

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Uma Abordagem Evolucionária para o
Posicionamento Dinâmico de Máquinas Virtuais em
Datacenters Privados, Aplicando Sobrecarga de
Recursos

Petrônio Carlos Bezerra

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Redes de Computadores

Dr. Reinaldo César de Moraes Gomes

(Orientador)

Campina Grande, Paraíba, Brasil

©Petrônio Carlos Bezerra, 28/06/2019

B574a

Bezerra, Petrônio Carlos.

Uma abordagem evolucionária para o posicionamento dinâmico de máquinas virtuais em *datacenters* privados, aplicando sobrecarga de recursos / Petrônio Carlos Bezerra. – Campina Grande, 2019.

248 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2019.

"Orientação: Prof. Dr. Reinaldo César de Moraes Gomes".

Referências.

1. Redes de Computadores.
2. Computação em Nuvem.
3. Virtualização.
4. Algoritmos Meméticos.
5. Sobrecarga de Recursos.
6. Fronteira de Pareto. I. Gomes, Reinaldo César de Moraes. II. Título.

CDU 004.7(043)

"UMA ABORDAGEM EVOLUCIONÁRIA PARA O POSICIONAMENTO DINÂMICO DE MÁQUINAS VIRTUAIS EM DATACENTERS PRIVADOS, APLICANDO SOBRECARGA DE RECURSOS"

PETRÔNIO CARLOS BEZERRA

TESE APROVADA EM 28/06/2019

REINALDO CÉZAR DE MORAIS GOMES, Dr., UFCG
Orientador(a)

ANDERSON FABIANO BATISTA FERREIRA DA COSTA, Dr., IFPB
Examinador(a)

MARCUS WILLIAMS AQUINO DE CARVALHO, Dr., UFPB
Examinador(a)

KELVIN LOPES DIAS, Dr., UFPE
Examinador(a)

GLAUCO ESTÁCIO GONÇALVES, Dr., UFRPE
Examinador(a)

CAMPINA GRANDE - PB

Resumo

O advento das tecnologias de Computação em Nuvem tem auxiliado empresas com soluções para manutenção de suas estruturas computacionais, oferecendo redução de custos, facilidade de gerenciamento e escalabilidade. Várias companhias têm adotado as soluções de Nuvem em ambientes privados, o que permite a redução dos custos ainda mantendo a posse e gerência de seus *datacenters*, mitigando alguns problemas que normalmente são enfrentados nas *Nuvens Públicas* como multilocação e segurança. Ambientes como esses fazem uso contínuo e intenso de Máquinas Virtuais, tirando proveito dos benefícios da virtualização. Máquinas virtuais podem migrar constantemente entre servidores físicos, com objetivos diversos, o que pode impactar de forma negativa no desempenho dos serviços nelas implantados, e no *datacenter* como um todo, contudo, migração é uma característica que favorece o gerenciamento dos ambientes de Nuvens. Portanto, faz-se necessário um monitoramento constante do ambiente, alocando da melhor forma possível as máquinas virtuais nos servidores físicos, para que os objetivos dos usuários do *datacenter* sejam alcançados. A abordagem aqui desenvolvida, chamada de *VMPOS*, realiza o posicionamento dinâmico das máquinas virtuais, através de um algoritmo memético que auxilia na busca da melhor alocação dessas máquinas, considerando múltiplos objetivos a serem atendidos de forma simultânea, além de aplicar sobrecarga de recursos com diferentes níveis nos servidores físicos, visando obter melhor desempenho para serviços considerados críticos. Assim, é possível configurar o nível de comprometimento das máquinas físicas, fornecendo flexibilidade ao *datacenter* e permitindo que mais ou menos máquinas virtuais sejam alocadas por máquina física de acordo com os serviços que executam em cada máquina virtual. Como resultados, nas comparações com outras três abordagens de posicionamento, foi possível confirmar que a *VMPOS* reduziu o consumo de energia e aumentou a consolidação de servidores, aplicando sobrecarga de recursos, em nuvem privada.

Palavras-chave: Computação em Nuvem; Virtualização; Algoritmos Meméticos; Sobrecarga de Recursos; Fronteira de Pareto.

Abstract

The advent of cloud computing technologies has helped companies with solutions to maintain their computing structures, offering cost savings, ease of management and scalability. Several companies have adopted cloud solutions in private environments, which allows them to reduce costs while still maintaining the ownership and management of their datacenters, mitigating some of the problems that are commonly encountered in Public Clouds such as multi-tenancy and security. Environments like these make continuous and intense use of Virtual Machines, taking advantage of the benefits of virtualization. Virtual machines can constantly migrate between physical servers, with different objectives, and may negatively impact the performance of the services deployed in them, and in the datacenter as a whole, however, migration is a feature that favors the management of Cloud environments. Therefore, constant monitoring of the environment is necessary, allocating the virtual machines in the best possible way in the physical servers, so that the objectives of the users of the datacenter are reached. The approach developed here, called VMPOS, performs the dynamic positioning of virtual machines, through a memetic algorithm that assists in the search for the best allocation of these machines, considering multiple objectives to be met simultaneously, in addition to applying resource overhead with different levels in physical servers, in order to obtain better performance for services considered critical. Thus, it is possible to configure the level of compromise of physical machines, providing flexibility to the datacenter and allowing more or less virtual machines to be allocated by physical machines according to the services that execute in each virtual machine. As a result, in comparisons with three other positioning approaches, it was possible to confirm that VMPOS reduced power consumption and increased server consolidation by applying resource overhead in the private cloud.

Keywords: Cloud Computing; Virtualization; Memetic Algorithms; Oversubscription; Pareto Front.

Dedicatória

Dedico este trabalho à minha amada esposa **Maria Emília Gonçalves da Rocha Bezerra**, meu porto seguro, que me apoiou incondicionalmente nas horas mais difíceis, mostrando que a vitória chegaria. Obrigado por fazer parte da minha vida. Amo muito você.

Dedico também este trabalho aos meus filhos **Caio José Rocha Bezerra** e **Cauã Rocha Bezerra**, grandes amores de minha vida, que durante esta trajetória estiveram sempre ao meu lado compreendendo as ausências e oferecendo apoio e carinho. Vocês fizeram o fardo parecer mais leve com seus sorrisos, beijos e abraços. Muito obrigado. Amo muito vocês.

Agradecimentos

Quero iniciar agradecendo à Deus por me permitir trilhar mais uma etapa importante de minha vida, sei que sempre esteve comigo ouvindo e cuidando de meus lamentos, dúvidas e medos. Onde sempre busquei forças nas vezes que pensei em desistir.

Ao meu orientador, Professor Reinaldo César de Moraes Gomes, por sua ajuda, paciência, amizade e profissionalismo durante esta longa caminhada que trilhei. Sem seus direcionamentos, jamais eu chegaria até aqui.

À minha mãe, Margarida Ferreira Bezerra, por seu amor, dedicação e orações constantes para que tudo desse certo no doutorado. Sua força ao longo de seus 87 anos sempre nos motivou. Ao meu pai, Manoel Agostinho Bezerra (*in memoriam*) que, apesar dos pouco mais que 13 anos em que esteve ao meu lado, sempre se mostrou um homem correto em tudo, com exemplos suficientes para que eu pudesse me espelhar e formar meu caráter. Ao meu irmão Paulo César Bezerra, minhas irmãs Maria Verônica Bezerra e Virgínia de Fátima Bezerra, por suas orações, palavras de força e pela ajuda, mesmo morando em outras cidades, em dividir comigo o cuidado de nossa mãe para que eu pudesse me dedicar melhor ao doutorado. A todos meus amados e amadas sobrinhas e sobrinhos, cunhados e cunhada, meu muito obrigado. À minha cunhada Rosélia Gonçalves Rocha, que tantos finais de semana cuidou de meus filhos, enquanto a mãe estava de plantão em outra cidade, para que eu pudesse trabalhar na minha pesquisa. Ao meu cunhado Valner da Silva Nogueira e minha irmã Virgínia pelas conversas sobre meu tema de pesquisa e a disposição em ajudar. À minha tia Hilda Barbosa (*in memoriam*), que nos deixou ao longo desta minha jornada, sempre em orações e com palavras de confiança e certeza na aprovação de meu trabalho, em todas as vezes que nos encontrávamos.

Aos pesquisadores que desenvolveram trabalhos relacionados ao tema de minha pesquisa, compartilhando conteúdo e ajudando nos momentos de dúvidas, em especial ao Dr. Fabio López Pires (coordenador do *Education Innovation Center at the Itaipu Technological Park* no Paraguai).

Aos professores e professora membros de minha banca de qualificação, Prof^ª. Joseana Fechine, Prof. Anderson Fabiano, Prof. Glauco Gonçalves e ao Prof. Marcus Carvalho,

que com suas importantes contribuições fizeram a pesquisa tomar outro rumo, chegando a um resultado muito melhor. Assim como, aos membros de minha banca final de defesa de tese que contou também com participação do Prof. Kelvin Lopes Dias, com todos fazendo importantes observações para os ajustes finos finais. O meu muito obrigado.

Aos colegas do Laboratório de Telecomunicações (LATEC), Gustavo Nóbrega, Fellype Cavalcante, Saulo Aislan, José Anderson Rodrigues e Teles Bezerra, pelas conversas descontraídas e apoio na resolução de problemas acadêmicos. Aos meus amigos Alex Sandro da Cunha Rêgo, Ana Cristina, César Rocha e a Elmano Ramalho, agradeço pela amizade e importantes contribuições com dicas, material e ferramentas que facilitaram o desenvolvimento do trabalho.

Aos professores do programa de Computação da UFCG, Eustáquio Rangel e Adalberto Cajueiro, pelo incentivo e descontração em saber sobre o andamento de meu trabalho. E em especial ao professor Anderson Fabiano do IFPB por sua participação e importantes contribuições para pesquisa.

Aos professores do IFPB por terem aprovado mais prazo em meu afastamento das atividades acadêmicas da instituição, possibilitando mais “tranquilidade” para conclusão da tese. Da mesma forma, ao IFPB por me possibilitar que a jornada fosse trilhada com dedicação integral ao doutorado.

Para concluir, agradeço aos funcionários da UFCG, nas pessoas dos professores e coordenadores do programa, dos funcionários da coordenação, dos terceirizados da limpeza e a todos os demais que direta ou indiretamente contribuíram com a elaboração deste trabalho.

“O importante e bonito do mundo é isso: que as pessoas não estão sempre iguais, ainda não foram terminadas, mas que elas vão sempre mudando. Afinam e desafinam.”

Guimarães Rosa

Conteúdo

1	Introdução	1
1.1	Objetivos	9
1.1.1	Objetivos Específicos	9
1.1.2	Questões de Pesquisa	10
1.2	Contribuições	11
1.3	Estrutura do Documento	12
2	Fundamentação Teórica	14
2.1	Computação em Nuvem	14
2.1.1	Definição	15
2.1.2	Tipos de Nuvens	16
2.1.3	Modelos de Serviços	16
2.1.4	Problemas das Nuvens Públicas	17
2.2	Provisionamento de Recursos	19
2.2.1	O modelo <i>top-down</i> de provisionamento de recursos em Nuvens	19
2.3	Abordagens Utilizadas para Validação	23
2.3.1	A Abordagem <i>First Fit</i>	23
2.3.2	A Abordagem <i>Random Fit</i>	23
2.3.3	A Abordagem <i>iMaVMP</i>	24
2.4	Virtualização	25
2.4.1	Definição	26
2.4.2	Tipos de Virtualização	27
2.4.3	Benefícios da Virtualização	30
2.5	<i>Containers</i>	31

2.6	Otimização com Múltiplos Objetivos	34
2.6.1	POM	35
2.6.2	Algoritmos Genéticos (AG)	37
2.6.3	Algoritmos Meméticos	43
2.7	Considerações Finais	44
3	Trabalhos Relacionados	45
3.1	Gerenciamento de Recursos	45
3.1.1	Otimização com Objetivo Único	46
3.1.2	Otimização de até Dois Objetivos	47
3.1.3	Otimização Não Simultânea de Múltiplos Objetivos	51
3.1.4	Otimização Simultânea de Múltiplos Objetivos	54
3.2	Considerações Finais	59
4	A Abordagem VMPOS	60
4.1	Estrutura do <i>Datacenter</i>	60
4.2	Visão Geral	63
4.2.1	Funções dos Componentes do <i>Datacenter</i>	64
4.3	Descrição da Abordagem Desenvolvida	70
4.3.1	Componente: Indivíduo Base	72
4.3.2	Componente: <i>Datacenter</i>	73
4.3.3	Componente: Configurações	73
4.4	Formulação do Problema de Posicionamento de VMs	78
4.4.1	Dados de Entrada	79
4.4.2	Dados de Saída	81
4.4.3	Aplicação de Pesos	82
4.4.4	Funções Objetivo	83
4.4.5	Restrições	89
4.5	Alocação com Múltiplos Objetivos no VMPOS	90
4.5.1	Codificação	91
4.5.2	Algoritmo Memético	92
4.5.3	Inicialização da População	94

4.5.4	Correção das Soluções não Factíveis	94
4.5.5	Aperfeiçoamento Local	97
4.5.6	Função de <i>Fitness</i>	99
4.5.7	Seleção e Cruzamento	99
4.5.8	Mutação	100
4.6	Considerações Finais	101
5	Resultados e Discussões	102
5.1	Características dos <i>Datacenters</i> nas Simulações	102
5.2	Componentes da Validação	104
5.2.1	A Abordagem <i>First Fit</i>	104
5.2.2	A Abordagem <i>Random Fit</i>	105
5.2.3	A Abordagem <i>iMaVMP</i>	105
5.3	Procedimento para Execução da VMPOS	105
5.4	Configurações Utilizadas nas Comparações	106
5.5	Procedimentos Estatísticos Adotados	109
5.5.1	Amostras Coletadas	111
5.6	Avaliação da Solução Proposta	112
5.6.1	Objetivo Consumo de Energia (VMPOS x FF e RF)	112
5.6.2	Objetivo Dispersão (VMPOS x FF e RF)	119
5.6.3	Objetivo Consumo de Energia (VMPOS x iMaVMP)	123
5.6.4	Quantidade de Máquinas Físicas Alocadas	126
5.7	VMPOS x VMPOS	128
5.7.1	Priorizando Redução no Consumo de Energia	129
5.7.2	Priorizando Redução na Dispersão	133
5.7.3	Priorizando Redução na Quantidade de Migração	137
5.7.4	Efeito da Priorização nos Demais Objetivos	140
5.8	Análise do Fator Tempo	143
5.9	Considerações Finais	148
6	Conclusões e Trabalhos Futuros	149
6.1	Publicações Científicas	152

6.2	Trabalhos Futuros	152
A	Configurações da VMPOS	166
A.1	Arquivo de Configuração	167
A.2	Simulador Gerador de <i>Datacenter</i>	169
A.3	Número de Indivíduos Utilizados	172
B	Priorização de Objetivo	176
B.1	Quando foi priorizado o objetivo Energia	177
B.1.1	Efeito no objetivo Dispersão	177
B.1.2	Efeito no objetivo Migração	179
B.2	Quando foi priorizado o objetivo Migração	182
B.2.1	Efeito no objetivo Energia	182
B.2.2	Efeito no objetivo Dispersão	183
B.3	Quando foi priorizado o objetivo Dispersão	185
B.3.1	Efeito no objetivo Energia	185
B.3.2	Efeito no objetivo Migração	186
C	Testes de Normalidade	189
C.1	Testes com os dados de Dispersão	190
C.2	Testes com os dados de Energia	193
D	Testes de <i>Kruskal</i>	196
D.1	Testes com os dados de Energia	196
D.2	Testes com os dados de Dispersão	198
E	Testes de <i>Nemenyi</i>	199
E.1	Testes com os dados de Energia	199
E.2	Testes com os dados de Dispersão	202
F	Testes Estatísticos das Diferenças	206
F.1	Testes unilaterais com os dados de Energia	207
F.2	Testes unilaterais com os dados de Dispersão	210

G Estatísticas VMPOS x VMPOS	212
G.1 Testes bilaterais com os dados de Energia	213
G.2 Testes unilaterais com os dados de Energia	214
G.3 Testes bilaterais com os dados de Dispersão	215
G.4 Testes unilaterais com os dados de Dispersão	216
G.5 Testes bilaterais com os dados de Migração	217
G.6 Testes unilaterais com os dados de Migração	218
H Gráficos	219
H.1 Dados de Energia, SC=1,0 e SNC=1,0	219
H.2 Dados de Energia, SC=1,0 e SNC=1,5	224
H.3 Dados de Energia, SC=1,5 e SNC=1,5	229
H.4 Dados de Dispersão, SC=1,0 e SNC=1,0	234
H.5 Dados de Dispersão, SC=1,0 e SNC=1,5	239
H.6 Dados de Dispersão, SC=1,5 e SNC=1,5	244

Lista de Figuras

1.1	Previsões para o percentual de crescimento Nuvens Públicas vs. Privadas de 2015 a 2020.	5
1.2	Pesquisa com percentual de intenção de uso dos diversos tipos de Nuvens em 2018.	6
2.1	Fases para a realização do posicionamento de uma VM.	20
2.2	Esquema com o escalonamento de recursos em nuvem.	22
2.3	Virtualização completa.	27
2.4	Para-virtualização.	28
2.5	Virtualização Assistida por <i>Hardware</i>	29
2.6	Virtualização no Nível de <i>Kernel</i>	30
2.7	Virtualização baseada em <i>Containers</i>	32
2.8	Exemplo de Fronteira de Pareto com dominância de soluções.	37
2.9	Ciclo evolutivo de um Algoritmo Genético.	39
2.10	Representação de uma roleta de seleção em AG. Cada fatia destinada a cada indivíduo é proporcional a seu <i>fitness</i>	40
2.11	Processo de evolução utilizado no NSGA-II.	42
4.1	Esquema do ambiente de <i>Datacenter</i> privado considerado na pesquisa.	61
4.2	Fluxograma de alocação de nova VM.	66
4.3	Fluxograma para exclusão de uma VM.	68
4.4	Fluxograma representativo da verificação sistemática do estado do sistema.	69
4.5	Fluxograma representativo do processo de posicionamento das VMs em MFs dentro do SDC.	70
4.6	Ampliação da abordagem VMPOS.	72

4.7	Ilustração com as possibilidades de configurações da VMPOS.	74
4.8	Exemplificação do cálculo do Nível de Dispersão.	88
4.9	Representação da alocação de nove VMs em três Máquinas Físicas e o cromossomo correspondente.	91
4.10	Cruzamento em ponto único.	100
4.11	Mutação em um gene.	101
5.1	Sequência realizada para execução dos experimentos com a VMPOS. . . .	105
5.2	Médias do consumo de Energia das três abordagens, com SC=1 e SNC=1, para todos os tamanhos de <i>Datacenter</i>	114
5.3	<i>Boxplots</i> com as amostras de Energia das três abordagens, com 10% de VMs críticas e com SC=1 e SNC=1.	114
5.4	Médias do consumo de Energia das três abordagens, com SC=1 e SNC=1,5, para todos os tamanhos de <i>Datacenters</i>	118
5.5	<i>Boxplots</i> com as amostras de Energia de três abordagens, com 10% de VMs críticas e com SC=1 e SNC=1,5.	119
5.6	Médias da Dispersão das VMs, nas três abordagens comparadas, com SC=1 e SNC=1,5, para todos os tamanhos de <i>Datacenters</i> e percentuais de VMs críticas.	120
5.7	<i>Boxplots</i> com as amostras de Dispersão das três abordagens, com 50% de VMs críticas e com SC=1 e SNC=1,5.	121
5.8	Médias do consumo de Energia das duas abordagens, com SC=1,0 e SNC=1,0, para todos os tamanhos de <i>Datacenter</i>	124
5.9	<i>Boxplots</i> com as amostras de Energia das abordagens VMPOS e iMaVMP, com 50% de VMs críticas e com SC=1 e SNC=1.	125
5.10	Número médio de máquinas físicas alocadas.	127
5.11	Número médio de máquinas virtuais não alocadas pelo iMaVMP.	128
5.12	Dados médios do consumo de Energia, em <i>Datacenter</i> Pequeno, favorecendo o parâmetro Energia e sem aplicação de pesos.	130
5.13	<i>Boxplots</i> com as amostras de Energia para o <i>Datacenter</i> Pequeno, com SC=1 e SNC=1.	131

5.14	Resultados médios das amostras de Dispersão para os níveis de compromisso SC=1,0 e SNC=1,0.	134
5.15	Resultados médios das amostras de Dispersão para os níveis de compromisso SC=1,0 e SNC=1,5.	134
5.16	Resultados médios das amostras de Dispersão para os níveis de compromisso SC=1,5 e SNC=1,5.	135
5.17	<i>Boxplots</i> com as amostras de Migração para o <i>Datacenter</i> Pequeno, com SC=1 e SNC=1.	139
5.18	Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em <i>Datacenter</i> Pequeno.	141
5.19	Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em <i>Datacenter</i> Pequeno.	142
5.20	Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em <i>Datacenter</i> Pequeno.	143
5.21	Durações médias de execução dos experimentos, com SC=1,0 e SNC=1,0, e com 10% de VMs críticas.	145
5.22	Durações médias de execução dos experimentos no VMPOS, com SC=1,0 e SNC=1,0.	146
A.1	Ampliação com detalhes da abordagem VMPOS.	166
A.2	Arquivo <i>vmpos_config.vmp</i> com exemplo de configuração da VMPOS. . .	167
A.3	Arquivo de entrada do Simulador.	169
A.4	Arquivo de saída do Simulador.	171
A.5	Escolha do número de indivíduos a ser utilizado nos experimentos.	173
A.6	Fator média de tempo vezes geração do melhor indivíduo.	174
B.1	Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em <i>Datacenter</i> Pequeno.	177
B.2	Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em <i>Datacenter</i> Médio.	178
B.3	Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em <i>Datacenter</i> Grande.	178

B.4	Resultado da Migração quando se aplicou prioridade para o objetivo Energia, em <i>Datacenter</i> Pequeno.	179
B.5	Resultado da Migração quando se aplicou prioridade para o objetivo Energia, em <i>Datacenter</i> Médio.	180
B.6	Resultado da Migração quando se aplicou prioridade para o objetivo Energia, em <i>Datacenter</i> Grande.	181
B.7	Resultado da Energia quando se aplicou prioridade para o objetivo Migração, em <i>Datacenter</i> Pequeno.	182
B.8	Resultado da Energia quando se aplicou prioridade para o objetivo Migração, em <i>Datacenter</i> Médio.	182
B.9	Resultado da Energia quando se aplicou prioridade para o objetivo Migração, em <i>Datacenter</i> Grande.	183
B.10	Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em <i>Datacenter</i> Pequeno.	183
B.11	Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em <i>Datacenter</i> Médio.	184
B.12	Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em <i>Datacenter</i> Grande.	184
B.13	Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em <i>Datacenter</i> Pequeno.	185
B.14	Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em <i>Datacenter</i> Médio.	185
B.15	Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em <i>Datacenter</i> Grande.	186
B.16	Resultado da Migração quando se aplicou prioridade para o objetivo Dispersão, em <i>Datacenter</i> Pequeno.	186
B.17	Resultado da Migração quando se aplicou prioridade para o objetivo Dispersão, em <i>Datacenter</i> Médio.	187
B.18	Resultado da Migração quando se aplicou prioridade para o objetivo Dispersão, em <i>Datacenter</i> Grande.	188

H.1	<i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,0 e SNC=1,0.	219
H.2	<i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,0 e SNC=1,0.	220
H.3	<i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,0 e SNC=1,0.	220
H.4	<i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,0 e SNC=1,0.	221
H.5	<i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,0 e SNC=1,0.	221
H.6	<i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,0 e SNC=1,0.	222
H.7	<i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,0 e SNC=1,0.	222
H.8	<i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,0 e SNC=1,0.	223
H.9	<i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,0 e SNC=1,0.	223
H.10	<i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,0 e SNC=1,5.	224
H.11	<i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,0 e SNC=1,5.	224
H.12	<i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,0 e SNC=1,5.	225
H.13	<i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,0 e SNC=1,5.	225
H.14	<i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,0 e SNC=1,5.	226
H.15	<i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,0 e SNC=1,5.	226

H.16 <i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,0 e SNC=1,5.	227
H.17 <i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,0 e SNC=1,5.	227
H.18 <i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,0 e SNC=1,5.	228
H.19 <i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,5 e SNC=1,5.	229
H.20 <i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,5 e SNC=1,5.	229
H.21 <i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,5 e SNC=1,5.	230
H.22 <i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,5 e SNC=1,5.	230
H.23 <i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,5 e SNC=1,5.	231
H.24 <i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,5 e SNC=1,5.	231
H.25 <i>Boxplots</i> com as amostras de Energia com 10% de VMs críticas e com SC=1,5 e SNC=1,5.	232
H.26 <i>Boxplots</i> com as amostras de Energia com 25% de VMs críticas e com SC=1,5 e SNC=1,5.	232
H.27 <i>Boxplots</i> com as amostras de Energia com 50% de VMs críticas e com SC=1,5 e SNC=1,5.	233
H.28 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,0 e SNC=1,0.	234
H.29 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,0 e SNC=1,0.	234
H.30 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,0 e SNC=1,0.	235

H.31 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,0 e SNC=1,0.	235
H.32 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,0 e SNC=1,0.	236
H.33 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,0 e SNC=1,0.	236
H.34 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,0 e SNC=1,0.	237
H.35 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,0 e SNC=1,0.	237
H.36 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,0 e SNC=1,0.	238
H.37 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,0 e SNC=1,5.	239
H.38 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,0 e SNC=1,5.	239
H.39 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,0 e SNC=1,5.	240
H.40 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,0 e SNC=1,5.	240
H.41 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,0 e SNC=1,5.	241
H.42 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,0 e SNC=1,5.	241
H.43 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,0 e SNC=1,5.	242
H.44 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,0 e SNC=1,5.	242
H.45 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,0 e SNC=1,5.	243

H.46 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,5 e SNC=1,5.	244
H.47 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,5 e SNC=1,5.	244
H.48 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,5 e SNC=1,5.	245
H.49 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,5 e SNC=1,5.	245
H.50 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,5 e SNC=1,5.	246
H.51 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,5 e SNC=1,5.	246
H.52 <i>Boxplots</i> com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,5 e SNC=1,5.	247
H.53 <i>Boxplots</i> com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,5 e SNC=1,5.	247
H.54 <i>Boxplots</i> com as amostras de Dispersão, com 50% de VMs críticas e com SC=1,5 e SNC=1,5.	248

Lista de Tabelas

3.1	Caracterização dos trabalhos relacionados.	58
4.1	Componentes existentes na formulação matemática da abordagem.	79
5.1	Características dos <i>datacenters</i> utilizados nos experimentos.	103
5.2	Parâmetros comparados entre as abordagens.	107
5.3	Níveis de comprometimento aplicados nas abordagens.	108
5.4	Características das amostras coletadas.	112
5.5	Resultados do <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Pequeno, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.	116
5.6	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Médio, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.	122
5.7	Valores médios e percentuais de reduções no consumo de Energia no <i>datacenter</i> Pequeno, aplicando-se peso com prioridade para Energia.	132
5.8	Dados médios registrados em algumas amostras aplicando-se peso com prioridade para Energia em comparação com os dados de referência.	133
5.9	Dados médios registrados nos experimentos aplicando-se peso com prioridade para Dispersão em comparação com os dados de referência.	136
5.10	Dados médios registrados nos experimentos aplicando-se peso com prioridade para Migração em comparação com os dados de referência.	138
5.11	Tempos médios, em segundos, registrados nos experimentos com as abordagens VMPOS e iMaVMP, com SC=1,0 e SNC=1,0.	145

C.1	Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1$, com os dados de Dispersão.	190
C.2	Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1,5$, com os dados de Dispersão.	191
C.3	Resultados dos testes de normalidade, usando $SC = 1,5$ e $SNC = 1,5$, com os dados de Dispersão.	192
C.4	Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1$, com os dados de Energia.	193
C.5	Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1,5$, com os dados de Energia.	194
C.6	Resultados dos testes de normalidade, usando $SC = 1,5$ e $SNC = 1,5$, com os dados de Energia.	195
D.1	Resultados dos testes de Kruskal com os dados de Energia considerando todas as abordagens utilizadas.	197
D.2	Resultados dos testes de Kruskal com os dados de Dispersão considerando todas as abordagens utilizadas.	198
E.1	Resultados do <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Pequeno, com $SC=1$ e $SNC=1$, para diferentes percentuais de VMs críticas.	199
E.2	Resultados do <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Médio, com $SC=1$ e $SNC=1$, para diferentes percentuais de VMs críticas.	199
E.3	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Grande, com $SC=1$ e $SNC=1$, para diferentes percentuais de VMs críticas.	200
E.4	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Pequeno, com $SC=1$ e $SNC=1,5$, para diferentes percentuais de VMs críticas.	200

E.5	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Médio, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.	200
E.6	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Grande, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.	201
E.7	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Pequeno, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.	201
E.8	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Médio, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.	201
E.9	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Energia, no <i>datacenter</i> Grande, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.	202
E.10	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Pequeno, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.	202
E.11	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Médio, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.	202
E.12	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Grande, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.	203
E.13	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Pequeno, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.	203
E.14	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Médio, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.	203

E.15	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Grande, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.	204
E.16	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Pequeno, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.	204
E.17	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Médio, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.	204
E.18	Resultados da função estatística <i>posthoc.kruskal.nemenyi.test</i> , com os dados de Dispersão, no <i>datacenter</i> Grande, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.	205
F.1	Resultados dos testes unilaterais de Wilcoxon, com os dados de Energia, entre o VMPOS e o <i>First Fit</i> ,	207
F.2	Resultados dos testes unilaterais de Wilcoxon, com os dados de Energia, entre o VMPOS e o <i>Random Fit</i>	208
F.3	Resultados dos testes unilaterais de Wilcoxon, com os dados de Energia, entre o VMPOS e o <i>iMaVMP</i>	209
F.4	Resultados dos testes unilaterais de Wilcoxon, com os dados de Dispersão, entre o VMPOS e o <i>First Fit</i>	210
F.5	Resultados dos testes unilaterais de Wilcoxon, com os dados de Dispersão, entre o VMPOS e o <i>Random Fit</i>	211
G.1	Resultados dos testes estatísticos bilaterais de Wilcoxon, com os dados de Energia, nos comparativos VMPOS x VMPOS, amostras com prioridade para Energia contra as amostras sem atribuir prioridade alguma.	213
G.2	Resultados dos testes estatísticos unilaterais de Wilcoxon, com os dados de Energia, nos comparativos VMPOS x VMPOS, amostras com prioridade para Energia contra as amostras sem atribuir prioridade alguma.	214

G.3	Resultados dos testes estatísticos bilaterais de Wilcoxon, com os dados de Dispersão, nos comparativos VMPOS x VMPOS, amostras com prioridade para Dispersão contra as amostras sem atribuir prioridade alguma.	215
G.4	Resultados dos testes estatísticos unilaterais de Wilcoxon, com os dados de Dispersão, nos comparativos VMPOS x VMPOS, amostras com prioridade para Dispersão contra as amostras sem atribuir prioridade alguma.	216
G.5	Resultados dos testes estatísticos bilaterais de Wilcoxon, com os dados de Migração, nos comparativos VMPOS x VMPOS, amostras com prioridade para Migração contra as amostras sem atribuir prioridade alguma.	217
G.6	Resultados dos testes estatísticos unilaterais de Wilcoxon, com os dados de Migração, nos comparativos VMPOS x VMPOS, amostras com prioridade para Migração contra as amostras sem atribuir prioridade alguma.	218

Lista de Algoritmos

1	Algoritmo exemplo de uma roleta	41
2	Algoritmo Memético	93
3	Verifica Viabilidade	95
4	Correção das Soluções Inviáveis	96
5	Aperfeiçoamento Local	98

Lista de Siglas

- AG Algoritmos Genéticos
- DC Centro de Dados (em inglês, *Datacenter*)
- EC Computação Evolutiva (em inglês, *Evolutionary Computation*)
- FF *First Fit*
- FM Fator de Migrações
- MF Máquina Física
- MOEA Algoritmo Evolucionário com Múltiplos Objetivos (em inglês, *Multi-Objective Evolutionary Algorithm*)
- POM Problemas de Otimização com Múltiplos objetivos
- QoS Qualidade de Serviço (em inglês, *Quality of Service*)
- RAS Sistemas de Alocação de Recursos (em inglês, *Resource Allocation System*)
- RF *Random Fit*
- RSA Algoritmos de Escalonamento de Recursos (em inglês, *Resource Scheduling Algorithms*)
- SC Serviços Críticos
- SLA Acordos de Níveis de Serviço (em inglês, *Service Level Agreement*)
- SNC Serviços Não Críticos

TI Tecnologia da Informação

VM Máquina Virtual (em inglês, *Virtual Machine*)

VMPOS Posicionamento de Máquinas Virtuais Aplicando Sobrecarga de Recursos (em inglês, *Virtual Machine Placement with OverSubscription*)

Capítulo 1

Introdução

Com o advento e a rápida evolução dos ambientes de Computação em Nuvem (em inglês, *Cloud Computing*) [1], a Internet também promoveu mudanças na forma como os sistemas computacionais são organizados. Empresas e pessoas comuns passaram a contar com a possibilidade de utilizar recursos computacionais disponibilizados por empresas provedoras de ambientes de Computação em Nuvem. Devido aos benefícios que são oferecidos por esses ambientes, Computação em Nuvem tem alcançado grande sucesso desde meados de 2005 [2]. Grandes empresas da área de tecnologia passaram a permitir que os recursos disponíveis em suas infraestruturas de Tecnologia da Informação (TI) de larga escala, passem a ser arrendados para usuários que desejam atender suas demandas computacionais [3]. São exemplos de provedores de Computação em Nuvem: no mundo, *Amazon*¹, *Microsoft*², *Rackspace*³, e no Brasil, *UOL*⁴. Essas empresas são chamadas de provedores de *nuvens públicas*, e oferecem diversos serviços como locação de recursos computacionais na forma *pay-as-you-go*, ou seja, pague pelos recursos e tempo que utilizar [4][5][6][7]. Segundo [8], os recursos em um ambiente de Computação em Nuvem podem ser arrendados da mesma forma que itens de utilidade pública, como gás, água e energia.

Na área acadêmica, uma grande variedade de trabalhos sobre Computação em Nuvem já foi escrita, muitos dos quais podem ser encontrados de forma comparativa e resumida em alguns levantamentos das pesquisas na área [9][10][11][12], e o tema continua atual com

¹<https://aws.amazon.com/pt/ec2/>

²<https://azure.microsoft.com/>

³<https://www.rackspace.com/>

⁴<http://www.uolhost.uol.com.br/uol-cloud-computing.html>

diversos desafios, como pode ser visto em vários artigos que foram publicados de 2016 até hoje [13][14][15][16][17].

Apesar dos avanços observados na área de Computação em Nuvem [18][19][2], várias empresas ainda preferem manter suas estruturas e seus dados/informações sob suas administrações [20][21][12], em suas próprias *nuvens privadas*. Para esse tipo de ambiente, o proprietário precisa adquirir e gerenciar todos os recursos computacionais de que necessita, ou locar os recursos de terceiros para uso exclusivo [22], diferentemente das *nuvens públicas* nas quais os recursos computacionais são mantidos e gerenciados pela empresa provedora contratada para fornecer a infraestrutura computacional desejada.

Independente do ambiente, público ou privado, as tecnologias de virtualização de máquinas são comumente utilizadas em infraestruturas de nuvem por trazer diversos benefícios [23][24][25][26], dentre os quais: consolidação de servidores, economia de energia, redução de custos em equipamentos, alta disponibilidade, dentre outros.

Segundo [27]: “*a consolidação de servidores tem se tornado parte integrante do planejamento de TI para reduzir custos e melhorar a eficiência atualmente nas empresas de datacenters*” (PADALA, P., 2007, p. 1, tradução nossa).

A redução no consumo de energia é um dos fatores mais estudados quando se trata de gerenciamento de Nuvens. Isso porque tal fator tem impacto significativo nos custos de manutenção desses ambientes. Portanto, reduzir esse consumo torna-se uma das principais metas dos estudos para gerenciamento de *datacenters*. As pesquisas têm ajudado a gerenciar os recursos dos *datacenters* de maneira energeticamente eficiente [28][29].

Ao se aplicar as técnicas utilizadas pelas tecnologias de virtualização é possível reduzir o número de equipamentos físicos, como computadores, necessários para manter um determinado número de serviços. Por exemplo, se cada serviço necessitar de uma Máquina Física (MF) para hospedá-lo, através da virtualização dos servidores e conseqüente consolidação, reduz-se a quantidade de máquinas físicas necessárias para manter os mesmos serviços. Com isso, o custo energético e com equipamentos pode ser reduzido.

A alta disponibilidade é alcançada na medida em que um serviço implantado em Máquina Virtual (em inglês, *Virtual Machine* ou VM) pode ser migrado, em tempo real, de uma máquina física para outra segundo as decisões do gerenciador do *datacenter*. Sendo assim, não há necessidade de interrupção do serviço da VM já que o processo de mudança ocorre

de modo transparente para a aplicação.

Não obstante, a inserção de uma camada de virtualização em um processo computacional acrescenta uma sobrecarga na execução de uma tarefa qualquer [24][30][31][9][32][21][27][26], que pode ser significativa e inviabilizar a realização de tal tarefa. Dessa maneira, o processo de manter serviços importantes (aqui chamados de *críticos*) de uma empresa em VM deve ser realizado com todo o cuidado, buscando alternativas que permitam priorizar, de alguma forma, o desempenho das máquinas virtuais que abrigam os principais serviços da empresa, ao mesmo tempo em que outros objetivos sejam também favorecidos, como reduzir o consumo de energia ou o número de migração de VMs na rede.

Em ambientes típicos de nuvens públicas ou privadas, os serviços são implantados em VMs isoladas que compartilham os recursos da máquina física hospedeira (como disco, CPU, memória e largura de banda) [33]. Essas VMs normalmente são migradas com os mais diversos objetivos: economia de energia, redução de custo de nó, otimização de desempenho, maximização de utilização, etc. [12].

Várias pesquisas já demonstraram que o aumento no número de VMs em um hospedeiro acarreta maior disputa pelos recursos físicos da máquina física, conseqüentemente, reduzindo o desempenho das VMs [10][11][34][6][35][36]. Por outro lado, para atender aos objetivos dos proprietários de *datacenters*, como ampliar a consolidação de servidores, a técnica de alocar VMs com uma demanda total de requisitos por recursos acima da capacidade física real existente, é uma estratégia que agrega redução de custos com equipamentos e energia e melhora a utilização das máquinas físicas, uma vez que já foi constatado que servidores físicos operam entre 10% e 50%, de suas capacidades plenas, na maior parte do tempo [37]. Outro fator agravante na questão de boa utilização dos recursos é que, mesmo os servidores físicos completamente ociosos ainda consomem cerca de 70% de sua potência de pico [37], o que aumenta o custo com energia e se mostra extremamente ineficiente devido a subutilização [35].

Portanto, separar VMs em dois tipos: as que *possuem serviços críticos*, não devendo ser alocadas com muitas outras VMs para evitar a concorrência por recursos, e as que *não possuem serviços críticos*, não oferecendo problemas mais sérios caso fique sem recursos disponíveis temporariamente por estarem dividindo a MF com outras VMs do mesmo tipo, agrega benefícios ao *datacenter* por atender aos dois pontos que são antagônicos. Ou seja,

priorizar o bom desempenho para as VMs críticas ao mesmo tempo em que se permite alocar VMs não críticas além da capacidade real da máquina física. Isso é feito neste trabalho buscando-se separar essas VMs em hospedeiros diferentes, o que não implica dizer que as VMs não críticas terão desempenho ruim, mas que as chances de não obter um determinado recurso físico, no momento que precise, é maior.

Assim, no processo de consolidação, entende-se ser benéfico poder realizar o procedimento de ampliar a alocação de VMs além da capacidade dos recursos da máquina física, desde que esta não esteja hospedando VMs com serviços críticos. Por outro lado, quando a necessidade for de oferecer melhor desempenho dos serviços nas VMs, deve-se restringir a alocação acima da capacidade da MF.

Para atender a essa flexibilização, o gerenciamento do centro de dados computacional (em inglês, *Datacenter* - DC) deve controlar, de forma dinâmica, a alocação de VMs com características diferentes. Algumas VMs destinam-se a uso geral, como um computador de trabalho (ou *desktop*), e outras são mais requisitadas em termos de demanda computacional, como VMs chamadas de servidores, que hospedam serviços importantes das empresas a exemplo dos bancos de dados, serviço de comércio eletrônico, serviços *web*, etc. Dessa forma, a maneira como as VMs são alocadas nas máquinas físicas pode fazer uma diferença significativa, tanto no desempenho delas, como também no desempenho das máquinas hospedeiras.

As nuvens públicas prometem, para empresas clientes, alguns benefícios como escalabilidade facilitada, confiabilidade, redução de custos tanto de operação quanto de inicialização, entre outros [5][18][19][22].

