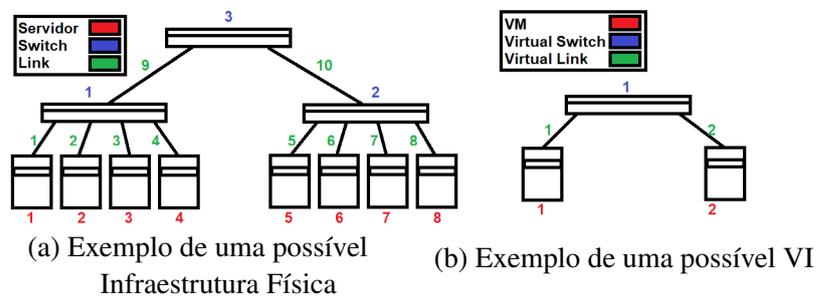
- Enumeração dos recursos virtuais e físicos: cada recurso físico e virtual recebe um

identificador iniciando do número zero para cada tipo de recurso;

- Representação dos recursos físicos: cada recurso físico enumerado é representado no cromossomo com um *locus*, sendo que os *allelos* são preenchidos em ordem crescente e respeitando as prioridades dos recursos: primeiro servidores, depois *switches* e por fim os *links*;
- Mapeamento dos recursos virtuais nos recursos físicos: cada *locus* do cromossomo recebe um valor ou *allelo* de forma aleatória, que nesse caso vai corresponder a um recurso virtual que será mapeado. Caso não haja mapeamento o *locus* vai receber um valor correspondente a vazio.

A codificação utilizada visa representar todo o espaço de busca do problema de alocação de VIs, ou seja, todos os mapeamentos possíveis. Sendo assim, nas Figuras 10a e 10b, é possível observar a codificação em cada *locus*. Considerando a quantidade de elementos da infraestrutura física apresentada na Figura 10a, os *locus* do cromossomo referente aos servidores recebem valores dentro do intervalo do número de servidores, ou seja, podem receber um valor $x \in N : 1 \leq x \leq 2$. Esse mesmo raciocínio vale para os *switches* e os *links*.

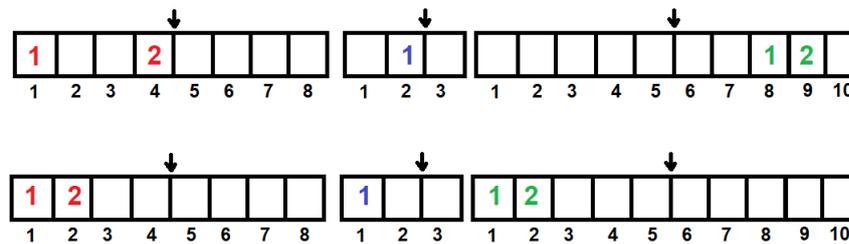
Figura 10 – Infraestrutura Física e Requisição de VI



Fonte: elaboração própria

Com a estrutura e codificação definidos, tem-se um arranjo de tamanho variado, sendo que essa variação é definida pela quantidade de elementos da infraestrutura física. Por outro lado, a codificação também varia de acordo com as requisições de VIs. Desse modo, tem-se uma estrutura flexível que se adapta às circunstâncias do problema. Na Figura 11 são apresentados dois possíveis cromossomos gerados aleatoriamente segundo o exemplo descrito. Os identificadores relativos aos servidores estão na cor vermelha, os dos *switches* estão em azul e o dos *links* estão na cor verde.

Figura 11 – Cromossomo e codificação utilizada.



Fonte: elaboração própria

Dado o exposto, vale mencionar que a cada nova requisição de VI, uma nova representação é construída para garantir que a estrutura seja adequada para a alocação. Além disso, todos os indivíduos gerados na etapa de geração da população inicial devem possuir essa estrutura. Em seguida a geração da população inicial, caso alguma das soluções geradas não atenda os critérios de parada, são aplicados o operador de seleção, *crossover* e mutação do GA.

3.3.2 Seleção

A seleção dos cromossomos é realizada por meio do método elitista, ou seja, as melhores soluções são sempre selecionadas. Uma visão geral dessa técnica pode ser encontrada em (LUKE, 2013).

3.3.3 Crossover

Depois da geração da população inicial é realizada a operação de *crossover* nos cromossomos selecionados (dois por vez). O tipo de *crossover* utilizado é o multiponto (SPEARS; JONG, 1991). A Figura 12 exemplifica a operação de *crossover* realizada.

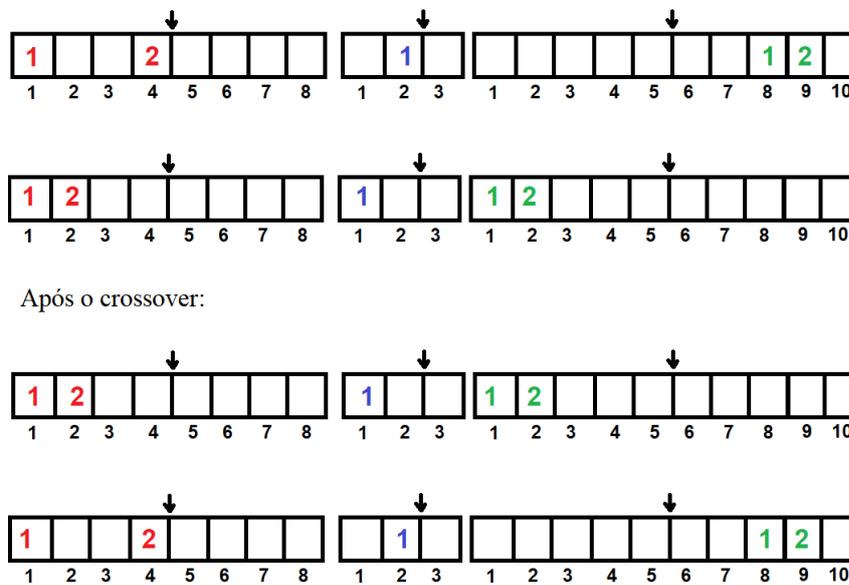
A troca de genes entre os cromossomos é realizada apenas entre recursos do mesmo tipo, ou seja, o cromossomo *a* só troca genes relacionados aos recursos de servidores com *b*. Isso vale para os demais recursos representados no cromossomo. Desse modo, um grande número de soluções inválidas não são geradas, o que contribui para a convergência do algoritmo para boas soluções.

3.3.4 Mutação

Após passar pela operação de *crossover* alguns cromossomos sofrem o processo de mutação. A mutação é apenas uma modificação aleatória em *allelos* do cromossomo. A Figura 13, exemplifica o processo de mutação empregado nos cromossomos.

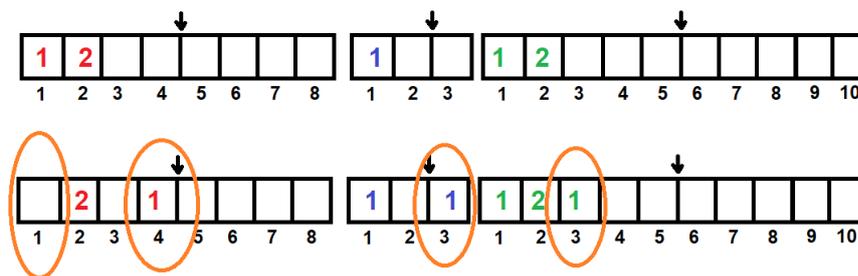
A mutação confere ao algoritmo características aleatórias que são essenciais para superar mínimos locais do espaço de busca.

Figura 12 – Operação de *crossover* multiponto.



Fonte: elaboração própria

Figura 13 – Operação de mutação.



Fonte: elaboração própria

3.3.5 Checagem de viabilidade

Inevitavelmente, grande parte das soluções geradas aleatoriamente ou por meio das operações são inválidas. Desse modo, as soluções são verificadas por meio de quatro checagens. As checagens retornam como resposta um sim ou não, para facilitar o cálculo após a aplicação das funções de qualidade.

- Checagem dos nós de computação: os tipos de nós de computação requisitados na VI estão presentes na potencial solução?
- Checagem dos nós de rede: os tipos de nós de rede solicitados na VI estão presentes na potencial solução?
- Checagem dos *links*: os tipos de *links* requisitados na VI estão presentes na potencial solução?

- Checagem de caminho: os links virtuais alocados formam um caminho virtual que consiga conectar todos os nós virtuais alocados, respeitando os requisitos definidos pelo usuário?

Essas verificações fazem parte do conjunto de operações que são realizadas para o cálculo efetivo da qualidade das soluções. Portanto, elas são essenciais para o algoritmo convergir para soluções adequadas para o problema.

3.3.6 Função de Avaliação Multiobjetivo

Os recursos residuais são parâmetros fundamentais do processo de alocação. O presente estudo também considera os recursos residuais da infraestrutura física como um parâmetro a ser otimizado. Logo, a solução resulta no mapeamento adequado das requisições de VI, objetivando o balanceamento da carga recebida na rede física e a redução do *overhead* de alocação. Para descrever as funções de qualidade, a Tabela 10 define as notações dos diferentes parâmetros utilizados.

As funções de avaliação tem como objetivo quantificar as melhores soluções considerando os objetivos de interesse. Cada solução contém todas as informações necessárias para a alocação de uma VI em uma infraestrutura física, as quais são: quantidade, características, e onde os recursos virtuais requisitados devem ser alocados. Desse modo, é possível verificar a qualidade da alocação quanto aos objetivos de balanceamento de carga e redução do *overhead* analisando a possível alocação em conjunto com os recursos disponíveis na infraestrutura física.

3.3.6.1 Balanceamento de carga

Balancear a carga é essencial na Computação em Nuvem. Sendo assim, a presente função de avaliação objetiva qualificar as melhores soluções quanto a esse quesito. Para avaliar o balanceamento da carga realizado por uma solução foi empregada a medida do coeficiente de variação. Empregando-o, é possível avaliar a dispersão dos recursos alocados na infraestrutura física. Assim sendo, com a análise da dispersão é possível quantificar o balanceamento de carga. Nesse caso, quanto menor for o coeficiente maior o balanceamento da carga na infraestrutura física. Ademais, o coeficiente de variação possibilita comparar as diferentes unidades utilizadas para quantificar os recursos alocados.

Para calcular a função de avaliação do balanceamento de carga, primeiro é necessário calcular as médias dos tipos de recursos da infraestrutura física:

$$\mu_c^H = \frac{1}{a} \sum_{i=1}^a H_i^c \quad (1)$$

$$\mu_m^H = \frac{1}{a} \sum_{i=1}^a H_i^m \quad (2)$$

Tabela 10 – Notação da formulação das funções de avaliação.

H_i^c	Recurso de CPU do servidor i ;
H_i^m	Recurso de Memória do servidor i ;
H_i^s	Recurso de <i>Storage</i> do servidor i ;
S_j^t	Recurso do <i>Switch</i> j ;
L_k^b	Recurso do <i>Link</i> k ;
a	Número de servidores;
b	Número de <i>Switches</i> ;
c	Número de <i>Link</i> ;
C_V	Coefficiente de variação;
μ_c^H	Média dos recursos de CPU disponíveis nos servidores;
μ_m^H	Média dos recursos de Memória disponíveis nos servidores;
μ_s^H	Média dos recursos de <i>Storage</i> disponíveis nos servidores;
μ_t^S	Média dos recursos dos <i>Switches</i> ;
μ_b^L	Média da largura de banda disponível nos <i>Links</i> ;
σ_c^H	Desvio padrão dos recursos de CPU disponíveis nos servidores;
σ_m^H	Desvio padrão dos recursos de Memória disponíveis nos servidores;
σ_s^H	Desvio padrão dos recursos de <i>Storage</i> disponíveis nos servidores;
σ_t^S	Desvio padrão dos recursos dos <i>Switches</i> disponíveis;
σ_b^L	Desvio padrão da largura de banda disponível nos <i>Links</i> ;
$C_{V_c}^H$	Coefficiente de variação dos recursos de CPU disponíveis nos servidores;
$C_{V_m}^H$	Coefficiente de variação dos recursos de Memória disponíveis nos servidores;
$C_{V_s}^H$	Coefficiente de variação dos recursos de <i>Storage</i> disponíveis nos servidores;
$C_{V_t}^S$	Coefficiente de variação dos recursos dos <i>Switches</i> disponíveis;
$C_{V_b}^L$	Coefficiente de variação da largura de banda disponível nos <i>Links</i> ;
f	Função de avaliação relacionada ao balanceamento de carga;
g	Função de avaliação referente ao <i>overhead</i> de alocação;
q	Função de avaliação multiobjetivo referente ao balanceamento de carga e <i>overhead</i> de alocação;

Fonte: elaboração própria

$$\mu_s^H = \frac{1}{a} \sum_{i=1}^a H_i^s \quad (3)$$

$$\mu_t^S = \frac{1}{b} \sum_{j=1}^b S_j^t \quad (4)$$

$$\mu_b^L = \frac{1}{c} \sum_{k=1}^c L_k^b \quad (5)$$

As Equações 1, 2, 3, 4 e 5 computam respectivamente: as médias dos recursos de CPU, Memória e *Storage* dos servidores e a média dos recursos dos *switches* e de Largura de banda dos links respectivamente.

Em seguida, são calculados separadamente o desvio padrão dos recursos da infraestrutura:

$$\sigma_c^H = \sqrt{\frac{1}{a} \sum_{i=1}^a (H_i^c - \mu_c^H)^2} \quad (6)$$

$$\sigma_m^H = \sqrt{\frac{1}{a} \sum_{i=1}^a (H_i^m - \mu_m^H)^2} \quad (7)$$

$$\sigma_s^H = \sqrt{\frac{1}{a} \sum_{i=1}^a (H_i^s - \mu_s^H)^2} \quad (8)$$

$$\sigma_t^S = \sqrt{\frac{1}{b} \sum_{j=1}^b (S_j^t - \mu_t^S)^2} \quad (9)$$

$$\sigma_b^L = \sqrt{\frac{1}{c} \sum_{k=1}^c (L_k^b - \mu_b^L)^2} \quad (10)$$

Por sua vez, as equações 6, 7, 8, 9 e 10 computam, o desvio padrão dos recursos de CPU, Memória e *Storage* dos servidores e a média dos recursos dos *switches* e de Largura de banda dos *links* respectivamente.

Computado os valores das médias e desvio padrão dos recursos, é possível calcular o valor do coeficiente de variação para cada recurso:

$$C_{V_c}^H = \frac{\sigma_c^H}{\mu_c^H} \quad (11)$$

$$C_{V_m}^H = \frac{\sigma_m^H}{\mu_m^H} \quad (12)$$

$$C_{V_s}^H = \frac{\sigma_s^H}{\mu_s^H} \quad (13)$$

$$C_{V_t}^S = \frac{\sigma_t^S}{\mu_t^S} \quad (14)$$

$$C_{V_b}^L = \frac{\sigma_b^L}{\mu_b^L} \quad (15)$$

As equações 11, 12, 13, 14 e 15 computam, o coeficiente de variação dos recursos de CPU, Memória e *Storage* dos servidores e a média dos recursos dos *switches* e de Largura de banda dos *links* respectivamente.