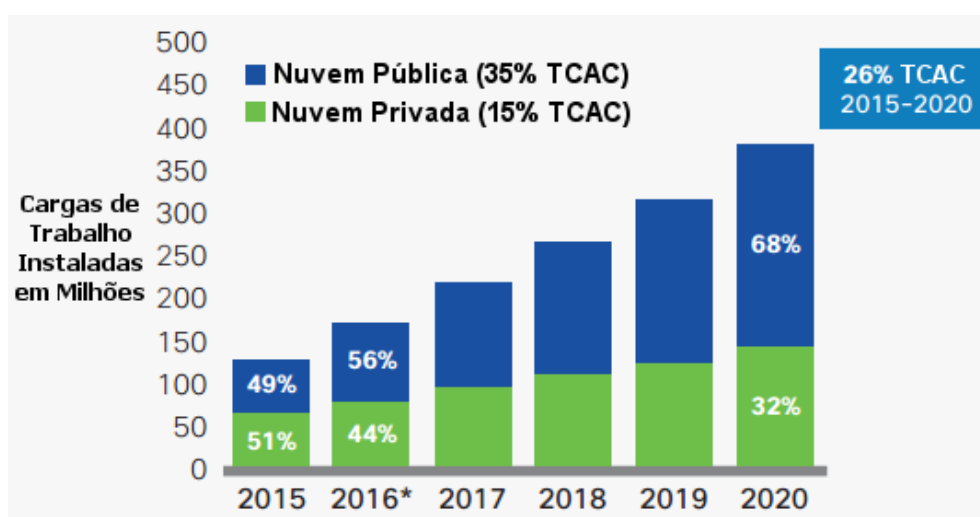
Apesar de bem estabelecida no mercado, as tecnologias relacionadas à Computação em Nuvem ainda apresentam lacunas que necessitam de investigações e ajustes, a exemplo da questão de segurança das informações presentes nas VMs que compartilham mesma MF com outras VMs de proprietários diferentes. Existem na literatura vários problemas e desafios em aberto associados aos mais diversos aspectos como: controle das identidades de acesso ao *datacenter* e multilocalização [1][6][18][22][38][39].

Possivelmente, devido a muitos dos problemas existentes citados nesses artigos referenciados e em vários outros (ver "Problemas das Nuvens" na seção: 2.1.4), muitas empresas e organizações têm criado ou mantido infraestruturas privadas de Computação em Nuvem

[20], indo na contramão do uso das nuvens públicas. Isso pode ser confirmado por pesquisas que indicam, ou preveem, a manutenção de elevados patamares de investimento e crescimento das nuvens privadas, como mostrado a seguir.

A Cisco Systems⁵, lançou um relatório [40] com previsões para a área de Computação em Nuvem para o período de 2015 a 2020. Nesse relatório, os autores fazem previsão de crescimento de 15% das nuvens privadas dentro do período, como pode ser visto na Figura 1.1. O termo TCAC, que aparece na figura, significa: Taxa de Crescimento Anual Composta.

Figura 1.1: Previsões para o percentual de crescimento Nuvens Públicas vs. Privadas de 2015 a 2020.



Fonte: Adaptado de CISCO, 2016, p. 23. (tradução nossa).

Um levantamento feito pelo *Uptime Institute*⁶ fornece uma visão geral das principais tendências que moldam as estratégias das infraestruturas de TI. A pesquisa foi realizada por *e-mail* entre fevereiro e maio de 2018 e inclui respostas de quase 900 operadores de *datacenters* e profissionais de TI de empresas e prestadores de serviços, globalmente. Uma das perguntas aos entrevistados procurava identificar como a organização deles atenderá a demanda por computação de ponta pelos próximos anos. Os resultados estão na Figura 1.2.

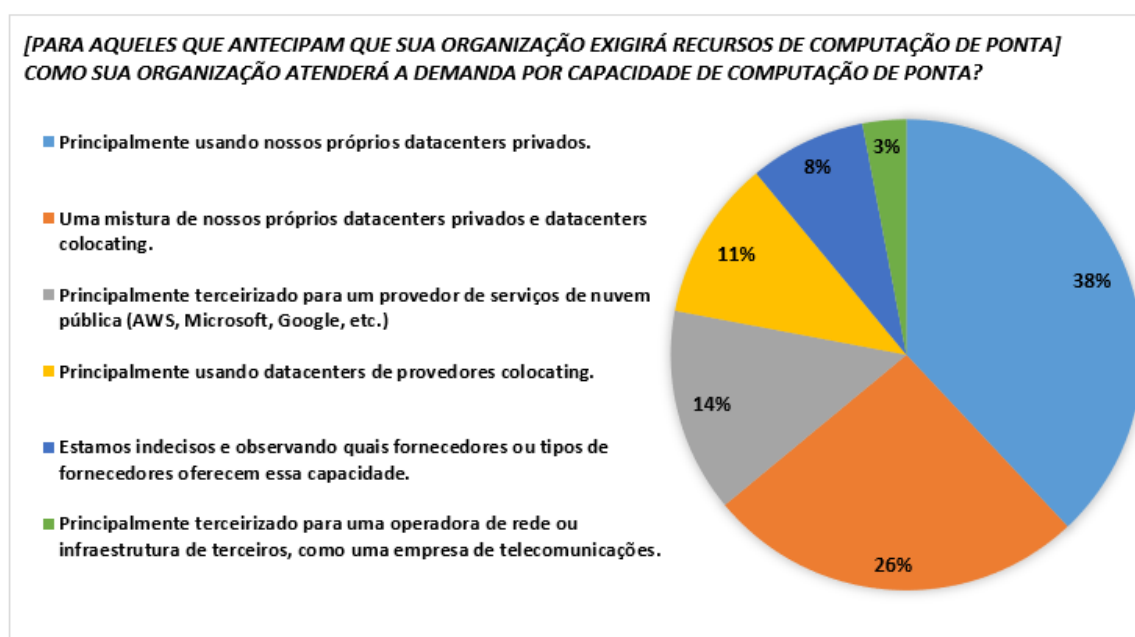
Como se pode constatar, a maior parte dos respondentes (38%) prevê utilização de infraestrutura própria, através de *datacenter* privado, para a demanda futura de computação

⁵<http://www.cisco.com/>

⁶Disponível em: <https://datacenter.com/wp-content/uploads/2018/11/2018-data-center-industry-survey.pdf>. Acesso em: 03 nov. 2018.

de ponta. Se for considerada a soma dessa parcela mais os que responderam que utilizarão nuvens híbridas (26%), o total que envolve *datacenter* privado sobe para mais da metade (64%) dos respondentes.

Figura 1.2: Pesquisa com percentual de intenção de uso dos diversos tipos de Nuvens em 2018.



Fonte: Adaptado de Uptime Institute, 2018, p. 9. (acesso em 03. nov. 2018) (tradução nossa).

De acordo com os dados apresentados até aqui, acredita-se que a tendência de crescimento, com altos investimentos nas nuvens privadas, deva continuar e atribui-se isso a dois fatores principais, que são:

- A contínua melhoria e popularização de *softwares* utilizados na gerência de ambientes de nuvens (públicas ou privadas), que estão se tornando cada vez mais fáceis de configurar, utilizar e com opção gratuita, a exemplo do OpenStack⁷ [21];
- E, aos problemas relacionados aos ambientes de nuvens públicas, a exemplo das questões de segurança e multilocação, conforme discutido na Seção 2.1.4, fazendo com que algumas empresas, clientes em potencial dessas nuvens públicas, desistam de migrarem seus serviços.

⁷<https://www.openstack.org/>

Vale ressaltar também que o crescimento das nuvens públicas gera maior interesse na busca por vulnerabilidades, nesses ambientes, por parte de pessoas mal intencionadas que visam ter acesso de forma indevida ao ambiente ou para realizar ataques [38].

Tudo isso serve de alerta e motivação para que sejam desenvolvidos produtos que possibilitem facilitar ainda mais a vida dos administradores das nuvens privadas, permitindo que os proprietários desses DCs, que tanto tem investido para manter suas estruturas computacionais localmente, sob seu total controle, possam se sentir mais seguros e confiantes de que seus ambientes de produção e serviços irão atender com qualidade a seus clientes.

As empresas de nuvens públicas podem, de acordo com as abordagens de posicionamento de VMs por elas implantadas, priorizar o lucro em detrimento de recursos suficientes para uma VM, ou seja, realizar um sub provisionamento, e dessa forma incorrer na não garantia dos níveis de serviços previamente contratados. Claro que pode haver sanções para o provedor, contudo, caso o fato ocorra, a imagem da empresa do cliente já foi afetada. Provedores, como empresas que são, desejam aumentar seus lucros maximizando a utilização de seus recursos, mesmo correndo o risco de terem que arcar com as penalidades. Os objetivos das empresas de nuvens e dos clientes são antagônicos [41]. Zhang et al. [2] tocam nesse ponto quando eles alertam para que se considere a propriedade aleatória das VMs na requisição de recursos, devendo-se manter baixa a probabilidade de ter uma utilização máxima de recursos, ou necessitar de mais recursos que os previamente alocados.

Tal situação é preocupante para uma empresa visto que sua imagem de confiabilidade e qualidade nos serviços prestados pode ser posta em dúvida se, por exemplo, uma simples página da *web* dessa empresa não estiver acessível, gerando desconfiança e insatisfação, podendo acarretar em abandono de atuais clientes ou a desistência de possíveis futuros clientes em estabelecer contratos com a empresa. Isso pode acontecer, por exemplo, se algum serviço importante da empresa estiver hospedado em um servidor implantado em VM e, devido a possível escassez temporária de recursos da máquina física na qual o servidor virtual está hospedado, este passar a não atender satisfatoriamente aos usuários do serviço específico.

Uma situação como a descrita pode ser remediada se houver uma forma de manter, no DC da empresa, servidores físicos que hospedam VMs de forma compartilhada (ou seja, que mantêm várias VMs ao mesmo tempo concorrendo pelos recursos disponíveis no hospedeiro), mas que se diferencie os tipos de VMs como algumas mais importantes (críticas)

e outras comuns (não críticas). Ao mesmo tempo, controla-se a forma de hospedar estes diferentes tipos aplicando sobrecarga de recursos às máquinas físicas hospedeiras, de forma diferenciada, dependendo dos tipos de VMs que a mesma estiver abrigoando em cada instante. A técnica de alocar mais recursos do que a capacidade real existente em uma máquina física é conhecida como sobrecarga de recursos (em inglês, *oversubscription*) [42].

A melhora esperada no desempenho dos serviços, mais especificamente dos servidores virtuais que abrigam serviços críticos, pode ser alcançada através de redimensionamento do nível aplicado para sobrecarga dos recursos disponibilizados pela MF hospedeira, sendo ajustado dinamicamente.

Isso pode trazer benefícios para as empresas que já estão optando por utilizar nuvem privada, evitando os possíveis problemas existente nas nuvens públicas como os de segurança, privacidade, desempenho, velocidade de acesso à informação (já que os serviços e dados estarão disponíveis localmente e não em DC distantes), dentre outros.

Segundo Singh e Chana [6], o gerenciamento de recursos de nuvem ainda é um tema relevante e muito pesquisado devido aos tempos elevados de execução de serviços e custos dos recursos.

Para se alcançar os benefícios obtidos com a redução de servidores físicos em funcionamento e assim reduzir o desperdício de recursos em um *datacenter*, múltiplas máquinas virtuais devem ser alocadas no menor número possível de máquinas físicas, possibilitando que as demais máquinas físicas possam ser desligadas ou mantidas em um estado de baixo consumo de energia. Essa abordagem sendo aplicada de forma eficiente evita desperdício de recursos, reduz a complexidade do *datacenter* facilitando a gerência, reduz o consumo de energia e conseqüentemente os custos [43]. A técnica de elaborar um mapeamento indicando quais máquinas físicas vão hospedar que máquinas virtuais, com base em critérios diversos, é chamada de *posicionamento de VMs*.

Considera-se que o trabalho, aqui apresentado, para o posicionamento dinâmico de máquinas virtuais em *datacenter* privado, aplicando sobrecarga de recursos e considerando múltiplos objetivos sendo evoluídos simultaneamente de forma pura através de algoritmo memético, até o melhor de nosso conhecimento, é inédito.

1.1 Objetivos

Tendo em vista o problema de posicionamento de VMs abordado na seção anterior e as restrições encontradas nas soluções existentes, este trabalho tem como objetivo principal desenvolver uma nova abordagem, intitulada **Posicionamento de Máquinas Virtuais Aplicando Sobrecarga de Recursos**, a qual possui acrônimo em inglês como **VMPOS** (*Virtual Machine Placement with OverSubscription*), para realizar a tomada de decisão de posicionamento das VMs nas MFs de uma nuvem privada.

Dessa forma, o objetivo geral do trabalho é desenvolver uma abordagem dinâmica de posicionamento de VMs em *datacenter* privado, considerando múltiplos objetivos, que deverão ser evoluídos de forma simultânea pura, aplicando diferentes níveis de sobrecarga de recursos nas máquinas físicas, de acordo com a criticidade das máquinas virtuais nelas hospedadas, visando melhorar a qualidade da alocação das máquinas virtuais no *datacenter* de acordo com os objetivos considerados.

1.1.1 Objetivos Específicos

- Desenvolver uma abordagem que elabore o mapeamento do posicionamento de máquinas virtuais em máquinas físicas. Essa abordagem será flexível o suficiente para que seja possível utilizá-la independente do gerenciador de recursos do *datacenter*;
- Possibilitar que a abordagem considere múltiplos objetivos, comparados de forma pura, na busca pelo melhor posicionamento de máquinas virtuais em máquinas físicas;
- Empregar níveis distintos de criticidade às máquinas virtuais, permitindo aloca-las de forma diferenciada, visando melhorar o desempenho das máquinas que abrigam serviços mais críticos;
- Permitir que sejam aplicados, de forma dinâmica, níveis de comprometimento às máquinas físicas, possibilitando assim o emprego de sobrecarga de recursos, de acordo com os níveis de criticidade das máquinas virtuais hospedadas em cada instante;
- Melhorar, através de algoritmos evolucionários, o posicionamento das máquinas virtuais em máquinas físicas de um *datacenter* privado;
- Realizar a avaliação da abordagem apresentada através de comparativos com outras abordagens utilizadas na literatura, tanto com relação ao posicionamento final obtido quanto

às questões de tempo de execução das abordagens.

1.1.2 Questões de Pesquisa

O objetivo geral pode ser desmembrado nas três seguintes questões de pesquisa:

- *Questão de Pesquisa 1 (QP1): Até que ponto usar a abordagem VMPOS para o posicionamento de VMs fornece melhores resultados que as outras abordagens avaliadas para alocação de VMs em um Datacenter?*

Nessa questão, deseja-se verificar se a abordagem VMPOS consegue melhorar os resultados obtidos por métodos conhecidos de posicionamento de VMs, como o *First Fit* ou o *Random Fit*.

- *Questão de Pesquisa 2 (QP2): Até que ponto usar a abordagem VMPOS ao invés de abordagens que representam o estado da arte, para otimizar a alocação de VMs considerando múltiplos objetivos através de algoritmos genéticos, permite melhorar os resultados em termos das funções objetivos consideradas na pesquisa?*

Nessa questão, deseja-se verificar se a abordagem VMPOS consegue melhorar os resultados obtidos por outra abordagem com as mesmas características, como utilizar evolução de múltiplos objetivos simultaneamente e algoritmos genéticos. Para isso, a abordagem de referência utilizada será a iMaVMP, que é atual e considerada a primeira a aplicar evolução de múltiplos objetivos simultaneamente com algoritmos genéticos, voltado ao posicionamento de VMs em *datacenters*.

- *Questão de Pesquisa 3 (QP3): O favorecimento de uma das funções objetivo consegue melhorar o objetivo em questão e qual a influência nos demais objetivos?*

Nessa questão, deseja-se verificar se é possível que, de acordo com escolhas pessoais do administrador do *datacenter*, alguma das funções objetivos seja priorizada em detrimento das demais. Por exemplo, que o objetivo de reduzir o consumo de energia tenha um peso maior no resultado final de escolha de uma única solução de posicionamento de VMs dentre as melhores possíveis (não no processo de evolução simultânea dos objetivos). Isso quer dizer que, ao se chegar ao resultado da abordagem VMPOS deseja-se priorizar, dentre os indivíduos selecionados como melhores soluções, aquele

que por exemplo, apresentar menor consumo energético independentemente dos resultados das demais funções objetivo consideradas.

1.2 Contribuições

Atendidos os objetivos postos, entende-se que este trabalho alcança as seguintes contribuições:

- Fortalece o crescimento da abordagem de nuvens privadas, considerando múltiplos objetivos de forma simultânea na busca da melhor solução para o posicionamento das VMs nas MFs. As pesquisas mais recentes na área, considerando a aplicação de múltiplos objetivos na decisão do posicionamento de VMs, têm trilhado esse caminho [44][45][46][47];
- Incorpora à literatura a aplicação de sobrecarga de recursos físicos juntamente com a utilização de algoritmos meméticos e a teoria de fronteira de Pareto na busca do posicionamento de VMs que melhor atenda as funções objetivos utilizadas;
- Fornece avaliação da abordagem VMPOS em comparação com outras três abordagens de posicionamento de VMs em *datacenters*, inclusive uma que representa o estado da arte. Os resultados mostraram que a VMPOS foi superior aos demais métodos confrontados;
- A aplicação de nova metodologia, desenvolvida empiricamente pelo autor, para avaliação do nível de concentração dos dois tipos de VMs utilizadas neste trabalho, que foram com serviços críticos e sem serviços críticos, por máquinas físicas. Esse parâmetro é um dos objetivos a ser minimizado, e foi chamada de Dispersão. É uma medida de espalhamento das VMs de cada tipo nas MFs do *datacenter*, quanto mais concentradas cada tipo de VMs em menor número de MFs, menor o valor da dispersão.
- Realização de ajuste no nível de comprometimento da MF de forma dinâmica, dependendo dos tipos de VMs hospedados a cada instante;
- O desenvolvimento dos algoritmos *First Fit* e *Random Fit* com ajustes, para que pudessem fornecer parâmetros utilizados nas comparações entre as abordagens, que foram:

inserir o cálculo de consumo de energia e do nível de dispersão. Além desses, foi desenvolvido um outro algoritmo para gerar, de forma aleatória, um *datacenter* com suas VMs e MFs. Tendo como entradas: as quantidades desejadas de cada tipo dessas máquinas e, as configurações dos possíveis tipos de MFs e de VMs que poderão ser gerados aleatoriamente, além do percentual de VMs críticas que se deseja entre as VMs que serão geradas.

Acredita-se que os administradores das nuvens privadas terão diferentes possibilidades de disposição de VMs em seus ambientes, sendo possível obter maior garantia nos níveis de serviços oferecidos pelas empresas, já que se espera desempenho superior dos serviços implantados em VMs críticas quando comparado ao desempenho das VMs não críticas. Isso está relacionado aos diferentes níveis de sobrecarga de recursos, aplicados às máquinas físicas dependendo dos tipos de VMs hospedadas, sendo possível alterar tais valores nas configurações da abordagem aqui desenvolvida.

1.3 Estrutura do Documento

O restante deste documento de tese está organizado da seguinte forma:

- **Capítulo 2 - Fundamentação Teórica:** capítulo que aborda conceitos básicos e terminologias inerentes ao entendimento do atual trabalho. Além disso, são abordados alguns problemas intrínsecos às tecnologias, como também os benefícios que podem ser obtidos como o uso de tais abordagens;
- **Capítulo 3 - Trabalhos Relacionados:** neste capítulo é realizado um levantamento bibliográfico dos principais trabalhos relacionados encontrados na literatura. Os trabalhos foram agrupados por temas: (i) pesquisas que realizaram Otimização com Objetivo Único, (ii) pesquisas que realizaram Otimização de até Dois Objetivos, (iii) pesquisas que realizaram Otimização Não Simultânea de Múltiplos Objetivos, e (iv) pesquisas que realizaram Otimização Simultânea de Múltiplos Objetivos;
- **Capítulo 4 - A abordagem VMPOS:** trata de detalhes acerca da estrutura de *datacenter* considerada, da arquitetura da abordagem de alocação com múltiplos objetivos e da formulação do problema de posicionamento de VMs em MFs dentro de um DC;

- **Capítulo 5 - Resultados e Discussão:** apresenta detalhes sobre configurações aplicadas, nas abordagens VMPOS e nas que foram empregadas para validação, para os experimentos. Descreve as características dos *datacenters* utilizados nos experimentos. Os procedimentos estatísticos aplicados nas análises dos dados são apresentados. Os resultados dos experimentos de posicionamento de VMs, utilizando a VMPOS, são exibidos e discutidos, assim como os comparativos entre a abordagem VMPOS e as demais;
- **Capítulo 6 - Conclusões e Trabalhos Futuros:** por fim, neste capítulo são apresentadas as conclusões alcançadas através da realização da pesquisa, limitações e direcionamentos para possíveis trabalhos futuros;
- **Apêndices:** Nos Apêndices encontram-se os detalhes sobre configurações de entrada da abordagem VMPOS, do gerador de *datacenter* desenvolvido, os testes estatísticos e os resultados completos com todas as comparações realizadas além de alguns gráficos que complementam os resultados apresentados durante o texto.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão apresentados conceitos e características, de forma mais detalhada, das principais tecnologias utilizadas no atual trabalho.

2.1 Computação em Nuvem

Empresas de nuvem são utilizadas com base na política de pague pelo tempo e recursos que utilizar, prometendo provisionamento e escalonamento de recursos, através de serviços garantidos e confiáveis [5]. É um modelo que trouxe grandes mudanças com profundas implicações na forma como os serviços de Internet passaram a ser fornecidos, assim como em todas as áreas de TI [18].

Nesses ambientes existem dois papéis: o do *provedor* e o do *cliente*. De um lado, a detentora de massivos recursos, em grandes *datacenters*, que locam essas estruturas para os clientes. Do outro lado, temos os clientes com suas demandas computacionais, aplicações que geralmente possuem características de cargas flutuantes, difíceis de serem previstas, e que contratam os serviços das nuvens para manter e/ou executar suas demandas computacionais. Esses dois papéis compartilham o mesmo ambiente com interesses específicos [41].

Alguns dos principais aspectos relacionados à Computação em Nuvem serão tratados nas seções a seguir.

2.1.1 Definição

Devido à confusão criada sobre o que realmente seria Computação em Nuvem quando o termo surgiu, e diferentes conceitos começaram a surgir, o artigo “*Above the Clouds: A Berkeley View of Cloud Computing*” [4] apresentou uma definição mais apropriada:

“Computação em Nuvem refere-se tanto aos aplicativos como aos serviços fornecidos pela Internet e ao *hardware* e sistemas de *software* nos *datacenters* que fornecem esses serviços. Os próprios serviços têm sido conhecidos como *Software as a Service* (SaaS), por isso usamos esse termo. O *hardware* e o *software* do *datacenter* são o que chamaremos de Nuvem”. (FOX, A., 2009, p. 1, tradução nossa).

Uma outra definição de Computação em Nuvem, bastante utilizada e mais recente, foi elaborada pelo Instituto Nacional de Padrões e Tecnologias (*National Institute of Standard and Technologies* - NIST), importante agência do departamento comercial dos EUA. A definição do NIST [48] engloba vários aspectos essenciais desse modelo [18]:

“Computação em Nuvem é um modelo que permite o onipresente, conveniente, acesso sob demanda a um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços), que podem ser rapidamente provisionados e lançados com o mínimo esforço de gerenciamento ou interação do provedor de serviços”. (MELL, P., 2011, p. 2, tradução nossa).

Como definição do ramo empresarial sobre Computação em Nuvem, publica-se aqui o que está disponível no glossário de TI da “principal empresa de pesquisa e consultoria do mundo”⁸, a Gartner⁹, que diz:

“A Gartner define Computação em Nuvem como um estilo de computação em que recursos escaláveis e elásticos habilitados para TI são fornecidos como um serviço usando tecnologias da Internet”¹⁰ (tradução nossa)

⁸<http://www.gartner.com/technology/about.jsp>

⁹<http://www.gartner.com/>

¹⁰<http://www.gartner.com/it-glossary/cloud-computing>

Considerando a diversidade de definições que podem ser encontradas e para deixar o foco no ambiente utilizado neste trabalho, pode-se afirmar que nuvem privada, no contexto desta tese, consiste em um conjunto de recursos de *hardware* (como discos, equipamentos de rede, etc.) e aplicativos, existentes em um *datacenter* privado, que são fornecidos aos usuários através do acesso a máquinas virtuais disponibilizadas utilizando-se tecnologias da Internet.

2.1.2 Tipos de Nuvens

Com relação aos tipos de nuvens, existem três categorias principais [38][49]:

1. **Nuvens Privadas:** é uma plataforma que é dedicada e pertence a uma determinada organização, podendo ser mantida e gerenciada por essa organização, por terceiros, ou por alguma combinação destes. Pode estar localizada dentro ou fora das instalações da empresa [38][22][48][49];
2. **Nuvens Públicas:** este tipo é dedicado ao uso por parte de usuários públicos que se cadastram e utilizam a infraestrutura disponibilizada por um provedor. Pode ser de posse, gerenciada e operada por empresa, organização acadêmica ou governamental, ou alguma combinação entre estas [38][22][48][49]; e
3. **Nuvens Híbridas:** é o tipo que utiliza parte dos recursos da nuvem privada e, ao mesmo tempo, pode atender suas necessidades por mais recursos utilizando uma nuvem pública. Ou seja, é uma composição dos dois tipos anteriores [38][22][48][49].

2.1.3 Modelos de Serviços

Os modelos de serviços utilizados em ambientes de nuvens são classificados como *Software as a Service* (SaaS), *Platform as a Service* (PaaS) e *Infrastructure as a Service* (IaaS) [22][48], também chamadas de camadas da arquitetura das nuvens [5].

Características desses modelos são dadas a seguir [48]:

- ***Software as a Service* (SaaS):** Nesse modelo, o provedor da nuvem utiliza sua capacidade computacional para fornecer aplicativos, que são executados nos *datacenters* da nuvem, aos usuários. Estes, por sua vez, não gerenciam configurações como as dos

dispositivos de rede, discos ou qualquer configuração relacionada ao Sistema Operacional (SO) ou à infraestrutura da nuvem. As aplicações ficam acessíveis através de diversos dispositivos com acesso à Internet, como computadores, *smartphones* ou *tablets*, utilizando-se uma interface cliente;

- **Platform as a Service (PaaS):** Aqui o cliente tem a permissão para implantar aplicativos próprios, ou adquiridos, dentro da infraestrutura da nuvem, desde que exista suporte com os pré-requisitos necessários à correta implantação e execução da aplicação. O usuário continua sem acesso à infraestrutura da nuvem como, gerenciamento de servidores, rede, armazenamento ou configurações no SO, mas possuem total acesso às configurações de suas aplicações;
- **Infrastructure as a Service (IaaS):** Esse modelo permite que o usuário gerencie suas próprias VMs, utilizadas nas nuvens, de forma livre para que atenda a suas necessidades. O usuário continua sem acesso à infraestrutura da nuvem, mas possui total acesso ao seu servidor em VM, podendo configurar livremente o SO, possivelmente algumas configurações restritas no dispositivo de rede, discos e aplicativos de seu servidor virtual [5].

2.1.4 Problemas das Nuvens Públicas

Uma série de problemas, em ambientes de nuvens, ainda se encontram em aberto conforme apontam estudos recentes. Segundo [38], as nuvens públicas são a implementação mais vulnerável por estarem disponíveis para usuários públicos hospedarem seus serviços, os quais podem ser maliciosos. Em seu levantamento bibliográfico sobre segurança na virtualização, AbdElRahem et al. [23] afirma que computação em nuvem ainda enfrenta muitos desafios, principalmente na questão de segurança da nuvem, e que essa é a principal razão pela não adoção das nuvens por usuários em potencial.

Na sequência, são apresentados, de forma resumida, os pontos que mais preocupam usuários e comunidade científica com respeito ao uso das nuvens públicas.

- Com relação às técnicas de balanceamento de cargas, na revisão sistemática da literatura realizada em [39] foram elencados alguns problemas em aberto, dentre os quais

estão: 1) a inexistência de uma técnica única independente considerando todos os problemas envolvidos em balanceamento de cargas, e 2) conceitos importantes como gerenciamento de falhas e recursos de migração de tarefas tiveram pouca atenção nas técnicas por eles revisadas;

- Singh e Chana [6] afirmam que ainda existem muitos desafios e problemas relacionados ao agendamento de recursos e que ainda precisam ser solucionados;
- No aspecto segurança, Yan e Yu [1] em seu recente levantamento bibliográfico afirmam que os ataques em ambientes de nuvens públicas estão crescendo devido às suas características essenciais. Segundo os autores: “*a segurança tem sido considerada como a barreira dominante para o desenvolvimento da computação em nuvem*” (YAN, Q., 2016, p.602, tradução nossa).
- Outro recente levantamento [18] cita a existência de muitos problemas ainda não resolvidos, tanto técnicos quanto relacionados ao negócio, com problemas específicos identificados para cada modelo de serviço, principalmente relacionados à segurança (integridade e segurança dos dados e segurança da rede), privacidade (confidencialidade de dados, conhecimento sobre a localização física dos dados) e Acordos de Níveis de Serviço ou SLA (do inglês, *Service Level Agreement*). Outro ponto preocupante das nuvens públicas é o fato da multilocação (em inglês, *multitenancy*) que também pode comprometer seriamente a segurança e levar ao vazamento de informações sensíveis [10]. Segundo os autores, **são fatores que afastam parte dos usuários em potencial;**
- No artigo de Almorsy et al. [38], eles realizaram uma análise dos problemas de segurança em Computação em Nuvem alertando que, apesar dos ganhos em potencial conseguidos com o uso de nuvens públicas, o modelo de segurança ainda é questionável, o que impacta na adoção dessa opção. Eles alertam que o problema de segurança se torna mais complicado com a entrada de novas dimensões no escopo do problema, relacionadas ao modelo de arquitetura, multilocação, elasticidade e dependências da pilha de camadas.

2.2 Provisionamento de Recursos

Uma tarefa crucial e ao mesmo tempo desafiadora para um ambiente de Computação em Nuvem é o gerenciamento de recursos. Atualmente, as capacidades computacionais existentes nos *hardwares* de servidores são muito superiores às demandas regulares geradas pelos usuários. Apesar dessas capacidades serem muito melhor aproveitadas graças à virtualização [35], há a possibilidade de exauri-las caso a alocação de VMs ultrapasse a capacidade real de recursos existentes.

Existem diversos trabalhos na literatura que tratam do assunto e, ainda assim, pesquisadores encontram dificuldades para selecionar uma abordagem apropriada e eficiente no que diz respeito a escalonar recursos para, por exemplo, cargas de trabalho específicas [6].

O provisionamento eficiente de recursos fará uma diferença significativa no desempenho dos serviços ofertados por uma empresa de nuvem pública ou em um *Datacenter* privado.

Os *Datacenters* são a base dos ambientes de Computação em Nuvem nos quais os servidores reais (máquinas físicas) que hospedam as máquinas virtuais (VMs) ficam localizados. Um fator relacionado aos DCs é sua localização. Quanto mais próximo aos usuários, mais rápido será o tempo de acesso através da rede, tornando perceptível um melhor desempenho nos serviços utilizados [12].

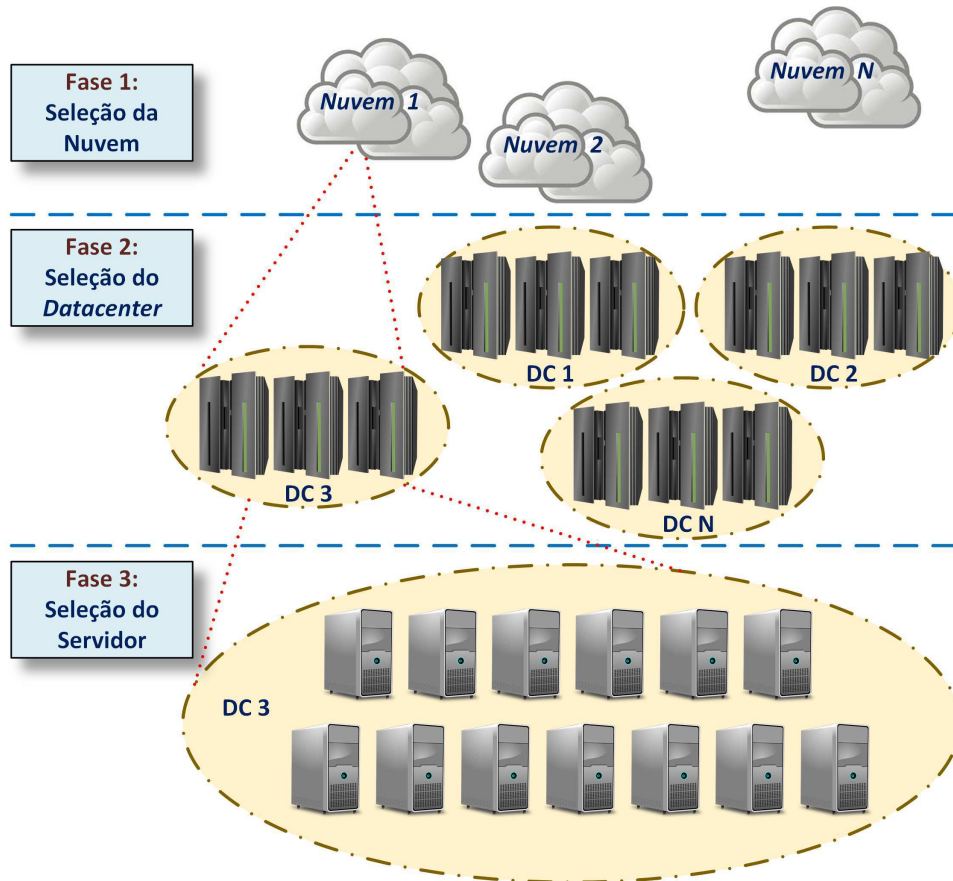
Além de tudo isso, é nos *datacenters* que ocorrem os reposicionamentos de VMs entre MFs, uma operação crítica que envolve migração de VMs, e tem por objetivo localizar a melhor máquina física para hospedar máquinas virtuais que estão sendo realocadas de posição. Uma operação desse tipo gera uma série de consequências, como efeito no desempenho dos serviços implantados na VM, elevação nos custos de energia e de refrigeração do ambiente, uma maior ou menor utilização de recursos, dentre outros. Muitos desses fatores influenciam nas despesas com a manutenção dos *datacenters* [11].

2.2.1 O modelo *top-down* de provisionamento de recursos em Nuvens

A decisão sobre a alocação de um recurso específico dentro de uma nuvem pode partir de um nível alto até a seleção de um recurso específico. No modelo de serviço IaaS, os recursos são disponibilizados como VMs na infraestrutura da nuvem, portanto, realizar provisionamento de recursos reduz-se a como posicionar uma VM na qual um usuário possa executar

suas demandas [12]. De forma esquemática, as decisões de onde posicionar uma VM podem ser vistas na Figura 2.1.

Figura 2.1: Fases para a realização do posicionamento de uma VM.



Fonte: Adaptado de ZHANG et al., 2016, p. 25.

Utilizando uma visão *Top-Down*, a decisão de alocação de uma VM em um servidor pode passar pelas seguintes fases, conforme ilustrado na Figura 2.1:

- **Fase 1:** Caso exista a possibilidade de utilizar recursos em mais de uma nuvem, esta será a primeira decisão a ser tomada, realizando a seleção de uma determinada nuvem, dentre diversas outras possíveis, na qual os recursos serão requisitados;
- **Fase 2:** Depois, passa-se a fase de seleção de um determinado DC no qual será posicionada a VM, dentro da nuvem previamente selecionada. Aqui, uma das decisões que podem influenciar na alocação é tentar alocar VMs de um mesmo usuário em um

mesmo *rack*. Com isso, restringe-se o tráfego de pacotes de rede dentro de uma mesma estrutura, o que ajuda na rede do DC como um todo;

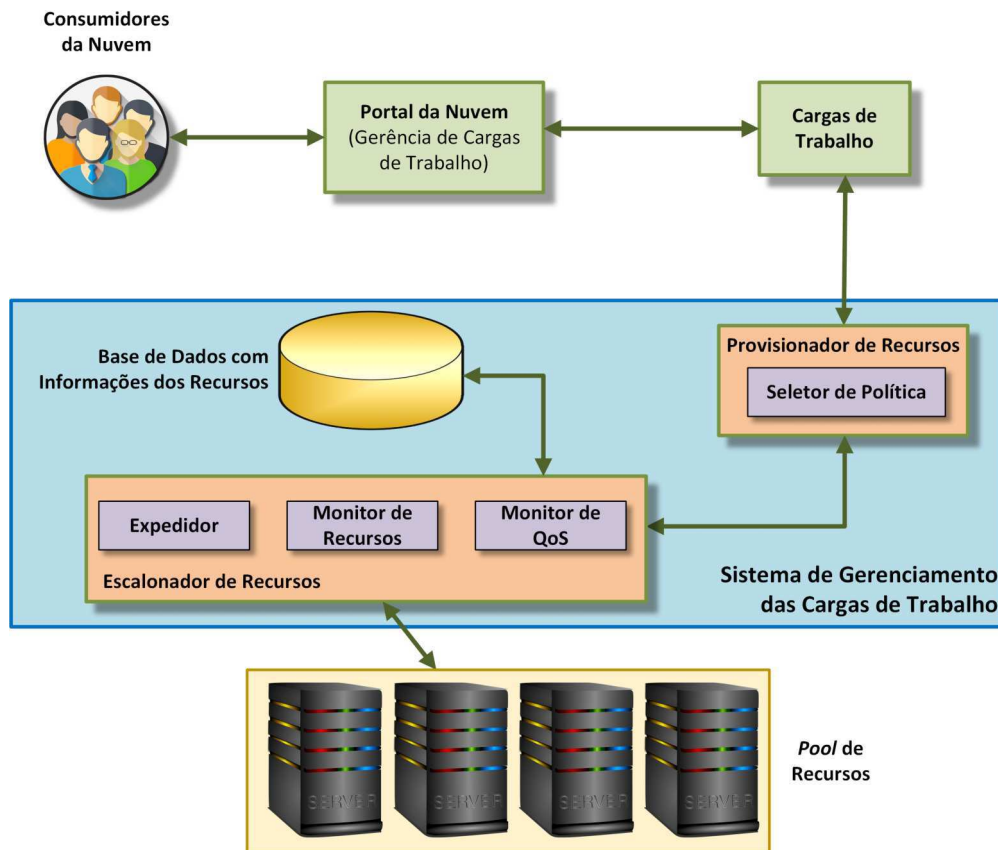
- **Fase 3:** Por último, chega-se à etapa de seleção de qual dos servidores físicos, dentro do DC selecionado, hospedará a VM [12]. Esta decisão pode ser pautada por diversos objetivos considerados pela empresa mantenedora do DC como, por exemplo, redução no custo energético, balanceamento de cargas, dentre outros. Como também ser gerenciada e ajustada dinamicamente, de acordo com a demanda por esses recursos [22].

Com relação às VMs, existem vários pontos de partida que vão depender, obviamente, desde o tipo de VM até a decisão de onde posicionar essa VM. Depende do tipo porque, por exemplo, as características de utilização de uma máquina *Desktop* são diferentes das de um Servidor, da mesma forma essa diferença ocorre quando esses tipos de máquinas estão implementadas em VMs. Entre os próprios servidores, os diversos tipos apresentam características diferentes (por exemplo, Bancos de Dados, Servidor de Arquivos, Servidor *Web*, etc.), cada um podendo requisitar mais, ou menos, de um ou outro recurso, ou seja, os aplicativos utilizados nesses sistemas demandam mais por alguns recursos computacionais do que por outros. Portanto, onde posicionar a VM é uma tarefa bastante complexa, que pode envolver vários objetivos como economia de energia, otimização do desempenho das aplicações que são executadas na VM, melhoria do desempenho da rede, otimização dos serviços da nuvem, etc.

O gerenciamento de recursos compreende então duas etapas: provisionamento e escalonamento de recursos. O *provisionamento de recursos* consiste em identificar recursos adequados, os quais receberão cargas de trabalho dos usuários, com base em critérios de Qualidade de Serviço (em inglês, *Quality of Service* - QoS) definidos pelo usuário.

Já o *escalonamento de recursos* refere-se ao mapeamento e implantação das VMs dos usuários, ou execução das cargas de trabalho, baseado na seleção de recursos previamente realizada na etapa de provisionamento. Essas etapas estão esquematizadas na Figura 2.2, considerando uma nuvem específica.

Figura 2.2: Esquema com o escalonamento de recursos em nuvem.



Fonte: Adaptado de SINGH e CHANA, 2016, p. 218.

Segundo o esquema da Figura 2.2, os usuários utilizam o portal da nuvem para submeterem suas cargas de trabalho e, com base nos detalhes fornecidos, o agente Provisionador de Recursos irá selecionar recursos, no *Pool de Recursos*, capazes de executar as cargas de trabalhos submetidas por um determinado usuário, levando em conta os critérios de QoS preestabelecidos. Terminada essa etapa, o agente envia requisições para o Escalonador de Recursos para que o agendamento do recurso seja realizado, após o provisionamento ter ocorrido com sucesso [6].

Diferentes critérios e parâmetros utilizados no gerenciamento de recursos levam a diferentes categorias de Algoritmos de Escalonamento de Recursos (em inglês, *Resource Scheduling Algorithms* ou RSAs) ou Sistemas de Alocação de Recursos (em inglês, *Resource Allocation System* ou RAS) [22]. Como este trabalho trata da questão de posicionamento de VMs em um *Datacenter*, ater-se-á apenas aos algoritmos que tratam de tal problema.

2.3 Abordagens Utilizadas para Validação

Para realizar a avaliação da abordagem desenvolvida decidiu-se por compará-la a dois métodos bem conhecidos de alocação de VMs, o *First Fit* e o *Random Fit*, além da abordagem desenvolvida por F. L. Pires e B. Barán [44], o iMaVMP.

Segundo [11] existem numerosos algoritmos de posicionamento de VM populares que são amplamente utilizados por outros esquemas de posicionamento para comparação e avaliação. O *First Fit* e o *Random Fit* são dois deles. Segundo [16] o *First Fit* é um algoritmo popular na literatura para alocação de recursos.

2.3.1 A Abordagem *First Fit*

A abordagem *First Fit (FF)* consiste em uma estratégia de alocação de VMs em máquinas físicas em que a lista de máquinas virtuais existentes, a serem alocadas nas máquinas físicas disponíveis, são lidas uma a uma e são hospedadas sequencialmente na primeira máquina física com recursos disponíveis para abrigá-la. Ou seja, enquanto a primeira MF tiver recursos suficientes para atender as demandas das VMs sendo alocadas, esta vai hospedando as VMs até o máximo que conseguir, respeitando a quantidade máxima de seus recursos. A partir do momento em que uma determinada VM requer mais recursos do que os existentes na primeira máquina física, o *First Fit* passa a alocar as VMs na próxima máquina física, e assim por diante. A escolha da próxima MF a ser utilizada segue a ordem sequencial das MFs de acordo com as informações do *datacenter* de entrada.

2.3.2 A Abordagem *Random Fit*

Uma outra forma de alocação de VMs em MFs consiste em ir alocando a lista de VMs de forma aleatória nos servidores físicos disponíveis. Então, para cada VM tomada sequencialmente seguindo a ordem das VMs de acordo com as informações do *datacenter* de entrada, sorteia-se um dos servidores físicos e tenta-se alocar a VM naquele servidor, caso as quantidades de recursos requeridas pela VM e disponíveis pela MF sejam atendidas. Em caso contrário, tenta-se alocar a VM em outra MF também selecionada aleatoriamente, e assim por diante, até que seja possível hospedar todas as VMs. Essa abordagem é chamada de *Random Fit (RF)*.

Por ser feita uma escolha aleatória do servidor físico que receberá a VM, utiliza-se o nome de *Random*, que significa aleatório.

2.3.3 A Abordagem *iMaVMP*

A sigla *iMaVMP* significa *Interactive Memetic Algorithm for Virtual Machine Placement*, que corresponde à *Algoritmo Memético Interativo para o Posicionamento de Máquinas Virtuais* (tradução nossa). Essa abordagem propõe otimização simultânea de cinco funções objetivos, quais sejam:

- i) *Minimização do Consumo de Energia*;
- ii) *Minimização do Tráfego de Rede*: representado pela soma do tráfego gerado por cada VM V_z , localizada em uma MF H_x , com outras VMs que estão hospedadas em uma MF diferente H_y ($x \neq y$);
- iii) *Maximização da Receita Econômica*: foi utilizada uma expressão para estimar a receita econômica total que um *Datacenter* recebe para atender as necessidades de seus clientes, representada pela soma da receita econômica obtida por cada máquina virtual que está efetivamente localizada para execução em uma máquina física;
- iv) *Qualidade de Serviço*: apenas utilizado como obrigatoriedade da VM estar alocada em alguma MF; e
- v) *Balanceamento de Carga da Rede*.

A abordagem aplica um algoritmo memético para encontrar os indivíduos na Fronteira de Pareto, oferecendo esta lista como solução do problema de posicionamento de VMs (mais detalhes no Capítulo 3 com os Trabalhos Relacionados).

O fato de ter um grande número de funções dificulta o processamento e qualidade das soluções visto que, quanto maior o número de funções objetivo, mais difícil de encontrar dominância entre as soluções [50]. Dessa forma, muitos casos irão ser classificados como não comparáveis, fazendo com que o número de indivíduos pertencentes a fronteira de Pareto cresça rapidamente (ver Seção 2.6).

2.4 Virtualização

Virtualização é uma técnica que não é nova em computação, foi inicialmente desenvolvida pela IBM nos anos 60 com o intuito de dividir logicamente a capacidade computacional dos antigos *mainframes* em unidades computacionais menores e independentes umas das outras [51]. Dessa forma, era possível reduzir os altos custos dos *mainframes* da época, já que o particionamento lógico permitia realizar o que se conhece como multitarefa, que é a capacidade de executar aplicações e processos múltiplos ao mesmo tempo. Com a popularização dos computadores pessoais e dos processadores x86, devido aos preços acessíveis, além dos sistemas operacionais terem convergido para umas poucas famílias (Microsoft, Macintosh e Unix), cada um com seus aplicativos e públicos alvos, as técnicas de virtualização perderam força [52][53].

O que se observou na sequência foi que as empresas passaram a investir em *desktops* de baixo custo para suas demandas computacionais. Com isso, novas questões surgiram:

- **Segurança:** cada *desktop* era utilizado de forma individualizada, e cada usuário era responsável por manter atualizações de sistemas, *softwares* e novos *patches* de segurança lançados, sendo difícil saber se esses procedimentos estavam sendo corretamente aplicados em todos os *desktops*, por todos os usuários de uma empresa;
- **Utilização:** o contínuo avanço das tecnologias nos processadores e dispositivos de memória RAM contribuíram para uma crescente subutilização dos *desktops*. Os usuários normalmente utilizavam um número reduzido de *software* no dia a dia, subutilizando os recursos disponíveis nos *desktops*;
- **Custos:** de acordo com a utilização comentada no item anterior, cada usuário possuía um *desktop* pessoal gerando alto custo total de equipamentos devido à contínua aquisição de novos *desktops*. Além disso, outros custos agregados eram gerados relacionados a atualização de *software*, atualização de *hardware* e consumo energético;
- **Ociosidade da Infraestrutura:** Com o tempo, o uso dos computadores conectados passou a exigir uma quantidade grande de servidores, os quais mantinham serviços heterogêneos demandados pelos usuários da rede. Muitas vezes, cada servidor era responsável por apenas um serviço e, com isso, a carga de trabalho submetida a tais servidores

normalmente não exigia todo o potencial dos recursos de *hardware* disponíveis, passando a ser um desperdício de capacidade computacional.

Devido a fatores como os citados, as técnicas de virtualização retornaram como uma possibilidade para melhor utilização das novas capacidades computacionais dos *desktops* e servidores que estavam sendo subutilizadas.

2.4.1 Definição

A virtualização de computadores permite executar, simultaneamente, múltiplas instâncias de sistemas operacionais independentes sobre o mesmo *hardware* compartilhado, de forma isolada, permitindo dessa forma uma melhor utilização dos recursos computacionais e uma maior flexibilidade [54][55]. A máquina física que recebe as VMs é chamada de *hospedeiro*, e as VMs que executam sobre esse *hardware* são chamadas de *convidados*. A virtualização é realizada adicionando-se uma camada de *software* posicionada entre o *hardware* da máquina física e o SO que o utiliza, oferecendo um ambiente completo muito semelhante ao que utilizamos em uma máquina física. Essa camada é chamada de Monitor de Máquina Virtual (em inglês, *Virtual Machine Monitor* - VMM) ou *hypervisor* [23][21][56]. Esse *hypervisor* é o responsável por implementar o *hardware* virtual e por executar os SOs convidados.

Virtualização é considerada uma tecnologia chave dentro de Computação em Nuvem para distribuir aos usuários recursos sob demanda como CPU, memória, rede, etc., de forma eficiente, flexível e transparente e a um custo também flexível. Com o uso de virtualização, administradores de nuvem podem gerenciar os servidores de forma dinâmica, escalando os recursos físicos para mais ou para menos, de acordo com as demandas existentes [21].

Com o passar do tempo, o termo máquina virtual englobou muitas abstrações, como por exemplo, a Máquina Virtual Java (em inglês, *Java Virtual Machine* - JVM) [57]. Aqui, os termos "*virtualização de servidor*" ou "*virtualização de máquina*" são utilizados para descrever a possibilidade de executar uma máquina virtual completa, incluindo seu próprio SO, em um servidor físico com SO próprio, podendo esse último ser igual ou diferente do SO da VM. Portanto, aqui, os termos *virtualização de servidor* ou *virtualização de máquina* serão utilizados para identificar técnicas de virtualização utilizadas por tecnologias de virtualização

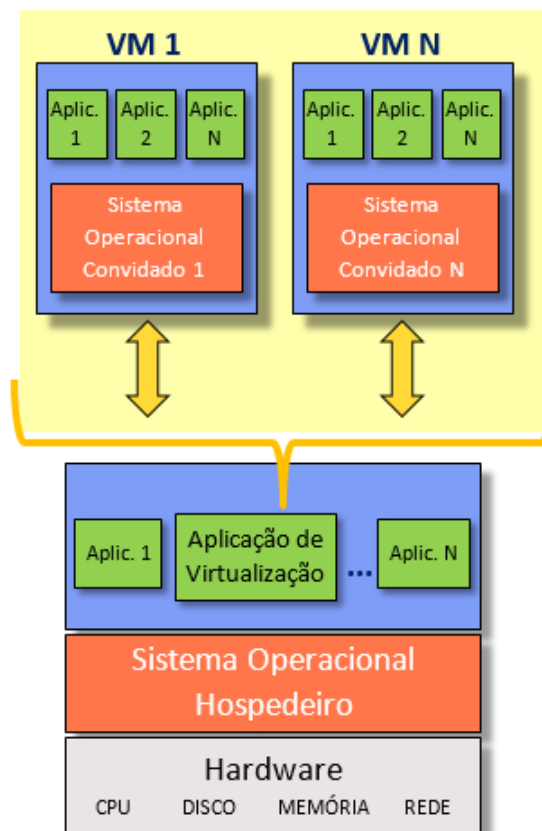
como, por exemplo, Xen, KVM, Virtual PC e VMware.

2.4.2 Tipos de Virtualização

As características de cada tecnologia de virtualização diferem na forma como a camada de virtualização está implementada. Em geral, as tecnologias de virtualização podem ser divididas em quatro tipos não disjuntos [21][23], descritos logo a seguir.

Na **virtualização completa** (Figura 2.3) uma aplicação padrão, que é executada no sistema operacional hospedeiro e gerencia as VMs, fornece uma réplica do *hardware* do hospedeiro, não sendo necessário modificações no *software* ou no SO da máquina convidada.

Figura 2.3: Virtualização completa.



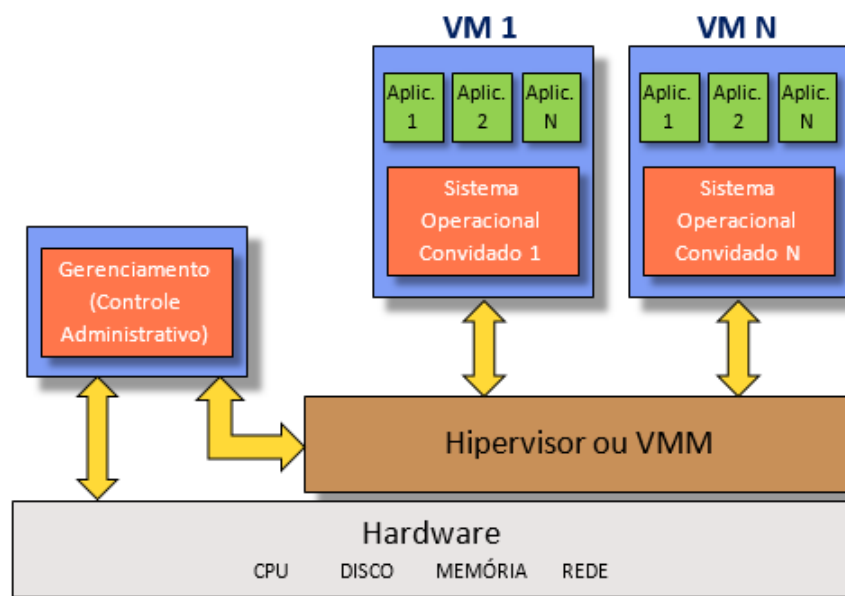
Fonte: Adaptado de CARISSIMI, 2008, p. 184.

Dessa forma, SO e *software* da VM executam sobre o *hardware* virtual, de forma isolada, como se fosse o *hardware* real. Contudo, as instruções precisam passar através do *hypervisor* para poder chegar aos recursos de *hardware* do hospedeiro. O fato de não precisar realizar

alterações no SO do convidado é considerada a principal vantagem dessa técnica. Exemplos de aplicações que realizam esse tipo de virtualização são o Virtual Box¹¹ e o *VMware Workstation Player* da VMware¹²;

No caso da **para-virtualização**, um *hypervisor* de baixo nível é carregado durante o processo de *boot*, antes das VMs, e executa direto sobre o *hardware* hospedeiro (Figura 2.4). O *hypervisor* trata as solicitações de acesso a recursos de *hardware* do sistema hospedeiro, captura e manipula instruções protegidas ou privilegiadas [58]. Nesse caso, o *hypervisor* realiza o que também é chamado de *para-virtualização*. Na para-virtualização, é necessário realizar alterações nos SOs do hospedeiro e dos convidados para oferecer desempenho superior através de acesso privilegiado ao *hardware* real. Como o SO do convidado é modificado para chamar o *hypervisor* quando necessário, espera-se desempenho superior dessa técnica se comparada a da virtualização completa.

Figura 2.4: Para-virtualização.



Fonte: Adaptado de CARISSIMI, 2008, p. 184.

Por outro lado, faz-se necessário que o código do SO convidado esteja disponível, sendo considerada a grande desvantagem dessa modalidade. Um *hypervisor* que utiliza essa forma de virtualização é o Xen¹³.

¹¹<https://www.virtualbox.org/>

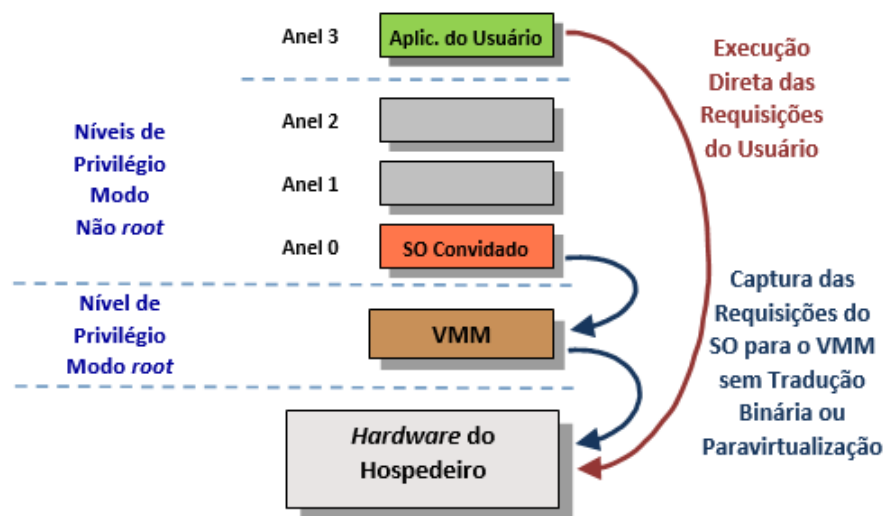
¹²<https://www.vmware.com/>

¹³<https://www.xenproject.org/>

Com a **virtualização assistida por hardware** os fabricantes de *hardware* começaram a melhorar os recursos de virtualização em seus processadores, adaptando seus dispositivos para melhorar a realização dos processos de virtualização. Como exemplos de processadores com esses recursos tem-se, o Intel VT e o AMD-V, desenvolvidos com novos modos de execução nos quais chamadas específicas de CPU não precisam ser tratadas pelo *hypervisor*, mas sim ser enviadas e executadas diretamente pela CPU. Dessa forma, um melhor desempenho das VMs é observado devido a diminuição nas tarefas desenvolvidas pelo *hypervisor* que não é demandado para realizar a tradução de algumas chamadas (Figura 2.5).

Na virtualização assistida por *hardware* se faz necessário o suporte explícito da CPU, e esta característica está disponível apenas em algumas famílias de processadores. Para aplicar o modelo de virtualização assistida por *hardware* e hospedar sistemas operacionais não modificados, é preciso realizar muitas capturas por parte do *hypervisor* causando muita carga na CPU, podendo influenciar na escalabilidade e eficiência de consolidação de servidores, o que pode ser considerado como uma desvantagem.

Figura 2.5: Virtualização Assistida por *Hardware*.

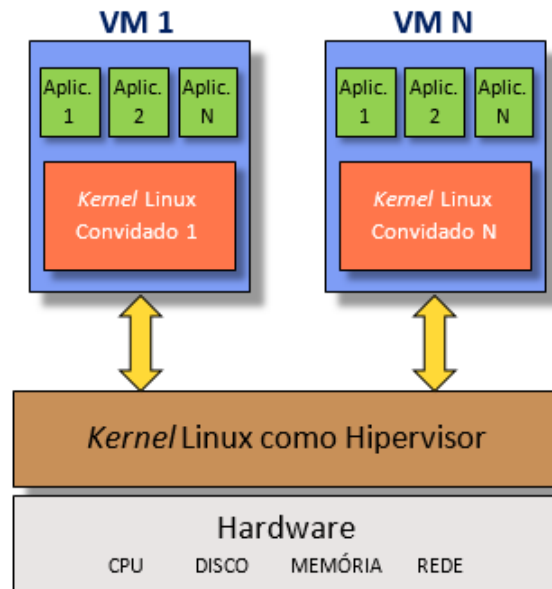


Fonte: Adaptado de ABDELRAHEM, 2016, p. 34.

Considerando a abordagem de **virtualização no Nível de Kernel**, o *kernel* Linux executa as máquinas virtuais como qualquer outro processo no espaço do usuário [59]. Contudo, para isso é necessário que os SOs, bibliotecas e utilitários dos convidados tenham sido compilados para o mesmo *hardware* e conjuntos de instruções do *kernel* Linux da máquina hospedeira,

o qual foi compilado para máquina física em que as VMs estão sendo executadas [58]. Esse tipo de virtualização pode ser visto na Figura 2.6. Um *hypervisor* que realiza essa forma de virtualização é o *Kernel-based Virtual Machine* ou KVM¹⁴.

Figura 2.6: Virtualização no Nível de *Kernel*.



Fonte: Adaptado de HUYNH, 2010, p. 2.

2.4.3 Benefícios da Virtualização

São muitos os benefícios alcançados com o uso da virtualização. Uma característica importante que fez com que houvesse uma ampla utilização dessa tecnologia, em ambientes computacionais, é a capacidade de isolamento e migração de cargas de trabalho entre *hardware* [60].

Como alguns dos outros benefícios no uso de virtualização em *datacenters* cita-se:

- Possibilita o provisionamento de novos servidores de forma automática, assim como, deixa o gerenciamento dos servidores mais rápido, permitindo que a equipe de TI utilize o tempo livre com desenvolvimento e outras atividades;
- Permite confiabilidade, já que se uma das VMs em um servidor físico apresentar problemas, as demais não serão prejudicadas. Aumenta efetivamente o nível de segurança

¹⁴<https://www.linux-kvm.org/>

no sentido de gerenciamento centralizado;

- Permite a migração de máquinas virtuais entre máquinas físicas, sem a necessidade de interrupção dos serviços implantados na VM. Essa característica possibilita realizar o balanceamento de cargas, migrando VMs entre diferentes hospedeiros, com diversas possibilidades de configuração, visando diferentes objetivos como: melhor desempenho, economia de energia, redução de custos, etc. Também é através da migração de VMs que ocorre a consolidação de servidores físicos, reduzindo custos como os de atualização de *hardware*, consumo de energia, equipamentos de refrigeração e gerenciamento, como também otimiza a utilização de espaço físico, contribuindo com a redução nas emissões de dióxido de carbono;
- Permite a realização de manutenção nos servidores físicos devido a problemas ou para realização de atualização de *hardware*, como também a redução do número de outros equipamentos de rede, como *switches*, o que também irá contribuir para a redução do consumo de energia. Da mesma forma, facilita na recuperação e reativação dos servidores em caso de desastres;
- Possibilita integrar ambientes de testes e desenvolvimento com os ambientes de produção.

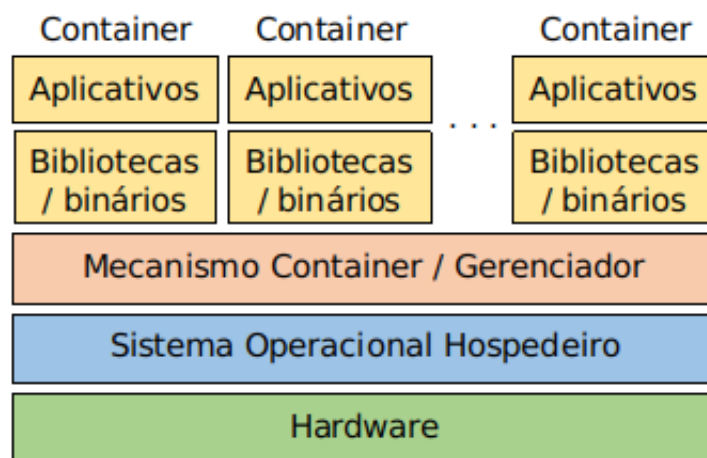
2.5 Containers

Uma forma de virtualização que surgiu e tem se popularizado desde sua implementação é a virtualização baseada em *containers*. Estes são leves em relação ao consumo de recursos da máquina física que os hospeda, proporcionam isolamento e permitem fácil processo de portabilidade. Uma aplicação que executa em um *container* tem acesso, de forma não compartilhada, a uma cópia do Sistema Operacional.

Desta forma, os *containers* virtualizam o SO enquanto a virtualização de máquinas (ver seção 2.4) virtualiza recursos de *hardware* [61][62]. Tal fato possibilita que os *containers* produzam pouca ou nenhuma sobrecarga e, conseqüentemente, maior grau de implantação, comparativamente à virtualização utilizando máquinas virtuais [63].

Como principal diferença entre as tecnologias de máquinas virtuais e a de *containers* pode-se apontar que, para as máquinas virtuais, cada aplicação virtualizada está em um ambiente que inclui um sistema operacional convidado completo e as bibliotecas/binários necessárias. Ao passo que, um ambiente *container* possui apenas a própria aplicação e suas dependências (bibliotecas/binários). Isso pode ser verificado no esquema da Figura 2.7, a qual representa *containers* como instâncias no espaço do usuário, que coexistem com um grupo de processos isolados, e compartilham o núcleo de um sistema operacional.

Figura 2.7: Virtualização baseada em *Containers*.



Fonte: Adaptado de GKORTZIS et al., 2016, p. 534.

Uma desvantagem importante dos *containers* é o fato de não poderem executar diretamente Sistemas Operacionais ou Núcleos diferentes daqueles da máquina hospedeira, por exemplo, *containers* Windows não podem executar sobre um hospedeiro Linux. Outras dificuldades importantes dos *containers*, apontadas por [62], estão relacionadas à problemas de I/O e segurança, no tocante às *permissões e privilégios e controle de acesso*, os quais foram os itens com maior número de vulnerabilidades reportadas.

É importante lembrar que, ainda nos dias atuais, se faz necessário dar suporte a uma variedade de diferentes sistemas operacionais, para acomodar a heterogeneidade observada nas aplicações populares [55].

Atualmente existem duas formas, não diretas, de executar *containers* Windows sobre hospedeiros Linux e vice versa, conforme indicado em roteiro da Microsoft disponível na

Internet¹⁵. Contudo, as duas formas apresentadas precisam usar máquinas virtuais para conseguir manter os dois sistemas sobre o mesmo SO.

Na questão da segurança, pensando em bancos de dados, a arquitetura dos *containers* não pode garantir segurança porque todos os *containers* compartilham o mesmo núcleo e, portanto, são menos isolados do que as máquinas virtuais. Isto significa dizer que um erro no *kernel* irá afetar todos os *containers* presentes na máquina hospedeira, podendo resultar em perda de dados [62].

Outro ponto relevante é que em ambientes de nuvem, tanto públicas quanto privadas, tem-se uma variedade de configurações predeterminadas de VMs, as quais são utilizadas de acordo com as necessidades das aplicações dos usuários. E esta variedade vai de configurações de *hardware* de VMs muito básicas até máquinas mais robustas, o que significa que em alguns casos as VMs podem ser bastante "leves". Além disto, como se tem um ambiente completo em uma única VM, é possível configurá-la de acordo com as necessidades das aplicações executadas pelo usuário dono da VM.

Dadas as características apresentadas sobre *containers* e levando-se em conta um ponto bastante relevante, que é o atual nível de maturidade das ferramentas de gerenciamento de máquinas virtuais, as quais possibilitam a realização de migração em tempo real, *snapshotting* e redimensionamentos, processos já bem maduros em monitores de máquinas virtuais [62], ainda é possível atribuir vantagem para utilização de máquinas virtuais em detrimento dos *containers*, ao menos para o cenário considerado neste trabalho.

As duas abordagens (máquinas virtuais e *containers*) possuem benefícios e desvantagens, como apontadas nas seções anteriores. Para este trabalho, foi escolhido utilizar as tecnologias associadas às máquinas virtuais por seu maior grau de maturidade, possibilidade de hospedar diferentes VMs, de diferentes proprietários e com possíveis diferentes SOs, sob um mesmo hospedeiro, além de proporcionar maior segurança aos dados, quando manipulados por VMs.

Porém, nada impede que *containers* sejam utilizados nas VMs do *datacenter*, se beneficiando dos dois mundos, já que isso é possível.

¹⁵Disponível em: <https://docs.microsoft.com/en-us/virtualization/windowscontainers/deploy-containers/linux-containers>. Acesso em: 07 maio 2018.

2.6 Otimização com Múltiplos Objetivos

Otimização é uma técnica matemática, utilizada na ciência e engenharia, para a qual diversos métodos já foram criados. Dentre eles, muitos dos métodos tradicionais foram desenvolvidos para otimização de tarefas com objetivo único. Contudo, existem diferentes problemas no mundo real que exigem otimização levando-se em conta vários propósitos que precisam ser otimizados simultaneamente, podendo haver restrições impostas a esses fins. Isto tem ocasionado o desenvolvimento de uma gama de algoritmos para resolver os Problemas de Otimização com Múltiplos objetivos (POM).

Algumas técnicas podem ser aplicadas para buscar a solução de um problema quando se deseja atender a um conjunto de objetivos simultaneamente. Uma delas consiste basicamente em converter um POM em um problema mono objetivo. Dependendo da especificação do problema, se de minimização ou maximização, escolhe-se como resultado a solução que obteve valor de acordo com essa especificação (a de maior ou de menor).

Uma outra técnica é aplicar a Fronteira de Pareto (ver Seção 2.6.1), em que as soluções são evoluídas até o ponto em que melhorar um objetivo necessariamente piora o resultado de outro objetivo, e escolher uma dentre as soluções que estão sobre essa fronteira. Uma forma de fazer isso é aplicar algoritmos evolucionários que partem de alguma(s) solução(ões) básica(s) e evoluem em direção às melhores soluções.

Como outra técnica também é possível otimizar cada objetivo individualmente, um após o outro, até que todos sejam evoluídos e se consiga chegar a uma solução final.

Segundo [44], quando se considera o contexto de alocação de máquinas virtuais em um *datacenter*, algumas abordagens de otimização são utilizadas na aplicação da função objetivo, como: (1) Objetivo único, (2) Múltiplos objetivos resolvidos como objetivo único e (3) Múltiplos objetivos pura. A abordagem com *objetivo único* considera a otimização de apenas um único objetivo, ou a otimização individual de vários, mas, aplicando um por vez.

No caso da abordagem que considera *múltiplos objetivos resolvido como objetivo único*, a estratégia é combinar os múltiplos objetivos em uma expressão única, que se reduz a um único objetivo. Isso pode ser feito aplicando-se pesos a uma soma dos objetivos normalizados. Por outro lado, a abordagem com *múltiplos objetivos pura*, considera uma otimização simultânea dos diferentes objetivos, possivelmente, mas não necessariamente conflitantes.

2.6.1 POM

Um POM é um problema que tem dois ou mais objetivos que precisam ser otimizados simultaneamente, e devem existir restrições impostas a estas metas, mesmo se houver conflitos entre elas [64]. Otimização pode envolver a maximização de todas as funções objetivo, a minimização de todas as funções, ou uma combinação de maximização e minimização dessas funções.

Um POM puro inclui um conjunto de p variáveis de decisão, q funções objetivo e r restrições. Com as restrições e funções objetivo sendo funções das variáveis de decisão.

Na formulação de um POM, x representa o vetor de decisão e y o vetor objetivo [65][64][44], da seguinte forma:

Otimize:

$$y = f(x) = [f_1(x), f_2(x), \dots, f_q(x)], \quad (2.1)$$

sujeito a,

$$e(x) = [e_1(x), e_2(x), \dots, e_r(x)] \geq 0, \quad (2.2)$$

em que:

$$x = [x_1, x_2, \dots, x_p] \in X, \quad (2.3)$$

$$y = [y_1, y_2, \dots, y_q] \in Y. \quad (2.4)$$

O espaço de decisão é representado por X e o espaço objetivo por Y .

Assim, $x \in \mathbb{R}^p$ é o vetor das p variáveis de decisão, $f_i(x)$ é uma das q funções objetivo e $e(x) \geq 0$ é o conjunto das r restrições que constituem o espaço de soluções possíveis. Denota-se por \mathbb{R} o conjunto dos números reais e por \mathbb{R}^p o conjunto de vetores reais p -dimensionais. Portanto, a otimização com múltiplos objetivos visa otimizar o vetor de funções objetivo.

Por conseguinte, a solução de um POM não é um ponto, mas, um conjunto de pontos que representam os melhores conflitos de escolha entre as funções objetivo.

Pode-se dividir o conjunto de possíveis soluções em: *Soluções dominadas* e *Soluções não-dominadas*. Dadas duas soluções viáveis x e $x^* \in X$, a solução x é dita uma *solução dominada* se houver uma solução x^* que obedeça a:

$$f_i(x^*) \leq f_i(x), \quad \forall i, 1 \leq i \leq q \quad (2.5)$$

$$\exists i \mid f_i(x^*) < f_i(x). \quad (2.6)$$

Ou seja, $f_i(x^*)$ é:

- Pelo menos tão boa quanto $f_i(x)$ em cada dimensão; e
- Melhor que $f_i(x)$ em pelo menos uma dimensão.

A expressão em 2.5 assume um caso de minimização para as funções objetivos, para os outros casos as definições são aplicadas de forma similar.

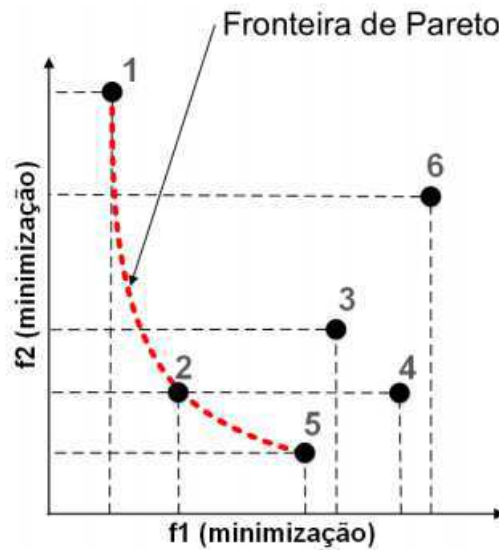
Quando uma solução x^* domina uma outra solução x , a representação utilizada é $x^* \prec x$, a qual indica que x^* domina x . Se x^* não domina x , e x não domina x^* , as soluções são ditas *não comparáveis*, e denota-se $x^* \sim x$.

Se forem reunidas em um conjunto todas as soluções não-dominadas de um POM, no espaço de decisão, esse conjunto é chamado de conjunto eficiente ou **Conjunto Ótimo de Pareto** P_c . O conjunto correspondente de vetores objetivo constitui a **Fronteira Ótima de Pareto** PF_c . Em um POM, busca-se um algoritmo que encontre um conjunto de soluções não-dominadas que se aproxime o máximo possível da fronteira ótima de Pareto, e que não possa ser melhorado.

Uma das possíveis técnicas de solução utilizadas para um POM é a **Fronteira de Pareto**. Cada uma das soluções em uma fronteira ótima de Pareto apresenta diferentes compromissos de qualidade entre os objetivos que estão sendo considerados.

Na Figura 2.8 encontra-se um dado conjunto de soluções no qual é possível realizar comparações entre quaisquer pares de soluções usando a definição de dominância apresentada. Como exemplo, se forem comparados os pontos 2 e 4 chega-se à conclusão que o ponto 2 é melhor em relação à função f_1 , o que não ocorre em relação à função f_2 visto que elas apresentam o mesmo resultado. Agora, considerando os pontos 2 e 3 nota-se que a solução 2 é melhor que a solução 3 em ambos os objetivos, e pode-se dizer que a solução 3 é uma solução dominada pela solução 2. Se a comparação for entre as soluções 4 e 5, é evidente que a solução 4 é dominada pela solução 5, sendo esta última melhor em ambos os objetivos, portanto, a solução 4 deve ficar fora da fronteira. De acordo com a Figura 2.8, os pontos que pertencem a Fronteira de Pareto, ou seja, não dominados por nenhuma outra solução, são os pontos 1, 2 e 5 [66].

Figura 2.8: Exemplo de Fronteira de Pareto com dominância de soluções.



Fonte: ISHIBASHI, 2013, p. 44.

Quando todos os objetivos considerados são importantes e fazem parte da fronteira de Pareto, não se pode selecionar uma solução favorecendo um objetivo sem piorar os demais. Esse compromisso de qualidade (em inglês, *tradeoff*) entre as soluções não dominadas é interessante por encontrar um leque de soluções consideradas as melhores dentre todas as soluções, mas exige a definição por parte de um tomador de decisão de qual solução será adotada para o problema em questão.

2.6.2 Algoritmos Genéticos (AG)

Para solucionar um POM, os Algoritmos Evolucionários com Múltiplos Objetivos (em inglês, *Multi-Objective Evolutionary Algorithms - MOEAs*) têm sido utilizados, visto que podem lidar simultaneamente com um conjunto de soluções possíveis em uma única etapa [64]. Dentro desse nicho, Algoritmos Genéticos (AG) despontam como os que são predominantemente aplicados [64][46][34][67].

Algoritmos Genéticos são inspirados na teoria da evolução de Darwin, sendo os mais antigos, mais utilizados e mais conhecidos dos algoritmos evolucionários [68]. Solucionar um problema com AG é fazer uso de um processo evolucionário. Decidiu-se usar AG neste trabalho por ser uma abordagem muito utilizada e conhecida, aplicada nos trabalhos do estado da arte [69][44][17] em problemas semelhantes ao tratado nesta pesquisa.

Um Algoritmo Genético é composto de uma cadeia de caracteres codificados, chamados de cromossomos, em que cada um deles representa uma solução no espaço de busca, aos quais são atribuídos valores de aptidão (em inglês, *fitness*) que reflete a qualidade da solução em relação aos objetivos de otimização.

As soluções são melhoradas através de uma população de soluções candidatas que é evoluída, de forma iterativa, para melhorar a solução. A cada iteração, chamada de **geração**, são criadas cadeias de caracteres a partir de operações genéticas, podendo ser por cruzamento ou por mutação, e inseridas no conjunto de soluções atual, formando a nova geração. Operações de mutação ocorrem quando são aplicadas, a um simples cromossomo, ações que selecionam e alteram parte de sua cadeia de caracteres, de forma aleatória. Para realizar as operações chamadas de cruzamento, duas soluções são selecionadas como pais e partes de suas cadeias de caracteres são misturadas para formar novas cadeias, que serão novas soluções [67].

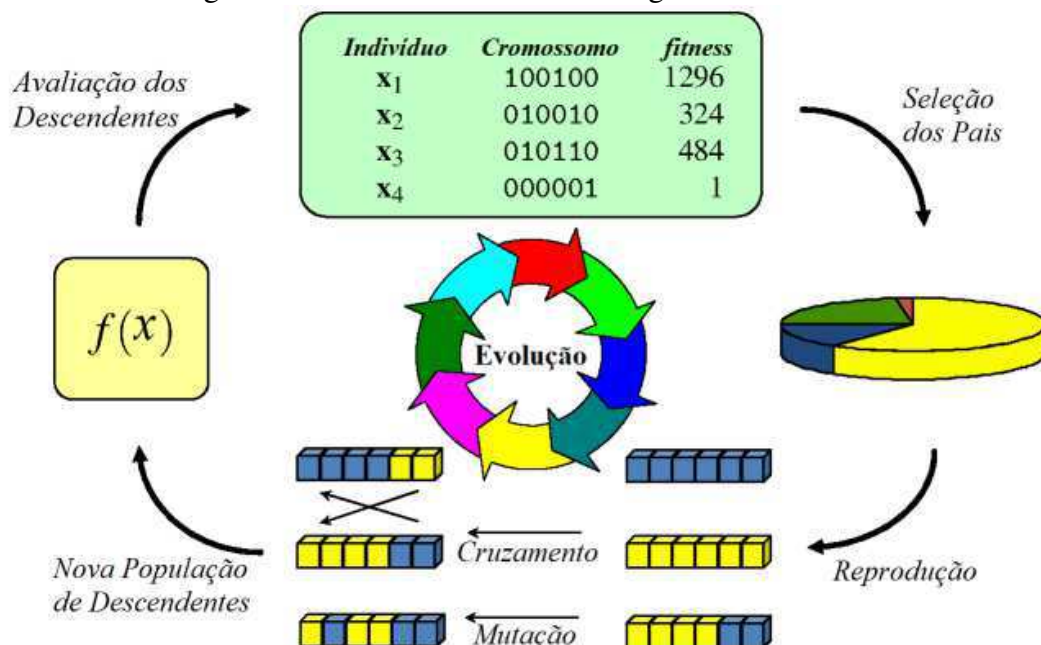
O processo básico de funcionamento de um Algoritmo Genético [65] é:

1. Inicialização: Criação de uma população inicial. Normalmente gerada de forma aleatória, podendo ser de qualquer tamanho desejado;
2. Avaliação: Cada membro da população é avaliado calculando-se seu *fitness*. É um valor que indica quanto o indivíduo se encaixa com os requisitos estabelecidos previamente;
3. Seleção: É desejável melhorar constantemente a aptidão geral da população. O processo de seleção é responsável por descartar os indivíduos ruins e manter apenas os melhores indivíduos na população. Existem alguns métodos para se fazer seleção, todos com o mesmo objetivo: fazer com que os indivíduos aptos tenham maiores chances de serem selecionados para a próxima geração. Por isso, cromossomos pais são selecionados com probabilidade relacionada a seu *fitness*, logo, os que tiverem maiores *fitness*, terão maiores chances de serem selecionados;
4. Cruzamento: É o processo pelo qual novos indivíduos são criados pela combinação de aspectos de outros indivíduos selecionados. A ideia é como na natureza, na qual combinando os melhores traços de cada um dos pais, espera-se criar descendentes aptos ainda melhores;

5. **Mutação:** Para evitar que cada combinação de soluções criada por cruzamentos acabe resultando na população inicial, é necessário incluir um pouco de aleatoriedade na população. Por isso, utiliza-se o processo de mutação, o qual realiza pequenas mudanças aleatórias ao genoma de um indivíduo;
6. **Repetição:** Terminada a sequência anterior, agora tem-se uma nova geração, podendo-se recomeçar a partir do passo 2, até que se alcance uma condição de parada determinada.

A Figura 2.9 ilustra esquematicamente o processo descrito nos passos anteriores.

Figura 2.9: Ciclo evolutivo de um Algoritmo Genético.



Fonte: adaptado de F. S. CAPARRINI, 2018. Disponível em: <http://www.cs.us.es/~fsancho/?e=65>. Acesso em: 04 out. 2017.

Uma condição de parada pode ser a de se conseguir a melhor geração possível de se alcançar, ou seja, o algoritmo encontrou uma solução boa o bastante, a qual atende aos requisitos mínimos exigidos e é não-dominada. Outras condições de parada, para que o algoritmo não continue buscando por uma solução indefinidamente, podem ser: o número de gerações sem obter evolução, custo ou tempo.

No processo de Seleção dos cromossomos pais, citado anteriormente, pode-se utilizar uma roleta, a qual será girada (através de uma simulação computacional) e, dependendo de onde parar, selecionar-se-á um indivíduo pai.