Finalmente, utilizando os cálculos anteriores, a função de avaliação relacionada ao balanceamento de carga é definida abaixo:

$$f : \frac{1}{3} \left[\frac{1}{3} (C_{V_c}^H + C_{V_m}^H + C_{V_s}^H) + C_{V_t}^S + C_{V_b}^L \right] \quad (16)$$

A Equação 16 foi construída dessa forma para que os recursos dos servidores, *switches* e *links* fossem considerados de forma equilibrada no momento do cálculo. Desse modo, mesmo o servidor possuindo três tipos de recursos representados no presente modelo, eles representam apenas um terço de todo o valor da função. Logo, não há prioridade ou peso para uma determinada entidade, servidores, *switches* e *links* tem participação equilibrada no cálculo dessa função.

3.3.6.2 Overhead de Alocação

O *overhead* de alocação mede os recursos provisionados em excesso para atender as requisições de VIs e seus requisitos. O cálculo do *overhead* empregado é simples, basicamente, é a divisão do somatório dos recursos alocados pelos requisitados de servidores, *switches* e *links*. A função que calcula o *overhead* é definida na Equação 17.

$$g : (|N_{aloc}^H/N_{req}^H - 1|) + (|N_{aloc}^S/N_{req}^S - 1|) + (|N_{aloc}^L/N_{req}^L - 1|) \quad (17)$$

Após a definição das funções de balanceamento de carga e *overhead* é possível descrever a função de avaliação multiobjetivo empregada nesse modelo na Equação 18.

$$q : f + g \quad (18)$$

A Equação 18, é a função de qualidade empregada por esse modelo, basicamente, a soma da função de balanceamento de carga com a de *overhead* de alocação. Desse modo, o objetivo do algoritmo é caminhar para soluções que minimizem essa função. Assim sendo, as soluções selecionadas pelo algoritmo vão proporcionar alocações que tenham o melhor impacto no balanceamento de carga e menor *overhead* na infraestrutura física.

3.4 Algoritmo

O pseudocódigo do MOEA empregado no presente trabalho está exposto no Algoritmo 1, ele é baseado no GA apresentado por Luke (2013, p. 36).

Algoritmo 1: MOEA BASEADO NO GA

Entrada: Infraestrutura Física e VI
Saída: Mapeamento de recursos virtuais da VI nos recursos físicos da infraestrutura, ou seja, a alocação

- 1 **início**
- 2 Critério de parada = n execuções
- 3 Criar estrutura do cromossomo
- 4 Gerar de forma aleatória população inicial P
- 5 **repita**
- 6 Avalia a aptidão dos indivíduos de P
- 7 Seleção em P
- 8 Crossover em indivíduos selecionados de P
- 9 Mutação em indivíduos selecionados de Crossover P
- 10 **até** atender o critério de parada;
- 11 **fim**
- 12 **retorna** Melhor indivíduo da população P

A avaliação da solução proposta para o problema de alocação é realizada por meio de um simulador. Sendo assim, o algoritmo fica responsável apenas por realizar o mapeamento dos recursos virtuais nos recursos físicos.

3.5 Simulador

O presente trabalho apresenta um simulador devido a necessidade de avaliar a solução proposta. Por meio do mapeamento sistemático executado na etapa de pesquisa bibliográfica dessa pesquisa, foi identificado que nenhum trabalho de alocação de VIs propôs um simulador que pudesse ser utilizado por terceiros para avaliar soluções. Sendo assim, foi desenvolvido um simulador para avaliar a solução proposta.

O simulador considera o conceito de infraestrutura habilitada para rede em nuvem (SOARES; APARÍCIO; SARGENTO, 2017). Nessa caracterização, os recursos de computação e rede fazem parte do mesmo conjunto de recursos, podendo ser virtualizados e gerenciados de maneira integrada. Portanto, empregando o conceito é possível simular a alocação de várias infraestruturas virtuais dentro de uma infraestrutura física previamente definida. Além disso, por sua estrutura ser baseada em um grafo, é exequível a criação de qualquer topologia na

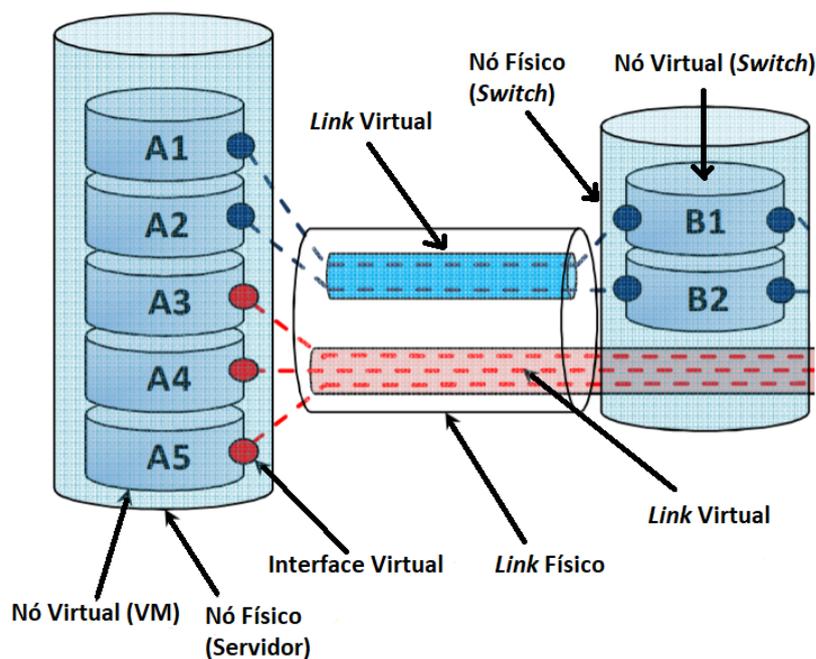
infraestrutura ou requisição de VI.

3.5.1 Estrutura

A estrutura do simulador é baseada em um grafo, sua implementação foi realizada por meio de uma lista de adjacências. Sendo assim, é possível adicionar uma quantidade de nós e arestas suficientes para representar qualquer topologia. Dois tipos de nós são considerados, os de computação e rede, cada um tem um conjunto específico de características. Os nós de computação, que representam os servidores são caracterizados por: número de CPUs, C ; quantidade de memória disponível, M ; capacidade de *storage* disponível, S . Os nós de rede, que representam os *switches* podem ser determinados por: quantidade de recursos disponíveis, R . E finalmente, as arestas que representam os *links* são caracterizados por: capacidade de largura de banda, B ; bidirecional com o *delay* máximo de D .

O comportamento de virtualização é simulado por meio da alocação de nós e *links* virtuais dentro dos nós físicos, ou seja, cada nó e *link* físico tem a capacidade de fatiar e compartilhar seus recursos entre vários nós e *links* virtuais. A Figura 14, ilustra a alocação dos nós virtuais nos físicos. O nó físico ou servidor aloca os nós virtuais ou VMs A1, A2, A3, A4 e A5; eles se comunicam com os *switches* virtuais B1 e B2 por meio do *link* virtual a ele conectado pela interface também virtual. Assim sendo, o simulador possibilita a representação de um cenário próximo do real para a avaliação das soluções.

Figura 14 – Alocação de diferentes tipos de nós virtuais em físicos numa infraestrutura virtualizada.



Fonte: adaptado de Carapinha e Jiménez (2009).

Para facilitar o gerenciamento dos recursos e avaliação do desempenho das soluções, o simulador disponibiliza um conjunto de métodos capazes de efetuar medições na infraestrutura física. As mensurações, podem ser locais, como por exemplo, na medida da quantidade de recursos físicos alocados em um determinado nó de computação ou rede, como também de um *link*. De outro modo, há métodos habilitados a oferecer medições que consideram toda a infraestrutura, como por exemplo, média dos recursos de computação alocados, média da largura de banda disponível, desvio padrão dos recursos de computação alocados, dentre outros. Portanto, o simulador é uma ferramenta que oferece um conjunto de serviços que facilitam a avaliação de soluções.

3.5.2 Serviços

Com o objetivo de simplificar a avaliação de desempenho de soluções para alocação de VIs, o simulador oferece um conjunto de serviços. Sendo os principais:

- Criação da topologia da infraestrutura física;
- Seleção das requisições de VI;
- Criação da carga de trabalho para avaliação;
- Medição do desempenho por meio de métricas.

Esses serviços fornecem os mecanismos base para uma avaliação de uma solução. Nas próximas seções cada um deles é descrito com mais detalhes.

Definição da Infraestrutura Física

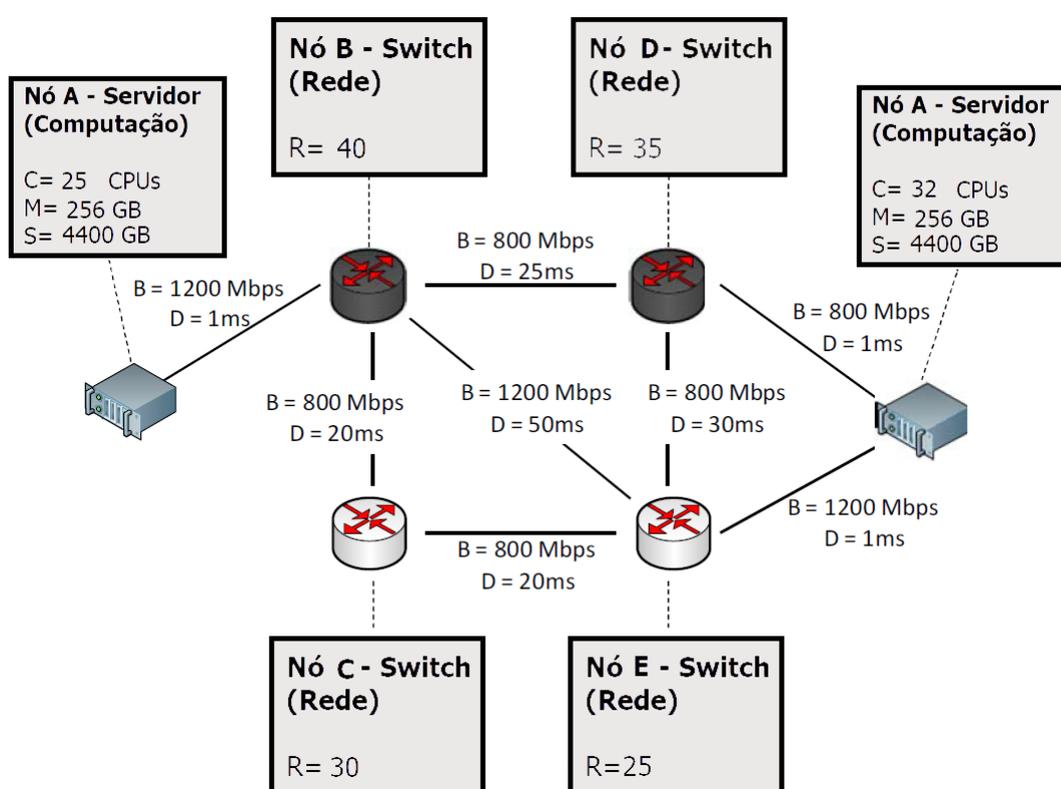
Esse serviço possibilita descrever a quantidade de nós e *links* que a infraestrutura vai possuir e o modo como todos estão ligados entre si. Ademais, a quantidade dos recursos físicos pode ser definida. Os tipos de recursos foram definidos baseando-se nos trabalhos de [Anhalt, Koslovski e Primet \(2010\)](#), [Manvi e Shyam \(2014\)](#). Sendo assim, os recursos dos nós de computação ou servidores são: número de CPUs, C ; quantidade de memória, M ; capacidade de *storage*, S . Os nós de rede, que representam os *switches* podem ser determinados por: quantidade de recursos (todos os recursos do switch foram simplificados para apenas um como forma de simplificar o problema), R ; As arestas que representam os *links* são caracterizados por: capacidade de largura de banda, B ; bidirecional com o *delay* máximo de D . Em suma, é possível descrever a quantidade de nós e *links*, o modo como estão conectados entre si, e as suas características.

A flexibilidade na definição da topologia possibilita a criação e avaliação de soluções nos mais variados cenários. A Figura 15, ilustra uma infraestrutura física heterogênea que pode ser criada por meio do simulador. Nela, é possível observar as características de cada recurso

físico, como também o modo como estão conectados. Sendo assim, na infraestrutura, é possível definir recursos físicos heterogêneos, aproximando ainda mais o cenário simulado do real.

Para cada alocação de VI na infraestrutura física, é cedida uma parcela dos recursos, ou seja, é subtraído de cada recurso físico o valor solicitado pela VI alocada. Além disso, para cada recurso virtual alocado fica um registro no recurso físico, para que após as alocações seja possível identificar onde uma determinada VI foi alocada. Desse modo, é possível um gerenciamento das VIs alocadas e dos recursos físicos disponíveis na infraestrutura.

Figura 15 – Exemplo de uma infraestrutura física que pode ser simulada por meio do simulador.



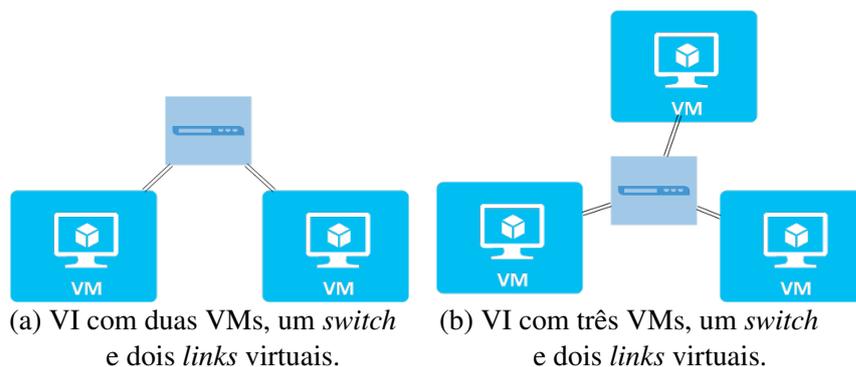
Fonte: adaptado de Soares, Aparício e Sargento (2017).

Definição da requisição de VI

De forma análoga a definição da infraestrutura física, na definição da requisição de VI também é possível descrever a quantidade de nós e links e o modo como eles estão conectados. Além disso, suas características com relação a quantidade de recursos que vão ser requisitados.

O simulador já disponibiliza duas requisições de VIs simples, elas estão exibidas nas Figuras 16a e 16b. Nesse caso, cada VM, switch e link já está configurada com uma quantidade de recursos que pode ser facilmente alterada.

Figura 16 – Requisições de VIs simples.



Fonte: elaboração própria

Para a elaboração de cenários mais próximos da realidade, o simulador também está habilitado para permitir a criação de requisições de VIs de maneira semelhante a da infraestrutura física. Desse modo, o simulador permite liberdade total para a definição de qualquer cenário de avaliação que se deseje empregar.

Criação da carga de trabalho para avaliação

Esse serviço tem como objetivo simplificar o teste de avaliação por meio da definição dos tipos e quantidades de VIs que precisam ser alocadas em uma determinada infraestrutura. A carga de trabalho pode ser definida, basicamente, como a quantidade de VIs que vão ser alocadas e quais os tipos. Além disso, existe a opção da carga ser homogênea ou heterogênea, ou seja, VIs do mesmo tipo ou de tipos diferentes.