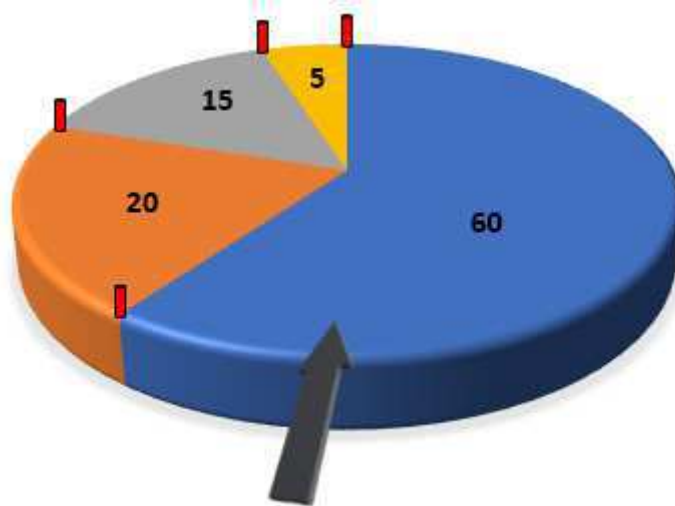
Para exemplificar o funcionamento da roleta, suponha uma população com quatro indivíduos e que a *fitness* de cada um seja como informado na sequência:

Indivíduo 1: Fitness = 5 *Indivíduo 2: Fitness = 15*

Indivíduo 3: Fitness = 20 *Indivíduo 4: Fitness = 60*

Suponha ainda que o objetivo seja encontrar o indivíduo de maior *fitness* logo, conforme indicam os valores, o indivíduo com maior *fitness* é o 4 e o que apresenta o menor, é o 1. Então, constrói-se uma roleta em que a área correspondente a cada indivíduo, é proporcional ao *fitness* individual. Para o caso deste exemplo, a roleta pode ser vista na Figura 2.10.

Figura 2.10: Representação de uma roleta de seleção em AG. Cada fatia destinada a cada indivíduo é proporcional a seu *fitness*.



Fonte: Elaborada pelo autor.

Da Figura 2.10, observa-se que o *Indivíduo 1* tem apenas 5% de chances de ser selecionado, ao passo que o *Indivíduo 4* tem 60% de chances de ser selecionado. As chances de ser selecionado estão diretamente ligadas ao *fitness* de cada indivíduo, quanto maior o *fitness*, maiores as chances.

Um exemplo de pseudocódigo de como selecionar um pai utilizando a roleta, para o exemplo com quatro indivíduos, pode ser visto no Algoritmo 1.

Algoritmo 1: Algoritmo exemplo de uma roleta

```

início
1  Gere aleatoriamente um valor real  $r \in [0, 1]$ ;
2  se  $r < 0,05$  então
3    | pai=1;
   senão
4    | se  $r < 0,15$  então
5      | pai=2;
     senão
6      | se  $r < 0,20$  então
7        | pai=3;
       senão
8        | pai=4;
        fim
     fim
   fim
fim

```

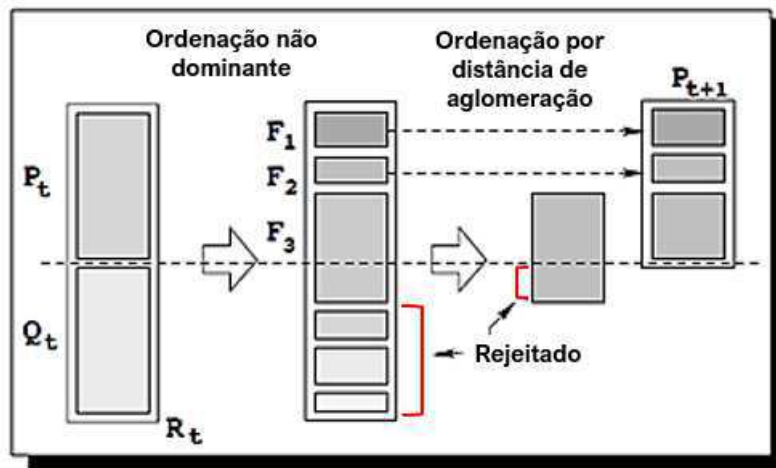
Elitismo é um procedimento importante quando utilizado com algoritmos evolucionários. O elitismo simples funciona da seguinte forma, o melhor indivíduo da geração k (o pai) é mantido na próxima geração $k + 1$ se seu filho tiver aptidão inferior que a do seu pai. Sem elitismo, os melhores resultados podem ser perdidos durante as operações de seleção, mutação e cruzamento. No caso de elitismo global, cada indivíduo na população da geração $k + 1$ pode substituir seu pai da geração k , se tiver aptidão superior a dele. Nesse caso, em uma geração $k + 1$, os indivíduos são melhores que os indivíduos da geração k [70].

Parte da abordagem desenvolvida neste trabalho aplica o processo de evolução baseado na ordenação não dominante do algoritmo NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) desenvolvido por Kalyanmoy Deb [71], que considera múltiplos objetivos e está ilustrado na Figura 2.11.

O funcionamento básico do NSGA-II consiste no seguinte. Considere M como o número de objetivos e N como o tamanho populacional. Inicialmente, uma população aleatória de pais é criada. A população é classificada com base na não dominância. A cada solução é atribuída uma condição de aptidão (ou classificação) igual ao seu nível de não dominância, de

acordo com a Equação 2.5. Assim, minimização de aptidão é assumida, ficando os melhores indivíduos nas fronteiras com menor nível de dominância. Inicialmente, os operadores usuais de seleção, cruzamento e mutação são usados para criar uma população de descendentes Q_0 também de tamanho N .

Figura 2.11: Processo de evolução utilizado no NSGA-II.



Fonte: Adaptado de DEB et al., 2002, p. 186. (tradução nossa).

Uma vez que o elitismo é introduzido pela comparação da população atual com as melhores soluções não dominadas encontradas anteriormente, o procedimento é diferente após a geração inicial. Descrever-se-á aqui a t -ésima geração do algoritmo.

Primeiro, uma população $R_t = P_t \cup Q_t$ combinada é formada. A população R_t tem tamanho $2N$. Então, a população R_t é classificada de acordo com a não dominância. Uma vez que todos os membros da população anterior e atual estão incluídos em R_t , o elitismo é assegurado. Agora, as soluções pertencentes ao melhor conjunto não dominado F_1 são as melhores soluções na população combinada e devem ser enfatizadas mais do que qualquer outra solução na população combinada. Se o tamanho de F_1 for menor que N , então, definitivamente todos os membros do conjunto F_1 foram escolhidos para a nova população P_{t+1} . Os membros restantes da população P_{t+1} são escolhidos entre as fronteiras subsequentes não dominadas, na ordem de suas classificações. Assim, as soluções do conjunto F_2 são escolhidas na sequência, seguidas de soluções do conjunto F_3 , e assim por diante. Este procedimento, que está ilustrado na Figura 2.11, é continuado até que nenhum conjunto mais possa ser acomodado.

A nova população P_{t+1} de tamanho N agora é usada para seleção, cruzamento e mutação

para criar uma população Q_{t+1} de tamanho N .

A complexidade geral do algoritmo é $O(MN^2)$, que é regida pela parte da classificação de não dominância do algoritmo. Se for realizado com cuidado, a população completa de tamanho $2N$ não precisa ser classificada de acordo com a não dominância. Assim que o procedimento de classificação tenha encontrado um número suficiente de fronteiras para ter N membros em P_{t+1} , não há motivo para continuar com o procedimento de classificação.

2.6.3 Algoritmos Meméticos

O termo Algoritmo Memético é utilizado para abranger uma ampla classe de meta-heurísticas (isto é, métodos de finalidade geral destinados a guiar uma heurística subjacente). O método provou ser de sucesso prático em uma variedade de domínios de problemas e, em particular, para a solução aproximada de problemas de otimização NP completos [72].

Ao contrário dos métodos tradicionais de Computação Evolutiva (em inglês, *Evolutionary Computing* - EC), os algoritmos meméticos se preocupam com a exploração de todo o conhecimento disponível sobre o domínio do problema em estudo. Essa incorporação não é um mecanismo opcional, mas um recurso fundamental que caracteriza os algoritmos meméticos. Essa forma de funcionalidade é ilustrada pelo termo “memético”. Segundo [72], a palavra “meme” foi criada por R. Dawkins [73] e denota um análogo ao gene no contexto da evolução cultural. Nas palavras de Dawkins:

"Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperms or eggs, so memes propagate themselves in the meme pool by leaping from brain to brain via a process which, in the broad sense, can be called imitation".

Os algoritmos meméticos são métodos de busca baseados na população que combinam algoritmos genéticos (busca global) e refinamentos locais (busca local), que é quando se incorpora alguma estratégia de busca local utilizando todo o conhecimento disponível sobre o domínio do problema. Esse casamento entre pesquisa global e busca local permite manter alta diversidade populacional via mutação (reduzindo assim a possibilidade de convergência

prematura) e aumenta a velocidade de convergência via busca local (a busca local pode melhorar muito a qualidade da solução e assim fazer com que a solução se aproxime da solução ideal mais rapidamente) [74].

Com a incorporação de busca local nos Algoritmos Evolucionários, esse tipo de algoritmo passa a ser chamado de Algoritmo Memético. Os operadores comuns como, cruzamento, mutação e seleção, dos Algoritmos Genéticos, continuam existindo. Contudo, a inserção de uma estratégia de otimização local permite obter boas soluções já nas fases iniciais da evolução populacional [44][68].

2.7 Considerações Finais

Neste capítulo, apresentou-se as terminologias, tecnologias e informações necessárias a um melhor entendimento do ramo de pesquisa aqui explorado. A ideia do provisionamento de recursos é ampla e, como foi abordado anteriormente, pode partir de uma visão macro envolvendo a escolha desde uma nuvem específica a ser selecionada para receber uma máquina virtual, até o servidor hospedeiro que abrigará a VM em seu destino final. A decisão de como prover e gerenciar os recursos existentes em um ambiente corporativo, com grandes quantidades de máquinas e recursos reais, como em uma nuvem pública, faz toda diferença em vários setores ou objetivos das empresas como, por exemplo, na visão que os clientes terão sobre a qualidade dos serviços, na economia com recursos de *hardware* e *software*, na economia com uso de energia, no desempenho dos serviços, etc.

Uma tecnologia chave que entra para facilitar no alcance de vários destes objetivos é a virtualização. Com a utilização de máquinas virtuais, opção largamente utilizada em ambientes de nuvens computacionais, é possível manipular o ambiente no sentido de obter o melhor arranjo de VMs dentro do *datacenter*, buscando o(s) objetivo(s) foco da empresa, o qual é de responsabilidade dos tomadores de decisão em seus ambientes.

Capítulo 3

Trabalhos Relacionados

Este capítulo trata dos trabalhos encontrados na literatura que abordam questões de posicionamento de VMs em *datacenters* com os mais variados objetivos. Destaca-se principalmente as pesquisas nas quais aplicou-se o tratamento de múltiplos objetivos a serem atendidos, além daquelas que utilizaram abordagens evolutivas no posicionamento de VMs em MFs nos *datacenters*.

3.1 Gerenciamento de Recursos

O provisionamento de recursos é parte essencial em um ambiente de DC. Nesta seção, tal processo será revisado através de levantamento do que tem sido produzido recentemente nessa área de pesquisa.

Para tal propósito, alguns levantamentos das pesquisas científicas sobre o tema foram escolhidos [75][10][11][76][39][77][12][49], os quais realizaram trabalhos abrangentes na pesquisa bibliográfica do estado da arte das formas de gerenciamento de recursos em *datacenters*, como sobre as técnicas aplicadas.

Vários e diferentes são os objetivos quando se realiza o gerenciamento de recursos de um *datacenter*. Quando se pensa em custo, a redução no número de servidores físicos em funcionamento é um dos principais objetivos a serem atingidos, assim como a redução na largura de banda que conecta os servidores, no número de componentes secundários da rede como *switches*, etc. Outro objetivo específico bastante considerado é a redução no consumo de energia, que também pode ser alcançado com a redução no número de servidores físicos

ligados, mas, também pode ser buscado de outras formas como, por exemplo, tratando o fluxo de circulação do calor dentro das salas dos DCs.

Como a atual pesquisa procura obter o melhor posicionamento de VMs em MFs, aplicando sobrecarga de recursos e considerando diversos objetivos simultaneamente, o que pode impactar no desempenho das aplicações implantadas nas VMs, aqui tratar-se-á dos trabalhos que se alinham nessa direção, buscando semelhanças e diferenças relacionadas a esse ponto.

3.1.1 Otimização com Objetivo Único

Song et al. [78] propuseram uma abordagem para o provisionamento adaptativo de recursos em nuvem usando *Bin Packing*. A abordagem utiliza as tecnologias de virtualização para alocar os recursos dos *datacenters* de forma dinâmica, baseado nas demandas das aplicações, buscando otimização dos servidores utilizados. Para conseguir tal intento, os autores elaboraram e avaliaram uma variante do problema *bin packing* em seus algoritmos, desenvolvendo o *Variable Item Size Bin Packing* (VISBP). Os autores afirmam que a solução funciona bem em sistemas reais. Como justificativa para o trabalho, lembrando que as demandas por recursos por parte das VMs mudam frequentemente, eles afirmaram que a mudança de tamanho de um item (uma VM) não se ajustava facilmente nos algoritmos existentes. Os resultados mostraram, quando comparados com algoritmos existentes, que o VISBP supera na mitigação de *hot spots* (quando a utilização de uma máquina física está acima de um limite preestabelecido) e no balanceamento de cargas, além de manter um equilíbrio das demandas entre estabilidade e consumo de energia.

Segundo Wang e Tianfield [79] uma estratégia reconhecida chamada de consolidação de máquinas virtuais dinâmicas (em inglês, *Dynamic Virtual Machine Consolidation - DVMC*), permite reduzir o consumo de energia sem comprometer o acordo de nível de serviço. No artigo, eles apresentaram uma nova estrutura de DVMC para nuvem através de nova política proposta de posicionamento de máquina virtual, chamada de SABFD (em inglês, *Space Aware Best Fit Decreasing*) e uma nova política de seleção de VM para migração, baseada na alta utilização da CPU, chamada de HS (do inglês, *High Selection*).

Simulações foram realizadas para avaliar o desempenho de diferentes estratégias DVMC com base em perfis de cargas de trabalho reais, aplicando DVMC com várias combinações de detecção de sobrecarga de máquina física, seleção de VM para migração e políticas de

posicionamento de VM. As simulações compararam as políticas NPA (*Non Power Aware*), DVFS (*Dynamic Voltage and Frequency Scaling*) e políticas de economia de energia. Duas políticas de posicionamento de máquinas virtuais foram comparadas nas simulações, uma foi a política proposta (SABFD), e a outra foi a PABFD (*Power Aware Best Fit Decreasing*).

O PABFD classifica todas as VMs em ordem decrescente de suas utilizações de CPU e aloca cada VM a um hospedeiro, de modo que a alocação cause aumento mínimo no consumo de energia. Cinco políticas de detecção de sobrecarga do hospedeiro: Intervalo Interquartil (IQR), Regressão Local (LR), Regressão Local Robusta (LRR), Desvio Absoluto da Mediana (MAD); e quatro políticas de seleção de migração de VM: Tempo Mínimo de Migração (MMT), Seleção Aleatória (RS), Máxima Correlação (MC) e HS foram usadas.

Os resultados das simulações mostraram que a estratégia DVMC com a política SABFD ou com a política HS superaram abordagens alternativas de DVMC. Outro resultado destacado pelos autores foi que a aplicação da estratégia DVMC com SABFD e HS juntas obtiveram o melhor desempenho. Não houve utilização do conceito de sobrecarga de recursos e apenas um objetivo (reduzir o consumo de energia) foi contemplado.

3.1.2 Otimização de até Dois Objetivos

Manvi e Shyam [10] elaboraram um levantamento bibliográfico sobre o gerenciamento de recursos para o modelo de serviço IaaS em computação em nuvem. No trabalho, eles mencionam ferramentas existentes que realizam monitoramento e/ou provisionamento (de recursos). Os autores atestam que os artigos mencionados não discutem os algoritmos necessários para resolver os desafios levantados. Afirmam também que existem vários problemas que precisam ser resolvidos no gerenciamento de recursos em nuvem com respeito à flexibilidade, escalabilidade, adaptabilidade, customização e reuso. Como um dos desafios no mapeamento de recursos os autores citam o desenvolvimento de algoritmos que possam encontrar um mapeamento rápido usando algoritmos genéticos para acelerar o processo e garantir o atendimento de todos os prazos da tarefa. O trabalho não cita, dentre os artigos avaliados, o uso de técnicas que realize posicionamento de VMs considerando múltiplos objetivos e sobrecarga de recursos, semelhante ao que aqui é proposto.

J. Xu e J. Fortes [67] propuseram um sistema de controle de dois níveis responsável por gerenciar os mapeamentos das cargas de trabalho para VMs e de VMs para recursos físicos.

O problema para posicionamento de VM foi tratado como um problema de otimização com múltiplos objetivos, que tenta minimizar simultaneamente o desperdício de recursos, o consumo de energia e os custos com dissipação térmica. Foi proposto um algoritmo genético melhorado com avaliação multiobjetivo para pesquisar eficientemente o grande espaço de solução e combinar, convenientemente, objetivos possivelmente conflitantes.

A avaliação de múltiplos objetivos com Lógica-Fuzzy foi aplicada para combinar os objetivos. Foram utilizados controles locais, em nível de aplicação, para garantir os recursos necessários para que o desempenho dessas aplicações se mantivesse em patamares esperados. Os experimentos basearam-se em simulações e indicaram que o modelo proposto melhorou o desempenho comparando com conhecidos algoritmos de empacotamento (*bin packing*) e abordagens com apenas um objetivo. Os autores não consideraram o posicionamento de VMs de forma dinâmica e não utilizaram o tratamento aqui proposto, que utiliza uma abordagem pura com múltiplos objetivos para melhorar o desempenho dos serviços das VMs, principalmente as consideradas críticas através da aplicação de sobrecarga de recursos, como também a alocação de forma mais eficiente.

Alguns artigos, a exemplo de [80], buscaram melhorar o desempenho agrupando VMs que realizam I/O de forma intensa. VMs que realizam troca intensa de dados pela rede podem, por exemplo, ser agrupadas em um mesmo *rack* reduzindo o tempo de transferência entre elas, assim como, diminuindo o tráfego no *backbone* da rede. Foi considerado um único provedor de nuvem e a questão de como melhor atender à demanda do cliente em termos de oferta e preço, buscando maximizar a receita do provedor e satisfação dos clientes, ao mesmo tempo, minimizar o custo da energia. O problema foi modelado como de controle de tempo discreto restrito e foi usado o MPC (*Model Predictive Control*) para encontrar a solução. As simulações usando cargas de trabalho de nuvem real indicaram que, sob condições dinâmicas de carga de trabalho, a solução proposta alcançou maior receita líquida do que as estratégias de alocação estática e minimizou o tempo médio de espera das solicitações.

O problema de fornecer serviços de conteúdo foi posto como um desafio em [81], principalmente levando-se em conta técnicas de colaboração entre os sistemas de armazenamento e as infraestruturas de rede, mais especificamente para aplicações que são sensíveis à latência como *streaming* de voz, vídeo e serviços de terminal, em Provedores de Serviços de Nuvens (CSP). Existem muitos fatores que afetam aplicações sensíveis à latência, como a carga nos

servidores de aplicações, o mecanismo de roteamento utilizado, a engenharia de tráfego e os gargalos no acesso aos dados. Os autores propuseram realizar uma distribuição de conteúdo com estratégias de roteamento de requisições, recebendo como entrada informações de armazenamento e de tráfego de rede. As estratégias adotadas permitiram balancear a capacidade de armazenamento economizada e o tráfego de rede para um melhor desempenho. Isso pôde ser observado nas simulações baseadas em perfis de rastreamento do YouTube.

A abordagem desenvolvida, chamada de *Space4time*, melhorou em média 22,3% da latência no acesso, em comparação com a abordagem de referência, chamada de *StaticScheme*. O artigo está relacionado a este trabalho no sentido de também visar melhoria no desempenho dos serviços que são executados nas nuvens, contudo, difere da atual pesquisa no fato dos autores terem considerado uma infraestrutura de nuvem geo-distribuída que consistia em diferentes *datacenters* distribuídos em vários locais geográficos. Além disso, por tratar de nuvens públicas e não utilizar diferentes objetivos na construção da solução de posicionamento das VMs.

Muitos trabalhos apresentam estratégias de gerenciamento de desempenho eficientes, em termos de energia, com base na tensão dinâmica e escala de frequência (em inglês, *Dynamic Voltage and Frequency Scaling - DVFS*). Wang e Wang [26], propuseram uma solução de otimização de desempenho controlada por energia para *clusters* de servidores virtualizados com aplicativos de várias camadas. Os autores alegam que a maioria dos trabalhos existentes dependem tanto de DVFS quanto de consolidação de servidores de forma separada, e a solução por eles desenvolvida utiliza ambas as estratégias para maximizar a economia de energia, integrando teoria de controle com estratégias de otimização. No nível da aplicação, um novo controlador com várias entradas e várias saídas foi projetado para alcançar o desempenho desejado para aplicações que abrangem várias VMs, em uma escala de tempo curto, realocando os recursos da CPU e conduzindo DVFS. Já no nível de *cluster*, um otimizador de energia é proposto para consolidar as VMs, de forma incremental, nos servidores mais eficientes em termos de energia em uma escala de tempo mais longa.

Os resultados empíricos obtidos pelos autores indicaram que a solução proposta foi superior às soluções que se apresentavam como estado da arte, com maior economia de energia e menor sobrecarga de consolidação, ao mesmo tempo em que alcança o desempenho requerido pela aplicação. A abordagem apresentada se diferencia da atual pesquisa por considerar

aplicações multicamadas, que são aquelas que podem ser particionadas em várias VMs, além de considerar DVFS para ajustes na frequência da CPU. Além disso, para a atual pesquisa, a questão principal é encontrar o melhor posicionamento de VMs considerando três objetivos evoluídos simultaneamente.

Segundo [37], do ponto de vista do provedor, o principal problema enfrentado é maximizar os lucros realizando a minimização dos custos, contudo, mantendo os níveis de serviços contratados pelos usuários. Dessa forma, o gerenciamento de energia nos *datacenters* dos provedores de computação em nuvem tem se tornado um problema crucial, já que tem grande influência nos custos operacionais. Segundo os autores, o alto nível de consumo de energia não se deve apenas à quantidade de recursos computacionais e a ineficiência energética do *hardware*, mas está no uso ineficaz desses recursos, pois na maior parte do tempo os servidores operam a 10-50% de suas capacidades.

Por isso, manter servidores subutilizados é extremamente ineficiente do ponto de vista de consumo de energia. Os autores propuseram uma abordagem baseada em enxame de partículas (em inglês, *Particle Swarm Optimization - PSO*) para realocar as VMs migradas de hospedeiros sobrecarregados e dinamicamente consolidá-las nos hospedeiros subcarregados objetivando economia de energia. Para isso, o algoritmo baseado em PSO mapeia uma VM para cada dimensão da posição da partícula, em que o valor da posição indica a potência do hospedeiro mais o custo da migração. Os objetivos da pesquisa foram melhorar a eficiência energética, em nuvens privadas, e garantir a qualidade de serviço para os usuários, através de tempos de respostas e prazos de trabalho.

A aplicação da técnica que permite sobrecarga de recursos pode ser encontrada em [42]. No trabalho, os autores citam que utilizar sobrecarga de recursos é um complemento necessário aos sistemas existentes. Os autores pretendiam melhorar um sistema de gerenciamento de *cluster* baseado no Apache Mesos para transcodificação de vídeo. Para ampliar a utilização, desenvolveram um módulo de sobrecarga de recursos no topo do Mesos visando maior eficiência do *cluster*. Os recursos eram requisitados com base no uso real de tarefas através de um modelo leve de previsão. Em seguida, as tarefas eram provisionadas novamente com os recursos recuperados, suprimindo as tarefas quando a carga do sistema excedia determinado limite. Outro procedimento que os autores aplicavam era a otimização dinâmica da CPU através da abordagem desenvolvida, chamada *Datom (Dynamic Throttling and Over-*

subscription on Mesos), que é uma extensão do módulo gerenciador de *cluster* do Mesos.

O *Datom* foi avaliado com cargas de trabalho reais de transcodificação de vídeo e simulações com perfis do *cluster* da Google. Os objetivos eram maximizar a sobrecarga de recursos enquanto minimizava penalidades na execução de tarefas. Os resultados mostram que, comparando com o Mesos original, o *Datom* aumentou a utilização da CPU, a utilização da memória e a taxa de transcodificação em até 22%, 23% e 20%, respectivamente, e o tempo de trabalho foi reduzido em até 20%, mas, com moderada penalidade na execução da tarefa. Apesar da aplicação de sobrecarga de recursos ser dinâmica, como neste trabalho de tese, o artigo apresentado funciona alocando tarefas aos recursos disponíveis, podendo uma mesma tarefa ser atendida por diferentes máquinas físicas, dependendo do tipo de tarefa.

Ashraf e Porres [82] apresentaram um novo algoritmo multiobjetivo, chamado MOACS (*Multi-Objective ACS*), que usa sistema de colônia de formigas (em inglês, *Ant Colony System - ACS*) através de meta-heurística chamada de otimização de colônia de formigas (em inglês, *Ant Colony Optimization - ACO*). O algoritmo visa otimizar dois objetivos, a consolidação de máquinas virtuais através da maximização do número de máquinas físicas liberadas, e a minimização das migrações de máquinas virtuais. Foram realizadas uma série de experimentos para avaliação do MOACS comparando o algoritmo proposto com dois algoritmos de consolidação de VM baseados em otimização de colônia de formigas publicados anteriormente, escolhidos devido ao seu excelente desempenho em relação a diferentes atributos. Foram considerados quatro cenários diferentes para testar os três algoritmos com diferentes configurações de VM. Os resultados experimentais mostraram que o algoritmo proposto forneceu uma solução eficiente para a consolidação de VMs em ambiente de computação em nuvem, superando os outros dois algoritmos de consolidação, também baseados em otimização de colônia de formigas, em termos de número de máquinas físicas liberadas e número de migrações de VMs. De acordo com a abordagem utilizada, o trabalho aplica a otimização dos objetivos seguindo uma ordem de prioridade dos objetivos, ou seja, realiza otimização de um objetivo por vez.

3.1.3 Otimização Não Simultânea de Múltiplos Objetivos

No trabalho [34], os autores reforçam a importância do uso de máquinas virtuais em ambientes de nuvens, principalmente a possibilidade de migração das VMs, como solução

para diversos problemas tais como, flutuações dinâmicas das cargas de trabalho, provocando situações indesejadas como desbalanceamento de cargas, baixa utilização de recursos ou concentração de cargas de trabalho em alguns pontos. Eles afirmam que muitas pesquisas exploram o processo de migração de VMs para melhorar desempenho com apenas um objetivo, como economia de energia ou aumento na utilização de recursos. Além disso, afirmam que são poucas as pesquisas que procuram utilizar mais de um objetivo ao mesmo tempo, acrescentando ainda haver algumas desvantagens nas metodologias desses trabalhos.

Eles consideraram o processo de migração como um problema multiobjetivo e propuseram uma nova política para migração de VMs, consolidada por uma nova estratégia de otimização com múltiplos e conflitantes objetivos, fornecendo flexibilidade na manipulação de diferentes casos. Os objetivos considerados simultaneamente foram: Volume de Cargas, Consumo de Energia, Estado Térmico, Desperdício de Recursos e Custo de Migração. A nova estratégia de otimização multiobjetivo baseou-se no *Static Bayesian Game* (SBG) que tem por base o *Multi-Objective Genetic Algorithm* (SBG-MOGA) sendo complementado pelo *Augmented Weighted Tchebycheff Program* (AWTP) para fornecer uma maneira flexível de ajustar a seleção de processos de migração para diferentes objetivos. Com essa abordagem, foi possível encontrar uma decisão de migração adequada para diferentes situações de carga de trabalho. Eles utilizaram a ferramenta de simulação CloudSim para validar as políticas de migração propostas e, como resultado, conseguiram controlar o desempenho do sistema ajustando os objetivos de migração para lidar com diversas situações de cargas de trabalho, limitando os objetivos aos valores desejados para cada critério.

Esse trabalho se difere do atual trabalho em alguns pontos, como: eles consideraram Estado Térmico no modelo, já que o estudo considera ambientes de nuvens públicas, que são consideravelmente maiores das que se considera na atual pesquisa, que são nuvens privadas, no qual entende-se que a quantidade de máquinas é reduzida, comparativamente à nuvens públicas, e portanto a questão do estado térmico seja menos preocupante; Na função objetivo por eles utilizada foram aplicados pesos aos diversos objetivos e depois somados em um único valor a ser confrontado com as demais soluções. Dessa forma, juntaram múltiplos objetivos combinando em uma expressão com objetivo único [44]. Esse tipo de abordagem é diferente da que aqui é sugerida, na qual se deseja otimização simultânea dos múltiplos objetivos.

Em seu recente artigo, o qual considera múltiplos objetivos para o posicionamento de VMs, Saber et al. [46] analisaram três pontos: *i*) sob quais condições a aplicabilidade de um solucionador de otimização de números inteiros misto (como o IBM ILOG CPLEX¹⁶) pode ser usado para o problema com múltiplos objetivos de rearranjo de máquinas; *ii*) quanto do espaço de busca uma solução híbrida como CBLNS (*Constraint-Based Large Neighbourhood Search*, é uma abordagem que usa LNS que por sua vez é uma meta-heurística para otimizar o problema de rearranjo de VMs com objetivo único) explora; e *iii*) se é possível desenvolver uma nova abordagem híbrida (GeNePi), para resolver o problema, combinando MILP (*Mixed Integer Linear Programming*) ou CBLNS com outra meta-heurística proposta recentemente. O problema colocado pelos autores é uma extensão, com múltiplos objetivos, do problema de rearranjo de VMs inspirado pelo Google/ROADEF¹⁷ (um desafio acompanhado de um conjunto de dados de vários *datacenters* com diferentes tamanhos).

Os autores mostraram que o MILP se aplica melhor apenas em *datacenters* de pequena e média escalas, mesmo assim com algumas simplificações. O CBLNS explora mais o problema como um todo, alcançando desempenho razoável para instâncias maiores do problema. Uma abordagem meta-heurística híbrida (GeNePi) foi então proposta para otimizar o problema de rearranjo de VMs com múltiplos objetivos, quais sejam: custo energético, custo de migração e custo de confiabilidade. GeNePi aplica, repetidamente, três meta-heurísticas: GRASP (*Greedy Randomized Adaptive Search Procedure*), NSGA-II (*Non-dominated Sorting Genetic Algorithm*) e PLS (*Pareto Local Search*). Os resultados dos experimentos mostraram que CPLEX se saiu melhor que GeNePi e CBLNS sendo, contudo, limitado às instâncias médias e pequenas. Por outro lado, GeNePi e CBLNS são melhores para grandes instâncias.

No geral, quando utilizando de forma híbrida, CPLEX+GeNePi superaram CBLNS+GeNePi em instâncias pequenas e médias, mas CBLNS+GeNePi se adapta melhor para grandes instâncias. Como diferencial para atual pesquisa, pode-se observar que os múltiplos objetivos utilizados na abordagem do artigo são diferentes, pois, no trabalho revisado, foi considerado custos de confiabilidade. Outra distinção é que o artigo revisado segue na direção de resolver a questão do posicionamento de VMs para grandes instâncias

¹⁶Disponível em: <https://www.ibm.com/br-pt/analytics/cplex-optimizer>. Acesso em: 09 nov 2017.

¹⁷Disponível em: <http://www.roadef.org/challenge/2012/en/index.php>. Acesso em: 09 nov 2017.

do problema, enquanto aqui ataca-se o problema para *datacenters* privados. A solução proposta no artigo não foi melhor do que soluções existentes, justamente para instâncias médias e pequenas, além de não considerar sobrecarga de recursos nem diferenciação das VMs.

O trabalho [16] procurou resolver alguns problemas de consolidação em Computação em Nuvem, como consolidação de servidores, consolidação de máquinas virtuais e consolidação de tarefas. Segundo os autores, Tecnologia de Grupo (em inglês, *Group Technology - GT*¹⁸) tem sido usada de forma efetiva para administrar problemas de alocação de recursos aumentando-se a produtividade do sistema e reduzindo custos de produção. Eles propuseram um algoritmo de otimização cuco discreto (*Discrete Cuckoo Optimization Algorithm - DCOA*) baseado no GT para consolidação em Computação em Nuvem. O algoritmo desenvolvido permite resolver problemas discretos de larga escala. Os resultados obtidos atestaram a superioridade de qualidade do DCOA em relação aos algoritmos *First Fit* e *Round Robin*, e a superioridade de eficiência e eficácia do DCOA em relação ao Algoritmo Genético desenvolvido pelos mesmos autores em 2016. No trabalho foi utilizada uma expressão única como função objetivo, considerando a soma dos custos de cada consolidação desejada, sendo, portanto, otimização de objetivos diversos sendo tratados em expressão única. Não foi utilizado sobrecarga de recursos.

3.1.4 Otimização Simultânea de Múltiplos Objetivos

De maneira geral

As principais críticas levantadas relacionadas aos MOEAs, que utilizam ordenamento não-dominante das soluções, são: 1) alta complexidade computacional no ordenamento não dominante ($O(MN^3)$, onde M é o número de objetivos e N é o tamanho populacional); 2) por utilizarem abordagem não elitista; e 3) por existir a necessidade de especificar um parâmetro de compartilhamento. Kalyanmoy Deb [71] propôs um MOEA chamado de Algoritmo Genético com Ordenação Não Dominante (em inglês, *Nondominated Sorting Genetic Algorithm II*, ou NSGA-II), que alivia as três questões elencadas como pontos críticos. O algoritmo proposto aplica uma abordagem de ordenação não dominante rápida, com complexidade computacional de $O(MN^2)$. Uma proposta de operador de seleção de população

¹⁸<https://hbr.org/1984/07/group-technology-and-productivity>

também é apresentada no trabalho de Deb, que consiste em uma piscina de cruzamento constituída pelas populações dos pais e descendentes, e na seleção dos N melhores indivíduos com relação à aptidão. Com isso, se garante a preservação de elitismo. Os resultados dos testes de simulação com problemas difíceis mostraram que o NSGA-II foi capaz de encontrar uma melhor disseminação de soluções e uma melhor convergência perto da frente ótima de Pareto em relação as estratégias *Pareto-archived Evolution Strategy* (PAES) e *Strength-Pareto EA* (SPEA), dois outros MOEAs elitistas que focam atenção especial na criação de uma frente ótima de Pareto diversificada.

Para o Posicionamento de VMs

Otimizar o posicionamento de VMs em um *datacenter* parece indicar que existe um arranjo ótimo de VMs para MFs que resolverá todos os problemas, contudo, em uma empresa o melhor posicionamento de VMs está relacionado com quais objetivos devem ser favorecidos, quem decide quais e como são esses objetivos, em outras palavras, a busca por um arranjo ótimo único que sirva em qualquer caso não é adequada. Cada responsável por alocar capital em uma empresa tem sua visão de como tornar o sistema melhor, alguns podem acreditar que reduzir custos com o gasto energético é o melhor caminho a seguir, outros considerarão que a confiabilidade dos serviços deve ser priorizada em detrimento de outros critérios (no caso de aplicações críticas espera-se confiabilidade total nos serviços), já alguns podem considerar o custo com licenças de *softwares* como o mais importante [46].

Alguns dos objetivos podem ser conflitantes, podendo acontecer do alocador de recursos decidir por otimizar dois critérios e alguns rearranjos de VMs atender melhor a um ou a outro critério, qual deles favorecer? Na sequência, contempla-se alguns dos trabalhos que consideraram tratar problemas com múltiplos objetivos na decisão do posicionamento de VMs .

O trabalho de Pires e Barán [83] de 2013 e a extensão desse em 2015 [44] e 2017 [84] são os que mais se assemelham ao trabalho atual. No artigo de 2013, López Pires levanta o problema de posicionamento de máquinas virtuais quando se considera múltiplos objetivos simultaneamente, quais sejam: minimização do consumo de energia, minimização do tráfego de rede e maximização das receitas de um *datacenter*. Além disso, a proposta considera Acordo de Níveis de Serviço para cada VM, contudo, esse parâmetro é utilizado apenas na

decisão de quais VMs devem ser alocadas de forma mandatória, iniciando imediatamente a execução de seus serviços. O autor desenvolve um algoritmo memético com múltiplos objetivos como solução ao problema colocado. Testes experimentais foram realizados com diversos cenários, em ambiente virtual simulado, considerando diferentes quantidades de máquinas virtuais e físicas, nos quais foram utilizados dados reais, disponibilizados pelo *Itaipu Technological Park Data Center*, do Paraguai.

Como principais diferenças do trabalho revisado, comparativamente à atual pesquisa, têm-se: (1) funções objetivo diferentes; (2) os autores geram um conjunto inicial de soluções aleatórias com posicionamentos das VMs, o qual será evoluído, não sendo possível partir da configuração de um ambiente já em funcionamento, através de uma leitura de um atual posicionamento das VMs nas MFs do *datacenter*. A criação da população inicial, na abordagem aqui desenvolvida, pode ser realizada de formas diversas, dependendo da existência ou não de um indivíduo base, sendo mais flexível que a abordagem de Pires; (3) O algoritmo proposto em Pires tem como um dos objetivos a serem atendidos a maximização das receitas, já que o público alvo do algoritmo desenvolvido são nuvens públicas. Para a atual pesquisa, considerou-se apenas *datacenters* privados, portanto, receita não foi um dos objetivos avaliados, apesar do tamanho do maior *datacenter* utilizado por Pires ser inferior ao maior tamanho utilizado nos experimentos desta tese; (4) Outra diferença para a atual pesquisa está relacionada ao tráfego de rede, que o artigo coloca como um dos objetivos a serem minimizados, enquanto aqui não é considerado. Entretanto, no atual trabalho foi considerado o custo de migração de forma que a característica de tráfego de rede pode ser ponderada; (5) No artigo revisado, não foram realizadas comparações de desempenho do algoritmo proposto com outras abordagens existentes. Na atual pesquisa foram realizadas comparações com outras soluções, inclusive com a de Pires; (6) Outro diferencial significativo é que aqui houve aplicação de sobrecarga de recursos através do ajuste dos diferentes níveis de comprometimento para as MFs. Em Pires, o máximo de VMs em uma MF é a quantidade em que a soma dos requisitos das VMs seja menor ou igual aos recursos reais disponíveis na MF; (7) A validação, no artigo revisado, foi realizada com tamanhos de ambientes não convencionais como 3 máquinas físicas com 5 máquinas virtuais e 4 máquinas físicas com 8 máquinas virtuais. Os resultados foram comparados com a própria abordagem, sendo executada a força bruta, em busca de todos os elementos da fronteira de Pareto, com

o intuito de mostrar que o algoritmo desenvolvido encontrou todas as soluções pertencentes à fronteira. No atual trabalho foram realizados experimentos com até 120 máquinas físicas no ambiente; (8) considera-se aqui dois tipos diferentes de serviços implantados nas VMs, sendo eles críticos ou não críticos; (9) Tentar reduzir o número de migrações no *datacenter* e diferenciar os tipos de VMs a serem migradas, aplicando diferentes custos de migração, foi uma estratégia usada nesta pesquisa; (10) Na abordagem de Pires não há preocupação com o desempenho das VMs que são marcadas com maior nível de SLA. Apenas considera essa informação para garantir que essas VMs sejam obrigatoriamente posicionadas em alguma MF em qualquer das soluções encontradas, o que pode ocasionar uma menor quantidade de VMs alocadas no *datacenter* se comparada com a abordagem aqui proposta. Na abordagem aqui desenvolvida, existem os controles de níveis de comprometimento que dão liberdade para que sejam ajustados individualmente com maior ou menor nível de sobrecarga de recursos, podendo tanto favorecer a alocação das VMs com serviços críticos, quanto deixar essa alocação indiferente se as VMs que serão posicionadas executam ou não serviços críticos; (11) Não é possível alterar parâmetros da abordagem desenvolvida por Pires como, por exemplo, o número de gerações ou o número de indivíduos a serem utilizados, sem que haja necessidade do código ser compilado novamente; (12) Em Pires, para extrair uma única solução de posicionamento, se faz necessário lançar mão de intervenção de alto nível que realize a escolha dentre diversas soluções encontradas na fronteira de Pareto; e (13) Não é possível, na abordagem desenvolvida em Pires, simplesmente indicar qual dos objetivos se deseja favorecer.

Em [84] houve uma generalização do trabalho anterior [83], apenas estendendo a opção de múltiplos objetivos para mais do que três objetivos. Os experimentos realizados foram da mesma maneira, porém, com um número máximo de máquinas físicas de 100 e com mais dois objetivos a serem otimizados, que foram: Qualidade dos Serviços e balanceamento de carga de rede, perfazendo cinco objetivos ao todo. Portanto, as diferenças já elencadas entre a abordagem aplicada nesta tese e o trabalho de Pires e Barán de 2013, também se aplicam ao trabalho deles de 2017.

Um resumo dos trabalhos revisados nesta tese encontra-se na Tabela 3.1. Na primeira coluna, observa-se as referências dos artigos. É possível observar, na segunda coluna, quais os objetivos foram considerados em cada pesquisa listada.

Tabela 3.1: Caracterização dos trabalhos relacionados.

Artigos	Objetivo (s)	Ambiente dos Experimentos	Tipo de Algoritmo	Aplicação de Sobrecarga de Recursos	Evolução Multi-Objetivos Pura
Song et al. (2014)	Reduzir o número de servidores ativos usados	Três servidores físicos	Bin Packing	Não	Não
Wang e Tianfield (2018)	Reduzir o consumo de energia via Consolidação de máquinas virtuais	Simulação de Nuvem Pública	Space Aware Best Fit Decreasing com HS	Não	Não
J. Xu e J. Fortes (2010)	Uso eficiente do recurso, Consumo de Energia	Simulação de Nuvem Pública	Lógica-Fuzzy	Não	Não
Zhang et al. (2011)	Custo com Energia, Aumento da Receita	Simulação de Nuvem Privada	MPC (Model Predictive Control)	Não	Não
Zeng et al. (2014)	Balancear a capacidade de armazenamento economizada e o tráfego de rede para um melhor desempenho de aplicações sensíveis à latência	Simulações baseadas em perfis de rastreamento do YouTube	Space4time: Nova solução de distribuição de conteúdo e de solicitação de solução de roteamento	Não	Não
Wang e Wang (2014)	Otimizar desempenho das VMs com redução do consumo de energia	Um cluster com quatro servidores físicos e simulador para grande escala	Algoritmo de otimização baseado em heurística	Não	Não
Dashti e Rahmani (2016)	Custo com Energia, QoS	Simulação de Nuvem Privada	PSO	Não	Não
Chen et al. (2018)	Maximizar a sobrecarga de recursos e minimizar a reserva de recursos de cada tarefa para melhorar a utilização de um cluster	Simulação em um Cluster local	Heurística	Sim	Não
Ashraf e Porres (2018)	Consolidação de máquinas virtuais através da maximização do número de máquinas físicas liberadas, e da minimização das migrações	Simulação de Nuvem Privada	Multi-objective ACS	Não	Não
Sallam e Li (2014)	Volume de carga, Consumo de energia, Estado térmico, Desperdício de recursos e Custo de Migração	Simulação de Nuvem Pública	SBG-MOGA	Não	Não
Saber et. al. (2017)	Custo de Confiabilidade (alocação sem sobrecarga de recurso), Custo com Energia, Custo de Migração	Simulação com Nuvens de diferentes tamanhos	Meta-heurística híbrida	Não	Não
Tavana et al. (2018)	Consolidação de servidores, de máquinas virtuais e de tarefas	Simulação com Nuvens de diferentes tamanhos	Discrete Cuckoo Optimization Algorithm - DCOA	Não	Não
Pires e Barán (2013) e (2017)	Minimizar consumo de energia, minimizar tráfego de rede e maximizar receita do provedor	Simulação de Nuvem Privada	Algoritmo Genético	Não	Sim
Esta Tese (2019)	Minimizar consumo de energia, minimizar dispersão e minimizar número de migrações	Simulação de Nuvem Privada	Algoritmo Genético	Sim	Sim

Ainda na Tabela 3.1, na terceira coluna estão os ambientes nos quais as pesquisas foram desenvolvidas. Os tipos de algoritmos empregados em cada técnica estão na quarta coluna. A quinta e sexta colunas indicam se houve, na técnica analisada, aplicação de sobrecarga de recursos e/ou evolução pura de múltiplos objetivos, respectivamente. Como se pode observar, apenas uma aplicou sobrecarga de recursos, e apenas uma aplicou evolução pura de múltiplos objetivos.

3.2 Considerações Finais

Como se pôde observar na revisão realizada, e também afirmado em [84], poucos artigos propuseram soluções para o problema de posicionamento de VMs considerando múltiplos objetivos sendo evoluídos de forma simultânea e, dentre os que resolveram o problema proposto considerando múltiplos objetivos, o fizeram combinando-os em apenas um, normalmente através de uma expressão em que se atribuiu pesos aos diferentes objetivos normalizados ou evoluindo um objetivo por vez. Esse tipo de solução difere do que é sugerido nesta pesquisa e do que foi apresentado em [84]. Uma solução para um problema de otimização através de evolução de múltiplos objetivos *pura*, realiza a otimização de forma *simultânea* dos objetivos. Portanto, a atual pesquisa se encontra posicionada em um grupo muito restrito de soluções elaboradas para o problema de posicionamento de VMs, utilizando uma abordagem pura de otimização com múltiplos objetivos, através do uso de algoritmos meméticos, com o diferencial de ir além das abordagens utilizadas no estado da arte, aplicando sobrecarga de recursos na decisão do posicionamento de VMs.

Até o melhor de nosso conhecimento, a evolução a ser realizada de forma simultânea, dos objetivos aqui propostos, é diferente das que até agora foram utilizadas na literatura, haja vista que aqui é apresentado o diferencial de aplicar sobrecarga de recursos na decisão do posicionamento de VMs junto à otimização simultânea de múltiplos objetivos de forma pura, além de ter sido desenvolvido um método empírico para mensuração de dispersão de VMs que é um dos objetivos a serem atendidos.

Capítulo 4

A Abordagem VMPOS

Neste capítulo serão apresentadas a estrutura geral do ambiente e a abordagem desenvolvida, chamada de VMPOS, descrevendo seus componentes e funções. A formulação matemática do problema, considerando múltiplos objetivos, assim como a proposta de solução com suas restrições e algoritmos.

4.1 Estrutura do *Datacenter*

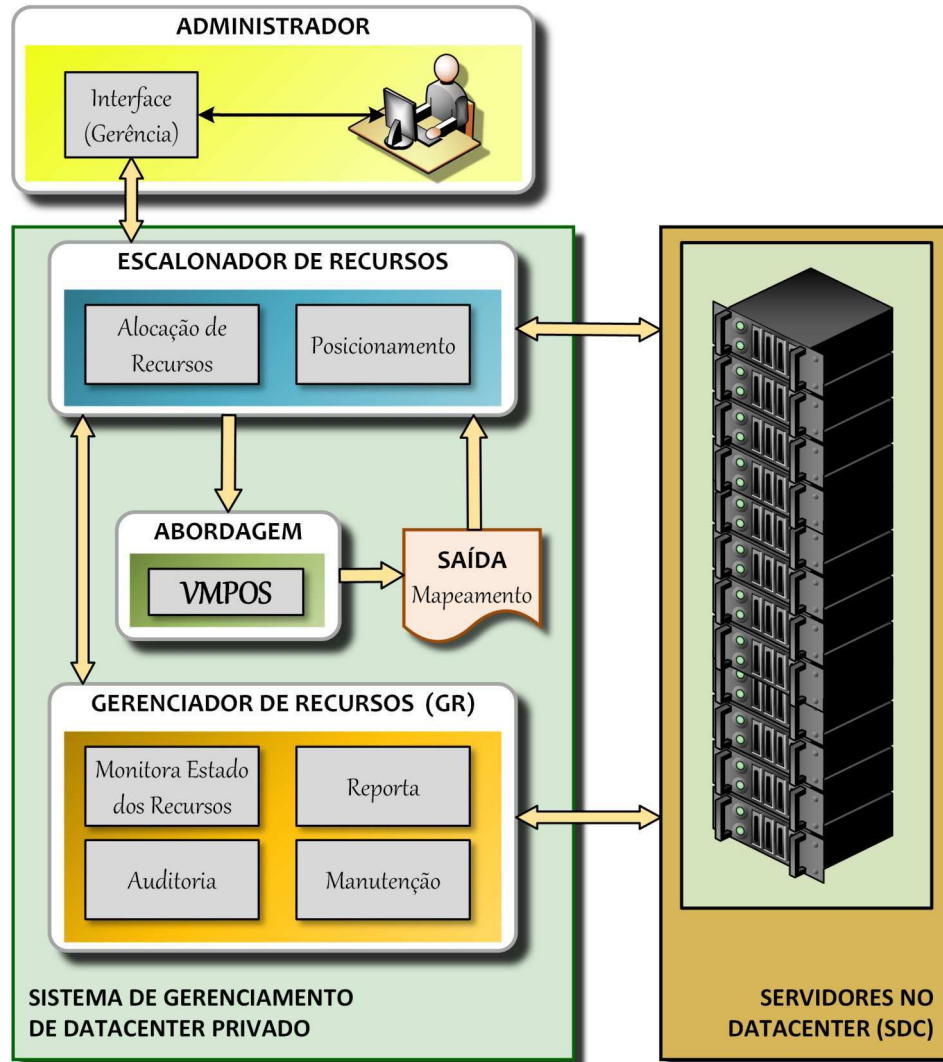
Um ambiente de *Datacenter* pode ser gerido com a utilização de módulos independentes e interligados compondo um sistema de gerenciamento maior. Isso permite independência entre os módulos, facilitando implementações/correções e reduz a complexidade da gerência da infraestrutura. Na Figura 4.1 está ilustrada a arquitetura geral com os componentes necessários para o gerenciamento dos recursos de um *Datacenter*.

De acordo com a Figura 4.1, o DC é composto por uma gama de máquinas físicas (*Servidores no Datacenter - SDC*). Cada um desses servidores físicos contém um *hypervisor* e podem hospedar VMs dos seguintes tipos:

- i) **VMs Críticas:** são os servidores, em máquina virtual, que abrigam um ou mais serviços considerados críticos (importantes) para empresa como, por exemplo, demandas relacionadas à *Big Data* ou serviços de *e-commerce*;
- ii) **VMs Não Críticas:** são as demais máquinas virtuais, de propósito geral, que não abrigam nenhum serviço considerado crítico como, por exemplo, uma máquina *desktop* de

um determinado setor da empresa.

Figura 4.1: Esquema do ambiente de *Datacenter* privado considerado na pesquisa.



Fonte: Elaborada pelo autor.

No momento da criação de uma nova VM será indicado se ela abrigará **Serviços Críticos (SC)** ou **Serviços Não Críticos (SNC)**. A decisão de quais serviços serão ou não críticos fica a cargo dos administradores do *datacenter*, já que estes são os detentores do *know-how* do negócio, dando mais flexibilidade ao sistema.

Para tentar atingir o máximo possível de alocação desejada, no sentido de manter em utilização praticamente a totalidade dos recursos físicos disponíveis, pode-se fazer uso de técnicas de sobrecarga de recursos. Isso é realizado atribuindo-se um nível de comprometimento à uma máquina física superior ao que teoricamente poderia suportar (*oversubscrip-*

tion) [42][85][86]. Por exemplo, supondo que uma máquina física dispõe de 4 CPUs e 8 GB de memória principal, atribuir a essa MF um nível de comprometimento de duas vezes seus recursos, permitiria alocar uma quantidade de máquinas virtuais em que a soma dos requisitos destas por recursos fosse equivalente a duas vezes os recursos reais existentes (até 8 vCPUs e 16 GB de memória).

Durante a maior parte do tempo, os usuários de recursos computacionais em nuvem não utilizam a totalidade dos recursos contratados [86], o que possibilita aos provedores aplicarem a técnica de sobrecarga dos recursos, conseguindo, assim, abrigar um maior número de usuários clientes. É uma técnica largamente utilizada em ambientes de nuvens e *datacenters* [85].

Os demais componentes presentes na infraestrutura, ilustrados na Figura 4.1, são:

- **Administrador:** Pessoa responsável por gerenciar todo o ambiente do *datacenter* através do *Sistema de Gerenciamento de Datacenter Privado*, ou seja, as VMs dos usuários e as configurações aplicadas aos parâmetros para execução da abordagem;
- **Gerenciador de Recursos (GR):** Componente responsável por realizar auditoria (como relatórios com as ocorrências de falhas registradas), informar sobre manutenção (indicar se existem máquinas em manutenção), monitorar (em busca de gargalos) e reportar (respondendo as solicitações do Escalonador) o estado dos recursos presentes no *Datacenter*;
- **Servidores no Datacenter (SDC):** O *Datacenter* é composto por uma gama de máquinas físicas, que podem ser vistas na parte de *Servidores no Datacenter*, as quais contêm um *hypervisor* e hospedam uma ou mais VMs. As máquinas físicas presentes no SDC podem abrigar VMs dos dois tipos simultaneamente, contudo, o processo de concentração das VMs de cada tipo favorece à aplicação de sobrecarga de utilização dos recursos físicos através da aplicação de níveis de comprometimento;
- **VMPOS:** É a parte principal do Sistema de Gerenciamento do *datacenter*, responsável por decidir o posicionamento das VMs nas MFs. A saída da execução da abordagem VMPOS, passada ao Escalonador de Recursos, é um mapeamento indicando onde cada máquina virtual deverá ser alocada, sendo migrada da atual posição, caso necessário;

- **Escalonador de Recursos:** Recebe as demandas provenientes do GR, geradas por algum alerta devido ao contínuo monitoramento de recursos, como também pode realizar consultas ao estado atual dos recursos no SDC para, caso necessário, solicitar à VMPOS um novo mapeamento através de uma execução da abordagem, podendo gerar alterações a serem feitas na atual distribuição das VMs. Responsável por aplicar os mapeamentos (posicionamentos) indicados pela VMPOS, além das alterações (criação, exclusão, etc. de VMs) e consultas realizadas pelo Administrador.

4.2 Visão Geral

Em um *datacenter* encontra-se uma variedade de recursos a exemplo das máquinas virtuais que são hospedadas em outros recursos que são as máquinas físicas. A existência dessa possibilidade de alocação de VMs em MFs permite uma gama de alternativas de posicionamento, ou alocação, dessas VMs nas MFs.

Dada essa característica, a decisão de posicionamento das VMs é tomada atendendo a algum objetivo, ou conjunto de objetivos, de acordo com as necessidades do negócio da empresa. Como exemplos de objetivos têm-se: reduzir o número de migrações de VMs, aumentar o lucro, reduzir o número de servidores físicos, reduzir o custo energético, aumentar a utilização dos recursos físicos, etc. Portanto, decidir por uma ou outra solução de posicionamento serve para refletir os objetivos traçados pelo tomador de decisão do ambiente.

A parte principal do Sistema de Gerenciamento de *datacenter* descrito nesta tese é a abordagem de *Posicionamento de Máquinas Virtuais Aplicando Sobrecarga de Recursos (VMPOS)*, foco deste trabalho, que decide o posicionamento das VMs nas MFs aplicando um algoritmo evolucionário para melhorar o posicionamento das VMs, tentando atender múltiplos objetivos simultaneamente, em *datacenter* privado. É empregado sobrecarga de recursos nas máquinas físicas através de diferentes níveis de comprometimento, visando favorecer os objetivos e o desempenho das VMs com serviços críticos.

Após a evolução dos múltiplos objetivos de forma simultânea pura, pesos definidos pelo administrador do *datacenter* são aplicados, a fim de selecionar solução única de posicionamento, favorecendo (pesos diferentes) ou não (pesos iguais) determinado objetivo.

Espera-se, com a aplicação da abordagem VMPOS, que os posicionamentos obtidos aten-

dam os objetivos traçados com desempenho superior às abordagens do estado da arte, uma vez que a VMPOS executa a evolução dos objetivos sendo considerados de forma simultânea pura, aplicando algoritmo memético, a técnica de fronteira de Pareto e empregando sobrecarga de recursos, em busca de melhores soluções para o problema.

Apesar da abordagem VMPOS poder trabalhar com quaisquer quantidade de objetivos, neste trabalho foram definidos três objetivos a serem considerados (detalhados na Seção 4.4.4) que são: *minimizar o consumo de energia*, através da redução do número de MFs utilizadas; *minimizar o fator de migração*, reduzindo o impacto dos processos de migrações dentro do DC, dando prioridade às soluções que gerarão uma menor quantidade de migrações, conseqüentemente, beneficiando aplicações executadas nas VMs; e, *minimizar a dispersão*, através desse objetivo que reflete o nível de agrupamento das VMs com e sem serviços críticos, visando permitir melhor desempenho das VMs com serviços críticos da empresa.

Na abordagem desenvolvida foram considerados dois tipos de VMs, as críticas e as não críticas, e utilizados dois níveis de comprometimento, um para máquinas físicas que hospedam pelo menos uma VM com *serviços críticos*, e outro para máquinas físicas que hospedam apenas VMs com *serviços não críticos*. Isso foi feito com o objetivo de manter as máquinas virtuais com serviços críticos em um ambiente mais “protegido”, em relação à disputa por recursos com outras VMs, sendo isso possível através da concentração de VMs críticas e do nível de comprometimento das MFs que as hospedam.

4.2.1 Funções dos Componentes do *Datacenter*

A seguir, estão descritas as funções de cada um dos componentes do *datacenter*, desde uma requisição para criação de uma máquina nova, até o monitoramento constante dos recursos no SDC.

Após receber uma requisição, proveniente do usuário, para criação/deleção de uma VM, ou receber uma demanda para realocação de VMs vinda do GR, o **Escalonador** realizará uma consulta sobre o estado atual dos recursos do sistema ao componente responsável, no caso, o Gerenciador de Recursos.

O **Gerenciador de Recursos**, por sua vez, mantém informações sobre o estado dos recursos (como memória, CPU, disco, etc.), através do *Monitoramento do Estado dos Recursos*, e

se alguma máquina física está em *manutenção*. O GR também *audita e reporta* as informações sobre o estado dos recursos auditados quando solicitado, alertando sobre sobrecarga ou baixa demanda.

Recebidas as informações sobre o estado atual dos recursos do sistema, o Escalonador iniciará o processo de *Alocação dos Recursos* chamando a abordagem **VMPOS** a qual, por sua vez, utiliza o quadro atual de posicionamento das VMs para descobrir como será possível melhorar a demanda existente, considerando múltiplos objetivos, e retorna o “novo” mapeamento gerado.

O mapeamento de quais VMs serão migradas e para onde, ou seja, a saída da abordagem VMPOS, será então devolvido ao Escalonador e, finalmente, o *Posicionamento* das VMs será realizado e as VMs passarão a ser executadas em MFs especificadas no resultado da abordagem.

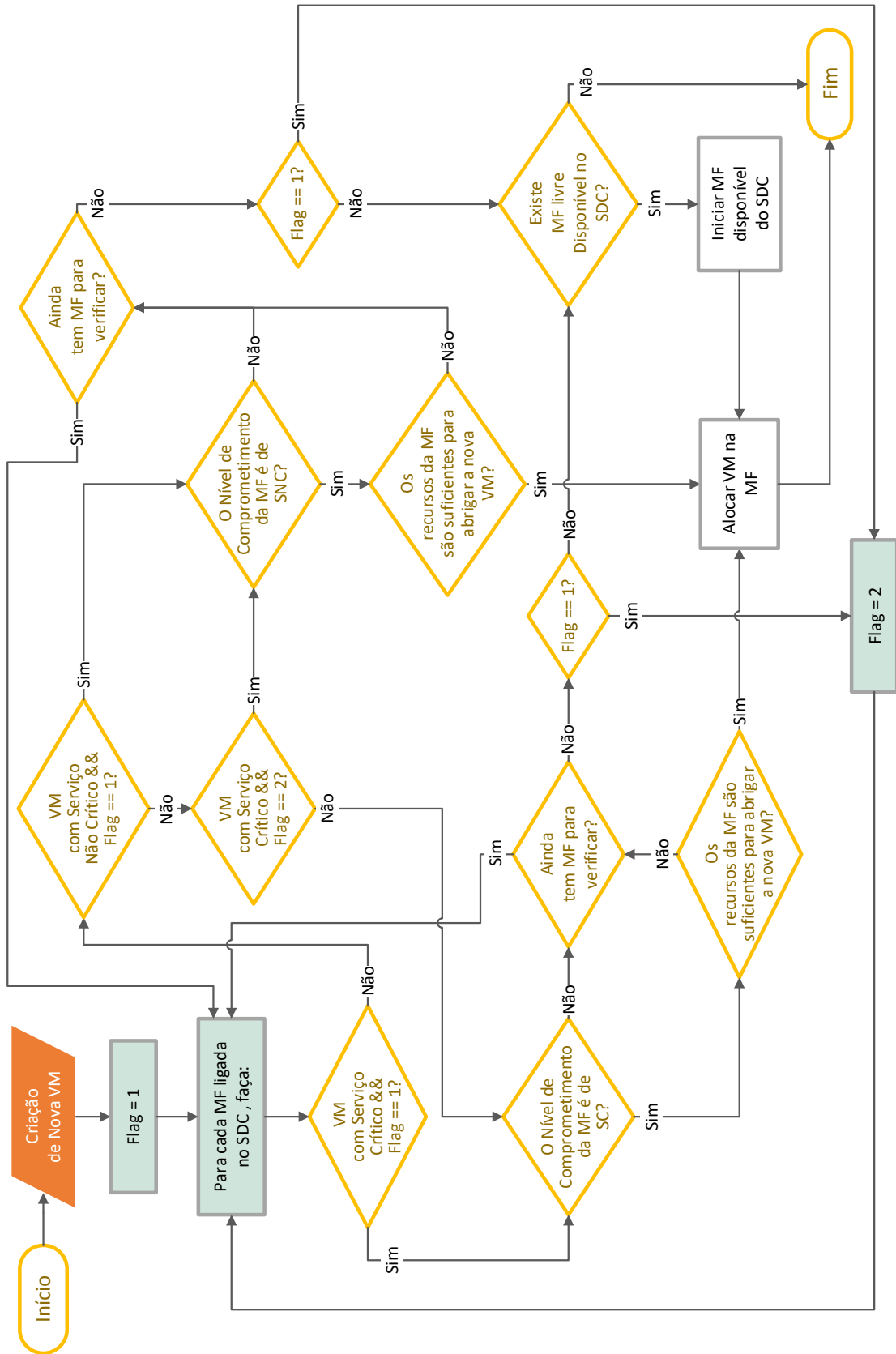
O Sistema de Gerenciamento de todo o *datacenter* então se baseia em 3 fases principais:

- **Fase 1:** quando se realiza periodicamente a avaliação das atuais condições de desempenho de todos os servidores no DC, tanto das MFs quanto das VMs, com o intuito de aferir se essas condições estão dentro de níveis preestabelecidos. Considera-se que a expectativa de tempo para avaliação periódica seja algo que deva ocorrer a cada 24 horas, ou a intervalos não muito curtos de forma que possibilite todas as movimentações de VMs antes da aplicação de novo posicionamento;
- **Fase 2:** dependendo do resultado da Fase 1, o mapa de alocação, que indica quais VMs estão alocadas em quais máquinas físicas no momento atual da leitura, é gerado e passado à VMPOS que, de acordo com seu processamento, irá produzir nova distribuição de VMs por MFs, ou indicará que nenhuma mudança deverá ser feita;
- **Fase 3:** quando acontecem os processos de migração das VMs para as MFs, caso necessário, resultante da Fase 2.

Deve-se levar em consideração duas operações comuns em *datacenters* que são a *inclusão* e *exclusão* de VMs.

A seguir, são apresentados os fluxogramas que detalham as fases dos eventos que ocorrem no *datacenter* e as ações realizadas em cada uma delas, começando com o fluxograma da Figura 4.2.

Figura 4.2: Fluxograma de alocação de nova VM.



Fonte: Elaborada pelo autor.

O fluxograma da Figura 4.2 representa a etapa de alocação de uma nova VM, dependendo do tipo de serviço que será executado por esta, se com ou sem serviços críticos, o que é indicado na solicitação de criação. A alocação acontece através da aplicação de algoritmo do tipo *First Fit*. Contudo, qualquer outro algoritmo de inserção pode ser utilizado para tal propósito.

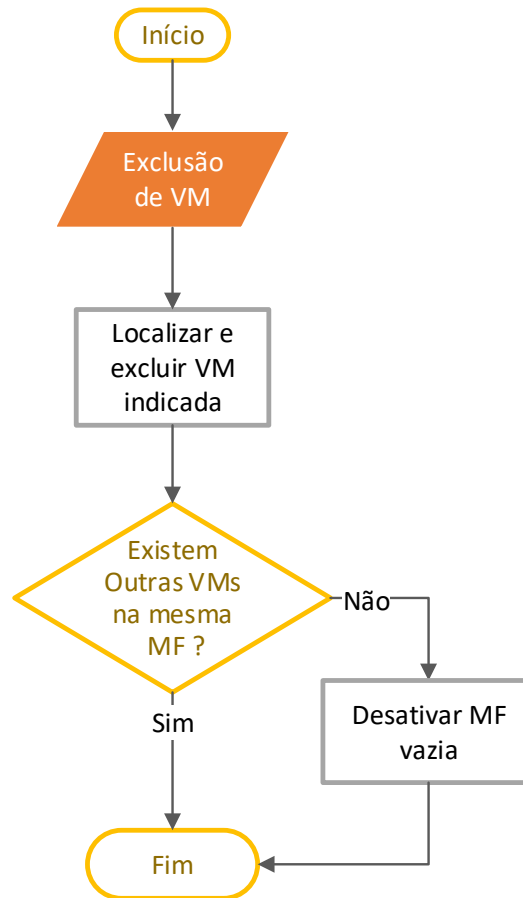
A variável *Flag* que aparece no fluxograma da Figura 4.2 indica em que tipo de MF está se tentando posicionar determinada VM, se em MF com mesmo nível de comprometimento da VM ou com nível de comprometimento diferente. Essas duas fases são diferenciadas pelo valor da variável *Flag*, sempre tentando posicionar primeiro em MF com mesmo nível de comprometimento da VM.

A abordagem tentará alocar a nova VM em MF já em funcionamento, com recursos suficientes e com nível de comprometimento equivalente ao que foi aplicado na VM. No caso de não conseguir alocar a VM em MF com mesmo nível de comprometimento, o algoritmo verifica as MFs com níveis de comprometimento iguais ao do tipo de VM oposto e tenta alocar. Caso consiga um hospedeiro, este terá atualizado o nível de comprometimento de acordo com as VMs alocadas naquele instante. Se não conseguir MF disponível para hospedar a VM, tentar-se-á através da inicialização de uma MF do SDC que não se encontrava ativa. Se nenhum desses casos for possível, a nova VM não será alocada. A princípio a MF recém ligada para abrigar uma VM ficará subutilizada, contudo, apenas até a próxima execução da VMPOS.

A Figura 4.3 representa o procedimento para a exclusão de uma VM de um dos servidores do SDC.

O processo é simples, consistindo na localização da VM indicada para exclusão, e verificação de existência de outra(s) VM(s) na mesma MF, o que acarretaria desativar ou manter a MF ativa.

Figura 4.3: Fluxograma para exclusão de uma VM.



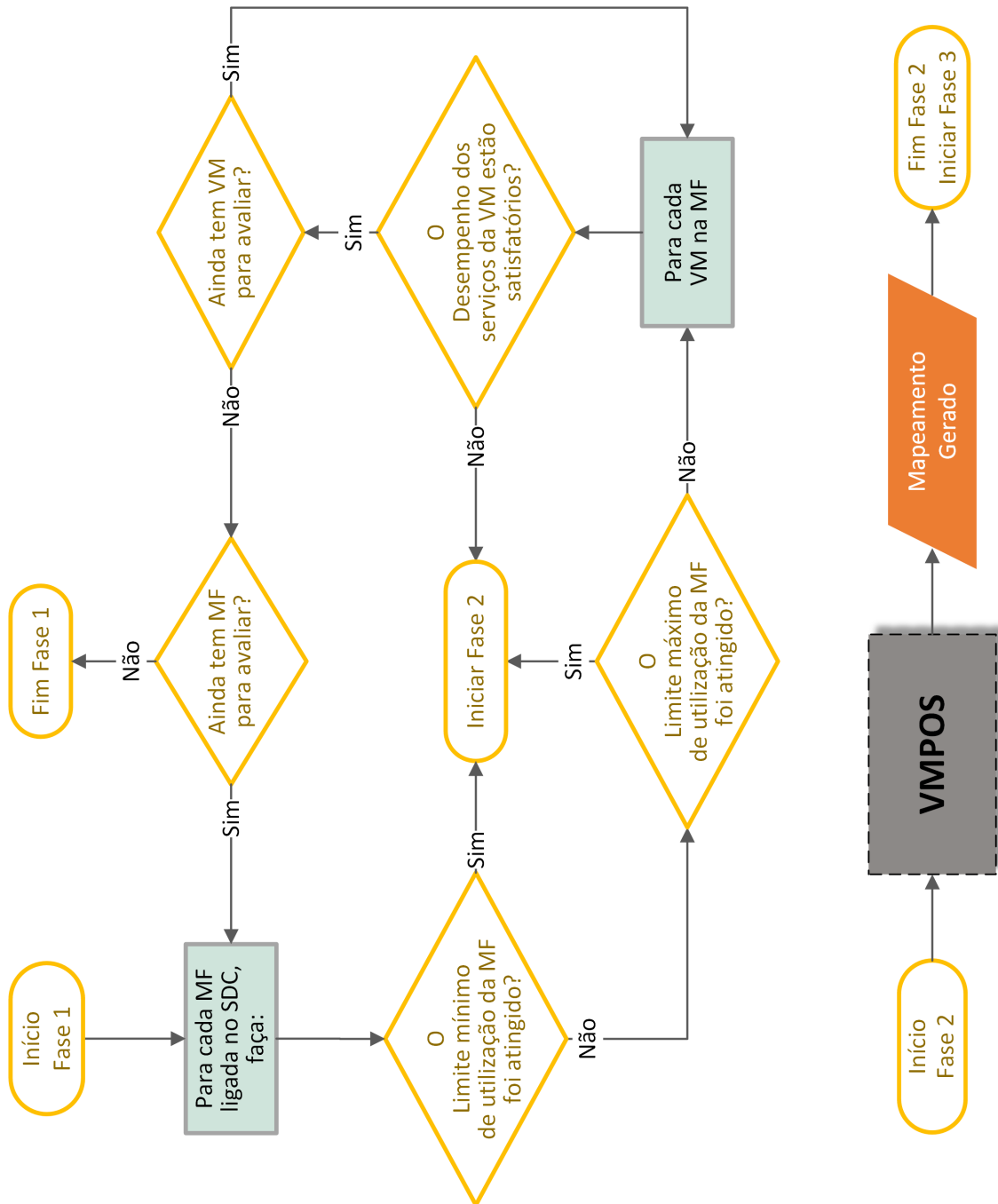
Fonte: Elaborada pelo autor.

Na Figura 4.4, o fluxograma (Fase 1, parte superior) ilustra o processo que deve ocorrer periodicamente e que tem por objetivo verificar se os desempenhos tanto das MFs quanto das VMs presentes no DC estão dentro de níveis aceitáveis.

Caso qualquer das MFs tenha atingido o limite mínimo de utilização, a fase 2 é iniciada (Figura 4.4, parte inferior), que corresponde à execução da abordagem VMPOS. No caso de limite máximo atingido, os desempenhos dos serviços das VMs alocadas na referida MF serão avaliados. Se estiverem satisfatórios, nada deve ser alterado. Caso alguma VM esteja com desempenho insatisfatório, inicia-se a fase 2.

A partir desse instante é iniciada a Fase 2, que aparece na parte inferior da Figura 4.4. Aqui, após a conclusão da Fase 1, será iniciada a execução da VMPOS a qual irá gerar uma solução ótima de posicionamento de VMs, com base nos objetivos aplicados na abordagem. A geração do mapeamento, com indicação de quais VMs estarão alocadas em quais MFs, finaliza a Fase 2 e dá início à Fase 3.

Figura 4.4: Fluxograma representativo da verificação sistemática do estado do sistema.

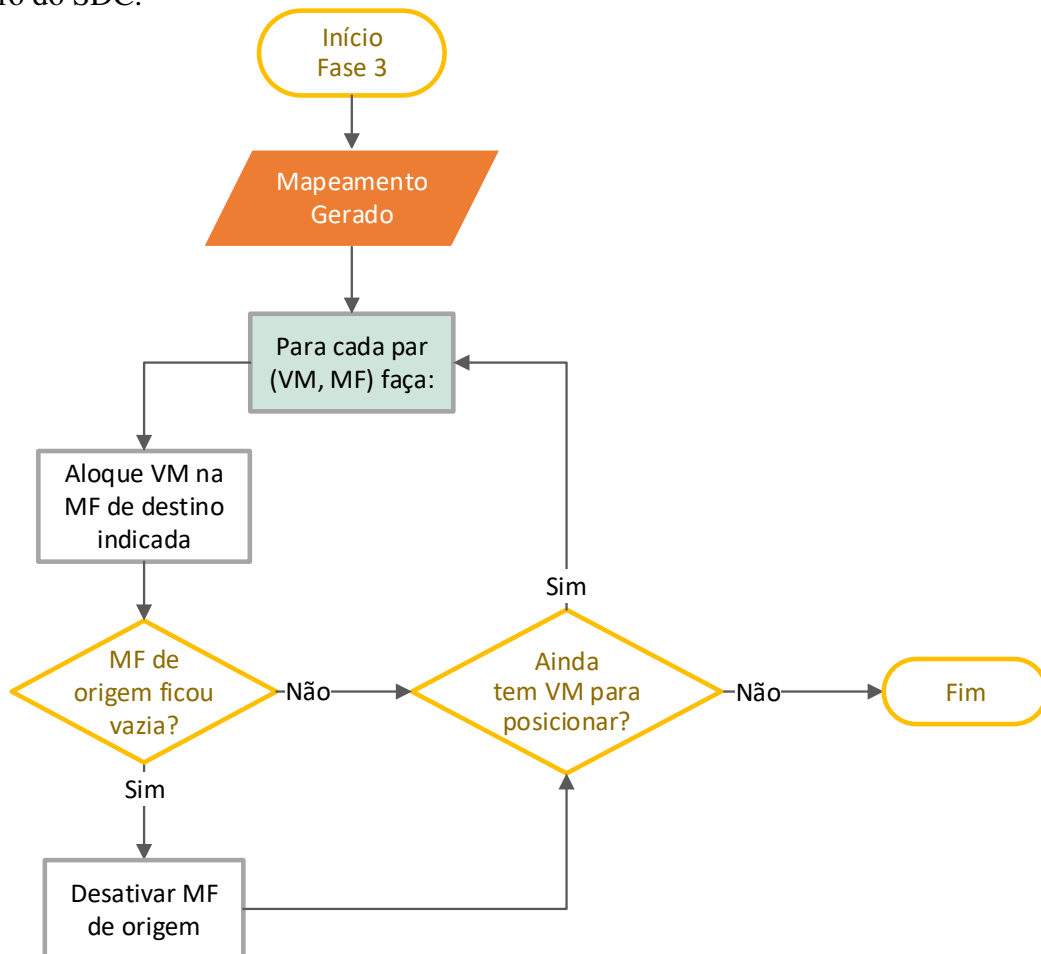


Fonte: Elaborada pelo autor.

Na Fase 3 (Figura 4.5), o processo consiste basicamente em migrar as VMs para as MFs indicadas, e verificar se a MF de origem ficou ou não vazia podendo, portanto, ser desativada. O problema de decidir como aplicar de fato o mapeamento gerado é um outro problema computacional bastante complexo, fora do escopo desta pesquisa. Além do mais,

a expectativa de tempo para uma nova etapa de reposicionamento de VMs pode ser fixada em espaço grande de tempo, como 24 horas. O fluxograma apresentado para Fase 3 é uma generalização simplificada apenas para as situações em que migrando todas as VMs de uma MF, a MF deve ser desativada.

Figura 4.5: Fluxograma representativo do processo de posicionamento das VMs em MFs dentro do SDC.



Fonte: Elaborada pelo autor.

4.3 Descrição da Abordagem Desenvolvida

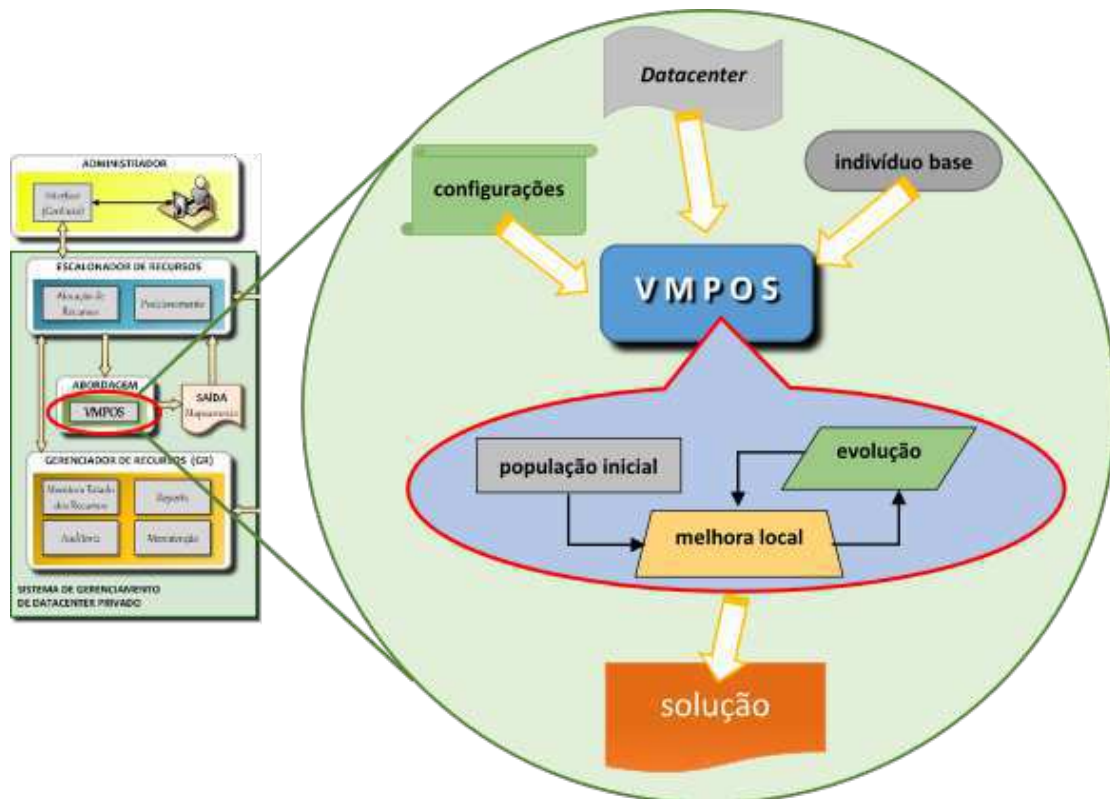
Do ambiente retratado na Seção 4.1, a pesquisa desenvolvida e descrita nesta tese teve foco no desenvolvimento da abordagem central chamada de **VMPOS**, responsável por tomar a decisão de posicionamento das VMs nas MFs do SDC. É a parte mais importante para o sistema de gerenciamento dos recursos do *datacenter* privado apresentado.

A abordagem permite uma gama de configurações realizadas pelo administrador, outras informações são dependentes do próprio *datacenter* ao qual a abordagem VMPOS está sendo aplicada como, por exemplo, a quantidade de MFs e de VMs. Essas quantidades são calculadas a partir da entrada de onde se obtém as especificações dos recursos das máquinas físicas e requisitos das máquinas virtuais presentes no ambiente.

Para uma visão mais próxima da estrutura da abordagem VMPOS, é possível visualizar na Figura 4.6 os componentes utilizados pela abordagem e necessários para sua execução. Segue uma breve descrição desses componentes:

- *Indivíduo base*: Informação com o posicionamento das VMs de uma solução, a partir da qual a abordagem irá calcular o valor representante do objetivo relacionado à migração, a cada novo indivíduo gerado.
- *Datacenter*: Informações sobre as máquinas do *datacenter*, composto por todas as MFs existentes no ambiente e seus recursos disponíveis, e informações de todas as VMs existentes com os requisitos de recursos de cada uma.
- *Configurações*: Um dos principais componentes é formado por um conjunto de configurações no qual se encontram todas as informações necessárias para a execução da abordagem, que são as seguintes:
 - o número de indivíduos a ser utilizado na evolução genética da população;
 - a quantidade máxima de gerações que se deseja realizar a evolução da população;
 - os pesos que devem ser aplicados a cada um dos objetivos, os quais serão utilizados na escolha de solução única de posicionamento;
 - os níveis de comprometimento aplicados nas MFs que abrigam pelo menos uma VM com serviços críticos e nas MFs que abrigam apenas VMs sem serviços críticos;
 - os custos de migração aplicados a cada um dos tipos de VMs, que são utilizados para o cálculo do valor representativo do objetivo relacionado à migração.

Figura 4.6: Ampliação da abordagem VMPOS.



Fonte: Elaborada pelo autor.

4.3.1 Componente: Indivíduo Base

O indivíduo base pode ser fornecido à abordagem VMPOS de diversas formas. No caso de um *datacenter* já estabelecido, em funcionamento, o indivíduo base inicial poderá ser uma transcrição fiel da alocação previamente existente no ambiente, indicando o posicionamento de cada VM em cada MF presente. Basta passar a configuração de posicionamento das VMs existentes, em um formato exatamente igual ao do cromossomo que está descrito na Seção 4.5.1.

Caso não se deseje partir de um mapeamento inicial, ou esse não exista, o indivíduo base também poderá ser gerado utilizando o resultado de alocação de uma outra abordagem de posicionamento qualquer como, por exemplo, um *First Fit*, que aloca as VMs existentes no *datacenter* de forma sequencial, hospedando o máximo de máquinas virtuais em cada uma das máquinas físicas disponíveis até que não haja mais VMs a serem alocadas. A solução gerada por outra abordagem pode então ser utilizada como indivíduo base para a VMPOS.

Como terceira e última opção, também é possível não fornecer nenhum indivíduo base, situação que a abordagem VMPOS se encarregará de gerar um posicionamento aleatório de VMs, utilizando tal indivíduo gerado como base.

Para qualquer um dos casos elencados, o número de máquinas físicas no indivíduo base, assim como a distribuição das VMs nessas máquinas, servirão de parâmetros para que a abordagem possa, por exemplo, calcular o objetivo de minimizar o número de migrações de VMs (detalhado na Seção 4.4.4), quando da busca pelo melhor posicionamento.