A aleatoriedade nos tipos e quantidade de requisições de VI é um fator provavelmente comum na realidade. Sendo assim, o presente serviço também possibilita a criação de cargas de trabalho com tipos, e quantidades de requisições definidas aleatoriamente. Logo, o simulador possibilita o emprego de cargas de trabalho próximas da realidade.

Medição do desempenho por meio de métricas

Para facilitar a mensuração do desempenho, o simulador oferece como um de seus serviços a medição do desempenho por meio de métricas. Por conveniência, são as mesmas que foram selecionadas para avaliar a solução proposta pelo presente trabalho (para saber mais sobre as métricas veja a seção 4.1.3). Ademais, novas medidas de desempenho podem ser implementadas por meio da combinação dos métodos já existentes para medição de nós e links, como também pela elaboração de novos.

Considerando o que foi exposto, o simulador possibilita a avaliação do desempenho de soluções para o problema de alocação de VIs de maneira simples e organizada. No pró-

ximo capítulo é descrita a avaliação experimental, que empregou o presente simulador para a experimentação e obtenção dos seus resultados.

4

Avaliação Experimental

No presente capítulo é apresentado o método de avaliação empregado para analisar o desempenho da solução proposta. Além disso, são apresentados os resultados obtidos na análise do desempenho de quatro algoritmos desenvolvidos para alocação de VIs utilizando o modelo proposto (veja o capítulo 3).

4.1 Método de Avaliação

O Método de Avaliação utilizado nesse trabalho é baseado na abordagem sistêmica para avaliação de desempenho de Jain (1991, p. 22). Empregada com o objetivo de avaliar soluções para o problema de alocação de VIs, a abordagem facilita a descrição do método de avaliação. Além disso, simplifica a organização e acelera a obtenção dos resultados. Desse modo, o método de avaliação utilizado para avaliar os algoritmos de alocação de VIs se divide nas seguintes etapas: definição dos objetivos e escopo do sistema, descrição dos serviços e saídas, métricas selecionadas, parâmetros, fatores de estudo, técnica de avaliação, carga de trabalho, planejamento dos experimentos e interpretação dos resultados. Cada uma dessas etapas é apresentada na presente seção.

4.1.1 Definição dos objetivos e escopo do sistema

Esse método de avaliação objetiva avaliar o desempenho de algoritmos para alocação de VIs. Basicamente, o algoritmo é responsável pelo mapeamento dos recursos virtuais das VIs na infraestrutura física. Portanto, analisar a maneira como esses recursos são alocados e o tempo necessário para realizar essa tarefa é fundamental para quantificar desempenho.

Os algoritmos avaliados são MOEAs que utilizam o modelo de alocação de VIs proposto no capítulo 3, sendo eles: o GA, algoritmo proposto por esse trabalho; o *Simulated Annealing* (SA), meta-heurística utilizada por (RAUGUST *et al.*, 2018) para alocar VIs em um cenário de

múltiplos InPs; o *Hill Climbing* (HA), um método de busca de um único estado (LUKE, 2013, p. 17); e finalmente, o algoritmo *Random Allocation* (RA), que basicamente realiza a alocação de maneira aleatória.

4.1.2 Definição dos serviços e saídas

O principal e único serviço considerado do algoritmo nessa avaliação é o mapeamento dos recursos virtuais das requisições de VIs na infraestrutura física. Logo, a saída a ser considerada é se a requisição de VI foi atendida ou não, ou seja, se a VI foi alocada com ou sem sucesso. Em suma:

- Serviço: alocação de VIs;
- Saídas: VI alocada com/sem sucesso.

4.1.3 Métricas

Métricas são critérios definidos para a avaliação de desempenho. Com o objetivo de analisar e comparar diferentes soluções para o problema em questão, foram selecionadas as métricas mais utilizadas na área segundo o mapeamento sistemático realizado (veja o capítulo 2). Nos itens abaixo estão apresentadas as métricas utilizadas e alguns dos estudos que também as aplicaram:

- Tempo para atender uma requisição de VI: é dado pelo cálculo do tempo médio necessário para atender a uma requisição de VI (OLIVEIRA; KOSLOVSKI, 2017).
- Lucro do provedor: é dada pela quantidade de requisições de VIs que foram atendidas pelo provedor (SOUZA *et al.*, 2017b; OLIVEIRA; KOSLOVSKI, 2017);
- Taxa de rejeição: quantifica a porcentagem de sucesso de VIs alocadas. Dado um intervalo, ela é calculada dividindo o número de VIs não-alocadas pelo número total de requisições (CHOWDHURY; RAHMAN; BOUTABA, 2012; NESI *et al.*, 2018; SOUZA *et al.*, 2017b);
- Eficiência do uso da infraestrutura física: mensurado pelo número de recursos alocados pelo número de requisitados (ZHENG *et al.*, 2017; YANG *et al.*, 2013).

4.1.4 Definição dos Parâmetros

Parâmetros são todos os valores que determinam o comportamento do sistema. Desse modo, os principais parâmetros para se considerar em um problema de alocação de VIs estão relacionados ao algoritmo responsável pelo mapeamento e a topologia da infraestrutura física e da VI. Os parâmetros do algoritmo têm influência direta no desempenho, pois eles são os

responsáveis pela alocação. Assim sendo, os parâmetros do algoritmo dependem da forma como ele foi implementado. Por exemplo, o algoritmo descrito na seção 3.4 pode receber como parâmetros: tamanho da população inicial, número de execuções ou tempo máximo de execução, requisição de VI e a infraestrutura física onde a VI será alocada.

A topologia da infraestrutura física e das VIs também são importantes para o avaliar o desempenho do algoritmo. A topologia define a quantidade e os tipos dos recursos (*servidores*, *switches* e *links*) disponíveis (Infraestrutura Física) ou que vão ser requisitados (VI). Além disso, descreve o modo como esses recursos estão ligados (por meio dos *links*) entre si. Segundo Oliveira e Koslovski (2017), a utilização de topologias que não representam cenários reais pode enviesar os resultados quanto a taxa de aceitação das requisições. Desse modo, a definição das topologias da infraestrutura física e das requisições de VIs também é essencial para avaliar o desempenho.

Pode-se classificar os parâmetros em dois tipos: parâmetros de sistema e da carga de trabalho. Considerando o que foi apresentado até o momento, a classificação em parâmetros de sistema e carga de trabalho pode ser definida como:

- Sistema: o conjunto de parâmetros relacionados ao algoritmo e a topologia da infraestrutura física e das requisições de VIs.
- Carga de trabalho: o número x de requisições de VIs que o algoritmo vai receber para alocar na infraestrutura física.

Vale ressaltar que devido a grande quantidade de parâmetros de sistema, eles foram descritos resumidamente.

4.1.5 Fatores de estudo

Nem todos os parâmetros enumerados devem ser considerados numa avaliação, apenas os mais relevantes. Caso forem levados em consideração todos os parâmetros identificados a análise pode ser tornar complexa e difícil de ser encaminhada. Portanto, dentro dos parâmetros identificados, foram selecionados os que trazem maior impacto e facilitam a análise de desempenho.

Os parâmetros que são variados durante a análise são denominados fatores e seus valores são chamados níveis. Sendo assim, nessa seção são apresentados os fatores e os níveis empregados na avaliação relacionados ao algoritmo, infraestrutura física e requisições de VIs.

4.1.5.1 Fatores de estudo dos algoritmos

Os algoritmos avaliados por esse método são de otimização. Conseqüentemente, eles dependem de um certo tempo de execução ou número de execuções para convergir para soluções

aceitáveis. Portanto, para todos os algoritmos, foram definidos os níveis apresentados na Tabela 11:

Tabela 11 – Fatores de estudo e níveis relacionado aos algoritmos.

Fator	Níveis
Tempo máximo para atender uma requisição	1800 segundos
Número de iterações	100 mil, 150 mil e 200 mil

Fonte: elaboração própria

4.1.5.2 Fatores de estudo da topologia da infraestrutura física

Com o objetivo de avaliar o desempenho da solução proposta em um cenário mais próximo o possível do real, foi selecionada uma topologia geralmente empregada em *Data Centers*. Desse modo, seguindo os trabalhos de Gustavo, Obelheiro e Koslovski (2014) e Oliveira e Koslovski (2017) a rede física empregada nos experimentos é baseada no Modelo Cisco de três camadas (CISCO, 2013).

A topologia de *Data Center* utilizada é comumente chamada de *Fat-tree*. Ela foi simplificada para essa avaliação. Sendo assim, ela pode ser definida da seguinte forma: considerando o modelo em camadas, tem-se três, (1) camada de acesso, composta por 4 *racks*, cada um contendo 4, 8 ou 12 servidores e um *switch* de topo de *rack*; (2) camada de agregação com 4 *switches*; e (3) camada core com 2 *switches*. Portanto, os fatores e níveis definidos podem ser vistos na Tabela 12.

Tabela 12 – Fatores de estudo e níveis relacionados a topologia da infraestrutura física.

Fator	Num de Servidores por Rack
Topologia <i>Fat-tree</i>	4, 8 e 12

Fonte: elaboração própria

4.1.5.3 Fatores de estudo da topologia da requisição de VI

Derivado do parâmetro da carga de trabalho, as requisições de VIs são essenciais para avaliar o desempenho dos algoritmos com relação ao mapeamento dos recursos. Assim sendo, foram utilizados dois tipos de requisição de VIs:

- VI1: composta por 2 servidores, 1 *switch* conectados por 2 *links*.
- VI2: composta por 3 servidores, 1 *switch* conectados por 3 *links*.

Esses tipos de requisições podem ser considerados níveis do fator requisição de VI. Desse modo, o fator e os níveis podem ser observados na Tabela 13.

Tabela 13 – Fatores de estudo e níveis relacionados aos tipos de requisição de VIs.

Fator	Tipo de requisição de VI
Requisição de VI	VI1 e VI2

Fonte: elaboração própria

4.1.6 Técnica de avaliação

Devido a natureza do problema e ampla aplicação em trabalhos anteriores (veja o capítulo 2), foi empregada no presente trabalho a técnica de simulação. Para Jain (1991, p. 30), a simulação é uma boa alternativa pela flexibilidade e facilidade de alteração das configurações. Além disso, afirma que simulações podem incorporar mais detalhes e requerem menos suposições e frequentemente estão mais próximas da realidade.

4.1.7 Carga de trabalho

A carga de trabalho é definida pelo número de requisições de VI que cada algoritmo vai alocar em uma dada infraestrutura física. Desse modo, foi selecionado como carga de trabalho 50 VIs. O tipo da VI é selecionado de forma aleatória, ou seja, para cada uma das 50, é selecionado um dos tipos descritos na seção 4.1.5.3, compondo assim um conjunto de requisições heterogêneo.

4.1.8 Planejamento dos experimentos

Os experimentos foram organizados de uma forma que se obtivessem os resultados com precisão aceitável e com o mínimo de esforço possível. Sendo assim, foram elaborados três cenários para experimentação que consideram todos os fatores e níveis descritos na seção 4.1.5. Ademais, para acelerar a obtenção dos resultados todos os experimentos foram automatizados por meio da utilização de *scripts*. Na Tabela 14, apresenta-se os três cenários produzidos.

Tabela 14 – Cenários elaborados para a execução dos experimentos.

Cenário	Característica
Cenário 1	Topologia com 4 <i>racks</i> e 4 servidores por <i>rack</i>
Cenário 2	Topologia com 4 <i>racks</i> e 8 servidores por <i>rack</i>
Cenário 3	Topologia com 4 <i>racks</i> e 12 servidores por <i>rack</i>

Fonte: elaboração própria

Para cada cenário um *script* foi criado e executado em uma máquina servidora. Todos os experimentos relacionados ao tempo de execução foram executados na mesma máquina a fim de garantir equidade de recursos computacionais para cada algoritmo. Sendo assim, na Tabela

15, é apresentado todos os recursos computacionais utilizados nos experimentos, como o nome fictício da máquina, o Sistema Operacional (SO) e os recursos de hardware que cada uma possui.

Tabela 15 – Recursos computacionais utilizados.

Nome	SO	Recursos
Máquina 1	Ubuntu <i>Server</i> 18.04	2 CPUs virtuais com 10 GB de RAM.
Máquina 2	Ubuntu 18.04	Intel Core i5 com 12 GB de RAM .
Máquina 3	Ubuntu 18.04	Intel Core i5 com 8 GB de RAM.
Máquina 4	Windows 10	Intel Core i7 com 8 GB de RAM.

Fonte: elaboração própria

Dentro desses cenários, os algoritmos foram executados considerando os fatores e os níveis definidos na Tabela 11. Na Máquina 1, foi executado o *script* responsável por realizar os experimentos quanto ao tempo de execução necessário para os algoritmos atenderem um conjunto de requisições de VIs. Em paralelo, nas Máquinas 2, 3 e 4 foram executados os demais *scripts* que experimentavam o desempenho dos algoritmos considerando os cenários da Tabela 14 e os fatores e níveis descritos na seção 4.1.5.2.

Para mensurar o desempenho por meio das métricas, dois tipos de experimentos foram realizados. O primeiro objetivou medir a quantidade de tempo que cada algoritmo necessita para alocar uma VI em uma determinada infraestrutura física. Por sua vez, o segundo mediu as demais métricas recebendo 50 requisições de modo sequencial e alocando-as na infraestrutura física. Desse modo, simplificou-se os experimentos e facilitou a utilização de máquinas diferentes, acelerando assim, o processo de obtenção dos resultados.

A quantidade de tempo necessário para que cada algoritmo realize a alocação de uma VI foi medido por meio de 50 requisições utilizando os níveis do fator de estudo definido para as requisições de VI (para cada VI, um nível é selecionado aleatoriamente). Desse modo, para cada requisição, contabiliza-se o tempo necessário para a alocação por cada algoritmo. No entanto, o tempo disponível que cada algoritmo pode empregar para alocação não é infinito, ele não deve ultrapassar tempo máximo que é 1800 segundos para cada requisição. Portanto, caso algum algoritmo chegue a esse valor a alocação é considerada sem sucesso e é atribuído esse valor como resultado.

As demais métricas foram mensuradas por meio do envio de 50 requisições de VI em sequência para serem alocadas por cada algoritmo em uma dada infraestrutura física. Sendo assim, vale destacar que a seleção das requisições é análoga ao experimento para mensurar o tempo que cada algoritmo necessita para a alocação. Além disso, para garantir uma comparação justa, todos os algoritmos alocaram o mesmo tipo de VI em infraestruturas físicas idênticas. Portanto, é garantida a igualdade de condições para uma comparação justa de cada solução.

Para garantir um nível de confiança adequado nos resultados os experimentos foram

realizados com a mesma configuração dez vezes. Ademais, todos os testes e experimentos realizados por essa pesquisa utilizaram os recursos do Laboratório Experimental em Redes de Computadores (ELAN - *Experimental Laboratory in computer Networks*) da Universidade Federal de Sergipe. O ELAN possui uma infraestrutura voltada para estudos e pesquisas em Computação em Nuvem e SDN. Portanto, forneceu todas as ferramentas necessárias para o desenvolvimento do trabalho.

4.1.9 Análise e interpretação dos resultados

Os resultados foram analisados utilizando técnicas estatísticas como o intervalo de confiança e o teste de Friedman (DERRAC *et al.*, 2011). Na próxima seção, todos os resultados obtidos com a aplicação do presente método de avaliação são apresentados.

4.2 Resultados Experimentais

Os resultados apresentados na presente seção mostram médias com 95% de intervalo de confiança. Ademais, nos experimentos relacionados a métrica tempo para atender uma requisição de VI foi empregado o teste de Friedman para garantir que as amostras são diferentes estatisticamente. Desse modo, os resultados são apresentados por métrica considerando cada cenário descrito na seção 4.1.8. Além disso, dentro de cada cenário são exibidos os resultados para cada um dos fatores definidos e os seus níveis.

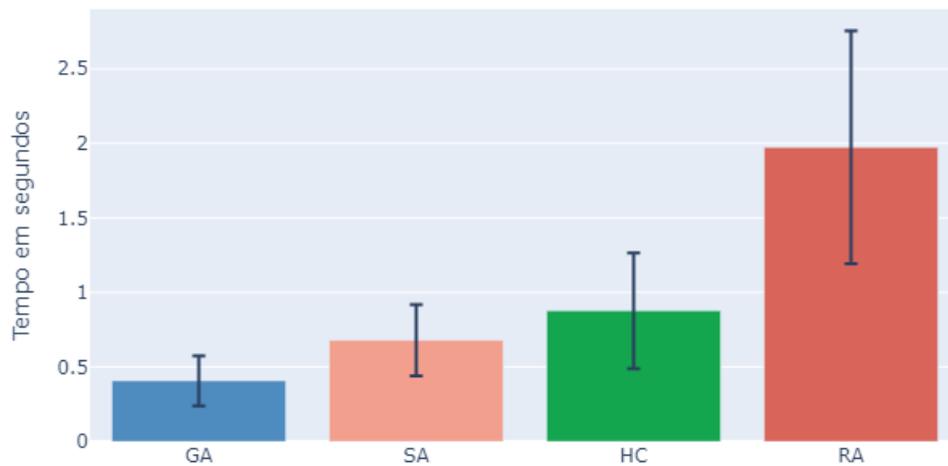
4.2.1 Cenário 1 - Topologia com 4 racks e 4 servidores por rack

Nesse cenário, a topologia da infraestrutura física é composta por 16 servidores, 10 *switches* e 22 *links*. Diante disso, nessa seção são apresentados os resultados para cada métrica de desempenho selecionada no cenário atual. Em adição, para as métricas lucro do provedor e taxa de rejeição, os resultados são apresentados para os níveis do fator número de execuções dos algoritmos.

4.2.1.1 Tempo necessário para alocação de uma requisição de VI

Na Figura 17, tem-se um gráfico apresentando o tempo médio necessário para cada algoritmo alocar uma determinada requisição de VI. Todos os algoritmos, no presente cenário, conseguiram atender a requisição de VI antes de alcançar o tempo máximo.

Figura 17 – Cenário 1: tempo para alocação de uma requisição de VI.



Fonte: elaboração própria

O tempo médio que cada algoritmo utilizou para realizar a alocação de um conjunto de requisições de VI está apresentado na Tabela 16.

Tabela 16 – Cenário 1: tempo médio utilizado por cada algoritmo para realizar uma alocação de VI.

Algoritmo	Tempo médio em segundos
GA	0,4071
SA	0,6793
HC	0,8770
RA	1,9733

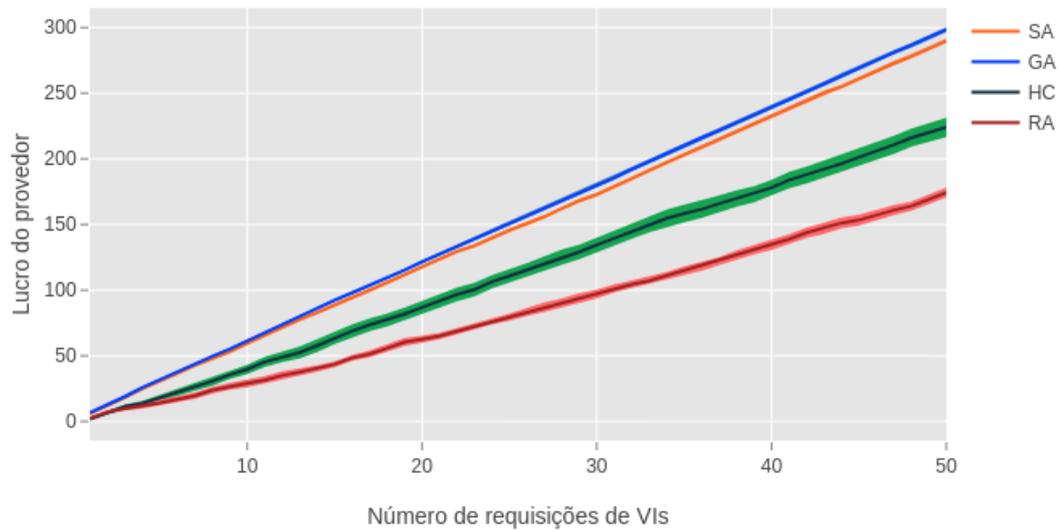
Fonte: elaboração própria

De posse dos resultados, foi empregado o teste de Friedman. O resultado do teste indicou que as amostras são diferentes estatisticamente. Desse modo, é possível inferir observando as médias que o GA é o melhor algoritmo nessas condições, pois apresenta o menor tempo utilizado para a alocação das requisições.

4.2.1.2 Lucro do provedor

A curva do crescimento do lucro do provedor de cada algoritmo para todos os níveis do fator número de iterações são exibidas nas Figuras 18, 19 e 20 respectivamente.

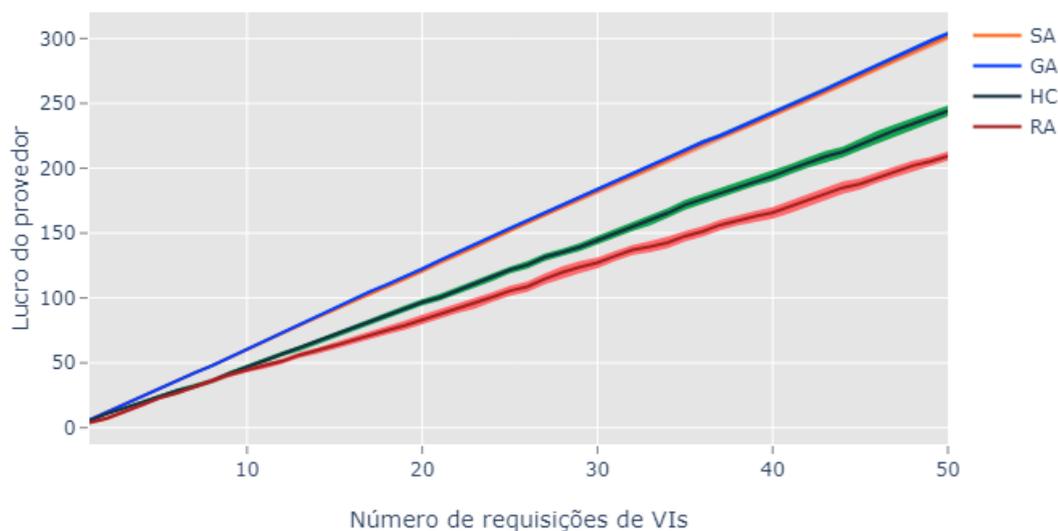
Figura 18 – Cenário 1: lucro do provedor para cada algoritmo empregando 100 mil iterações.



Fonte: elaboração própria

Para o primeiro nível, 100 mil iterações, o GA se apresenta com os melhores resultados, seguido pelo SA, HC e RA respectivamente.

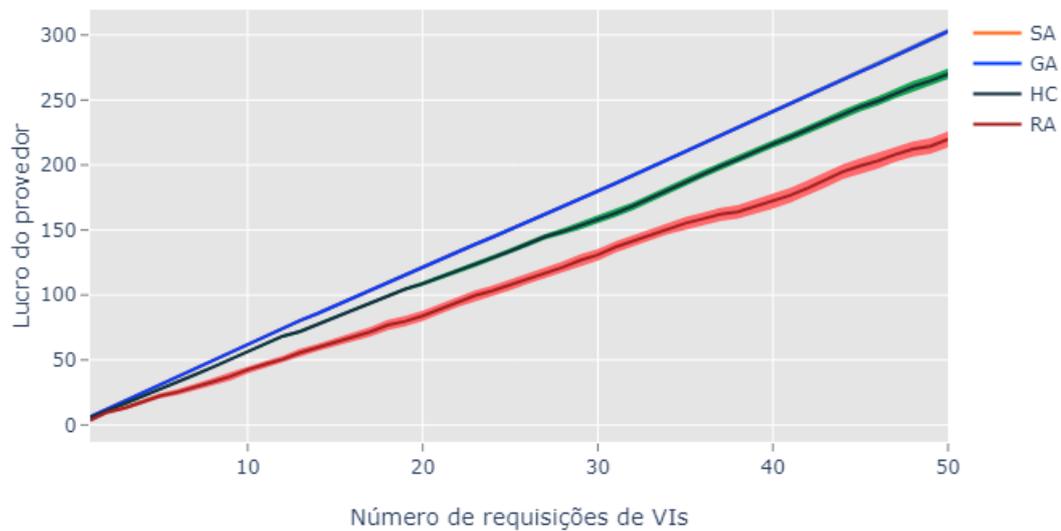
Figura 19 – Cenário 1: lucro do provedor para cada algoritmo empregando 150 mil iterações.



Fonte: elaboração própria

Na Figura 19, são exibidos os resultados considerando para cada algoritmo 150 mil iterações. Nesse caso, há uma melhora perceptível no desempenho dos algoritmos. Desse modo, o GA também é o melhor algoritmo para esse nível.

Figura 20 – Cenário 1: lucro do provedor para cada algoritmo empregando 200 mil interações.



Fonte: elaboração própria

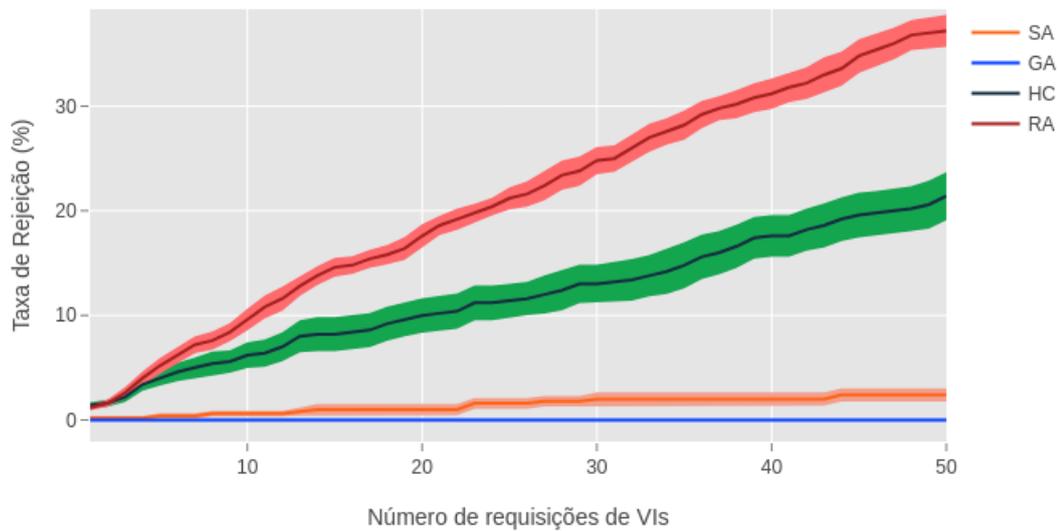
Observando a Figura 20, que apresenta os resultados dos algoritmos para o último nível, é possível inferir um ganho de desempenho do SA para essa métrica. Desse modo, considerando o intervalo de confiança, o GA e SA obtiveram desempenhos semelhantes, sendo que o GA apresenta melhores resultados com uma pequena vantagem.

Analisando o conjunto dos resultados obtidos pode-se inferir que para a métrica lucro do provedor o GA é o algoritmo que apresentou os melhores resultados. Desse modo, o GA é o melhor algoritmo nesse cenário para todos os níveis do fator número de interações do algoritmo.

4.2.1.3 Taxa de rejeição

A taxa de rejeição obtida por cada algoritmo empregando todos os níveis do fator número de interações nesse cenário é apresentada das Figuras 21, 22 e 23 respectivamente.

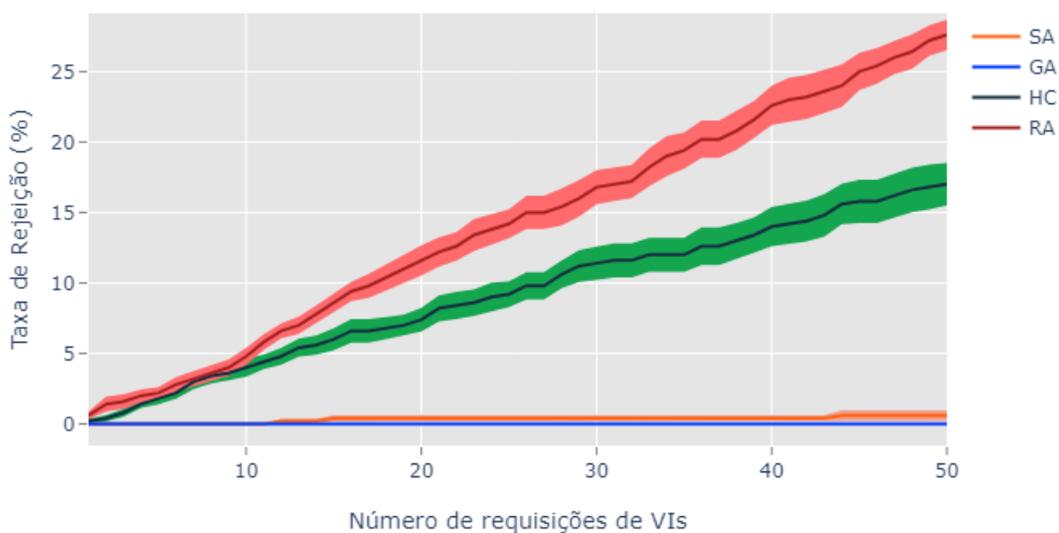
Figura 21 – Cenário 1: taxa de rejeição para cada algoritmo empregando 100 mil interações.



Fonte: elaboração própria

Observando a Figura 21, é possível observar a alta taxa de rejeição dos algoritmos HC e RA, ou seja, esses algoritmos não são efetivos para alocar VIs nesse cenário. Por outro lado, o GA obtém os melhores resultados possíveis.