4.3.2 Componente: *Datacenter*

O componente de entrada chamado *Datacenter*, conforme Figura 4.6, é responsável por informar à VMPOS os detalhes de recursos disponíveis em cada máquina física existente, assim como todos os detalhes de requisitos de recursos das máquinas virtuais existentes.

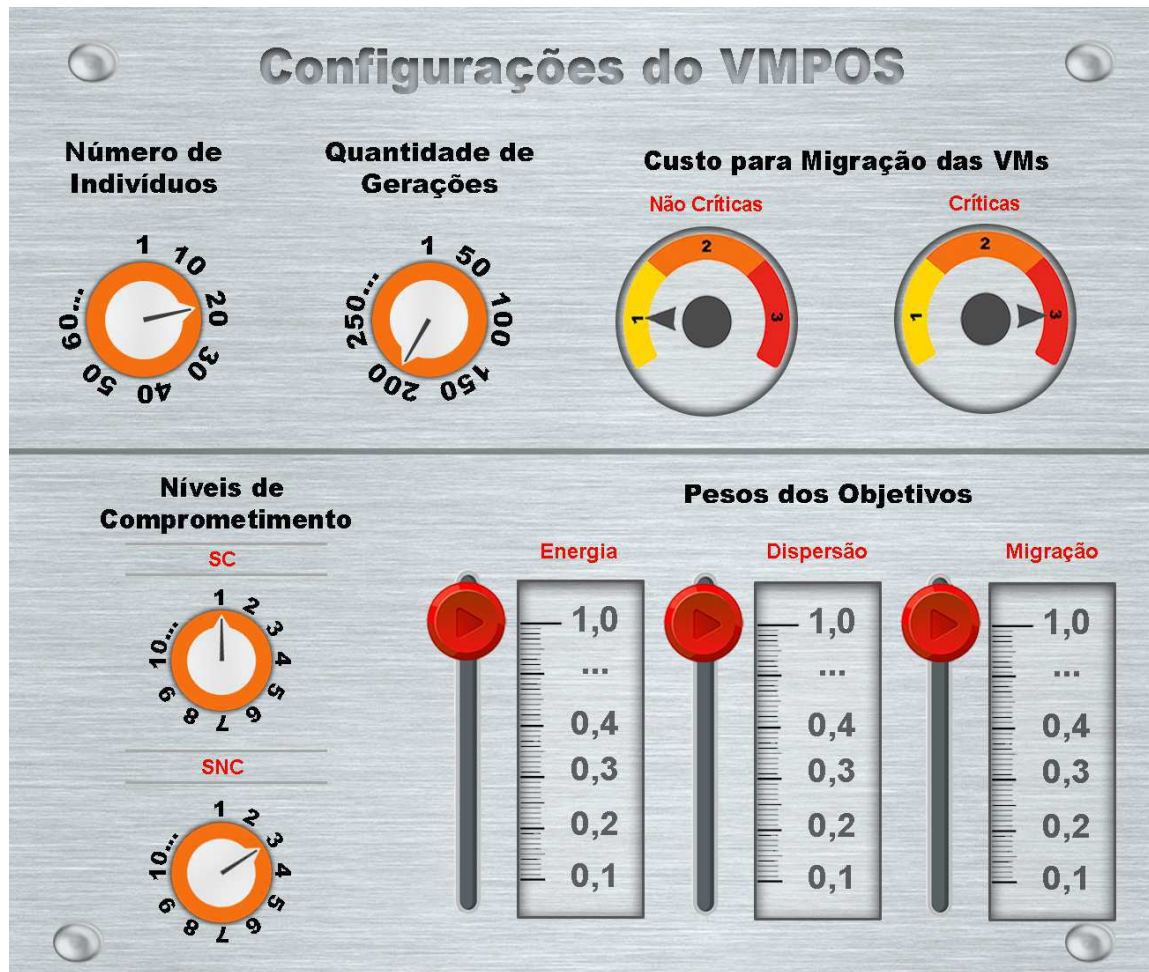
Durante a leitura das informações das máquinas presentes no *datacenter*, tanto físicas quanto virtuais, a VMPOS realiza a contagem das quantidades existentes de cada tipo de máquina. Esses valores são utilizados em diversas operações durante a busca por soluções de posicionamento.

No caso das máquinas físicas, são necessárias as informações de recursos de CPU, quantidade de memória RAM, capacidade de armazenamento e consumo máximo de energia. Para as máquinas virtuais, são necessárias as informações dos requisitos de CPU, quantidade de memória RAM requerida, capacidade de armazenamento requerida e informação se a VM está ou não executando serviços críticos.

4.3.3 Componente: Configurações

Para uma visão mais aprofundada das possibilidades de configurações que podem ser feitas na abordagem VMPOS, na Figura 4.7 é apresentada uma ilustração dos possíveis ajustes para execução da abordagem. Os parâmetros são aplicados com os valores desejados pelo administrador do DC, para: os níveis de comprometimento, os pesos dos objetivos, o número de indivíduos da população, a quantidade de gerações e os custos de migração de cada tipo de VM.

Figura 4.7: Ilustração com as possibilidades de configurações da VMPOS.



Fonte: Elaborada pelo autor.

Número de Indivíduos

É ajustado de acordo com a quantidade de indivíduos que se deseja utilizar para formação da população, que será evoluída segundo algoritmo evolucionário aplicado. A quantidade de indivíduos vai influenciar em maior (ou menor) cobertura do espaço de soluções devido à maior (ou menor) diversidade de indivíduos disponíveis para o processo de cruzamento. Por outro lado, quanto mais indivíduos na população mais demorado será o processo de evolução.

Quantidade de Gerações

Um algoritmo evolucionário, como o elaborado nesta pesquisa, necessita parar em algum momento. Várias decisões podem ser utilizadas para finalizar o processo evolutivo como,

por exemplo, se não conseguir melhorar ainda mais o melhor dos indivíduos já encontrado durante as próximas X gerações, ou se o tempo transcorrido desde o início da busca por soluções já ultrapassou T unidades de tempo, etc.

Na implementação da solução optou-se por utilizar, como condição de parada, a quantidade de gerações que se deseja evoluir a população, podendo esse aspecto ser modificado de maneira simples de acordo com a necessidade do usuário.

Custo de Migração

De forma semelhante ao que se realiza com os níveis de comprometimento, custos são aplicados a cada tipo de VM (crítica ou não crítica) para poder realizar o cálculo do objetivo minimização de migração.

A ideia utilizada é procurar reduzir o número de migrações de VMs com serviços críticos, deixando as VMs com serviços não críticos mais disponíveis para mudança de hospedeiro. Como será apresentado na Seção 4.4.4 com as funções objetivo, é possível atribuir qualquer valor a esses dois custos, contudo, o diferencial será efetivo se for atribuído maior custo de migração às VMs com serviços críticos, e menor custo às VMs sem serviços críticos.

A expressão final do objetivo custo de migração aplica uma soma simples levando em consideração os custos associados a cada tipo de VM, sempre comparando a solução encontrada (indivíduo atual) em relação ao indivíduo base. Ou seja, para sair do posicionamento existente representado pelo indivíduo base e aplicar o posicionamento da solução atual, encontrado em determinada geração pelo VMPOS, quantas VMs com serviços críticos e quantas com serviços não críticos terão que migrar de suas posições atuais.

Aplicar um valor de custo de migração maior para VMs com serviços críticos implica em tentar não migrar essas VMs, o que favorece os serviços nelas implantados, haja vista que não sofrerão o processo de *downtime* observado nas migrações em tempo real, com consequente degradação do desempenho [49]. Se o objetivo é minimizar migrações, procedendo dessa forma espera-se que as soluções com menor número de migrações de VMs com serviços críticos sejam favorecidas. O que será reflexo da menor soma com custos. Mais informações sobre os custos aplicados podem ser obtidas no Apêndice A.

Níveis de Comprometimento

Como mencionado anteriormente, foram utilizados dois níveis de comprometimento, um para máquinas físicas que hospedam pelo menos uma VM com *serviços críticos*, e outro para máquinas físicas que hospedam *apenas* VMs com *serviços não críticos*.

A decisão do posicionamento das VMs dentro da nuvem privada¹⁹ passa pelo processo de evolução realizado por algoritmo memético, objetivando manter um bom desempenho dos principais serviços implantados nos servidores virtuais, os quais serão chamados aqui de críticos. Tal melhora é alcançada através da concentração dos dois tipos de VMs utilizadas, que são com ou sem serviços considerados críticos, juntamente com o nível de comprometimento aplicado a cada máquina física.

Acredita-se que, aplicar níveis de comprometimento baixos para MFs hospedando VMs críticas favorece o desempenho dos serviços implantados nas VMs, haja vista que aplicar níveis de comprometimento possibilitando que mais VMs sejam alocadas na mesma MF, oferecendo mais recursos do que realmente disponíveis fisicamente, faz com que a concorrência pelos recursos de fato existentes seja maior entre as VMs hospedadas, podendo causar lentidão e redução no desempenho de uma aplicação crítica.

Pensando nisso, a abordagem VMPOS procura atender ao objetivo de tentar concentrar o máximo possível de VMs críticas em algumas MFs e VMs não críticas em outras. Dessa forma, obtêm-se o benefício de aplicar maiores níveis de comprometimento para MFs que não abrigam serviços críticos, podendo alocar muito mais VMs não críticas no DC, e as VMs críticas concentradas também são beneficiadas pelo fato de utilizarem o *hardware* real com nível de comprometimento 1 (um), ou não tão alto quanto o aplicado para as VMs não críticas. Claro que a decisão de que níveis utilizar ficará a cargo do administrador do ambiente, através de configurações da abordagem.

É importante ressaltar aqui que, na abordagem VMPOS, o nível de comprometimento associado a uma determinada máquina física é dinâmico, dependendo da distribuição dos tipos de VMs que estão hospedadas em determinado momento. Por exemplo, se na distribuição das VMs por MFs uma determinada MF teve alocadas uma VM com SC e duas com SNC, a MF em questão terá associado o nível de comprometimento SC, ou seja, existe pelo

¹⁹ou *Datacenter*. Para o contexto atual, e deste ponto em diante, estes termos serão utilizados de forma intercambiável.

menos uma VM crítica hospedada na MF. Caso ocorra uma redistribuição e a VM crítica seja migrada, aquela MF passa a ter o nível de comprometimento ajustado para SNC, considerando que a mesma não estará mais hospedando VMs com serviços críticos. Com isso, poderá abrigar um número maior de VMs com serviços não críticos devido a aplicação de maior nível de comprometimento dos recursos.

A abordagem VMPOS trabalha favorecendo as soluções que refletem maiores agrupamentos de VMs de cada tipo, assim, as VMs críticas tendem a serem alocadas paralelamente no mesmo *hardware*, favorecendo todas as VMs que compartilham a mesma MF. Além disso, contribui-se com o processo de consolidação de servidores. Vale lembrar que quanto mais VMs compartilhando o mesmo hospedeiro, menor o desempenho alcançado por cada uma delas [87][27]. Isso quer dizer que parece uma boa estratégia não permitir que muitas VMs sejam hospedadas ao mesmo tempo em MFs que abrigam VMs com serviços críticos.

Por outro lado, aplicar sobrecarga de recursos traz benefícios à medida que possibilita alocar uma quantidade de VMs em que a soma dos requisitos de recursos ultrapassa o que realmente existe na MF, ampliando a quantidade total de VMs alocadas no *datacenter*. A abordagem VMPOS atende essa característica concentrando as VMs marcadas como não críticas, assim, as MFs que estiverem hospedando apenas VMs não críticas, terão níveis de comprometimento ajustados para SNC.

Pesos dos Objetivos

A escolha das soluções que mais se aproximam da fronteira de Pareto, neste trabalho, é feita através da evolução dos três objetivos considerados de forma simultânea pura. Ao final do processo, são obtidas algumas soluções que pertencem à melhor fronteira encontrada. Nesse ponto se faz necessária a intervenção de tomador de decisão de nível mais alto que irá escolher que solução única adotar.

Com o objetivo de não depender de escolha de tomador de decisão, optou-se por selecionar apenas uma das soluções, dentre as que pertencem a melhor fronteira encontrada, como resultado final. Para isso, são aplicados pesos aos objetivos normalizados, escolhendo a solução que apresenta a menor soma desses objetivos.

Tal procedimento também possibilita o favorecimento de um dos objetivos em detrimento dos demais. Aplicando pesos diferenciados aos objetivos utilizados, é possível favorecer

aquele que se deseja melhores resultados. Por exemplo, se é desejo do administrador reduzir o consumo de energia, independentemente do que acontece com o objetivo número de migrações ou com a dispersão, ele aplica valores de peso maiores para o objetivo minimizar o consumo de energia.

Até o melhor de nosso conhecimento, a aplicação desse tipo de abordagem para o posicionamento de VMs, considerando diferentes objetivos a serem atendidos, principalmente o de dispersão, em *datacenter* privado, com a aplicação de diferentes níveis de comprometimento dos recursos físicos disponíveis e diferenciação dos tipos de VMs, é um trabalho inédito. Considerou-se nesta afirmação que os objetivos utilizados, de forma simultânea, não foram encontrados em outros trabalhos científicos sendo tratados na forma pura de otimização com múltiplos objetivos, ao mesmo tempo em que se aplica a técnica de sobrecarga de recursos.

Como se pôde ver, a abordagem desenvolvida neste trabalho é muito versátil, possibilitando uma gama de configurações que permitem ao administrador do *datacenter* ir ajustando os parâmetros de acordo com suas preferências.

Essa flexibilidade, que a princípio parece complicada, permite um leque de ajustes finos que irão trabalhar a favor dos administradores destes ambientes. O código da abordagem VMPOS desenvolvida está disponível na página pessoal do autor²⁰ na plataforma GitHub²¹ na Internet.

Mais detalhes sobre as características dos parâmetros de entrada para o modelo podem ser encontrados no Apêndice A.

4.4 Formulação do Problema de Posicionamento de VMs

Esta seção apresenta a forma como o problema de posicionamento de VMs, em *Datacenter* Privado, foi modelado. Ao buscar um posicionamento de VMs em MFs deseja-se, segundo alguns objetivos específicos, encontrar um mapeamento no qual estão as indicações de quais VMs devem ser alocadas em quais MFs. Tal resultado poderá gerar possíveis migrações de VMs entre MFs, alterações no consumo de energia do *datacenter* e uma maior ou menor dispersão de cada um dos tipos de VMs.

²⁰<https://github.com/petroniocg/VMPOS>

²¹<https://github.com/>

Na Tabela 4.1 estão resumidos todos os componentes definidos para a abordagem VMPOS.

Tabela 4.1: Componentes existentes na formulação matemática da abordagem.

Componentes da Abordagem	
Representação	Descrição
V	Conjunto com m máquinas virtuais.
m	Quantidade de VMs em V .
H	Conjunto com n máquinas físicas que hospedam VMs.
n	Número de máquinas físicas de H .
r	Conjunto com t recursos, como CPU, Memória e Disco.
t	Quantidade de recursos utilizados na abordagem.
U_{cpu_j}	Representa a utilização dos recursos de processamento, utilizados por H_j
FM	Fator de Migrações, leva em conta o número de migrações de VMs de cada tipo necessárias para aplicar uma dada solução.
V_i	Representa uma VM i , com $V_i \in V$ e $(1 \leq i \leq m)$.
H_j	Representa uma máquina real hospedeira j , com $H_j \in H$ e $(1 \leq j \leq n)$.
P_{ij}	Indica se a VM V_i está posicionada ou não para ser executada na MF H_j .
Y_j	Variável que indica se uma máquina real está ou não ligada.
NC_j	Nível de comprometimento de uma máquina física H_j .
f_1	Função objetivo que representa o Fator de Migração.
α	Quantidade de VMs com Serviços Críticos que migraram.
β	Quantidade de VMs com Serviços Não Críticos que migraram.
θ	Quantidade total de VMs com Serviços Críticos no <i>datacenter</i> .
δ	Quantidade total de VMs com Serviços Não Críticos no <i>datacenter</i> .
f_2	Função objetivo que representa o Consumo de Energia total das máquinas físicas do DC;
f_3	Função objetivo que representa o Nível de Dispersão total calculado para uma determinada solução.
TMF	Total de Máquinas Físicas que abrigam pelo menos uma VM com Serviços Críticos (TMF_{SC}) ou apenas VMs com Serviços Não Críticos (TMF_{SNC}).

4.4.1 Dados de Entrada

Vale ressaltar aqui que um problema de otimização pode estar relacionado tanto a minimização quanto a maximização [68]. Para comparar duas soluções em um contexto que

envolve múltiplos objetivos, o conceito de *dominância de Pareto* foi utilizado.

Considere um conjunto V com m máquinas virtuais e um conjunto H com n máquinas reais que hospedam as VMs de V :

$$V = \{V_1, V_2, \dots, V_m\}; \quad (4.1)$$

$$H = \{H_1, H_2, \dots, H_n\}. \quad (4.2)$$

Desta forma, V_i indica a máquina virtual i ($1 \leq i \leq m$), com $V_i \in V$ e H_j representa a máquina física j ($1 \leq j \leq n$), com $H_j \in H$.

Além das configurações do sistema, as outras duas informações necessárias para a execução da abordagem são: as informações das MFs e VMs presentes no DC e o indivíduo base, conforme visto anteriormente no esquema da Figura 4.6.

Dado um *datacenter* privado, constituído por um conjunto H de MFs e as indicação de seus recursos, e de um conjunto V de VMs com indicações se estão executando ou não serviços críticos, além da indicação de seus requisitos, deseja-se obter um conjunto de soluções em forma de mapeamento eficiente de VMs para MFs, satisfazendo um grupo de restrições e ao mesmo tempo otimizando todas as funções objetivo, que são: (1) *Minimização do Número de Migrações*, (2) *Minimização do Consumo de Energia* e (3) *Minimização da Dispersão*. Isso em um contexto de otimização simultânea pura de múltiplos objetivos.

Aqui, foram consideradas informações de: *CPU*, *memória*, *disco* e *consumo de energia*.

Seja r um conjunto com t recursos: $r = \{r_1, r_2, \dots, r_t\}$, em que r_s indica um determinado recurso s ($1 \leq s \leq t$), e $r_s \in r$.

Nesta tese os recursos considerados foram: $r' = \{CPU, memória, disco\}$, para o estabelecimento das restrições, acrescidas da informação de consumo total de energia no *datacenter* para decisão de alocação das VMs, assim como na escolha de indivíduo único no final do processo.

O conjunto de máquinas físicas será representado pela matriz H com dimensões ($n \times 4$). Cada H_j possui indicação de recursos de CPU, quantidade de memória RAM, capacidade de armazenamento e consumo máximo de energia, como:

$$H_j = [H_{cpu_j}, H_{ram_j}, H_{arm_j}, H_{pmax_j}], \quad \forall j \in \{1, \dots, n\}, \quad (4.3)$$

em que,

H_{cpu_j} : Informa a quantidade de recursos de processamento de H_j ;

H_{ram_j} : Informa a quantidade de memória RAM de H_j ;

H_{arm_j} : Informa a capacidade de armazenamento do disco de H_j ;

H_{pmax_j} : Informa o consumo máximo de energia de H_j ;

n : Número de máquinas físicas de H ;

Considerar-se-á uma diversidade de MFs, existentes no ambiente, e de tipos de VMs que poderão ser requisitadas pelos usuários (mais detalhes na Seção 5.1, com as características das VMs e MFs utilizadas). O conjunto V será representado como uma matriz com dimensões ($m \times 4$) em que cada linha possui informações de uma VM V_i , como: indicação dos requisitos de CPU, quantidade de memória RAM requerida, capacidade de armazenamento requerida e informação se está ou não executando serviços críticos. Logo:

$$V_i = [V_{cpu_i}, V_{ram_i}, V_{arm_i}, V_{sc_i}], \quad \forall i \in \{1, \dots, m\}, \quad (4.4)$$

em que,

V_{cpu_i} : Informa a quantidade de recursos de processamento requerida por V_i ;

V_{ram_i} : Informa a quantidade de recursos de memória requerida por V_i ;

V_{arm_i} : Informa a capacidade de armazenamento requerida por V_i ;

V_{sc_i} : Informa se V_i executa ou não serviços críticos. Será igual a 1 se V_i está executando algum serviço crítico, caso contrário, será igual a 0;

m : Quantidade de VMs em V .

4.4.2 Dados de Saída

Como saída da abordagem, será retornada a melhor solução encontrada (dentre as que mais se aproximam da Fronteira de Pareto) constituída dos posicionamentos de cada VM V_i em cada MF H_j . Esse resultado virá após a evolução realizada através do algoritmo memético, desde a situação inicial, considerando os critérios dos múltiplos objetivos a serem otimizados.

A escolha de solução única, dentre as possíveis soluções encontradas na evolução, acontece aplicando-se pesos atribuídos a cada um dos três objetivos normalizados, conforme detalhado na próxima Seção 4.4.3 e no Apêndice A.

Um posicionamento (ou solução, ou indivíduo), como o indivíduo base de entrada ou uma solução encontrada, será representado como uma matriz $P = \{P_{ij}\}$ de dimensão $(m \times n)$, em que $P_{ij} \in \{0, 1\}$ indica se a VM V_i está localizada ($P_{ij} = 1$) ou não ($P_{ij} = 0$) para ser executada na MF H_j . Ou seja, um elemento P_{ij} indica um mapeamento de uma V_i em uma H_j .

4.4.3 Aplicação de Pesos

Quando se utiliza um método de busca por soluções como o da fronteira de Pareto, o tomador de decisões se depara ao final do processo com algumas soluções possíveis, todas não comparáveis. São as soluções que pertencem a fronteira número 1 de Pareto, no caso da abordagem aqui desenvolvida.

Neste trabalho de doutorado, decidiu-se eleger a estratégia de se aplicar pesos para os objetivos como uma forma de se escolher apenas uma, dentre as possíveis soluções presentes na fronteira de Pareto, tanto favorecendo-se um dos objetivos em detrimento dos demais (aplicando valores diferentes para os pesos), como sem favorecimento nenhum dos objetivos utilizados (aplicando valores iguais para os pesos).

É necessário, porém, atentar para o fato de que o processo de escolha das soluções que fazem parte da fronteira de Pareto é executado sem a aplicação de nenhum tipo de peso, com evolução dos três objetivos ocorrendo absolutamente de forma simultânea. Só após a conclusão dessa fase é que ocorre a escolha de solução única, aplicando os pesos definidos na etapa de configuração da abordagem, descrita no Apêndice A.

Essa propriedade da abordagem VMPOS permite que o administrador do *datacenter* possa decidir se quer, por exemplo, favorecer a solução em que o consumo de energia seja o menor possível, independentemente do número de migrações que ocorram ou da dispersão que existirá para aplicar a solução com gasto energético mínimo. Da mesma forma, pode-se decidir por uma solução com menor valor de Migração, independentemente do gasto energético ou da dispersão de VMs que será obtida após aplicada a solução desejada.

A função objetivo utilizada para os comparativos da abordagem VMPOS, com e sem

favorecimento, tem a forma:

$$fitness = (E_N \cdot PE) + (D_N \cdot PD) + (M_N \cdot PM) \quad (4.5)$$

em que,

E_N , D_N e M_N : São os valores normalizados do consumo de Energia, do nível de Dispersão e do número de Migrações, respectivamente;

PE , PD e PM : São os valores dos pesos atribuídos aos objetivos: consumo de Energia, nível de Dispersão e número de Migrações, respectivamente.

4.4.4 Funções Objetivo

Ao se deparar com o problema de alocação de VMs em um *datacenter*, uma grande quantidade de objetivos para otimização do posicionamento das VMs pode vir à mente. Os critérios utilizados para obter a melhor alocação possível vai depender das prioridades e dos objetivos de otimização [44].

O problema de posicionamento de VMs em *datacenter* pode ser definido como um problema de otimização com múltiplos objetivos. Com o intuito de contemplar objetivos considerados importantes para qualquer *datacenter*, priorizando a qualidade dos serviços críticos alocados em VMs, a abordagem desenvolvida utiliza *Algoritmos Meméticos*, que evolui as soluções encontradas atendendo a múltiplos objetivos de forma simultânea pura (no sentido de tratar os objetivos simultaneamente sem reduzi-los a um só objetivo).

Aqui, considerou-se a otimização de forma simultânea de três objetivos que estão descritos na sequência. Contudo, vale ressaltar que o mecanismo aqui proposto é genérico o suficiente para que sejam escolhidos mais, ou outros objetivos a serem atendidos de acordo com a necessidade específica de cada *datacenter*.

Minimização do Número de Migrações

O processo de transferência de uma VM de uma máquina física para outra, chamado de migração, pode acontecer em tempo real (conhecido como *live migration*) e é bastante utilizado nos *datacenters* sendo uma tecnologia importante para realizar o balanceamento de carga entre os servidores físicos existentes [22]. A técnica também é utilizada para evitar

descumprimentos de acordos de SLA, que estabelecem os níveis mínimos de qualidade de serviços prestados por uma empresa a um cliente, em nuvens públicas [88].

Apesar dos benefícios obtidos, não se pode desprezar a sobrecarga gerada com o processo de migração. A abordagem proposta levará em consideração o número de migrações das VMs tentando, dessa forma, fazer com que esse número seja o menor possível. Com os objetivos sendo atendidos de forma simultânea, existe uma força que tensiona na direção de cada objetivo.

No caso do número de migrações, quando da análise para criação de nova geração através do algoritmo memético, as soluções encontradas que irão gerar um menor número de migrações de VMs são as que dominarão as outras soluções. Com isso, beneficia-se a rede como um todo, devido ao menor número de VMs se deslocando dentro do ambiente, assim como, minimiza-se o número de VMs que sofrerão impactos associados aos processos de migração nos serviços nelas implantados.

Para contabilizar o número de migrações de cada solução, e auxiliar no processo de tomada de decisão de qual solução é dominante, será introduzida uma variável *FM* (Fator de Migrações) que está associado à quantidade de migrações que serão necessárias realizar para aplicar uma determinada solução gerada pelo algoritmo memético, saindo do posicionamento de um indivíduo base para o posicionamento da solução encontrada. Portanto, um dos objetivos será minimizar o número de migrações em cada nova solução gerada, através do fator de migração.

O *FM* será calculado sempre em relação ao indivíduo base, levando em consideração os tipos de VMs existentes através da aplicação de custos. O fator é então calculado comparando-se quantas VMs mudaram de posição no indivíduo gerado com relação ao indivíduo base.

Para exemplificar, suponha que duas VMs com SC e três VMs com SNC precisariam mudar de MF em relação ao indivíduo base, caso fosse decidido aplicar o posicionamento encontrado com o novo indivíduo gerado. Essas quantidades serão representadas por α e β , respectivamente. Ou seja, para o exemplo, $\alpha = 2$ e $\beta = 3$.

Suponha também que os custos aplicados para cada tipo de VM são 2 e 1 para VMs com SC ($Custo_{VM_{SC}}$) e com SNC ($Custo_{VM_{SNC}}$), respectivamente. Portanto, o *FM*, para se mudar da posição existente no indivíduo base para a do indivíduo gerado, seria calculado

por:

$$FM = (Custo_VM_{SC} \times \alpha) + (Custo_VM_{SNC} \times \beta) = 2 \times 2 + 1 \times 3 = 7$$

Logo, um $FM = 7$ representaria o “número de migrações” que seriam necessárias realizar para sair do posicionamento representado pelo indivíduo base e aplicar o posicionamento do novo indivíduo encontrado após o processo de evolução. No exemplo, precisaria migrar cinco VMs, em que duas executam serviços críticos e as outras três, serviços não críticos.

A função objetivo da expressão 4.6 determina como é realizado o cálculo do Fator de Migração. Um dos objetivos desta pesquisa é minimizar esta função.

$$f_1 = (Custo_VM_{SC} \times \alpha) + (Custo_VM_{SNC} \times \beta), \quad (4.6)$$

em que,

f_1 : Número que representa o Fator de Migrações levando em consideração o número de migrações necessárias, de cada tipo de VMs, para sair da configuração do indivíduo base para uma nova configuração, representada por uma determinada solução.

Os valores α e β na expressão de f_1 são calculados da seguinte forma:

$$\alpha = \sum_{i=1}^m Vsc_i [(P_{ij})_{NI} - (P_{ij})_{IB}], \forall j \in \{1, \dots, n\} \text{ e } [(P_{ij})_{NI} - (P_{ij})_{IB}] > 0, \quad (4.7)$$

$$\beta = \sum_{i=1}^m [(P_{ij})_{NI} - (P_{ij})_{IB}], \text{ se } Vsc_i = 0, \quad (4.8)$$

$$\forall j \in \{1, \dots, n\} \text{ e } [(P_{ij})_{NI} - (P_{ij})_{IB}] > 0.$$

em que,

α : quantidade de VMs com Serviços Críticos que migraram;

β : quantidade de VMs com Serviços Não Críticos que migraram;

P_{ij} : indica se a VM V_i está posicionada ou não para ser executada na MF H_j .

NI : índice que indica Novo Indivíduo;

IB : índice que indica Indivíduo Base;

Minimização do Consumo de Energia

Um dos principais objetivos encontrados nos trabalhos relacionados ao posicionamento de VMs em *datacenters* é o de minimização do consumo de energia [88][89][22][83][49]. A energia consumida por um servidor físico provém de componentes como CPU, memória e disco. O valor total de energia consumida flutua dinamicamente de acordo com a carga das CPUs do servidor [90].

A expressão aplicada, especificamente para calcular o consumo de energia de uma máquina física, leva em consideração um parâmetro chamado de *Utilização de CPU*. Esse parâmetro representa a utilização dos recursos de processamento, utilizados por uma máquina física do SDC, em determinado instante. Será representado por U_{cpu} e calculado como a razão entre o total de vCPUs, requisitadas pelas VMs hospedadas na MF, pelo total de CPUs disponíveis para utilização na MF, considerando os níveis de comprometimento.

Assim,

$$U_{cpu_j} = \frac{\sum_{i=1}^m V_{cpu_i} \times P_{ij}}{H_{cpu_j} \times NC_j}, \quad \forall j \in \{1, \dots, n\}. \quad (4.9)$$

Note que, se uma MF H_j não estiver sendo utilizada, não haverá utilização de vCPUs, o que significa $U_{cpu_j} = 0$.

Seguindo o que foi proposto em [44], o atual trabalho também considera o objetivo de minimizar o consumo de energia conforme a expressão:

$$f_2 = \sum_{j=1}^n ((H_{pmax_j} - H_{pmin_j}) \times U_{cpu_j} + H_{pmin_j}) \times Y_j. \quad (4.10)$$

em que,

f_2 : Consumo de Energia total das máquinas físicas do DC, em *Watt-hora* (Wh);

H_{pmin_j} : Representa o consumo de energia mínimo de H_j . A expressão utilizada para o cálculo é conforme apresentada em [90]: $H_{pmin_j} = H_{pmax_j} \times 0,6$. Segundo [90], dados reais apontam que, tipicamente, H_{pmin_j} vale aproximadamente $H_{pmax_j} \times 0,6$, e dificilmente é menor que $H_{pmax_j} \times 0,5$ [91];

U_{cpu_j} : Representa a utilização dos recursos de processamento, utilizados por H_j conforme Equação 4.9.

Y_j : Variável binária que indica se a máquina física H_j está ligada (será igual a 1) ou desativada (será igual a 0). Tal valor será atribuído dependendo do valor de U_{cpu_j} , se esse for diferente de zero, $Y_j = 1$, caso seja igual a zero, $Y_j = 0$.

Minimização do Nível de Dispersão

Com o objetivo de conseguir obter o melhor aproveitamento possível da estratégia de sobrecarga de recursos (*oversubscription*), aplicando os diferentes níveis de comprometimento como discutido anteriormente, neste trabalho propôs-se um método para tentar concentrar os dois diferentes tipos de VMs, críticas junto com críticas e não críticas junto com não críticas, no menor número de MFs possível. Para isso, foi desenvolvida de forma empírica uma equação que mede o grau de concentração das VMs críticas e não críticas. Quanto **menor o valor desse parâmetro, maior a concentração dos dois tipos de VMs** em um menor número de máquinas físicas.

Uma das funções objetivo que se deseja minimizar é a que calcula o nível de dispersão de VMs, segundo a expressão empiricamente estabelecida e apresentada em 4.11.

$$f_3 = \frac{TMF_{SC}}{\theta} + \frac{\delta}{TMF_{SNC}}, \quad (4.11)$$

em que,

f_3 : Representa o Nível de Dispersão total calculado para uma determinada solução de posicionamento;

θ : quantidade total de VMs com Serviços Críticos no *datacenter*;

δ : quantidade total de VMs com Serviços Não Críticos no *datacenter*;

TMF_{SC} : Total de Máquinas Físicas, na solução em questão, que abrigam pelo menos uma VM com Serviços Críticos;

TMF_{SNC} : Total de Máquinas Físicas, na solução em questão, que abrigam apenas VMs com Serviços Não Críticos.

Para reduzir o valor da dispersão (f_3) é necessário aumentar o valor da razão de VMs com SC por MF com SC, ou a razão de VMs com SNC por MF com SNC ou essas duas razões ao mesmo tempo. Isso se faz exatamente reduzindo o número de MFs com SC ou de MFs com SNC sendo utilizadas para hospedar as VMs de cada tipo. Ou seja, concentrando

as VMs críticas e não críticas.

Para exemplificar, suponha um *datacenter* com 3 MFs, e 8 VMs para serem alocadas. Considere os seguintes dados e representações: O Nível de Comprometimento (NC) será igual a 1 nas Máquinas Físicas que estiverem alocando pelo menos uma VM com Serviços Críticos, e será igual a 2 nas MFs que estiverem alocando apenas VMs com Serviços Não Críticos; Que as MFs são iguais, cada uma com 6 CPUs e 12 GB de Memória; Que as VMs são iguais, cada uma requisitando 2 vCPUs e 4 GB de Memória; e que existem 2 VMs com SC das 8 VMs presentes no *datacenter*.

Possíveis posicionamentos das VMs nas MFs podem ser vistos nas situações *a*, *b* e *c* da Figura 4.8.

Figura 4.8: Exemplificação do cálculo do Nível de Dispersão.



Fonte: Elaborada pelo autor.

Na Figura 4.8, observa-se que: na Situação *a*) tem-se 2 VMs com SC ocupando 2 MFs e 6 VMs com SNC ocupando 3 MFs; Na Situação *b*) tem-se 2 VMs com SC ocupando 1 MF e 6 VMs com SNC ocupando 3 MFs; E, na Situação *c*) tem-se 2 VMs com SC ocupando 1 MF e 6 VMs com SNC ocupando 2 MFs;

Calculando os resultados dos valores de Dispersão, segundo a expressão 4.11, obtêm-se:

$$\text{Na Situação a): Nível de Dispersão} = \frac{2}{\binom{2}{2} + \binom{6}{3}} = 0,67$$

$$\text{Na Situação b): Nível de Dispersão} = \frac{1}{\binom{2}{1} + \binom{6}{3}} = 0,25$$

$$\text{Na Situação c): Nível de Dispersão} = \frac{1}{\binom{2}{1} + \binom{6}{2}} = 0,20$$

De acordo com os resultados obtidos, constata-se que o menor Nível de Dispersão é o do arranjo da Situação c), sendo também o que realiza melhor utilização dos recursos físicos disponíveis e maior concentração de VMs de cada tipo (com SC e com SNC).

Portanto, diante do que foi exposto até o presente, para resolver o problema com múltiplos objetivos, deseja-se:

Otimizar:

$$y = [f_1, f_2, f_3], \quad (4.12)$$

em que,

f_1 : Número de Migrações;

f_2 : Consumo de Energia;

f_3 : Nível de Dispersão.

Sujeito às restrições de 4.13 a 4.15, descritas na Seção 4.4.5.

Este tipo de problema normalmente é resolvido utilizando-se Programação Linear, Algoritmos Evolucionários ou heurística [92][93].

4.4.5 Restrições

Para estabelecer algumas restrições, é necessário definir os níveis de comprometimento que, como dito anteriormente, é dinâmico e depende dos tipos de VMs alocados na MF em determinado instante. Será utilizada a notação *SC*, para MFs com Nível de Comprometimento de Serviços Críticos, ou seja, que abrigam no mínimo uma VM com serviços críticos,

e *SNC*, para MFs com Nível de Comprometimento de Serviços Não Críticos, ou seja, que abrigam apenas VMs com serviços não críticos.

De forma geral, será utilizado apenas *NC* para indicar que um dos Níveis de Comprometimento, correspondente à situação atual da MF, deverá ser aplicado.

Nas expressões a seguir são estabelecidas as restrições associadas ao problema formulado.

$$\sum_{j=1}^n P_{ij} \leq 1, \quad \forall i \in \{1, \dots, m\} \text{ e } P_{ij} \in \{0, 1\}; \quad (4.13)$$

$$\sum_{i=1}^m V_i(r_s) \times P_{ij} \leq H_j(r_s) \times NC_j, \quad \forall j \in \{1, \dots, n\}, \quad \forall s \in \{1, \dots, t\}; \quad (4.14)$$

$$\sum_{j=1}^n P_{ij} = 1, \quad \forall i \in \{1, \dots, m\}, \quad sc_i \in \{0, 1\} \text{ e } V_{sc_i} = 1; \quad (4.15)$$

A restrição (4.13) garante que cada VM só pode ser alocada em uma única MF de *H*.

A restrição (4.14), relacionada aos limites de utilização dos recursos de uma máquina física, garante que a soma das capacidades de todas as VMs, considerando algum dos recursos r_s (de CPU, memória ou armazenamento), deve ser compatível com a capacidade, do mesmo recurso r_s existente na MF que está hospedando essas VMs, considerando o nível de comprometimento aplicado no instante da verificação.

A restrição (4.15) garante que cada VM V_i que abriga serviços críticos (ou seja, $V_{sc_i} = 1$) tem necessariamente que estar alocada em alguma MF H_i .

4.5 Alocação com Múltiplos Objetivos no VMPOS

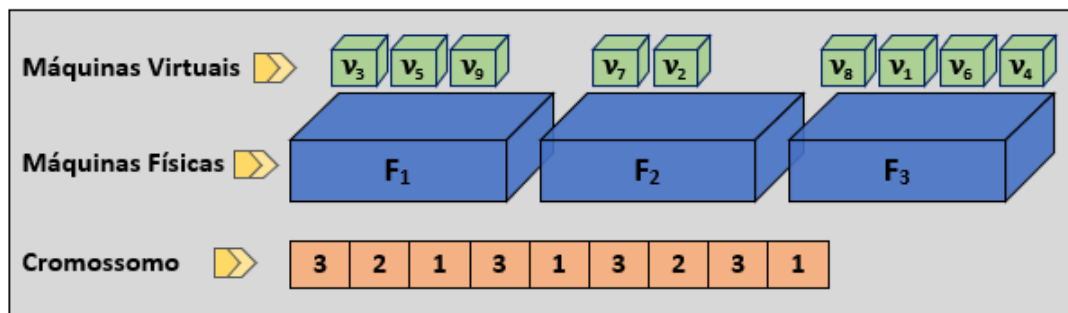
Inicialmente será descrita a configuração utilizada para codificação do cromossomo que representa um indivíduo (solução para o problema de posicionamento) no processo evolutivo do Algoritmo Memético. Logo em seguida, serão tratados os algoritmos que compõem a alocação com múltiplos objetivos.

4.5.1 Codificação

Neste trabalho foi utilizada uma estrutura de cromossomos como proposta em [94], de tal forma que a representação utiliza um vetor M , de dimensão m , tal que $M = [M_1, M_2, \dots, M_m]$, com $M_i \in \mathbb{N}$, e $i \in \{1, \dots, m\}$. Cada M_i representa a máquina física H_j que pertence ao conjunto H , na qual a VM V_i está hospedada para execução. Se $M_i = 0$, a máquina virtual V_i não foi alocada para executar em nenhuma MF.

Para exemplificar, considere a Figura 4.9 com uma simulação de posicionamento de VMs, e seu respectivo cromossomo.

Figura 4.9: Representação da alocação de nove VMs em três Máquinas Físicas e o cromossomo correspondente.



Fonte: Adaptado de PIRES e BARÁN, 2013, p. 4.

Como se observa na Figura 4.9, têm-se nove VMs particionadas em três MFs, e o cromossomo correspondente com nove genes. Cada gene codifica qual VM está alocada em qual MF. A posição no vetor do cromossomo indica a VM, e o valor armazenado no vetor indica em qual MF está hospedada a VM.

Portanto, cada cromossomo M pode ser mapeado de maneira única para uma solução P , correspondendo ao posicionamento das VMs no *datacenter*, como definido na Seção 4.4.2. Por exemplo, para o cromossomo da Figura 4.9, $M = [3, 2, 1, 3, 1, 3, 2, 3, 1]$, tem-se a solução mostrada em 4.16.

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (4.16)$$

Ou seja, se $M_i = 0$, então $P_{ij} = 0, \forall j \in \{1, 2, \dots, n\}$. O que significa que a VM V_i não está em execução em nenhuma MF;

Se $M_i = j$, então $P_{ij} = 1$, o que implica que $P_{ik} = 0, \forall k \neq j$, ou seja, a VM V_i está alocada e em execução apenas na MF H_j .

4.5.2 Algoritmo Memético

O algoritmo memético utilizado neste trabalho (Algoritmo 2), toma por base o que foi desenvolvido por em [83] e posteriormente expandido em 2015 [44] e em 2017 [84].