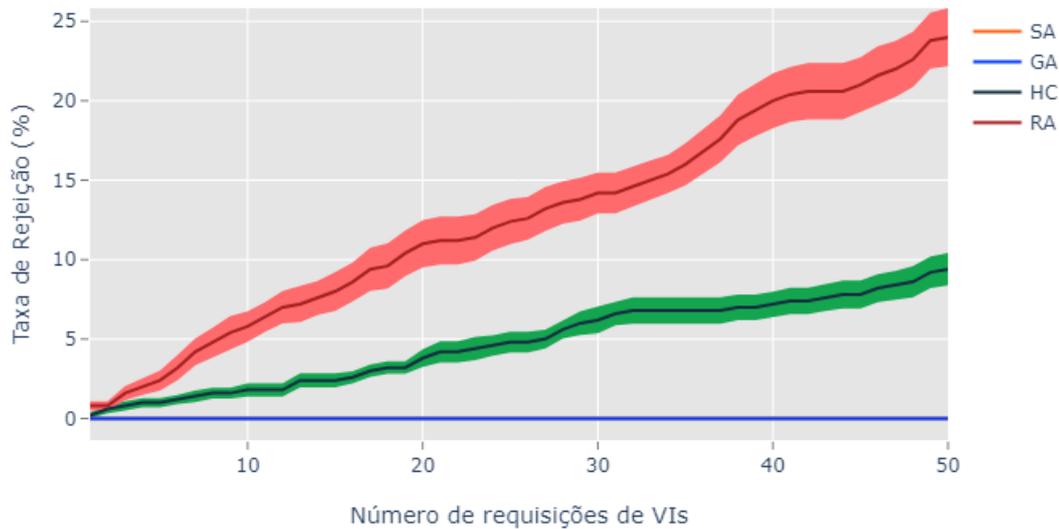
Figura 22 – Cenário 1: taxa de rejeição para cada algoritmo empregando 150 mil interações.



Fonte: elaboração própria

Analisando a Figura 22, é possível observar queda nas taxas de rejeição de todos os algoritmos. Apesar dessas melhorias, segundo os resultados apresentados, pode-se deduzir que o GA continua sendo o melhor algoritmo dentre os avaliados.

Figura 23 – Cenário 1: taxa de rejeição para cada algoritmo empregando 200 mil interações.



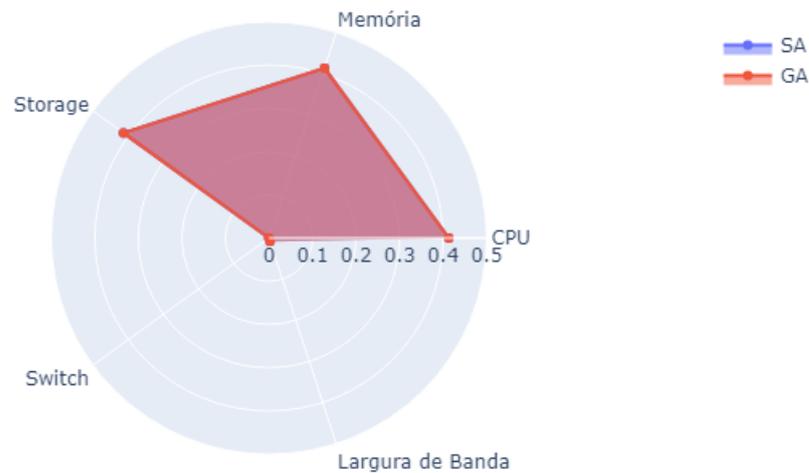
Fonte: elaboração própria

Estudando a Figura 23, fica claro a queda das taxas de rejeição de todos os algoritmos considerando esse número de interações. Desse modo, considerando todos os fatores, pode-se afirmar que o GA é o melhor algoritmo dentre os avaliados nesse cenário. Ademais, há uma relação direta entre alto lucro e baixa taxa de rejeição.

4.2.1.4 Eficiência do uso da infraestrutura física

A eficiência do uso da infraestrutura física para os algoritmos GA e SA é apresentada na Figura 24. Os demais algoritmos alocaram mais de duas vezes a quantidade de recursos de largura de banda solicitadas pelas requisições. Sendo assim, eles estão longe de possuir um desempenho adequado para serem comparados diretamente com o GA e SA nessa métrica. Consequentemente, os seus resultados foram omitidos e apenas os do GA e SA são apresentados.

Figura 24 – Cenário 1: eficiência do uso da infraestrutura física dos algoritmos GA e SA.



Fonte: elaboração própria

Observando a Figura 24, é possível observar que a eficiência do uso do substrato do GA e SA são idênticas nessas condições. Portanto, para essa métrica o desempenho do GA e SA são equivalentes.

Efetuada uma análise conjunta dos resultados obtidos em cada métrica nesse cenário pode-se afirmar que o GA foi o algoritmo que apresentou os melhores resultados para esse cenário. Em seguida veio o SA, HC e RA respectivamente.

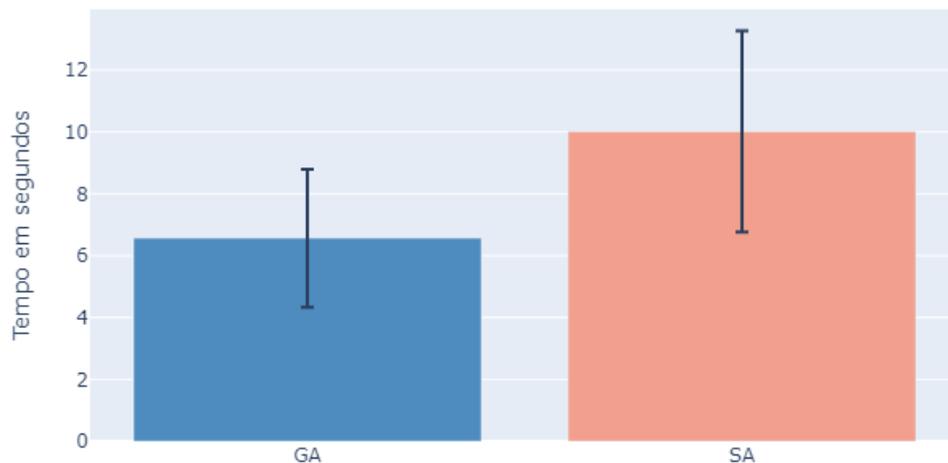
4.2.2 Cenário 2 - Topologia com 4 racks e 8 servidores por rack

No presente cenário, a topologia da infraestrutura física é composta por 32 servidores, 11 switches e 38 links. Sendo assim, na seção atual são apresentados os resultados para cada métrica de desempenho selecionada. Além disso, semelhante ao cenário anterior, as métricas: lucro do provedor e taxa de rejeição, apresentam os resultados para os níveis do fator número de execuções dos algoritmos.

4.2.2.1 Tempo necessário para alocação de uma requisição de VI

Na Figura 25, tem-se o gráfico apresentando o tempo necessário para cada algoritmo alocar uma determinada requisição de VI. O HC e o RA não conseguiram atender 14% e 18% das requisições respectivamente. Assim sendo, foram omitidos os seus resultados para uma melhor visualização dos que obtiveram desempenho superior.

Figura 25 – Cenário 2: tempo para alocação de uma requisição de VI.



Fonte: elaboração própria

Analisando a Figura 25, e empregando os testes estatísticos utilizados pelo método de avaliação, pode-se afirmar que as duas amostras são diferentes estatisticamente. Ademais, o tempo médio de alocação de cada algoritmo está exibida na Tabela 17.

Tabela 17 – Cenário 2: tempo médio utilizado por cada algoritmo para realizar uma alocação de VI.

Algoritmo	Tempo médio em segundos
GA	6,56744
SA	10,01294
HC	257,8698
RA	330,1166

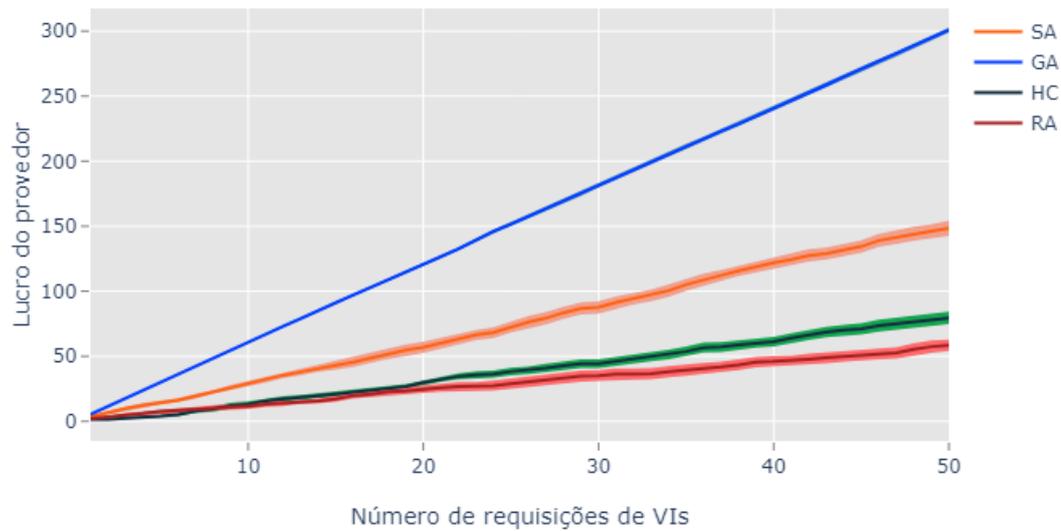
Fonte: elaboração própria

Considerando os testes estatísticos e o tempo médio para alocação de cada algoritmo, pode-se afirmar que o GA apresenta melhores resultados para esse caso específico.

4.2.2.2 Lucro do provedor

O lucro do provedor para todos os níveis do fator número de interações do algoritmo é apresentado nas Figuras 26, 27 e 28.

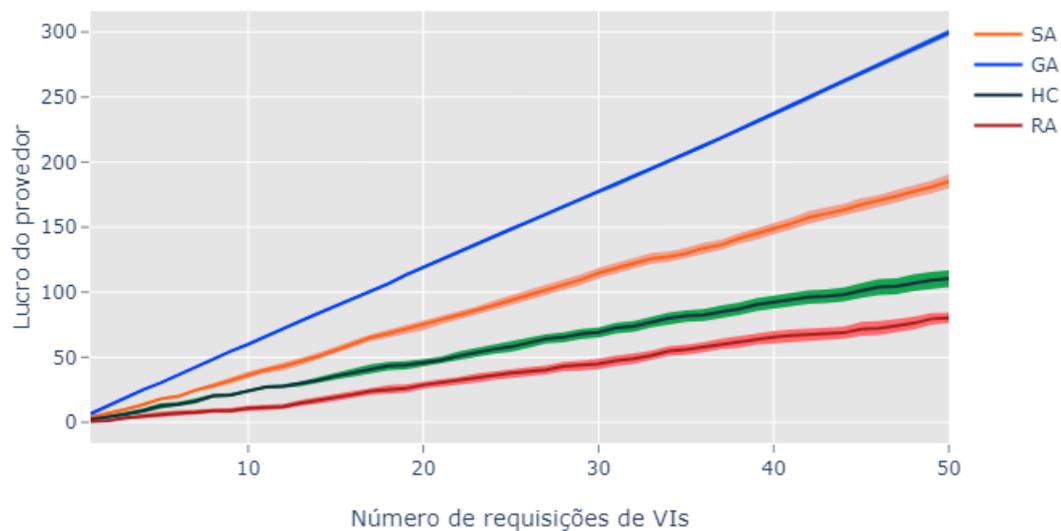
Figura 26 – Cenário 2: lucro do Provedor para cada algoritmo empregando 100 mil interações.



Fonte: elaboração própria

Observando a Figura 26, que apresenta o desempenho dos algoritmos executando 100 mil interações, chama a atenção a grande vantagem do GA sobre os outros algoritmos.

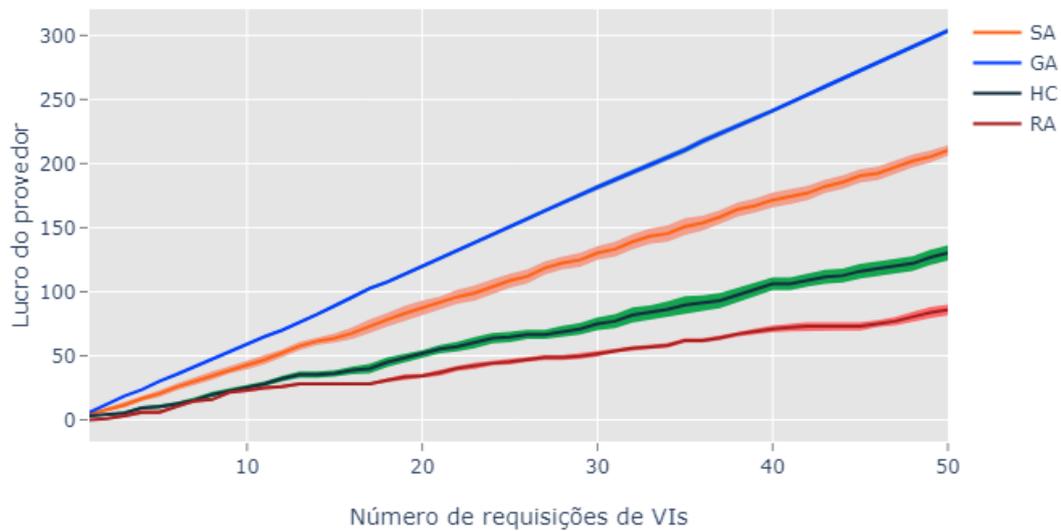
Figura 27 – Cenário 2: lucro do provedor para cada algoritmo empregando 150 mil interações.



Fonte: elaboração própria

No nível intermediário do fator número de interações, ou seja, para 150 mil interações, apresentado na Figura 27, a vantagem do GA sobre os demais algoritmos é atenuada. No entanto, ainda continua grande.

Figura 28 – Cenário 2: lucro do provedor para cada algoritmo empregando 200 mil interações.



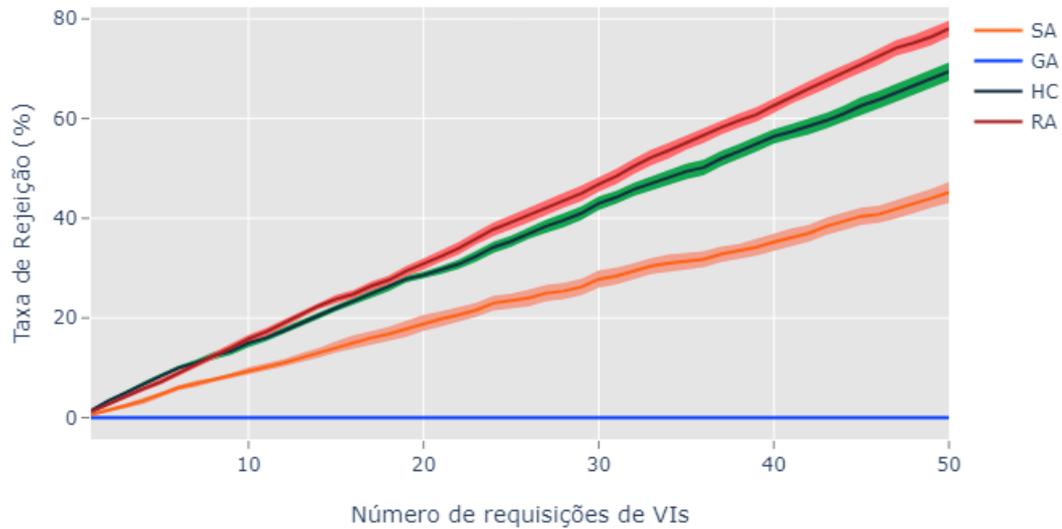
Fonte: elaboração própria

Finalmente, para esse cenário, na Figura 28, está exibido os resultados de lucro do provedor para cada algoritmo empregando 200 mil interações. Nesse resultado, é possível identificar uma maior atenuação das vantagens do GA sobre os demais algoritmos. Entretanto, o GA obtém uma larga vantagem sobre os demais. Sendo assim, o GA apresenta os melhores resultados para essa métrica.

4.2.2.3 Taxa de Rejeição

Os resultados quanto a métrica de taxa de rejeição para o presente cenário são exibidos nas Figuras 29, 30 e 31.

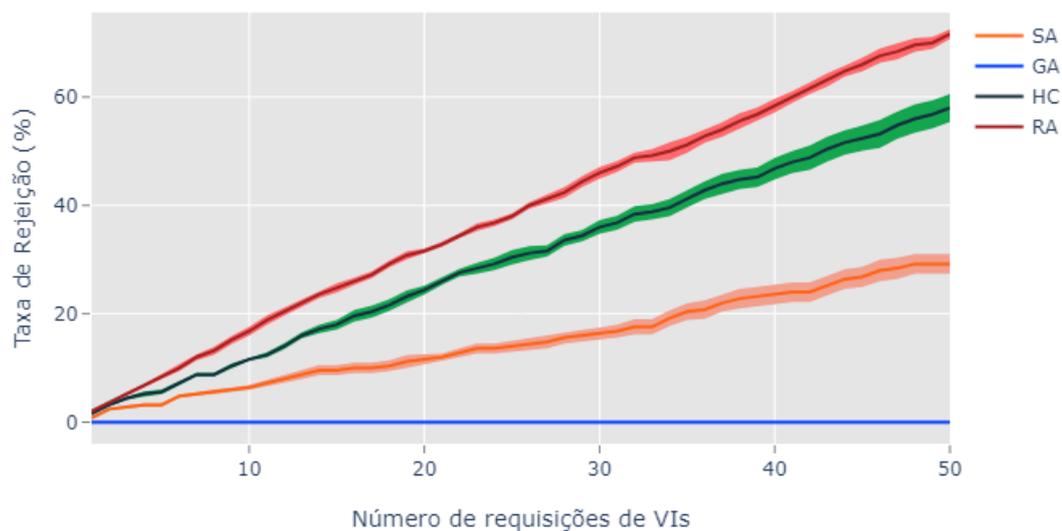
Figura 29 – Cenário 2: taxa de rejeição para cada algoritmo empregando 100 mil interações.



Fonte: elaboração própria

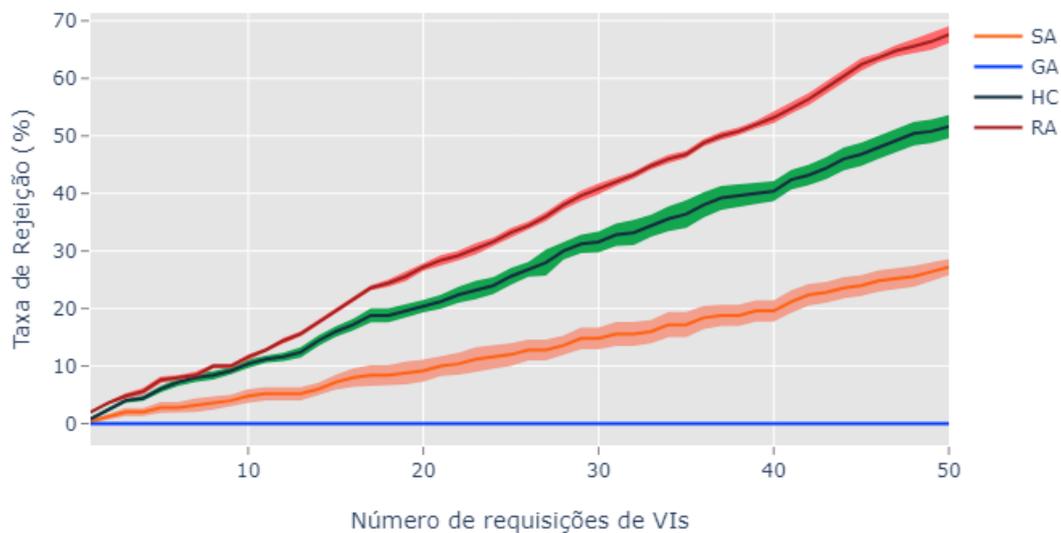
Para 100 mil interações, na Figura 29, fica perceptível a grande vantagem do GA sobre os demais algoritmos. Ademais, mesmo para 150 mil, na Figura 30, os valores continuam semelhantes aos da Figura 29.

Figura 30 – Cenário 2: taxa de rejeição para cada algoritmo empregando 150 mil interações.



Fonte: elaboração própria

Figura 31 – Cenário 2: taxa de rejeição para cada algoritmo empregando 200 mil interações.



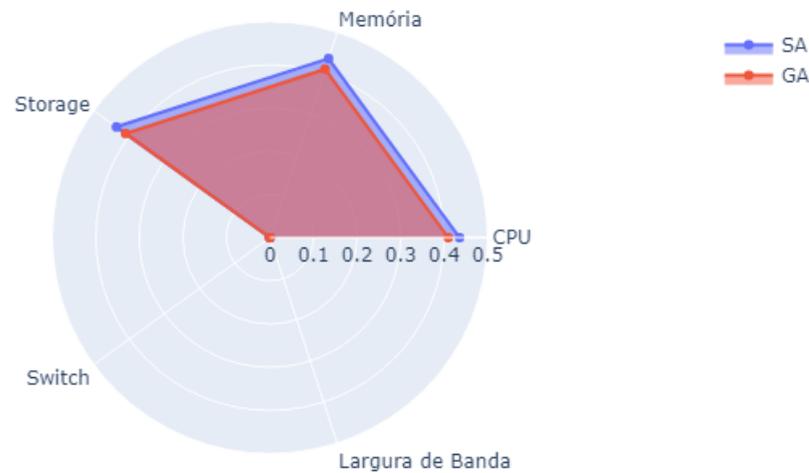
Fonte: elaboração própria

Analisando em conjunto as Figuras 29, 30 e 31, fica evidente que os valores pouco mudam mesmo com o acréscimo no número de interações do algoritmo, ou seja, um maior número de interações para encontrar soluções melhores. Sendo assim, é possível afirmar, considerando os resultados apresentados até agora para esse cenário que os algoritmos SA, HC e RA apresentam altas taxas de rejeição. Logo, essas altas taxas podem ser explicadas pela maior facilidade desses algoritmos de caírem em mínimos locais, ou seja, não conseguem encontrar melhores soluções a partir das que já possuem no momento.

4.2.2.4 Eficiência do uso da infraestrutura física

De maneira análoga ao cenário 1, os resultados apresentados para essa métrica consideraram apenas os algoritmos GA e SA. Portanto, os resultados de HC e RA foram omitidos pois eles foram muito inferiores. Logo, apenas os resultados do GA e SA são apresentados na Figura 32.

Figura 32 – Cenário 2: eficiência do uso da infraestrutura física dos algoritmos GA e SA.



Fonte: elaboração própria

Analisando os resultados, pode-se afirmar que o algoritmo GA foi mais eficiente na economia de recursos computacionais que o SA. Ademais, eles ficaram empatados nos demais tipos de recursos. Portanto, considerando todas as métricas e os fatores e seus níveis, pode-se afirmar que o GA é o melhor algoritmo para esse cenário.

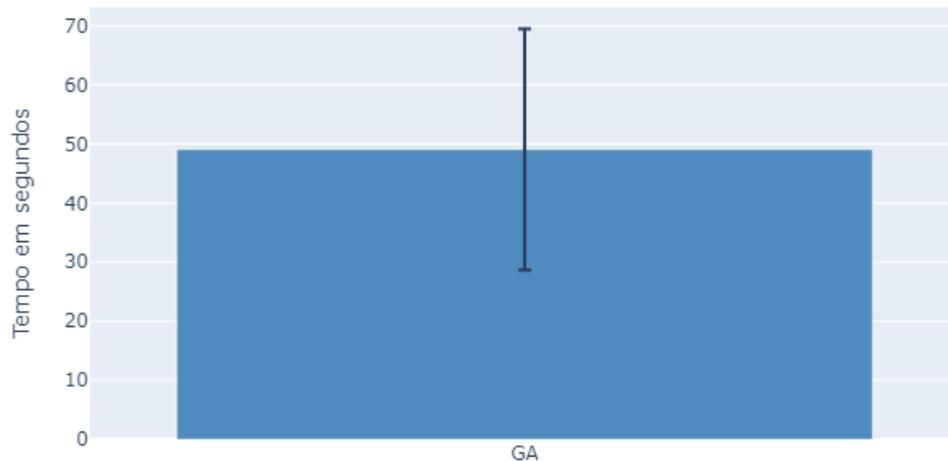
4.2.3 Cenário 3 - Topologia com 4 racks e 12 servidores por rack

No terceiro e último cenário, a topologia da infraestrutura física é composta por 48 servidores, 11 switches e 54 links. Sendo assim, na presente seção, são apresentados os resultados para cada métrica de desempenho selecionada. Além disso, para as métricas: lucro do provedor e taxa de rejeição, os resultados são apresentados para os níveis do fator número de execuções dos algoritmos.

4.2.3.1 Tempo necessário para alocação de uma requisição de VI

Os resultados relacionados ao tempo médio necessário para cada algoritmo alocar uma determinada requisição de VI nesse cenário estão apresentados na Figura 33. Os algoritmos SA, HC e RA não conseguiram atender 12%, 28% e 36% das requisições respectivamente. Desse modo, os resultados desses algoritmos foram omitidos, pois não atenderam o tempo mínimo exigido de 1800 segundos para o atendimento de uma requisição de VI.

Figura 33 – Cenário 3: tempo para alocação de uma requisição de VI.



Fonte: elaboração própria

Mesmo omitindo os algoritmos SA, HC e RA do gráfico, as médias para o atendimento de uma requisição de VI estão exibidas na Tabela 18. Assim sendo, vale ressaltar que quando um determinado algoritmo não atendia a requisição dentro do tempo máximo definido, a avaliação considerou esse tempo no cálculo da média.

Tabela 18 – Cenário 3: tempo médio utilizado por cada algoritmo para realizar uma alocação de VI.

Algoritmo	Tempo médio em segundos
GA	49,07158
SA	241,5241
HC	513,7746
RA	660,6503

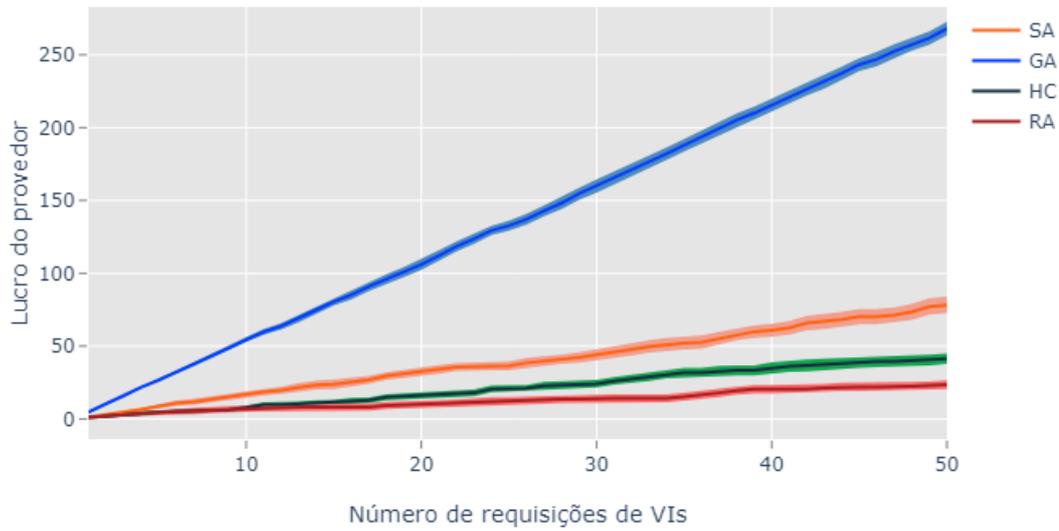
Fonte: elaboração própria

Estudando as médias, os valores dos testes estatísticos e o gráfico apresentado, pode-se afirmar com segurança que o GA é o algoritmo que atende mais rápido as requisições de VIs dentro do conjunto de algoritmos avaliados.

4.2.3.2 Lucro do provedor

Os resultados para a métrica lucro do provedor são exibidos nas figuras 34, 35 e 36.

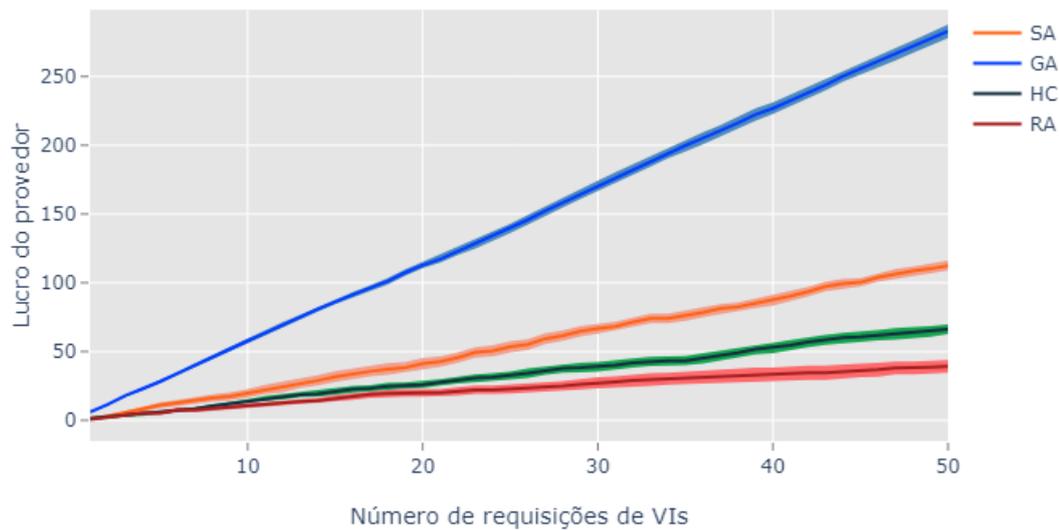
Figura 34 – Cenário 3: lucro do provedor para cada algoritmo empregando 100 mil interações.