O algoritmo memético desenvolvido tem o seguinte funcionamento básico:

- Linhas 1 e 2: Inicia realizando leituras das informações necessárias para execução;
- Linha 3: É verificado se foi fornecido ou não um indivíduo base. Se existir ele será lido, caso não exista, este indivíduo é gerado de forma aleatória na Linha 5;
- Linha 6: A população inicial P_0 de soluções candidatas é gerada, conforme o que está descrito na seção 4.5.3;
- Linha 7: As possíveis soluções geradas são reparadas, corrigindo aquelas não factíveis, segundo a série de restrições impostas ao problema (Seção 4.4.5), e garantido continuação apenas com soluções possíveis;

Algoritmo 2: Algoritmo Memético**Entrada:** Infraestrutura do *datacenter*, possível indivíduo base e configurações**Saída:** Melhor indivíduo obtido**início**

```

1   Lê infraestrutura do datacenter
2   Lê configurações da abordagem VMPOS
3   se Existe indivíduo base então
4     |   Lê indivíduo base
5     |   senão
6     |     |   Gera indivíduo base aleatoriamente
7     |   fim
8     |   Inicializa o conjunto de soluções  $P_0$ 
9     |    $P'_0$  = correção das soluções não viáveis de  $P_0$ 
10    |    $P''_0$  = aplica melhora local para as soluções de  $P'_0$ 
11    |   Atualize o conjunto de soluções não dominadas  $P_c$  a partir de  $P''_0$ 
12    |    $t = 0, P_t = P''_0$ 
13    |   enquanto Condição de parada não for satisfeita faça
14    |     |    $Q_t$  = seleção de soluções a partir de  $P_t \cup P_c$ 
15    |     |    $Q'_t$  = cruzamento e mutação das soluções de  $Q_t$ 
16    |     |    $Q''_t$  = correção das soluções não viáveis de  $Q'_t$ 
17    |     |    $Q'''_t$  = aplica melhora local para as soluções de  $Q''_t$ 
18    |     |   incremente  $t$ 
19    |     |   Atualize o conjunto de soluções não dominadas  $P_c$  a partir de  $Q'''_t$ 
20    |     |    $P_t$  = seleção por fitness a partir de  $P_t \cup Q'''_t$ 
21    |     |   fim
22    |     Escolha de indivíduo único como solução, dentro do Conjunto de Pareto gerado  $P_c$ , com base nos
23    |     pesos aplicados aos diferentes objetivos
24    |     retorna Melhor indivíduo obtido
25  fim

```

- Linha 8: O algoritmo tenta melhorar as soluções corrigidas na linha 7, usando a busca de melhoria local;
- Linha 9: A primeira aproximação conhecida de Pareto (P_c) é alcançada na linha 9, com as soluções não dominadas obtidas;
- Linha 11: A evolução é iniciada na repetição que começa nessa linha. Vale lembrar que

o processo de atualização das soluções não dominadas garante a aplicação de Elitismo, conforme explicado na Seção 2.6.2;

- Linha 19: Finalizada a evolução, o algoritmo encontra, na linha 19, o melhor indivíduo presente na fronteira P_c de Pareto gerada, de acordo com os pesos aplicados para cada um dos três objetivos considerados;
- Linha 20: Finalmente, o melhor indivíduo obtido é retornado.

Mais detalhes e algoritmos da abordagem são descritos a seguir.

4.5.3 Inicialização da População

A população inicial sempre será gerada de forma aleatória, composta por uma quantidade de indivíduos que deve ser indicada nas configurações de entrada. Caso não exista um indivíduo base, a população inicial deve ser criada com um indivíduo a mais do que a quantidade indicada nas configurações de entrada. Esse indivíduo a mais será utilizado como indivíduo base (ver Seção 4.3.1).

Cada uma das soluções, ou indivíduos no contexto de algoritmos meméticos, que formam a população inicial, terão a forma $M = [M_1, M_2, \dots, M_m]$. Para todas as VMs que estiverem executando serviços críticos, os valores possíveis de M_i para a VM V_i serão gerados no intervalo $[1, n]$. E para as VMs que não estiverem executando serviços críticos, os valores possíveis serão gerados no intervalo $[0, n]$. Dessa maneira, na fase de inicialização da abordagem, utilizando a informação em V_{sc_i} , o algoritmo vai garantir que todas as VMs V_i , que foram indicadas como *executando* serviços críticos, sejam hospedadas em alguma MF H_j . Para todas as demais V_i , que foram indicadas como *não executando* serviços críticos, existe a possibilidade de não serem alocadas para execução em alguma MF. Assim, obtém-se a população inicial P_0 .

4.5.4 Correção das Soluções não Factíveis

Com a utilização de geração aleatória de soluções, na fase de inicialização da população (linha 6 do Algoritmo 2), ou através dos operadores genéticos (cruzamento ou mutação - linha 13 do Algoritmo 2), é possível que sejam geradas soluções que não atendam às restrições

(de 4.13 a 4.15) impostas ao problema. Por exemplo, posicionar no mesmo hospedeiro mais VMs que a capacidade dos recursos desta MF pode suportar, mesmo aplicando a sobrecarga de recursos.

Para corrigir as soluções não viáveis (linhas 7 e 14 do Algoritmo 2), que porventura tenham sido geradas, são utilizadas duas etapas: primeiro verifica-se a condição de viabilidade das soluções, classificando a população em dois grupos: viáveis e inviáveis. O Algoritmo 3 faz a verificação de viabilidade das soluções.

Algoritmo 3: Verifica Viabilidade

Entrada: Conjunto de soluções P_t

Saída: Conjunto de soluções viáveis P'_t

```

início
1  enquanto existir solução para verificar faça
2      viavel = true, j = 1
3      enquanto  $(j \leq n) \ \&\& \ (viavel)$  faça
4          se a solução não satisfaz as restrições (4.13 a 4.15) então
5              viavel = false, break
6              senão
7                  incremente j
8              fim
9          fim
10         se viavel = false então
11             Chama o Algoritmo 4 para corrigir a solução
12         fim
13     fim
14 retorna Conjunto de Soluções Viáveis  $P'_t$ 
fim

```

Como processo de tentativa de reparação, as soluções que foram classificadas como inviáveis são submetidas às tentativas de correção que passam por diversas estratégias, como: (1) migrar algumas VMs entre MFs já em utilização, (2) “ligar” algumas MFs e então migrar algumas VMs para essas MFs ou (3) “desligar” algumas VMs com serviços não críticos. Lembrando que essas etapas não são realizadas fisicamente, são fases que existem no algoritmo no processo de busca pela melhor solução de posicionamento. A solução final com a distribuição de VMs por MFs, encontrada pela abordagem, é repassada para o Escalona-

dor de Recursos que é quem efetivamente realiza o procedimento de migrações das VMs, conforme discutido na Seção 4.1.

As tentativas de migrações entre hospedeiros procuram alocar as VMs a serem migradas preferencialmente em MFs que possuam nível de comprometimento igual ao da VM em deslocamento. Isso ajuda a manter os grupos dos dois tipos de VMs concentrados o máximo possível, contribuindo com o objetivo de minimizar dispersão.

O Algoritmo 4 realiza a correção das soluções identificadas como não viáveis.

Algoritmo 4: Correção das Soluções Inviáveis

Entrada: Solução inviável

Saída: Solução viável

início

```

1  viavel = false, i = 1
2  enquanto ( (i ≤ m) && (viavel = false) ) faça
3      se for possível então
4          Migre  $V_i$  para alguma  $H_{j'}$  ( $j' \neq j$ ) com mesmo nível de comprometimento do de  $V_i$ 
5          senão
6              se for possível então
7                  Migre  $V_i$  para alguma  $H_{j'}$  ( $j' \neq j$ ) com nível de comprometimento diferente do de
8                   $V_i$ 
9                  senão
10                     se  $V_i$  não executar serviços críticos então
11                         Desligue  $V_i$  em  $H_j$ 
12                     fim
13                 fim
14             fim
15         fim
16     se a solução satisfaz as restrições (4.13 a 4.15) então
17         viavel = true, break
18     fim
19     incremente i
20 fim
21 retorna Solução viável
22 fim
```

Note que, na linha 4 desse algoritmo, tenta-se migrar a VM independentemente do tipo de serviço que esta executa, se crítico ou não, contudo a tentativa é de alocar a VM em MF

com mesmo nível de comprometimento que a VM a ser migrada.

Note que as VMs podem migrar para outra MF mesmo se esta MF estiver desativada. Caso não seja encontrada MF que atenda ao requisito de mesmo nível de comprometimento, na linha 6 tenta-se migrar a VM para MF com nível de comprometimento diferente do aplicado na VM sendo migrada. Caso não tenha sido possível migrar a VM até essa fase, a VM é “desligada” na linha 7. Desligar aqui significa apenas atribuir zero como conteúdo no cromossomo, na posição relativa à da VM em questão.

4.5.5 Aperfeiçoamento Local

Após toda a população ser verificada e reparada, estando cada indivíduo com disposições de VMs em MFs que reflete apenas soluções viáveis, o algoritmo memético executa o Aperfeiçoamento Local, ou melhora local. O objetivo desse passo é tentar melhorar as soluções encontradas na população P_t em evolução. Esse procedimento está descrito no pseudocódigo do Algoritmo 5.

Para tentar melhorar cada uma das soluções da população P_t , o algoritmo gera um número aleatório de 0 a 1, e o utiliza como probabilidade. Caso o número tenha sido maior ou igual a $1/2$, tenta-se primeiro maximizar o número de VMs hospedadas nas MFs, alocando todas as possíveis VMs que ainda não foram alocadas, levando em consideração seus respectivos níveis de comprometimento. Com isso, o objetivo de reduzir o Nível de Dispersão será beneficiado. Depois, tenta desligar todas as MFs possíveis, migrando as VMs para outros hospedeiros com recursos disponíveis.

Caso o número aleatório gerado tenha sido menor que $1/2$, o algoritmo tenta primeiro minimizar o número de MFs ligadas agrupando as VMs em um menor número de hospedeiros, favorecendo assim o objetivo de minimizar o Consumo de Energia, desligando servidores físicos. Depois, tenta alocar as VMs marcadas como desligadas nas MFs ligadas.

Algoritmo 5: Aperfeiçoamento Local

Entrada: Conjunto de soluções viáveis P'_t

Saída: Conjunto de soluções possíveis otimizada P''_t

- 1 probabilidade = gere um número aleatório no intervalo [0,1]

início

- 2 **enquanto** (*existir soluções não verificadas*) **faça**
- 3 **se** *probabilidade* < 0.5 **então**
- 4 Tente “desligar” todas as H_j possíveis, migrando todas as V_i para $H_{j'}$ com recursos disponíveis ($j \neq j'$) e mesmo nível de comprometimento que a VM a ser migrada.
- 5 Depois, tente ligar todas as possíveis V_i alocando-as em H_j com recursos disponíveis e mesmo nível de comprometimento que a VM a ser “ligada”.
- 6 **senão**
- 7 Tente “ligar” todas as possíveis V_i alocando-as em H_j com recursos disponíveis e mesmo nível de comprometimento que a VM a ser “ligada”.
- 7 Depois, tente “desligar” todas as H_j possíveis, migrando todas as V_i para $H_{j'}$ com recursos disponíveis ($j \neq j'$) e mesmo nível de comprometimento que a VM a ser migrada.
- 8 **fim**
- 9 **fim**
- 10 **retorna** *Conjunto de soluções possíveis otimizada* P''_t

fim

A título de exemplo, considere que uma população apresente grande dispersão de VMs alocadas nas MFs, ou seja, existem muitas MFs sendo utilizadas, porém com muitos recursos sub utilizados. Segundo o aperfeiçoamento local, o algoritmo tentará migrar todas as VMs de uma determinada MF para outras MFs, permitindo que a MF de origem seja desligada. No outro passo, o algoritmo tentará alocar possíveis VMs que foram desligadas no algoritmo de correção de soluções, alocando-as nas MFs que estiverem ligadas. Este procedimento é repetido para todas as soluções.

Nesse exemplo, como existem “espaços” nas MFs do *datacenter*, muito provavelmente será possível desligar MFs, assim como alocar VMs que haviam sido desligadas. Dessa forma, algumas soluções serão melhoradas, tanto no quesito Nível de Dispersão quanto na redução do Consumo de Energia. Caso a população apresente grande concentração de VMs nas MFs ligadas, o aperfeiçoamento local talvez não consiga melhorar as soluções viáveis

existentes (desligando MFs ou alocando VMs desligadas), e este passo não ajudará a beneficiar ou antecipar o resultado do algoritmo memético. É importante destacar aqui que a abordagem VMPOS tenta migrar uma VM levando em consideração seu tipo e o nível de comprometimento da MF de destino, diferentemente da outra abordagem utilizada nos comparativos (a iMaVMP) que migra a VM para qualquer outra MF que tenha recursos suficientes para abrigá-la.

4.5.6 Função de *Fitness*

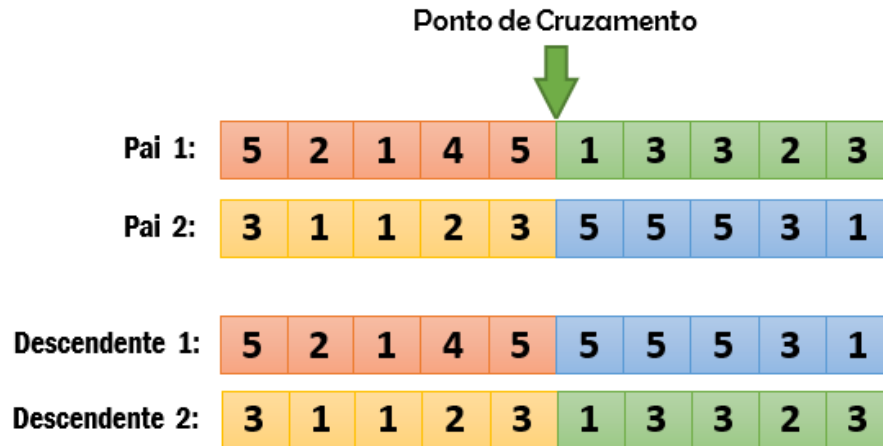
No atual trabalho, foi utilizada a função de *fitness* proposta por [71], na qual é criada uma classificação de soluções não dominadas, em que um valor igual ao nível de dominância de Pareto é atribuído a cada indivíduo da população de tal forma que: 1 representa o nível de maior dominância, 2 o próximo nível, e assim por diante. Logo, quando da comparação de dois indivíduos com diferentes classificações, aquele que tiver menor classificação (maior nível de dominância), será considerado melhor. Se os indivíduos possuírem classificações de não-dominância iguais, estes são considerados como não comparáveis e, portanto, estarão sob o mesmo nível de dominância.

4.5.7 Seleção e Cruzamento

Como explicado anteriormente, o objetivo do cruzamento é criar as novas gerações, mais especificamente, gerar descendentes com as melhores características herdadas de ambos os pais. A forma utilizada para realizar a seleção dos indivíduos para o cruzamento e mutação, na abordagem proposta, foi através de sorteio.

Por simplicidade, o cruzamento é realizado utilizando-se o método de ponto único de cruzamento [65]. É uma forma de recombinação que opera nas cadeias de números dos cromossomos dos dois pais. Cada pai é dividido em duas partes, em um ponto único de corte, e uma dessas partes é recombinada com a parte equivalente do outro pai. A Figura 4.10 ilustra o procedimento. Dessa forma, os dois descendentes gerados passam a carregar partes (genes) de cada um dos pais.

Figura 4.10: Cruzamento em ponto único.



Fonte: Adaptado de COELLO et al., 2007, p. 27.

O ponto de cruzamento aqui utilizado ocorre exatamente no meio da representação do indivíduo, se esse tiver uma quantidade par de genes, deixando a quantidade de genes que fica fixa igual a quantidade que será trocada. No caso da representação do indivíduo ter uma quantidade ímpar de genes, o gene central fica fixo e os demais após essa posição serão trocados. Nas duas situações, a parte permutada é sempre a parte do final dos genes.

4.5.8 Mutação

Apesar de ter uma baixa utilização nas implementações de Algoritmos Genéticos, em torno de 2% [65], a mutação é uma estratégia importante por permitir explorar novas potenciais soluções do problema [68].

Como processo de mutação, a abordagem aqui desenvolvida considera a aleatoriedade para aplicá-lo, ou seja, na criação de cada novo cromossomo, a sorte é lançada para decidir se será executado ou não o processo de mutação em cada filho gerado. Caso ocorra, a mutação será realizada em apenas um gene, sendo esse selecionado por sorteio, assim como o valor que o mesmo irá assumir.

A Figura 4.11 mostra uma possibilidade de mutação em um cromossomo, em que o gene com valor “5” foi alterado para “2”.

Figura 4.11: Mutação em um gene.



Fonte: Adaptado de COELLO et al., 2007, p. 26.

4.6 Considerações Finais

Muitas abordagens de posicionamento de VMs foram encontradas durante a revisão bibliográfica realizada (Capítulo 3). Algumas poucas tentando resolver o problema considerando a utilização de múltiplos objetivos simultaneamente. A atual pesquisa trilha esse caminho, porém, com diferenças significativas, já que sugere o desenvolvimento de um sistema diferenciado para o posicionamento das VMs em *datacenter*, através da utilização de diferentes tipos de VMs e aplicação de sobrecarga de recursos.

Neste capítulo, foi apresentada a abordagem VMPOS desenvolvida para posicionamento de VMs em *datacenter* privado. Descreveu-se a formulação geral do problema considerando múltiplos objetivos no posicionamento de VMs, evoluídos de forma simultânea pura, utilizando algoritmos meméticos como técnica na solução do problema de posicionamento de VMs e empregando a técnica de sobrecarga de recursos com níveis de comprometimento aplicados de forma dinâmica. Foi também descrita a forma de aplicação do algoritmo memético, suas restrições, operadores e os algoritmos mais importantes que foram usados.

A abordagem proposta é flexível o suficiente sendo possível utilizá-la independente do gerenciador de recursos aplicado ao *datacenter*.

Capítulo 5

Resultados e Discussões

Neste capítulo são apresentados e discutidos os detalhes de como foram elaborados os experimentos realizados, com informações dos valores que foram aplicados aos parâmetros da abordagem VMPOS. São apresentados os resultados experimentais quando da comparação da abordagem VMPOS com os outros três métodos de posicionamento de VMs. Aqui também se encontram as comparações dos resultados da VMPOS sem o favorecimento de objetivo contra a VMPOS sendo utilizada aplicando pesos que favorecessem cada um dos objetivos individualmente, além das comparações relacionadas às questões de tempo de execução, em que foram comparadas as abordagens VMPOS e iMaVMP. Com isso, espera-se ser possível reproduzir os experimentos em condições controladas, atendendo aos objetivos almejados neste trabalho.

5.1 Características dos *Datacenters* nas Simulações

As configurações dos *datacenters* utilizadas nos experimentos seguiram três tamanhos básicos, detalhados na Tabela 5.1.

Os *datacenters* P, M e G, fornecidos como entrada na abordagem VMPOS, foram gerados utilizando-se um aplicativo simulador desenvolvido exclusivamente para esse propósito. O simulador gera os dados de VMs e MFs de maneira aleatória dentro de uma faixa de valores pré estabelecida para os recursos aqui utilizados, de forma a realizar uma representação aproximada de um *datacenter*. Mais detalhes sobre o funcionamento do simulador podem

ser encontrados no Apêndice A.2 e na página do autor no GitHub²².

Tabela 5.1: Características dos *datacenters* utilizados nos experimentos.

Classificação do <i>Datacenter</i>	Quantidade de Máquinas Físicas	Quantidade de Máquinas Virtuais
Pequeno (P)	12	60
Médio (M)	60	300
Grande (G)	120	600

Fonte: Elaborada pelo autor.

Para cada tamanho de *datacenter* (P, M e G) foram criadas três entradas, cada um com uma quantidade diferente de VMs críticas. As quantidades utilizadas foram 10%, 25% e 50% do total de VMs críticas existentes no ambiente. Com isso, foram gerados com o simulador nove entradas de *datacenter* ao todo: P_10%, P_25%, P_50%, M_10%, M_25%, M_50%, G_10%, G_25% e G_50%. Essas entradas foram a base para *todos os experimentos realizados* com todas as abordagens utilizadas, ou seja, exatamente o mesmo *datacenter* era aplicado a cada uma das abordagens.

Decidiu-se por utilizar tais valores por entender ser o maior percentual aqui utilizado uma quantidade bastante significativa (50%) de VMs executando serviços críticos em um *datacenter*. Sendo assim, acredita-se que os três percentuais escolhidos refletem a maioria das situações encontradas nas nuvens privadas, podendo ter variações que giram em torno dos valores selecionados.

As características das VMs, com relação aos requisitos de recursos, tomaram por base valores aplicados atualmente em alguns tipos de instâncias utilizadas na nuvem pública da *Amazon Elastic Compute Cloud (EC2)*²³, como por exemplo, os tipos *t2.small*, *t2.medium* e *t2.large*. Segundo informações da Amazon: "As instâncias T2 são uma boa opção para diversas cargas de trabalho de uso geral, incluindo microsserviços; aplicativos interativos de baixa latência; bancos de dados de pequeno e médio portes; desktops virtuais; ambientes de desenvolvimento, compilação e preparação; repositórios de código e protótipos de produtos". Acredita-se que essas características estão bem dimensionadas ao tipo de *datacenter*

²²<https://github.com/petronioci/simulator>

²³<https://aws.amazon.com/pt/ec2/instance-types/>

privado que este trabalho se propõe a atender.

5.2 Componentes da Validação

A abordagem aqui apresentada possui código completamente desenvolvido em linguagem de programação C [95], partindo de uma base já desenvolvida (chamada de iMaVMP) por F. L. Pires e B. Barán [44], que se encontra disponível na página pessoal do Pires²⁴, na plataforma GitHub.

Apesar dessa base preexistente, a abordagem atual tem grandes diferenças para a proposta por F. L. Pires e B. Barán. As principais diferenças da abordagem VMPOS desenvolvida nesta tese para o trabalho referenciado já foram detalhadamente elencadas na Seção 4.5.2.

Para validação dos resultados obtidos com a abordagem desenvolvida, foram utilizadas três outras abordagens de posicionamento de VMs, já comentadas na Seção 2.3. Aqui, alguns detalhes precisam ser acrescentados.

5.2.1 A Abordagem *First Fit*

A abordagem *First Fit* utilizada neste trabalho também foi completamente desenvolvida pelo autor, visto que seria necessário realizar alguns cálculos para obter os valores de alguns objetivos utilizados na VMPOS, a exemplo do nível de dispersão, e assim possibilitar a comparação de resultados. A FF é completamente determinística, visto que: o *datacenter* de entrada da abordagem nos experimentos é o mesmo (uma entrada para cada tamanho de DC) para todas as execuções, e que o *First Fit* não trata com aleatoriedade. Logo, a sequência de alocação é sempre a mesma, resultando sempre no mesmo posicionamento final na distribuição das VMs nas MFs.

É importante ressaltar que o *First Fit* desenvolvido aplica sobrecarga de recursos nas máquinas físicas, quando o nível de comprometimento é configurado para tal.

O código da abordagem *First Fit* desenvolvido nesta tese também está disponível na página pessoal do autor no GitHub²⁵.

²⁴<https://github.com/flopezpires/iMaVMP>

²⁵<https://github.com/petroniocg/FFVMP>

5.2.2 A Abordagem *Random Fit*

A outra abordagem também aqui desenvolvida e utilizada é chamada de *Random Fit* (*RF*). Por ser necessário coletar algumas informações, além do resultado de alocação das VMs, para comparar com a VMPOS, a abordagem *Random Fit* foi adaptada de como é normalmente aplicada, assim como ocorreu como o FF.

O *Random Fit* aqui desenvolvido também aplica sobrecarga de recursos nas máquinas físicas, quando o nível de comprometimento configurado para tal.

O código da abordagem *Random Fit* aqui desenvolvida está disponível na página pessoal do autor no GitHub²⁶.

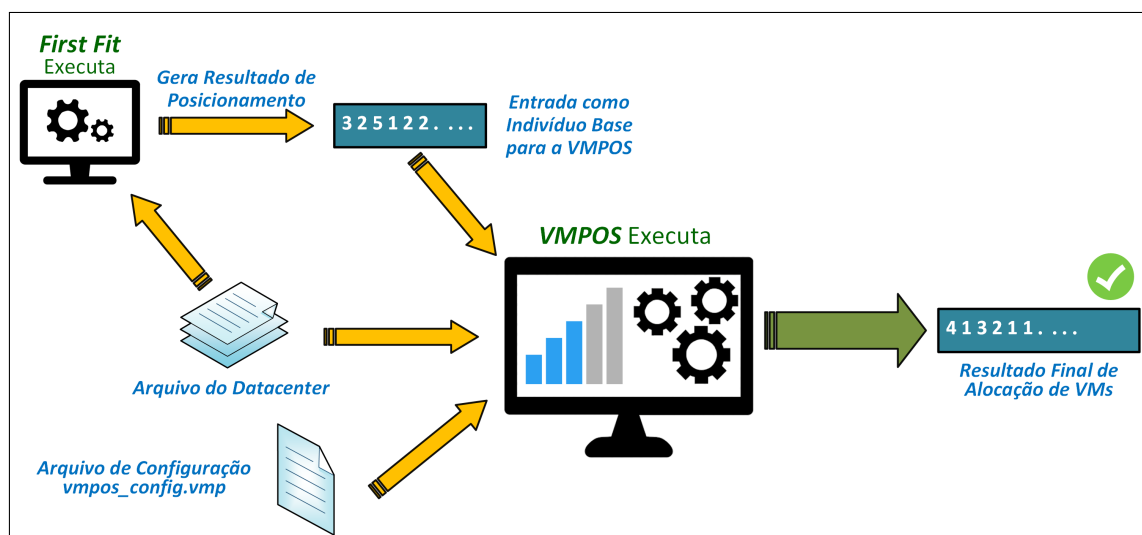
5.2.3 A Abordagem *iMaVMP*

Alguns detalhes da abordagem *iMaVMP* já foram tratados na Seção 2.3 e no Capítulo 3.

5.3 Procedimento para Execução da VMPOS

O esquema da Figura 5.1 ilustra o procedimento básico utilizado na bateria de experimentos realizados.

Figura 5.1: Sequência realizada para execução dos experimentos com a VMPOS.



Fonte: Elaborada pelo autor.

²⁶<https://github.com/petronioci/RFVMP>

Apesar da abordagem VMPOS gerar um mapeamento de posicionamento de VMs em MFs, independentemente de existir ou não um arranjo prévio, aqui chamado de *indivíduo base*, decidiu-se partir de uma situação preexistente de alocação e para isso foi utilizado o resultado da aplicação do *First Fit* no mesmo *datacenter* que seria também submetido à VMPOS.

Assim sendo, de acordo com a Figura 5.1, para cada tamanho de *datacenter*, com percentuais preestabelecidos de VMs críticas, o *First Fit* era executado e o resultado posteriormente utilizado tanto para comparação nas análises finais como fornecido de indivíduo base para a VMPOS, como se fosse uma “fotografia” de uma situação inicial de posicionamento do *datacenter*. Além disso, informações de configurações, como descrito anteriormente, eram fornecidas à abordagem VMPOS com os parâmetros descritos na Seção 4.3 e detalhados no Apêndice A.1.

5.4 Configurações Utilizadas nas Comparações

Para comparar as abordagens foi necessário deixar algumas possibilidades de confrontação de fora. Com relação ao parâmetro Dispersão, esse foi calculado tanto na abordagem *First Fit* quanto na *Random Fit*, mas não foi possível mensurá-lo no iMaVMP por não ser uma das funções objetivo utilizada pela referida abordagem, e por não se desejar alterar o código de sua forma original para calcular esse componente.

Dispersão é um parâmetro que apresenta mudança quando se altera o número de VMs críticas/não críticas pois, essa quantidade influencia no cálculo. E como o *First Fit* quanto o *Random Fit* foram aqui desenvolvidos completamente, foi possível inserir esse cálculo nessas abordagens. Além disso, como não existe o processo de evolução, que acontece nos algoritmos genéticos e que modifica o posicionamento das máquinas virtuais buscando otimizar continuamente o sistemas, o parâmetro Migração não pôde ser calculado nas abordagens FF e RF, por não aplicarem evolução, e nem para o iMaVMP pelo mesmo motivo já descrito em relação à dispersão.

Na Tabela 5.2 observa-se as marcações que indicam quais comparações, entre quais parâmetros e entre quais abordagens, foram possíveis de serem realizadas (✓) e quais não foram possíveis (x).

Na abordagem apresentada neste trabalho é possível ajustar os níveis de sobrecarga de recursos aos quais a *hardware* existente no *datacenter* estará submetido. Quaisquer valores podem ser passados como parâmetro para a abordagem e, portanto, aplicado às máquinas físicas. Isso é feito ajustando-se, nas configurações de entrada, a informação do nível de comprometimento das máquinas físicas com Serviços Críticos (SC) - que será aplicado às MF que estão hospedando pelo menos uma VM com serviços críticos -, e das com Serviços Não Críticos (SNC) - que será aplicado às MF que estão hospedando apenas VMs com serviços não críticos.

Tabela 5.2: Parâmetros comparados entre as abordagens.

Abordagens a Serem Comparadas	Funções Objetivo		
	Energia	Dispersão	Migração
<i>VMPOS x First Fit</i>	✓	✓	x
<i>VMPOS x Random Fit</i>	✓	✓	x
<i>VMPOS x iMaVMP</i>	✓	x	x

Fonte: Elaborada pelo autor.

Sendo assim, como tal facilidade não está presente nas demais abordagens, para ser justo em algumas comparações realizadas, esses níveis de comprometimento foram ajustados para $SC = 1$ e $SNC = 1$ na abordagem VMPOS. Com esses valores não se aplica nenhuma sobrecarga além dos recursos já existentes nas máquinas físicas.

Também foi possível confrontar a abordagem VMPOS aplicando diferentes níveis de comprometimento, mas, para esses casos, apenas as abordagens *First Fit* e a *Random Fit* entraram na comparação, haja vista que o iMaVMP não faz uso de tal método. Os níveis de comprometimento que foram aplicados nos experimentos se encontram na Tabela 5.3.

Como já apresentado, há na abordagem VMPOS a possibilidade de configuração do custo que deve ser aplicado a uma VM no caso de migração, o que vai depender do tipo de VM a ser migrada, se com ou sem serviços críticos. Essa configuração foi deixada fixa para todos os experimentos que foram realizados nesta pesquisa.

Tabela 5.3: Níveis de comprometimento aplicados nas abordagens.

Abordagens a Serem Comparadas	Níveis de Comprometimento		
	SC = 1,0 SNC = 1,0	SC = 1,0 SNC = 1,5	SC = 1,5 SNC = 1,5
<i>VMPOS x First Fit</i>	✓	✓	✓
<i>VMPOS x Random Fit</i>	✓	✓	✓
<i>VMPOS x iMaVMP</i>	✓	x	x

Fonte: Elaborada pelo autor.

Os valores utilizados foram: para VMs com serviços críticos custo $2,0$, e para VMs sem serviços críticos custo $1,0$. Dessa forma, quando do cálculo do objetivo Migração, um maior custo de migração para as VMs com serviços críticos tenderá a mantê-las sem realizarem muitas migrações favorecendo os serviços nelas implantados, já que estas VMs não sofrerão com a etapa de *downtime* que ocorre durante o processo de migração em tempo real de uma VM e pelo fato de se tentar minimizar o objetivo Migração.

No caso do número de indivíduos aplicados para formação da população a ser evoluída, decidiu-se por utilizar 20 (vinte) indivíduos em todos os experimentos. Para se chegar a esse valor, simulações foram feitas com algumas quantidades de indivíduos na população, desde 5 até 50 indivíduos, em busca de um valor que favorecesse a abordagem VMPOS a encontrar o melhor indivíduo o mais cedo possível.

Outro parâmetro importante para a VMPOS é por quantas gerações a população inicial deve ser evoluída. Neste trabalho, esse valor foi a decisão de parada da evolução genética, estando relacionado ao número de indivíduos discutido no parágrafo anterior. Os mesmos experimentos realizados para encontrar o número de indivíduos, também serviram para definir a quantidade de 200 (duzentos) como o número máximo de gerações da evolução genética, aplicado em todos os experimentos com a abordagem VMPOS. Os experimentos e os resultados obtidos, na decisão dessas duas quantidades, podem ser encontrados no Apêndice A.3.

5.5 Procedimentos Estatísticos Adotados

Para responder as questões de pesquisa anteriormente levantadas se faz necessário estabelecer algumas hipóteses, que devem ser investigadas através de testes estatísticos específicos, buscando identificar se existem ou não diferenças significativas entre a abordagem aqui proposta e as demais. Caso tal diferença exista, o objetivo será identificar qual dentre as que estão sendo comparadas se sobressai.

Um teste estatístico comum de ser aplicado para comparações de alternativas é o *T* de *Student*. Contudo, essa verificação estatística exige algumas premissas como, por exemplo, que as amostras a serem avaliadas sejam provenientes de populações com distribuição normal dos dados sendo, portanto, uma abordagem paramétrica.

Todo o tratamento e análises estatísticas apresentados neste trabalho se baseiam no livro de Raj Jain [96]. Os testes e verificações estatísticas, assim como a elaboração de alguns gráficos, foram realizados utilizando o *software R Project*²⁷ para cálculos estatísticos.

As amostras coletadas neste trabalho foram todas testadas no intuito de se verificar se as mesmas seguiam distribuição normal, procedimento realizado através do teste de *Shapiro-Wilk* no qual a hipótese nula atesta que os dados das amostras verificadas provêm de população normalmente distribuída, com hipótese alternativa indicando que não.

A estatística de *Shapiro-Wilk*, chamada de *W*, foi calculada para todas as amostras coletadas. Com *p-valor* menor do que o nível de significância de 0,05 (5%) e com o parâmetro *W* < 1, se rejeita a hipótese nula, ou seja, os dados não são provenientes de uma população normalmente distribuída. Tabelas com todos os resultados, das diversas possíveis combinações de configurações do ambiente, podem ser encontradas no Apêndice C.

Como dito anteriormente, a abordagem *First Fit* é determinística, apresentando um mesmo resultado independentemente de quantas vezes a abordagem seja executada, desde que recebendo o mesmo *datacenter* como entrada. As demais abordagens, apesar de terem sido submetidas aos mesmos *datacenter* de entrada que o FF, tomam a decisão posicionamentos das VMs de forma aleatória (RF) e/ou através de evolução genética (VMPOS e iMaVMP), o que conseqüentemente pode gerar resultados diferentes a cada nova execução. Sendo assim, é possível para um mesmo *datacenter* de entrada obter uma diversidade de

²⁷<https://www.r-project.org/>

posicionamentos como saída. Devido a esse fato, para conseguir comparar os resultados obtidos pelas abordagens, utilizando amostras de mesmo tamanho, fez-se necessário repetir o resultado único de posicionamento obtido pelo FF, compondo uma amostra com 30 valores iguais, para cada *datacenter* submetido à esta abordagem.

Os resultados obtidos dos testes em busca de normalidade, considerando todas as abordagens e todas as amostras coletadas, indicaram que, com os dados de Dispersão, em 32 dos casos não se rejeitou a hipótese nula e em 22 dos casos se rejeitou a hipótese nula. Já com dados de Energia, em 28 dos casos não se rejeitou a hipótese nula e em 35 dos casos se rejeitou a hipótese nula. Ou seja, tem-se evidências estatísticas de que algumas das amostras não provêm de população com distribuição normal e de que, para algumas outras, não se pode refutar a hipótese de as mesmas terem vindo de população com distribuição normal. Como os casos foram bem divididos e se desejava avaliar todas as amostras, optou-se por não depender da premissa de normalidade abrindo-se mão de aplicar abordagem paramétrica.

Desta feita, não se pode utilizar o teste paramétrico T de *Student* para comparação das abordagens. Neste caso, é necessário partir para aplicação de testes não paramétricos. Mesmo que para algumas amostras não tenha sido possível refutar a hipótese de serem provenientes de população normal, para os comparativos estatísticos realizados entre pares de amostras, ambas teriam que ser normais. Assim, é preferível partir para estatísticas sem esta condição prévia de normalidade dos dados.

O teste estatístico de *Kruskal-Wallis* é um método *não paramétrico* utilizado para comparar k amostras independentes. É uma alternativa ao ANOVA *paramétrico*, o qual é utilizado quando se deseja verificar se múltiplas médias amostrais vêm de mesma população. A análise de *Kruskal-Wallis*, ou o teste de classificação (*ranking*) de *van der Waerden*, pode ser empregado se os dados não atenderem às premissas para a *One-Way* ANOVA como, por exemplo, os dados serem provenientes de uma população com distribuição normal [97]. Neste trabalho, o *Kruskal-Wallis* foi aplicado para avaliar as hipóteses levantadas descritas nas seções 5.6.1 e 5.6.2.

A hipótese nula do teste de *Kruskal-Wallis* é que as k amostras foram geradas aleatoriamente a partir da mesma população. A hipótese alternativa é que pelo menos uma das k amostras não vem da mesma população.

Caso a hipótese nula seja rejeitada, deseja-se saber quais pares de amostras são significa-

tivamente diferentes. Para alcançar tal propósito, utilizou-se aqui o *Kruskal-Wallis post-hoc tests after Nemenyi* [98], presente no pacote *The Pairwise Multiple Comparison of Mean Ranks Package* (PMCMR) [97][99] do *software R*, que fornece os resultados dos testes de hipótese comparativos entre todas as amostras utilizadas no *Kruskal-Wallis*. O resultado dessa função permite dizer entre quais pares existe diferenças significativas, entretanto sem apontar qual das abordagens utilizadas é a melhor. O Nemenyi não oferece teste unilateral, portanto, para comparar duas amostras com mais propriedade, finalmente utiliza-se o teste unilateral não paramétrico de Wilcoxon para dizer qual dentre duas abordagens é a melhor.

5.5.1 Amostras Coletadas

Os dados aqui coletados são amostras independentes, obtidas através de trinta repetições (exceto para o *First Fit*), em cada uma de todas as possíveis configurações de percentuais de VMs críticas, tamanhos de *datacenters* e diferentes arranjos dos níveis de comprometimento aplicados às máquinas físicas. As características das amostras que foram coletadas são formadas por todos os possíveis arranjos das opções que podem ser vistas na Tabela 5.4.

Foram realizados 90 (noventa) experimentos, correspondendo a todas as possíveis configurações de cenários, com cada experimento sendo repetido 30 (trinta) vezes para compor as amostras de dados. Assim sendo, um total de 2.700 (dois mil e setecentos) experimentos foram realizados. Na coluna Métrica, da Tabela 5.4, constam as informações coletadas em cada configuração aplicada.

Tabela 5.4: Características das amostras coletadas.

Tamanho do Datacenter	% VMs Críticas	Nível de Comprometimento	Abordagem	Métrica
Pequeno, Médio, Grande	10, 25, 50	SC = 1,0 SNC = 1,0	VMPOS	Energia, Dispersão, Migração
			First Fit	Energia, Dispersão
			Random Fit	Energia, Dispersão
		iMaVMP	Energia	
		SC = 1,0 SNC = 1,5	VMPOS	Energia, Dispersão, Migração
			First Fit	Energia, Dispersão
	Random Fit		Energia, Dispersão	
	SC = 1,5 SNC = 1,5	VMPOS	Energia, Dispersão, Migração	
		First Fit	Energia, Dispersão	
		Random Fit	Energia, Dispersão	

Fonte: Elaborada pelo autor.

5.6 Avaliação da Solução Proposta

De acordo com as questões de pesquisa, definidas na Seção 1.1.2, buscou-se avaliar a qualidade da abordagem desenvolvida através de comparações dos resultados obtidos pela VMPOS com os das outras três abordagens, o *First Fit*, o *Random Fit* e outra abordagem mais atual chamada de iMaVMP. A primeira questão de pesquisa colocada é lembrada aqui:

- *QP1: Até que ponto usar a abordagem VMPOS para o posicionamento de VMs fornece melhores resultados que as outras abordagens avaliadas para alocação de VMs em um Datacenter?*

O tópico de pesquisa representado pela QP1 é abordado nas próximas subseções.