Fonte: elaboração própria

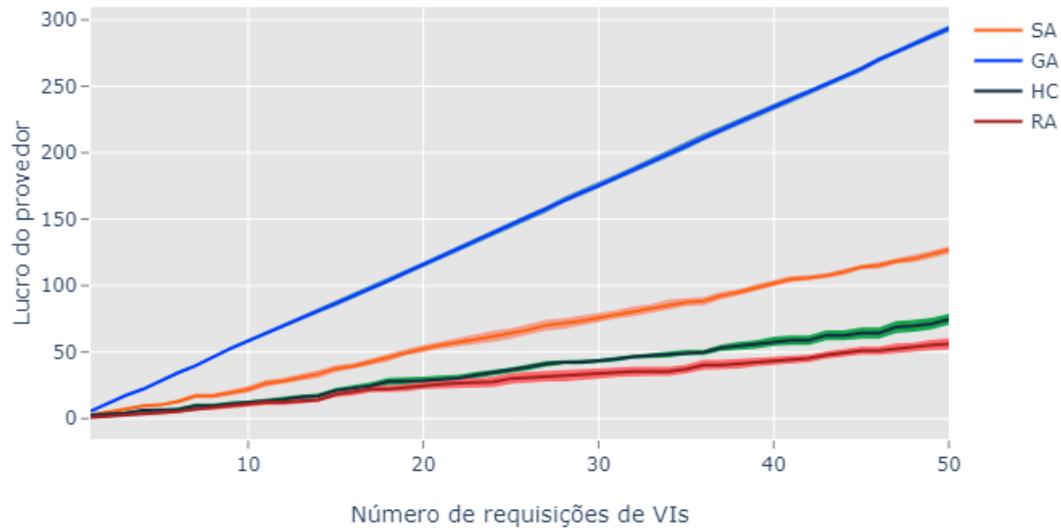
Observando os resultados das Figuras 34 e 35, fica claro que quando se aumenta o número de possíveis mapeamentos o GA obtém uma larga vantagem sobre os demais algoritmos.

Figura 35 – Cenário 3: lucro do provedor para cada algoritmo empregando 150 mil interações.



Fonte: elaboração própria

Figura 36 – Cenário 3: lucro do provedor para cada algoritmo empregando 200 mil interações.



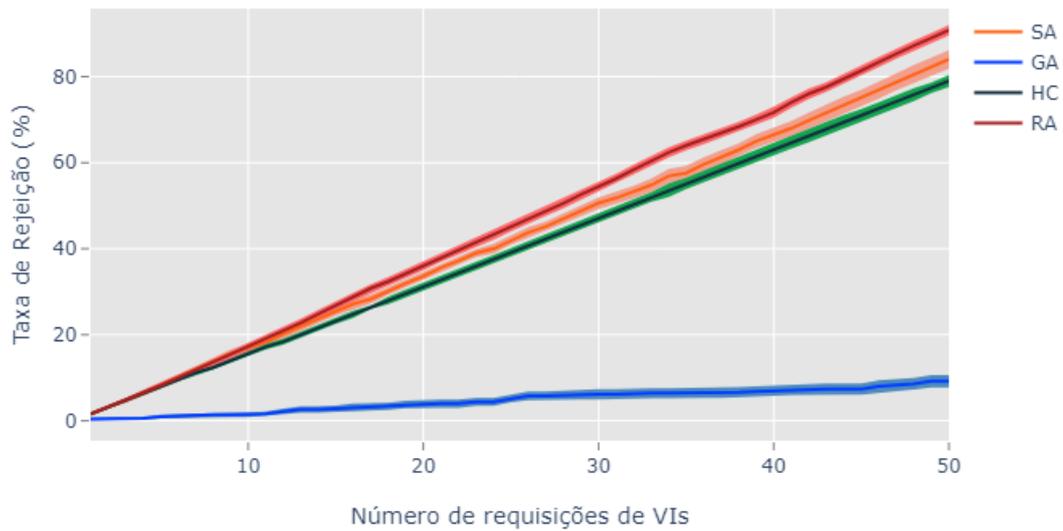
Fonte: elaboração própria

Analisando a Figura 36, onde são apresentados os resultados para o maior número de interações nesse cenário, a tendência apresentada nas Figuras 34 e 35 se mantém. Logo, pode-se afirmar que o GA é o algoritmo que apresenta os melhores resultados. Em adição, para cenários mais complexos é necessário o emprego de algoritmos mais sofisticados para realizar os mapeamentos.

4.2.3.3 Taxa de rejeição

Para o presente cenário, as taxas de rejeição para os níveis do fator número de interações dos algoritmos são apresentadas nas Figuras 37, 38 e 39.

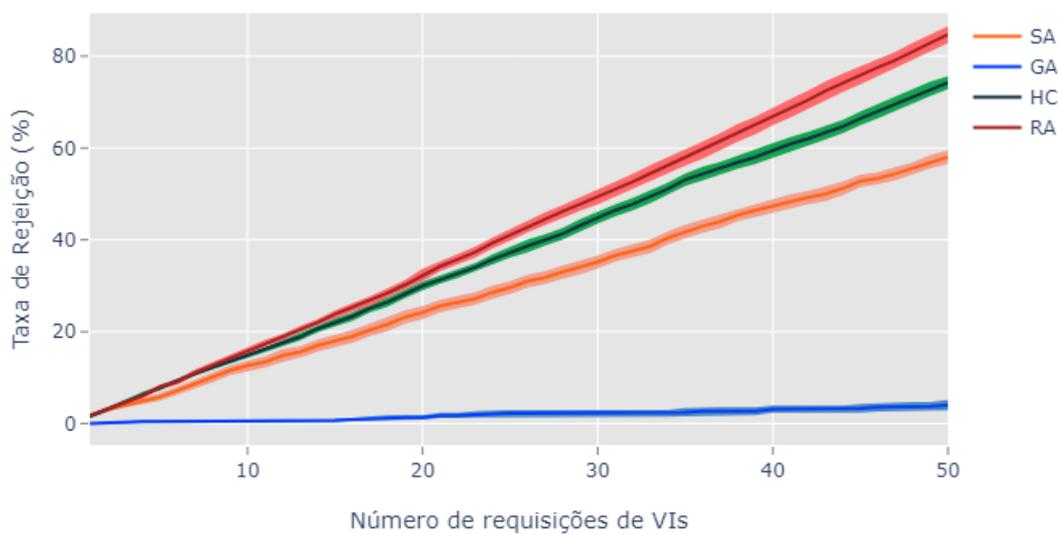
Figura 37 – Cenário 3: taxa de rejeição para cada algoritmo empregando 100 mil interações.



Fonte: elaboração própria

Observando a Figura 37, chama a atenção a alta taxa de rejeição dos algoritmos SA, HC e RA. Além disso, pela primeira vez, o SA apresenta resultados piores que o RA para essa métrica considerando os outros cenários.

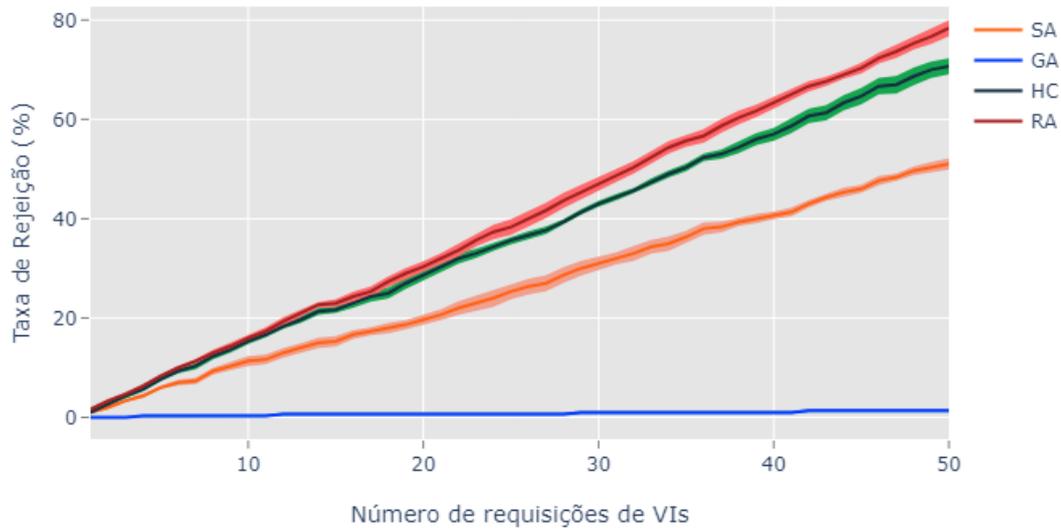
Figura 38 – Cenário 3: taxa de rejeição para cada algoritmo empregando 150 mil interações.



Fonte: elaboração própria

Estudando as Figuras 38 e 39, fica evidente a melhoria das taxas de rejeição do SA. Além disso, o GA que já apresentava os melhores resultados, melhora as suas taxas para números próximos de zero.

Figura 39 – Cenário 3: taxa de rejeição para cada algoritmo empregando 200 mil interações.



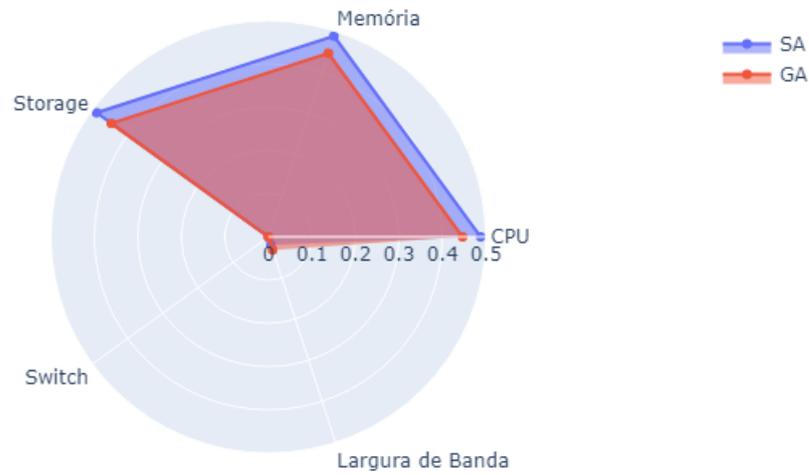
Fonte: elaboração própria

Analisando o conjunto de resultados desse cenário pode-se afirmar que o GA é o melhor algoritmo para a presente métrica. Isso se justifica novamente pelo grande número de possíveis mapeamentos que esse cenário apresenta. Sendo assim, o GA se mostrou com maior capacidade de convergir para melhores soluções, conseqüentemente, ele apresenta melhores resultados que os demais algoritmos.

4.2.3.4 Eficiência do uso da infraestrutura física

Semelhante ao resultado apresentado no cenário 1, os resultados apresentados na Figura 40, para essa métrica, consideram apenas os algoritmos GA e SA. Assim sendo, os demais foram omitidos dessa exibição e foram desconsiderados pois alocaram para alguns dos recursos mais de 200% da quantidade requisitada. Logo, estão longe de serem considerados por InPs como possíveis soluções para o problema de alocação de VIs.

Figura 40 – Cenário 3: eficiência do uso da infraestrutura física dos algoritmos GA e SA.



Fonte: elaboração própria

Analisando os resultados apresentados, pode-se afirmar que o GA é o algoritmo mais eficiente na utilização da infraestrutura física. Além disso, considerando as métricas e os cenários anteriores, pode-se dizer com segurança que o GA é o melhor algoritmo dentre o conjunto avaliado para todos os cenários analisados por esse estudo.

5

Conclusão

O presente capítulo conclui o trabalho dissertando sobre as considerações finais, as principais contribuições e os trabalhos futuros.

5.1 Considerações Finais

Para atender os requisitos dos usuários, os InPs começaram a oferecer VIs. Basicamente, uma VI é composta por VMs interconectadas por recursos de rede e sua configuração pode ser ajustada de acordo com requisitos pré-definidos. Desse modo, é possível para os InPs fornecer ao usuário solicitante controle total sobre a agregação dos recursos de TI e de rede, mantendo o usuário a parte do gerenciamento e manutenção dos recursos físicos, cabendo a ele apenas o gerenciamento dos recursos virtuais.

Dentre as tarefas necessárias para oferecer VIs como serviço para os usuários, a mais importante é a alocação do conjunto de VIs solicitadas na infraestrutura física. Esse processo consiste, basicamente, em identificar dentro da infraestrutura um conjunto de recursos para hospedar componentes da VI provisionada respeitando os requisitos de rede e computação pré-definidos. Além disso, para um gerenciamento efetivo da infraestrutura, o balanceamento da carga e a redução do *overhead* de alocação são essenciais.

No processo de alocação, alguns objetivos dos InPs e usuários devem ser atendidos. Geralmente guiados por suas perspectivas financeiras, os InPs desejam maximizar a sua receita alocando o maior número de VIs usando a menor infraestrutura possível. Por outro lado, os usuários desejam majoritariamente VIs eficientes e de baixo custo. Sendo assim, a complexidade do processo de alocação não é trivial, pois deve-se mapear os recursos atendendo os requisitos. E além disso, atender objetivos do provedor e usuário que em determinadas situações podem entrar em conflito.

Para resolver o problema de alocação, o presente trabalho apresentou um MOEA para

alocar VIs em uma infraestrutura física, atendendo os requisitos de computação e rede, avaliando soluções que atendam objetivos como: balanceamento de carga e baixo *overhead* de alocação. Sendo assim, para avaliar o algoritmo foi desenvolvido um simulador e conseqüentemente, empregada a técnica de simulação para avaliar o desempenho da solução proposta.

É importante ressaltar que a presente dissertação inova na alocação de VIs, apresentando uma solução que emprega uma combinação de técnicas de otimização multiobjetivo que atenda os objetivos de balanceamento de carga e *overhead* de alocação, com algoritmos evolucionários, considerando os desafios indicados pela literatura. Além disso, supera algumas limitações de trabalhos anteriores como a utilização de topologias aleatórias na avaliação da solução e a necessidade de várias etapas de pré-processamento para a execução do algoritmo de alocação.

5.2 Principais Contribuições

Resumidamente, as principais contribuições desse trabalho são descritas abaixo:

- Um Simulador para auxiliar na avaliação de novas soluções para o problema de alocação de VIs.
- Um modelo para alocação de VIs utilizando MOEAs.
- Um MOEA baseado no GA para alocar VIs.

Por meio da avaliação experimental, pode-se afirmar que as contribuições propostas foram alcançadas. Os resultados experimentais demonstraram que o MOEA baseado no GA proposto apresentou resultados promissores para diversos cenários, combinando rapidez e eficiência no processo de alocação de VIs. Por sua vez, o modelo de alocação proposto é útil para o mapeamento dos recursos virtuais das VIs para os recursos físicos das infraestruturas. Além disso, o simulador atendeu o seu papel de: i) simular os mapeamentos; ii) facilitar a criação e aplicação das cargas de trabalho selecionadas; e iii) calcular as métricas de desempenho considerando as requisições e a infraestrutura física simulada.

5.3 Trabalhos Futuros

Os resultados promissores apresentados abrem caminho para os seguintes trabalhos futuros:

- Aplicar o algoritmo em conjunto com modelo proposto para alocar VIs considerando múltiplos provedores (RAUGUST *et al.*, 2018);

- Segundo [Oliveira e Koslovski \(2017\)](#), os usuários buscam respostas imediatas para as suas requisições. Sendo assim, pode ser interessante acelerar o tempo de resposta por meio da aplicação de técnicas de programação paralela ([LUKE, 2013](#), p. 99).
- Formalmente, o problema de alocar VIs sobre infraestruturas físicas pode ser descrito como uma extensão do problema VNE ([OLIVEIRA; KOSLOVSKI, 2017](#)). Portanto, o simulador pode ser adaptado para avaliar o desempenho de soluções para VNE.

Referências

- ANHALT, F.; KOSLOVSKI, G.; PRIMET, P. V.-B. Specifying and provisioning virtual infrastructures with hipernet. *International Journal of Network Management*, Wiley Online Library, v. 20, n. 3, p. 129–148, 2010. Citado 2 vezes nas páginas 17 e 61.
- AYOUBI, S. *et al.* Minted: Multicast virtual network embedding in cloud data centers with delay constraints. *IEEE Transactions on Communications*, IEEE, v. 63, n. 4, p. 1291–1305, 2015. Citado 3 vezes nas páginas 35, 42 e 44.
- BARHAM, P. *et al.* Xen and the art of virtualization. In: ACM. *ACM SIGOPS operating systems review*. [S.l.], 2003. v. 37, n. 5, p. 164–177. Citado na página 16.
- BIANCHI, F.; PRESTI, F. L. A markov reward model based greedy heuristic for the virtual network embedding problem. In: IEEE. *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. [S.l.], 2016. p. 373–378. Citado 4 vezes nas páginas 35, 42, 43 e 46.
- BUYYA, R.; YEO, C. S.; VENUGOPAL, S. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In: IEEE. *2008 10th IEEE international conference on high performance computing and communications*. [S.l.], 2008. p. 5–13. Citado na página 16.
- CALHEIROS, R. N. *et al.* Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*, Wiley Online Library, v. 41, n. 1, p. 23–50, 2011. Citado na página 46.
- CARAPINHA, J.; JIMÉNEZ, J. Network virtualization: a view from the bottom. In: ACM. *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*. [S.l.], 2009. p. 73–80. Citado na página 60.
- CARVALHO, A. B. d. *Novas estratégias para otimização por nuvem de partículas aplicadas a problemas com muitos objetivos*. Tese (Doutorado) — Universidade Federal do Paraná, 2013. Citado 2 vezes nas páginas 19 e 48.
- CHENG, X. *et al.* Virtual network embedding through topology-aware node ranking. *ACM SIGCOMM Computer Communication Review*, ACM, v. 41, n. 2, p. 38–47, 2011. Citado 4 vezes nas páginas 35, 42, 44 e 45.
- CHENG, X. *et al.* Virtual network embedding through topology awareness and optimization. *Computer Networks*, Elsevier, v. 56, n. 6, p. 1797–1813, 2012. Citado 4 vezes nas páginas 35, 42, 44 e 46.
- CHOWDHURY, M.; RAHMAN, M. R.; BOUTABA, R. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. *IEEE/ACM Transactions on Networking (TON)*, IEEE Press, v. 20, n. 1, p. 206–219, 2012. Citado 11 vezes nas páginas 16, 21, 23, 26, 36, 41, 42, 43, 44, 46 e 66.
- CHOWDHURY, N. M. K.; BOUTABA, R. Network virtualization: state of the art and research challenges. *IEEE Communications magazine*, IEEE, v. 47, n. 7, p. 20–26, 2009. Citado 2 vezes nas páginas 17 e 19.

- CISCO. *Cisco Data Center Infrastructure 2.5 Design Guide*. 2013. Disponível em: <https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCI_SRND_2_5a_book.html>. Citado 2 vezes nas páginas 45 e 68.
- COELLO, C. A. C. *et al.* *Evolutionary algorithms for solving multi-objective problems*. [S.l.]: Springer, 2007. v. 5. Citado 2 vezes nas páginas 19 e 49.
- DEB, K. *et al.* A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: SPRINGER. *International conference on parallel problem solving from nature*. [S.l.], 2000. p. 849–858. Citado na página 49.
- DERRAC, J. *et al.* A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, Elsevier, v. 1, n. 1, p. 3–18, 2011. Citado na página 71.
- FISCHER, A. *et al.* Virtual network embedding: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 15, n. 4, p. 1888–1906, 2013. Citado 3 vezes nas páginas 19, 21 e 27.
- GHOMI, E. J.; RAHMANI, A. M.; QADER, N. N. Load-balancing algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, Elsevier, v. 88, p. 50–71, 2017. Citado na página 18.
- GUAN, X. *et al.* Ant colony optimization based energy efficient virtual network embedding. In: IEEE. *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*. [S.l.], 2015. p. 273–278. Citado 2 vezes nas páginas 35 e 44.
- GUSTAVO, A. D. S.; OBELHEIRO, R. R.; KOSLOVSKI, G. Optimal resource allocation for survivable virtual infrastructures. In: IEEE. *Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on the*. [S.l.], 2014. p. 1–8. Citado 6 vezes nas páginas 18, 22, 24, 26, 34 e 68.
- HAERI, S.; TRAJKOVIĆ, L. Vne-sim: a virtual network embedding simulator. In: ICST (INSTITUTE FOR COMPUTER SCIENCES, SOCIAL-INFORMATICS AND *Proceedings of the 9th EAI International Conference on Simulation Tools and Techniques*. [S.l.], 2016. p. 112–117. Citado na página 46.
- HOUIDI, I.; LOUATI, W.; ZEGHLACHE, D. Exact multi-objective virtual network embedding in cloud environments. *The Computer Journal*, Oxford University Press, v. 58, n. 3, p. 403–415, 2015. Citado 5 vezes nas páginas 21, 26, 35, 42 e 44.
- HOUIDI, I.; ZEGHLACHE, D. Exact adaptive virtual network embedding in cloud environments. In: IEEE. *2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. [S.l.], 2013. p. 319–323. Citado 3 vezes nas páginas 35, 42 e 44.
- JAIN, R. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. [S.l.]: John Wiley & Sons, 1991. Citado 3 vezes nas páginas 45, 65 e 69.
- KARMOSHI, S. *et al.* Vne-greedy: Virtual network embedding algorithm based on openstack cloud computing platform. In: IEEE. *2016 6th International Conference on Digital Home (ICDH)*. [S.l.], 2016. p. 143–149. Citado 3 vezes nas páginas 35, 42 e 43.

KOSLOVSKI, G. *et al.* Locating virtual infrastructures: Users and inp perspectives. In: IEEE. *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*. [S.l.], 2011. p. 153–160. Citado 6 vezes nas páginas 18, 22, 24, 26, 35 e 43.

KOSLOVSKI, G. *et al.* Reliability support in virtual infrastructures. In: IEEE. *2010 IEEE Second International Conference on Cloud Computing Technology and Science*. [S.l.], 2010. p. 49–58. Citado na página 17.

LI, S.; SAIDI, M. Y.; CHEN, K. A cloud-oriented algorithm for virtual network embedding over multi-domain. In: IEEE. *2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops)*. [S.l.], 2016. p. 50–57. Citado 3 vezes nas páginas 35, 42 e 43.

LUIZELLI, M. C. *et al.* Caracterizando o impacto de topologias no mapeamento de redes virtuais. *Anais do XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 75–88, 2013. Citado 2 vezes nas páginas 25 e 45.

LUKE, S. *Essentials of Metaheuristics*. second. [S.l.]: Lulu, 2013. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>. Citado 6 vezes nas páginas 45, 48, 53, 59, 66 e 92.

MAIYAMA, K. M. *et al.* Performance modelling and analysis of an openstack iaas cloud computing platform. In: IEEE. *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*. [S.l.], 2017. p. 198–205. Citado na página 16.

MANVI, S. S.; SHYAM, G. K. Resource management for infrastructure as a service (iaas) in cloud computing: A survey. *Journal of network and computer applications*, Elsevier, v. 41, p. 424–440, 2014. Citado 2 vezes nas páginas 19 e 61.

MAO, Y. *et al.* Sharing based virtual network embedding algorithm with dynamic resource block generation. *IEEE Communications Letters*, IEEE, v. 19, n. 12, p. 2126–2129, 2015. Citado 3 vezes nas páginas 35, 42 e 44.

MATLAB. 2019. Disponível em: <<https://www.mathworks.com/products/matlab.html>>. Citado na página 46.

MELL, P.; GRANCE, T. *et al.* The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National . . . , 2011. Citado 2 vezes nas páginas 16 e 27.

NESI, L. L. *et al.* Gpu-accelerated algorithms for allocating virtual infrastructure in cloud data centers. In: IEEE. *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. [S.l.], 2018. p. 364–365. Citado 5 vezes nas páginas 19, 34, 41, 43 e 66.

OLIVEIRA, R. de; KOSLOVSKI, G. P. A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements. *International Journal of Network Management*, Wiley Online Library, v. 27, n. 1, p. e1958, 2017. Citado 13 vezes nas páginas 17, 21, 24, 25, 26, 27, 35, 41, 43, 66, 67, 68 e 92.

PATHAK, I.; VIDYARTHI, D. P. A model for virtual network embedding across multiple infrastructure providers using genetic algorithm. *Science China Information Sciences*, Springer, v. 60, n. 4, p. 040308, 2017. Citado 8 vezes nas páginas 22, 26, 36, 42, 44, 46, 49 e 51.

PETERSEN, K. *et al.* Systematic mapping studies in software engineering. In: *Ease*. [S.l.: s.n.], 2008. v. 8, p. 68–77. Citado na página 28.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, Elsevier, v. 64, p. 1–18, 2015. Citado na página 28.

QING, S.; FENG, M.; JIN, H. Distributed virtual network embedding with bloom filter. In: IEEE. *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*. [S.l.], 2015. p. 1073–1078. Citado 4 vezes nas páginas 35, 42, 43 e 46.

RAUGUST, A. S. *et al.* Allocation of virtual infrastructures on multiple iaas providers with survivability and reliability requirements. In: IEEE. *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. [S.l.], 2018. p. 1147–1154. Citado 7 vezes nas páginas 24, 25, 26, 35, 43, 65 e 91.

RUCK, D. B.; OLIVEIRA, R. de; KOSLOVSKI, G. P. Comparação de algoritmos para alocação de infraestruturas virtuais. *Revista Brasileira de Computação Aplicada*, v. 6, n. 2, p. 98–112, 2014. Citado na página 18.

SAHOO, J.; MOHAPATRA, S.; LATH, R. Virtualization: A survey on concepts, taxonomy and associated security issues. In: IEEE. *2010 Second International Conference on Computer and Network Technology*. [S.l.], 2010. p. 222–226. Citado na página 16.

SHEN, M. *et al.* Availability-aware virtual network embedding for multi-tier applications in cloud networks. In: IEEE. *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on CyberSpace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*. [S.l.], 2015. p. 1–6. Citado 3 vezes nas páginas 35, 42 e 44.

SHERWOOD, R. *et al.* Flowvisor: A network virtualization layer. *OpenFlow Switch Consortium, Tech. Rep*, v. 1, p. 132, 2009. Citado na página 17.

SHI, Y. *et al.* Particle swarm optimization: developments, applications and resources. In: IEEE. *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*. [S.l.], 2001. v. 1, p. 81–86. Citado na página 49.

SOARES, J.; APARÍCIO, J.; SARGENTO, S. Dynamic strategies for the optimal embedding of virtual infrastructures. In: IEEE. *2017 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.], 2017. p. 272–277. Citado 7 vezes nas páginas 23, 26, 34, 41, 43, 59 e 62.

SOUALAH, O. *et al.* Pr-vne: Preventive reliable virtual network embedding algorithm in cloud's network. In: IEEE. *2013 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2013. p. 1303–1309. Citado 3 vezes nas páginas 35, 42 e 44.

SOUALAH, O. *et al.* A reliable virtual network embedding algorithm based on game theory within cloud's backbone. In: IEEE. *2014 IEEE International Conference on Communications (ICC)*. [S.l.], 2014. p. 2975–2981. Citado 4 vezes nas páginas 35, 42, 43 e 46.

SOUALAH, O. *et al.* A batch approach for a survivable virtual network embedding based on monte-carlo tree search. In: IEEE. *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. [S.l.], 2015. p. 36–43. Citado 3 vezes nas páginas 35, 42 e 43.

SOUALAH, O. *et al.* A novel virtual network embedding scheme based on gomory-hu tree within cloud's backbone. In: IEEE. *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. [S.l.], 2016. p. 536–542. Citado 3 vezes nas páginas 35, 42 e 44.

SOUZA, F. R. de *et al.* Alocação de infraestruturas virtuais em data centers implementados com redes definidas por software. 2017. Citado na página 19.

SOUZA, F. R. de *et al.* Qos-aware virtual infrastructures allocation on sdn-based clouds. In: IEEE. *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. [S.l.], 2017. p. 120–129. Citado 9 vezes nas páginas 17, 18, 19, 24, 26, 35, 41, 43 e 66.

SPEARS, W. M.; JONG, K. A. D. An analysis of multi-point crossover. In: *Foundations of genetic algorithms*. [S.l.]: Elsevier, 1991. v. 1, p. 301–315. Citado na página 53.

WANG, C. *et al.* Virtual network embedding with pre-transformation and incentive convergence mechanism. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 29, n. 14, p. e3947, 2017. Citado 3 vezes nas páginas 36, 44 e 46.

WEN, K.; QIAN, Z.; LU, S. Breaking the atomicity of virtual network embedding. In: IEEE. *2013 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2013. p. 2307–2312. Citado 3 vezes nas páginas 35, 42 e 44.

WHITLEY, D. A genetic algorithm tutorial. *Statistics and computing*, Springer, v. 4, n. 2, p. 65–85, 1994. Citado na página 49.

XU, M.; TIAN, W.; BUYYA, R. A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 29, n. 12, p. e4123, 2017. Citado na página 18.

XU, Z.; LIANG, W.; XIA, Q. Efficient embedding of virtual networks to distributed clouds via exploring periodic resource demands. *IEEE Transactions on Cloud Computing*, IEEE, v. 6, n. 3, p. 694–707, 2016. Citado 2 vezes nas páginas 35 e 44.

YANG, Y. *et al.* Effective and efficient ai-based approaches to cloud resource provisioning. In: IEEE. *2013 19th IEEE International Conference on Networks (ICON)*. [S.l.], 2013. p. 1–6. Citado 4 vezes nas páginas 35, 42, 44 e 66.

YU, R.; XUE, G.; ZHANG, X. Towards min-cost virtual infrastructure embedding. In: IEEE. *2015 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2015. p. 1–6. Citado 5 vezes nas páginas 16, 23, 26, 35 e 43.

ZEGURA, E. W.; CALVERT, K. L.; BHATTACHARJEE, S. How to model an internetwork. In: IEEE. *Proceedings of IEEE INFOCOM'96. Conference on Computer Communications*. [S.l.], 1996. v. 2, p. 594–602. Citado na página 45.

ZHENG, H.-K. *et al.* Link mapping-oriented ant colony system for virtual network embedding. In: IEEE. *2017 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2017. p. 1223–1230. Citado 4 vezes nas páginas 35, 42, 44 e 66.