5.6.1 Objetivo Consumo de Energia (VMPOS x FF e RF)

Para responder a QP1, com relação ao consumo de Energia no *datacenter*, e determinar se as diferenças médias dos resultados obtidos com a abordagem VMPOS para os resultados

das abordagens FF e RF eram estatisticamente significantes, testes estatísticos de confronto de alternativas foram executados para avaliar as seguintes hipóteses *H1* levantadas:

H1₀: O consumo médio de Energia no datacenter aplicando o posicionamento de VMs encontrado pela abordagem VMPOS ou por outra abordagem (First Fit, Random Fit ou iMaVMP) são iguais;

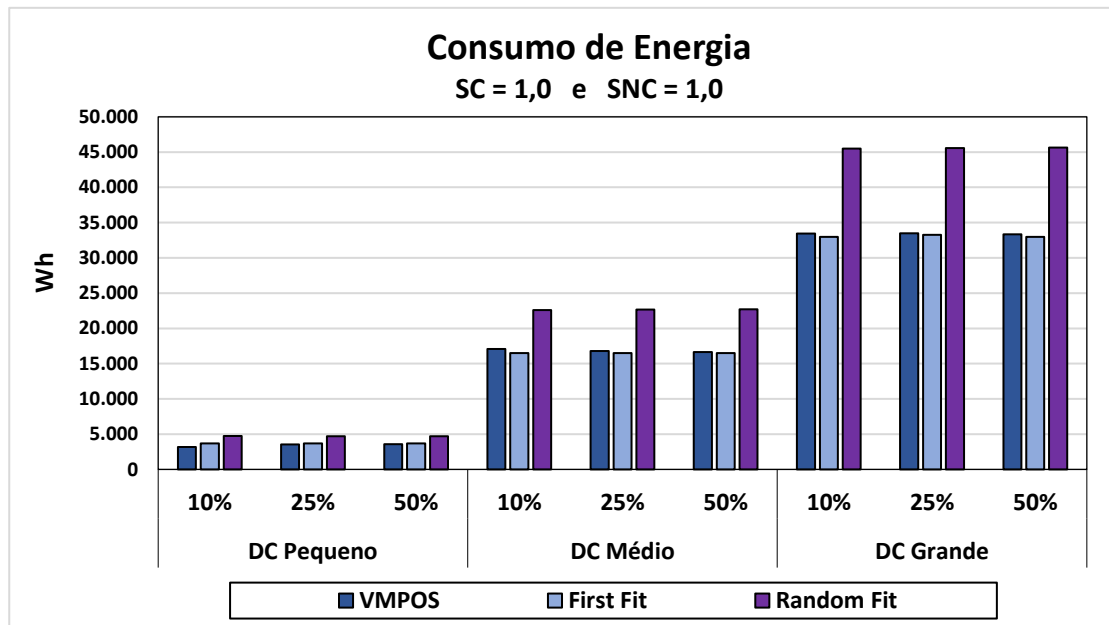
H1₁: O consumo médio de Energia no datacenter aplicando o posicionamento de VMs encontrado pela abordagem VMPOS ou por outra abordagem (First Fit, Random Fit ou iMaVMP) são diferentes.

Um gráfico com os dados médios de consumo de Energia para as três abordagens, com configurações dos níveis de comprometimento para MFs hospedando VMs com Serviços Críticos SC=1 e com Serviços Não Críticos SNC=1, pode ser visto na Figura 5.2. Como se pode observar, e já era esperado, o consumo de energia aumenta com o aumento do tamanho do *datacenter*.

Destacam-se os resultados obtidos pelas abordagens VMPOS e *First Fit*, apresentando menores consumos médios de energia, em relação aos do *Random Fit*, principalmente para os tamanhos de *datacenters* médio e grande. Atribui-se esse fato à metodologia aplicada pela abordagem RF que, ao decidir em qual MF será alocada determinada VM, faz isso aleatoriamente o que acaba deixando as VMs mais espalhadas em uma maior quantidade de MFs, gerando maior consumo de energia.

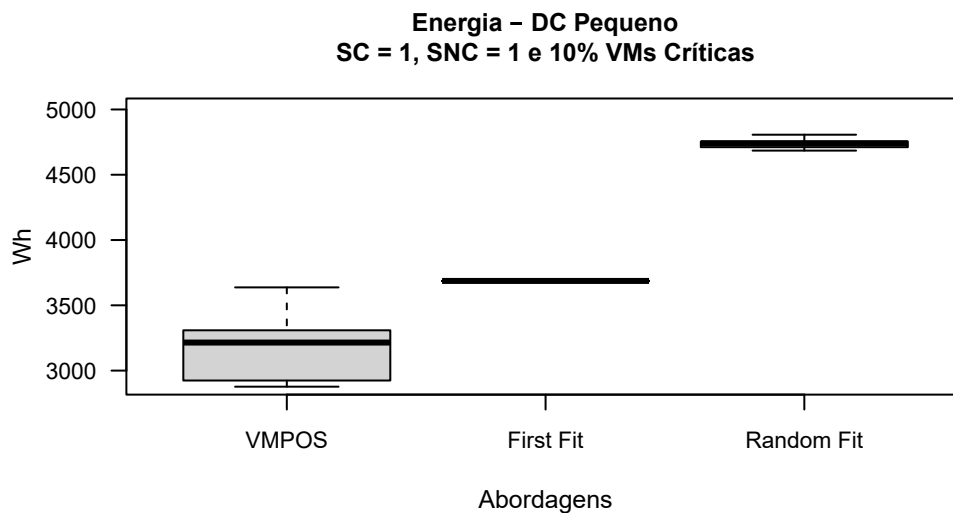
Olhando o gráfico da Figura 5.3, com os *boxplots* dos dados de Energia em *datacenter* Pequeno, com 10% de VMs críticas, e com SC=1 e SNC=1, para as três abordagens, pode-se inferir que os resultados da abordagem VMPOS foram melhores que os das abordagens FF e RF, com as configurações aplicadas. Os demais gráficos de *boxplots*, de todos os experimentos realizados, estão disponíveis no Apêndice H.

Figura 5.2: Médias do consumo de Energia das três abordagens, com $SC=1$ e $SNC=1$, para todos os tamanhos de *Datacenter*.



Fonte: Elaborada pelo autor.

Figura 5.3: *Boxplots* com as amostras de Energia das três abordagens, com 10% de VMs críticas e com $SC=1$ e $SNC=1$.



Fonte: Elaborada pelo autor.

Devido ao *First Fit* ser determinístico, os resultados obtidos de consumo de energia e de dispersão foram repetidos 30 vezes. Assim, no gráfico de *boxplot* apresentado, a informação relativa ao *First Fit* aparece como uma linha reta, visto que o menor valor, o maior e a mediana são todos iguais.

Para verificar se a suspeita apontada pelo resultado gráfico se confirmava, foi então executado o teste não paramétrico de *Kruskal-Wallis*, com 95% de nível de confiança. Logo, um $p\text{-valor} < 0,05$ rejeita a hipótese nula de que as abordagens comparadas têm origem em uma mesma população e, portanto, geram posicionamentos de VMs que provocam consumo de energia semelhantes no *datacenter*.

Com o objetivo de não deixar este documento muito extenso, serão descritos e analisados os testes e gráficos para apenas alguns experimentos realizados. Todos os demais resultados dos testes estatísticos, com todos os arranjos possíveis, poderão ser encontrados nos Apêndices referenciados ao longo deste Capítulo.

O resultado do teste de *Kruskal-Wallis* com os dados de Energia, para o *datacenter* Pequeno, com 10% de VMs críticas e com níveis de comprometimento: SC = 1 e SNC = 1, foi de $\chi^2(3) = 103,71$ e $p\text{-valor} < 2,2e-16$. Com a mesma configuração, apenas mudando-se o percentual de VMs críticas para 25%, obteve-se: $\chi^2(3) = 105,62$ e $p\text{-valor} < 2,2e-16$. E, por fim, com 50% de VMs críticas, obteve-se: $\chi^2(3) = 102,87$ e $p\text{-valor} < 2,2e-16$. Como todos os $p\text{-valores}$ foram menores que o nível de significância adotado de 0,05, rejeita-se a hipótese nula do teste, a qual afirma que todas as amostras provêm de uma mesma população. Todos os demais resultados dos testes de *Kruskal-Wallis*, relacionados a todos os arranjos possíveis, se encontram no Apêndice D.

Desde que os testes estatísticos de *Kruskal-Wallis* apresentaram resultados com forte significância, rejeitando a hipótese nula, faz sentido aplicar os testes de *post-hoc* utilizando a função *posthoc.kruskal.nemenyi.test* para descobrir quais amostras são significativamente diferentes. Os resultados desse teste encontram-se na Tabela 5.5.

A função de Nemenyi permite comparar várias amostras em paralelo, segundo [99], desde que cada uma das amostras tenha no mínimo 6 (seis) observações e existam no mínimo 4 (quatro) amostras, isso no caso de existir a presença de valores repetidos. Essas condições são todas atendidas com as amostras que foram utilizadas nesta tese.

Tabela 5.5: Resultados do *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *data-center* Pequeno, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.

	10%			25%			50%		
	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit
iMaVMP	4.1e-07	-	-	1.3e-08	-	-	3.6e-09	-	-
Random Fit	0.00023	0.63812	-	0.00099	0.16666	-	0.00038	0.17644	-
VMPOS	0.01005	<2e-16	5.0e-13	0.01967	<2e-16	3.8e-11	0.06422	<2e-16	1.6e-10

Fonte: Elaborada pelo autor.

O teste retorna o triângulo inferior da matriz que contém os p-valores das comparações pareadas. Como se pode observar, olhando o comparativo da abordagem VMPOS com as outras abordagens (linha com o nome VMPOS), o p-valor foi menor que 0,05, refutando-se a hipótese nula, e confirmando a existência de diferenças significativas entre essa abordagem e as demais analisadas. O único caso em que isso não ocorreu foi quando existiam 50% de VMs críticas no *datacenter*, e se comparou a abordagem aqui desenvolvida com o *First Fit*. O resultado (0,064) foi maior que o nível de significância estabelecido (0,05), o que leva a aceitar a hipótese nula do teste de que não há diferença estatisticamente significativa entre as abordagens para essa configuração específica do ambiente.

Se olharmos os dados da situação em destaque (DC Pequeno, 50% de VMs críticas, SC=1 e SNC=1), as médias de consumo de Energia foram muito próximas entre essas duas abordagens, ainda assim, com a VMPOS apresentando resultado médio melhor (3.584,55 Wh) do que o do *First Fit* (3.686,67 Wh). Mas, devido à proximidade das medianas, o teste estatístico não conseguiu inferir diferença entre elas. Vale ressaltar aqui que, no caso dos *datacenters* médio (casos com 25% e 50%) e grande (caso com 25%), o teste estatístico de Nemenyi não rejeitou a hipótese de medianas iguais, não conseguindo estabelecer diferença significativa, porém, é possível identificar através do gráfico 5.2 que o *First Fit* obteve médias ligeiramente menores das obtidas pela abordagem VMPOS. As tabelas com estes comparativos para os *datacenters* médio e grande estão contidas no Apêndice E.

Porém, quando se utilizou o teste estatístico unilateral de Wilcoxon, que compara apenas duas amostras para indicar qual tem mediana menor ou maior e poder afirmar qual das abordagens é melhor, só não houve resultado estatístico de redução no consumo de energia favorável à VMPOS justamente nos comparativos com o *First Fit*, e apenas nos casos com

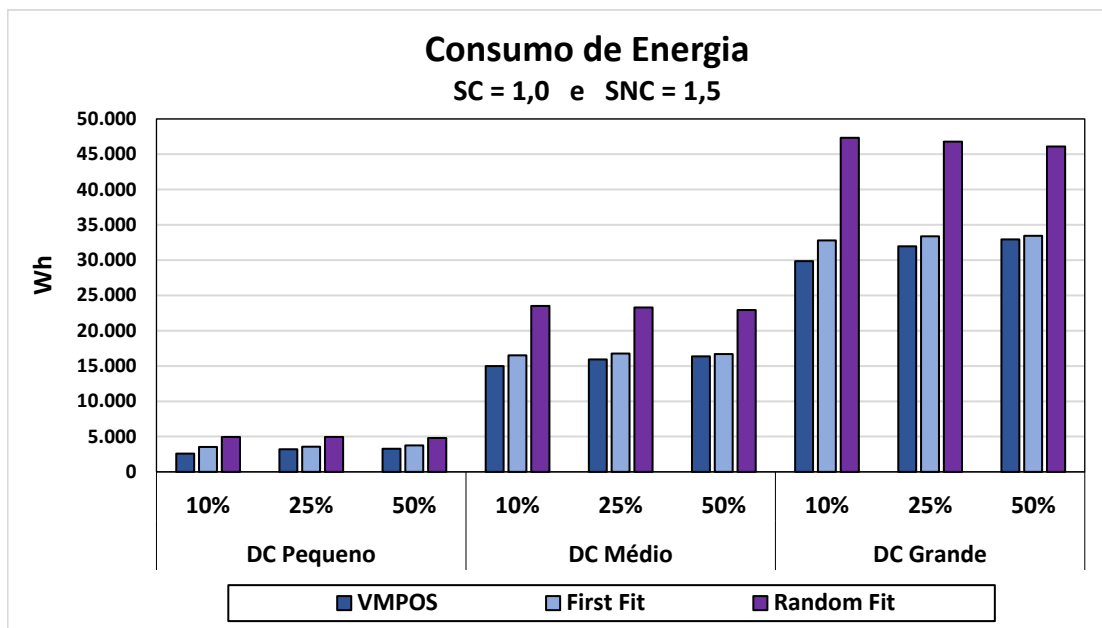
SC = 1 e SNC = 1, para os *datacenters* médio e grande. Até para a situação descrita no parágrafo anterior o teste unilateral conseguiu capturar as medianas menores da VMPOS e atestou a superioridade dessa abordagem. Esses resultados podem ser confirmados na Tabela F.1 do Apêndice F.

Os experimentos analisados em relação aos dados da Figura 5.3 foram realizados utilizando-se níveis de comprometimento: SC = 1 e SNC = 1, ou seja, nenhuma sobrecarga de recursos foi aplicada. Com essa configuração a abordagem VMPOS não se beneficia de uma das suas principais capacidades, que é justamente aplicar um nível diferenciado de comprometimento dos recursos das máquinas físicas. Mas, mesmo assim, devido ao mecanismo de melhora local, ainda foi possível obter melhores resultados na abordagem VMPOS do que nas outras abordagens em diversas configurações do ambiente. Isso pode ser confirmado observando-se os gráficos das Figuras 5.2 e 5.3.

Para se ter uma ideia da melhora alcançada pela abordagem VMPOS, em relação as abordagens *First Fit* e *Random Fit*, no gráfico da Figura 5.4 é possível observar menores médias no consumo de energia na abordagem VMPOS quando se aplicou níveis de comprometimento de SC = 1 e SNC = 1,5. Para todos os tamanhos de *datacenters* e percentuais de VMs críticas. Ressalta-se que a mesma sobrecarga de recursos foi aplicada no FF e no RF.

Comparando os gráficos 5.2 e 5.4, a VMPOS que praticamente apresentou mesmo consumo de energia que o *First Fit* no DC Pequeno e consumo de energia ligeiramente maior nos DCs Médio e Grande quando se aplicou SC=1 e SNC=1 (Figura 5.2), passou a apresentar menores médias em todos os comparativos quando se aplicou maiores níveis de comprometimento (Figura 5.4). Tendo mais recursos disponíveis, através da sobrecarga de recursos, a VMPOS consegue agrupar melhor os dois tipos de VMs utilizados, fazendo com que seja reduzido o consumo de energia no ambiente.

Figura 5.4: Médias do consumo de Energia das três abordagens, com $SC=1$ e $SNC=1,5$, para todos os tamanhos de *Datacenters*.



Fonte: Elaborada pelo autor.

Para concluir a questão do comparativo com as abordagens FF e RF, em relação ao objetivo consumo de energia, e com as configurações do ambiente utilizadas até agora, foram realizados testes estatísticos unilaterais comparando as abordagens duas a duas, sendo uma delas a VMPOS. Para isso, o teste não paramétrico unilateral de Wilcoxon foi utilizado. No procedimento aplicou-se 95% de nível de confiança para verificar se as medianas das amostras do *First Fit* e do *Random Fit* (realizando os testes dois a dois) eram significativamente maiores do que a mediana da abordagem VMPOS.

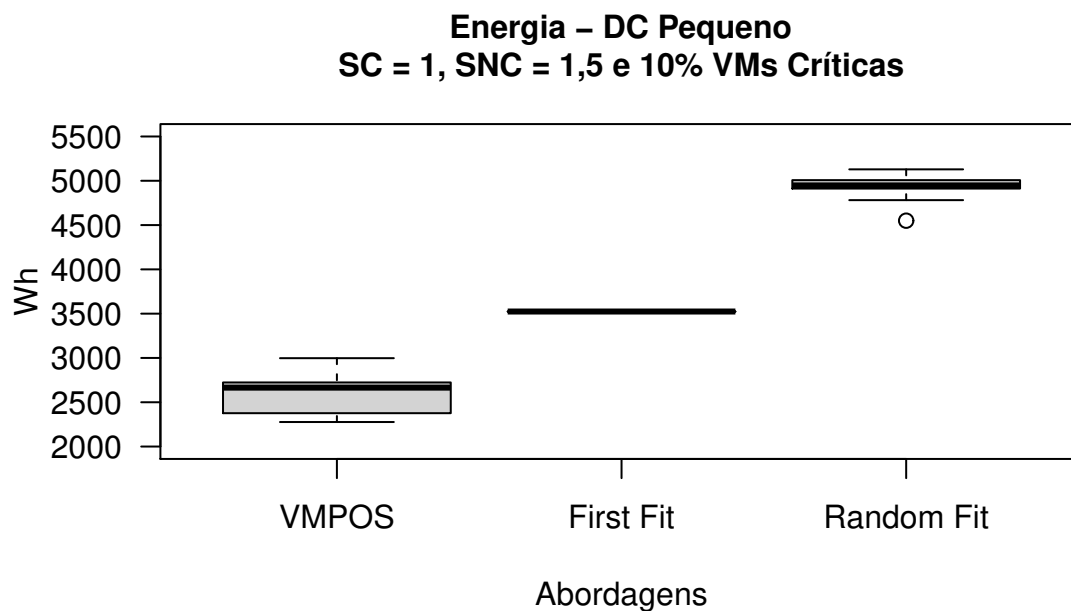
O resultado obtido, contrastando VMPOS e *First Fit* (em *datacenter* Pequeno, com 10% de VMs críticas, $SC=1$ e $SNC=1,5$), estabelecendo como hipótese alternativa de que a mediana da abordagem VMPOS é menor do que a do FF, foi de $W = 900$ e $p\text{-valor} = 6,049e-13$. E na comparação da VMPOS com a abordagem *Random Fit*, $W = 900$ e $p\text{-valor} = 1,503e-11$. Todos os demais testes realizados encontram-se no Apêndice F.

Com resultados de $p\text{-valor} < 0,05$ nos dois testes, aceita-se a hipótese alternativa e *pode-se afirmar com 95% de nível de confiança que a abordagem VMPOS é superior tanto ao First Fit quanto ao Random Fit no objetivo de reduzir o consumo de Energia, para as configurações apresentadas*. Ressalta-se, nesses resultados, que foi aplicado sobrecarga de recursos

apenas para as MFs hospedando VMs não críticas ($SNC=1,5$), em todas as soluções avaliadas.

Tal resultado é enfatizado pelo gráfico da Figura 5.5, com os *boxplots* para o caso de *datacenter* Pequeno, com 10% de VMs críticas, $SC=1$ e $SNC=1,5$. As mesmas conclusões foram obtidas ao se alterar as configurações do ambiente para os demais casos possíveis conforme pode ser confirmado através das demais figuras disponíveis no Apêndice H e nas Tabelas F.1 e F.2 também no Apêndice.

Figura 5.5: *Boxplots* com as amostras de Energia de três abordagens, com 10% de VMs críticas e com $SC=1$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

5.6.2 Objetivo Dispersão (VMPOS x FF e RF)

Continuando na QP1, agora com relação à Dispersão no *datacenter*, procurou-se determinar se a diferença média da abordagem VMPOS para as abordagens FF e RF é estatisticamente significativa, através do confronto de alternativas para avaliar as seguintes hipóteses $H2$ levantadas:

$H2_0$: Os níveis de dispersão das VMs no *datacenter* aplicando o posicionamento en-

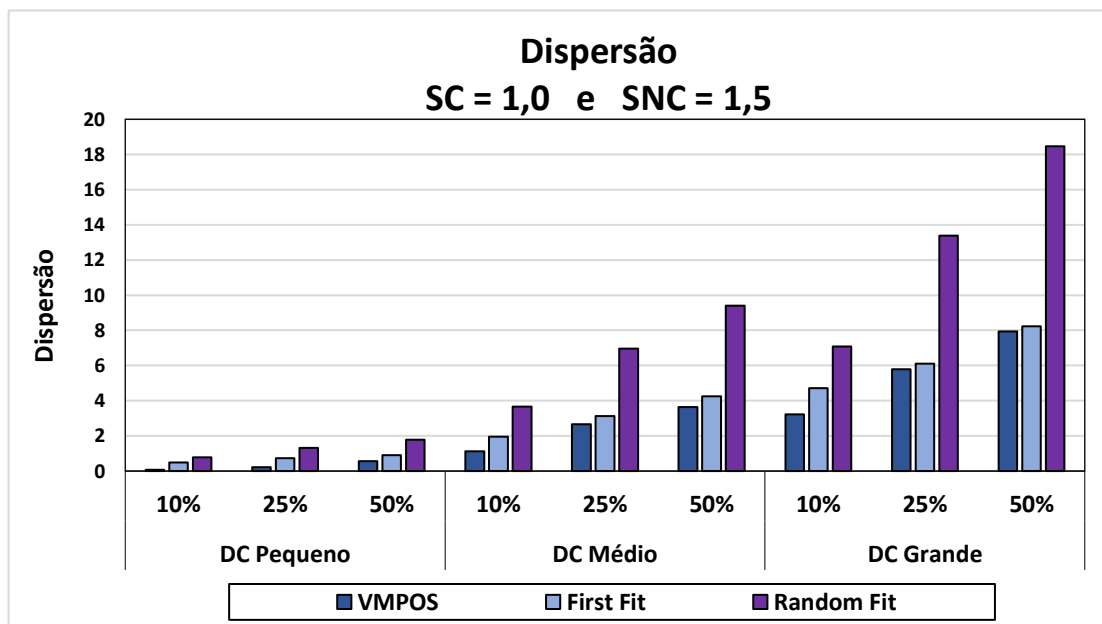
contrado pela abordagem VMPOS ou por outra abordagem (First Fit ou o Random Fit) são iguais;

$H2_1$: Os níveis de dispersão das VMs no datacenter aplicando o posicionamento encontrado pela abordagem VMPOS ou por outra abordagem (First Fit ou o Random Fit) são diferentes.

Nas análises das amostras de Energia foram utilizados os dados do *datacenter* pequeno (P), com $SC = 1 - SNC = 1$ e $SC = 1 - SNC = 1,5$. Com o intuito de variar as análises, e cobrir ambientes com características diferentes, serão utilizados para o caso de Dispersão, as amostras coletadas em *datacenter* médio (M), com 50% de VMs críticas e níveis de dispersão $SC = 1$ e $SNC = 1,5$.

Antes, para se ter uma visão geral das outras configurações aplicadas, foi elaborado o gráfico da Figura 5.6, com os dados médios de Dispersão para todos os tamanhos de *datacenters* e percentuais de VMs críticas utilizados nos experimentos.

Figura 5.6: Médias da Dispersão das VMs, nas três abordagens comparadas, com $SC=1$ e $SNC=1,5$, para todos os tamanhos de *Datacenters* e percentuais de VMs críticas.



Fonte: Elaborada pelo autor.

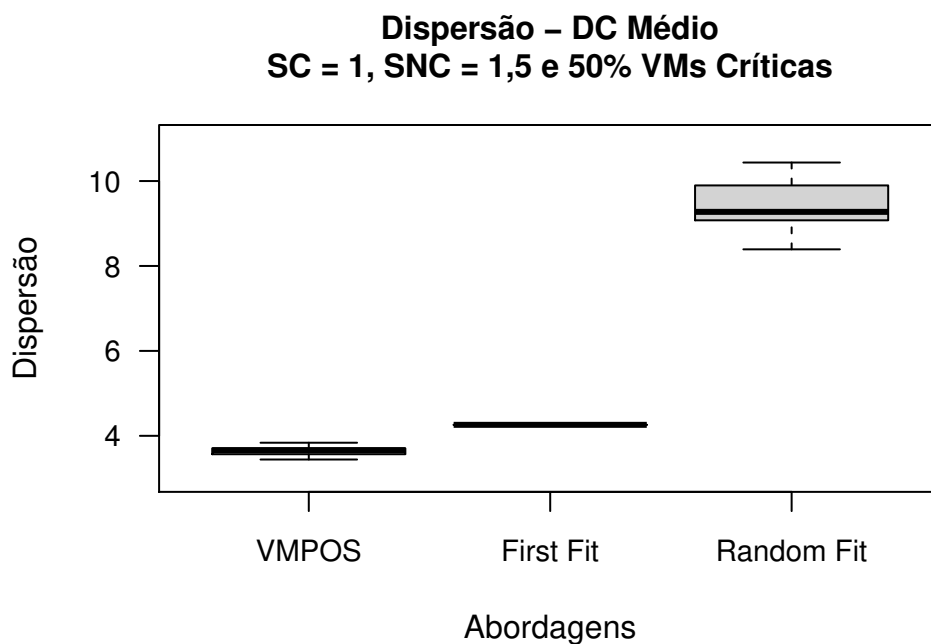
Os níveis de dispersão de VMs da Figura 5.6 indicam que houve maior concentração, dos tipos de VMs utilizados na pesquisa, nas amostras da abordagem VMPOS. Independente dos tipos de VMs utilizados na pesquisa, nas amostras da abordagem VMPOS.

dentemente do tamanho do *datacenter* ou dos percentuais de VMs críticas existentes no ambiente.

Atribui-se esses resultados ao processo aplicado no algoritmo de melhora local desenvolvido, o qual decide alocar uma VM (crítica ou não) dependendo do nível de comprometimento já aplicado à MF que irá receber o novo hóspede. Ou seja, VMs críticas só migram para MFs que possuam nível de comprometimento equivalente ao de serviços críticos (SC), e as VMs não críticas podem migrar para MFs com níveis de comprometimento equivalentes a serviços críticos ou não críticos (SNC), com preferência para esse último.

Passando para a análise específica do nível de dispersão encontrado no ambiente, com as amostras coletadas em *datacenter* médio (M), com 50% de VMs críticas e níveis de dispersão $SC = 1$ e $SNC = 1,5$, foi elaborado o gráfico de *Boxplots* dessa variável que pode ser visto na Figura 5.7.

Figura 5.7: *Boxplots* com as amostras de Dispersão das três abordagens, com 50% de VMs críticas e com $SC=1$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

Assim como no caso para os dados de consumo de Energia, o gráfico de Dispersão indica que a abordagem VMPOS apresenta menores valores de dispersão, ou seja, aparentemente

consegue realizar melhor agrupamento das VMs críticas e das não críticas.

Como nas análises para o caso de Energia, aqui foi repetido o teste não paramétrico de *Kruskal-Wallis*, com 95% de nível de confiança, com os dados de Dispersão, para o *data-center* Médio, com 50% de VMs críticas e com níveis de comprometimento: SC = 1 e SNC = 1,5. O resultado obtido foi de $\chi^2(3) = 105,56$ e p-valor $< 2.2e-16$. Ou seja, rejeita-se a hipótese nula, a qual afirma que todas as amostras provêm de uma mesma população.

A etapa seguinte foi aplicar o teste *post-hoc* utilizando a função anteriormente aplicada de Nemenyi, cujos resultados estão presentes na Tabela 5.6.

Tabela 5.6: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Médio, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas					
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.00998	-	0.01002	-	0.01001	-
VMPOS	9.0e-08	$< 2e-16$	7.8e-07	$< 2e-16$	8.3e-08	$< 2e-16$

Fonte: Elaborada pelo autor.

Como se pode constatar dos resultados da VMPOS em relação as abordagens FF e RF, os p-valores foram todos inferiores ao nível de significância de 5%, inclusive para os percentuais de 10% e 25%, confirmando diferenças estatísticas significantes entre as abordagens. Portanto, para concluir, realizou-se os testes unilaterais de Wilcoxon.

O resultado obtido, contrastando VMPOS e *First Fit*, foi de $W = 900$ e p-valor = $5,252e-13$. E na comparação da VMPOS com a *Random Fit*, $W = 900$ e p-valor = $1,319e-11$.

Com resultados de p-valor $< 0,05$ nos dois testes, *pode-se afirmar com 95% de nível de confiança que a VMPOS é superior tanto ao First Fit quanto ao Random Fit no objetivo de reduzir a Dispersão de VMs no datacenter, para as configurações apresentadas*. As mesmas conclusões foram obtidas ao se alterar, da atual configuração de ambiente, os percentuais de VMs críticas para 10% e 25%. Os resultados para essas outras configurações encontram-se no Apêndice F.

Terminadas estas etapas, é possível responder a primeira questão de pesquisa (QP1) em relação as abordagens FF e RF de posicionamento de VMs. Assim, conclui-se que:

As hipóteses $H1_0$ e $H2_0$ estabelecidas foram refutadas em favor das hipóteses alternativas $H1_1$ e $H2_1$, as quais atestam que a abordagem VMPOS de posicionamento de VMs é superior as abordagens FF e RF, tanto no objetivo de economia de Energia quanto na redução de Dispersão. Assim sendo, respondendo a QP1: **é comprovadamente melhor utilizar a abordagem VMPOS do que as abordagens FF ou RF.**

5.6.3 Objetivo Consumo de Energia (VMPOS x iMaVMP)

A abordagem denominada iMaVMP é a que mais se aproxima da VMPOS por ser uma abordagem que também utiliza algoritmos genéticos e otimização simultânea de múltiplos objetivos em sua metodologia. Tal algoritmo também aplica um mecanismo de melhora local. Por conta de diferenças nas funções objetivos que se deseja otimizar, além de que o iMaVMP não trabalha com aplicação de diferentes níveis de comprometimento, só foi possível comparar as duas abordagens no único objetivo em comum que é o de redução do consumo de Energia no *datacenter*, sem aplicação de sobrecarga de recursos por parte da VMPOS, ou seja, utilizou-se $SC=1$ e $SNC=1$.

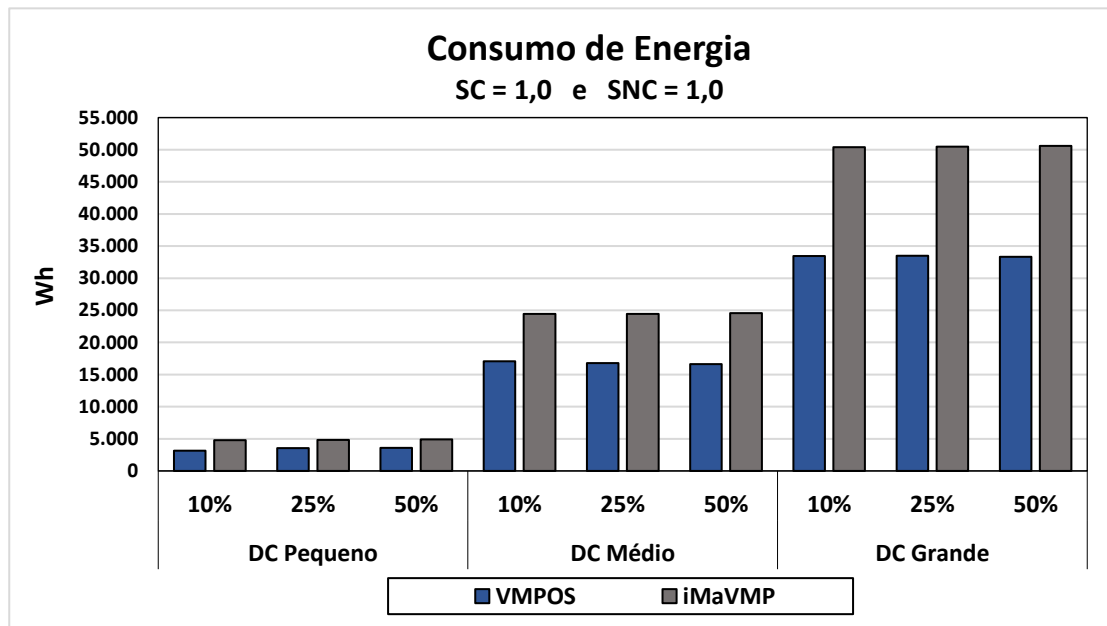
Voltando as questões de pesquisa, definidas na Seção 1.1.2, a segunda questão de pesquisa levantada está relacionada ao iMaVMP, e é relembrada aqui:

- *QP2: Até que ponto usar a abordagem VMPOS ao invés de abordagens que representam o estado da arte, para otimizar a alocação de VMs considerando múltiplos objetivos através de algoritmos genéticos, permite melhorar os resultados em termos das funções objetivos consideradas na pesquisa?*

Para responder a essa questão procedeu-se como anteriormente em relação as abordagens FF e RF. Para confrontar VMPOS e iMaVMP, em relação ao consumo de Energia, foram levantadas as hipóteses $H1_0$ e $H1_1$, já descritas na seção 5.6.1.

A Figura 5.8 ilustra todos os resultados médios comparativos dos dados de Energia entre as abordagens VMPOS e iMaVMP. Como é possível perceber, em todas as possibilidades de configurações das abordagens, a capacidade de redução no consumo de energia conseguido pela abordagem VMPOS foi superior à da abordagem iMaVMP.

Figura 5.8: Médias do consumo de Energia das duas abordagens, com $SC=1,0$ e $SNC=1,0$, para todos os tamanhos de *Datacenter*.



Fonte: Elaborada pelo autor.

Objetivando apresentar os resultados obtidos utilizando uma maior diversidade de configurações no ambiente, utilizar-se-á nas análises desta seção os resultados de consumo de energia no *datacenter* grande, com níveis de comprometimento configurados em $SC = 1$ e $SNC = 1$ e com 50% de VMs críticas.

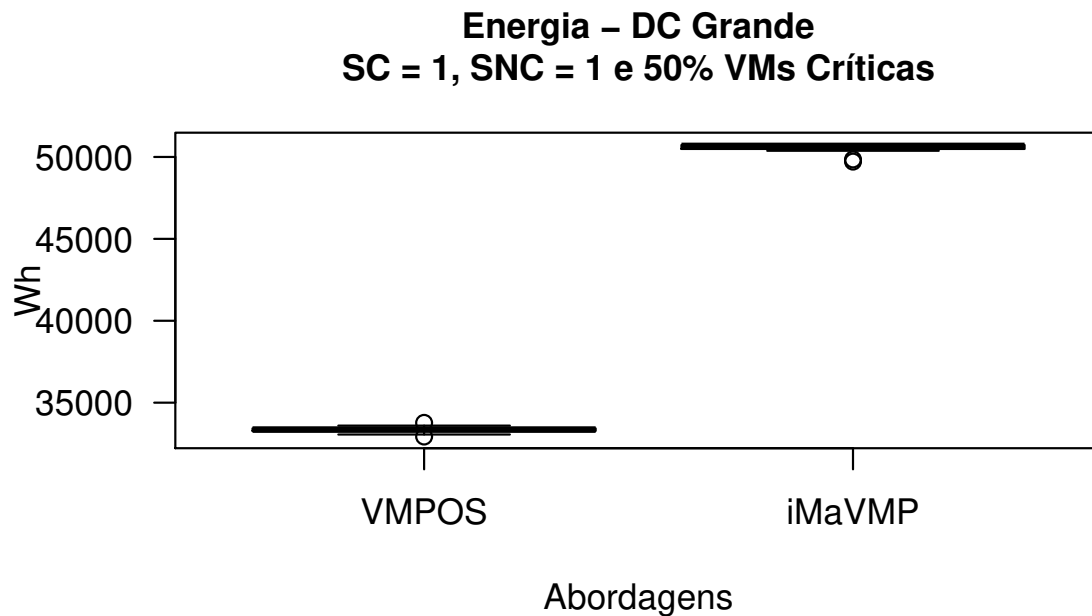
Como não é possível aplicar diferentes níveis de comprometimento, o comparativo acaba sendo diferenciado na capacidade de concentração de VMs críticas e não críticas, ou seja, de fato o que promove algum diferencial é o poder das abordagens na parte de melhora local dos dois algoritmos.

As duas abordagens buscam desligar o máximo possível de MFs, migrando VMs para outras MFs com disponibilidades de recursos. Um diferencial, que ajuda a VMPOS nesse comparativo, é a possibilidade de partir de uma alocação prévia através do indivíduo base, o que é bastante positivo por forçar a abordagem a conseguir alocar as VMs na mesma quantidade de MFs que existem sendo utilizadas no indivíduo base, como também até conseguir diminuir essa quantidade. Com isso, o resultado do posicionamento da abordagem aqui desenvolvida acaba apresentando resultados superiores.

Construindo o gráfico de *boxplots*, que fornece uma boa visão dos dados das amostras,

pode-se verificar que toda a amostra da abordagem VMPOS apresentou resultados menores que os do iMaVMP, conforme Figura 5.9.

Figura 5.9: *Boxplots* com as amostras de Energia das abordagens VMPOS e iMaVMP, com 50% de VMs críticas e com SC=1 e SNC=1.



Fonte: Elaborada pelo autor.

De acordo com o gráfico anterior, observa-se diferença significativa existente entre as duas abordagens. Também é possível inferir grandes diferenças de consumo médio de energia, entre as duas abordagens em comparação, observando-se a Figura 5.8 para os outros tamanhos de ambientes e percentuais de VMs críticas. Entretanto, não se pode confiar unicamente em suposições com base apenas em gráficos. Por isso, as análises estatísticas são imprescindíveis para demarcar o poder das afirmações realizadas.

Os resultados dos testes com a função de Nemenyi, constantes da Tabela 5.5, já confirmavam a existência de diferença significativa quando da comparação da abordagem VMPOS com a iMaVMP, em *datacenter* Pequeno, com 50% de VMs críticas (assim como para 10% e 25%).

Como a comparação utiliza apenas duas amostras (VMPOS e iMaVMP), utilizou-se o teste não paramétrico unilateral de Wilcoxon, após confirmar que existe diferença estatisticamente significativa entre elas, para determinar qual das duas abordagens apresenta resultados de posicionamento de VMs que consomem menos energia.

No resultado, obteve-se $W = 900$ e $p\text{-valor} = 1,51e-11$. Tendo o $p\text{-valor} < 0,05$, *pode-se afirmar com 95% de nível de confiança que a abordagem VMPOS é superior à iMaVMP no objetivo de reduzir o consumo de Energia, para as configurações apresentadas*. As mesmas conclusões foram obtidas ao se alterar, na atual configuração de ambiente, os percentuais de VMs críticas para 10% e 25%. Todos os resultados dos testes estatísticos, com todas as possíveis configurações de ambientes, se encontram na Tabela F.3 do Apêndice F.

Finalizada a análise estatística, é possível então responder a segunda questão de pesquisa (QP2) com relação ao objetivo de redução do consumo de Energia. Assim, conclui-se:

A hipótese $H1_0$ estabelecida foi refutada em favor da hipótese alternativa $H1_1$, atestando que o consumo de energia resultante dos posicionamentos de VMs encontrados pela abordagem VMPOS e pela iMaVMP *são diferentes*. Sendo diferentes, foi utilizado teste unilateral que confirmou a VMPOS superior à abordagem iMaVMP no objetivo de economia de Energia. Assim sendo, respondendo a QP2: **é comprovadamente melhor utilizar a abordagem VMPOS que a iMaVMP para reduzir o custo energético em um *datacenter***.

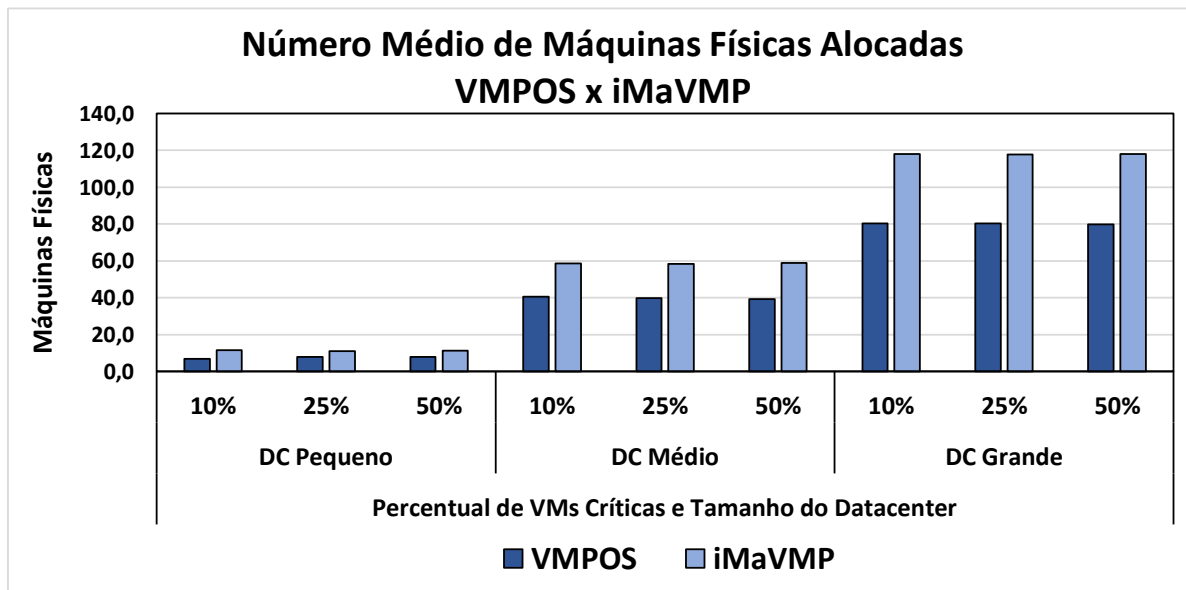
5.6.4 Quantidade de Máquinas Físicas Alocadas

Um parâmetro importante de ser analisado em comparações entre abordagens, como a que aqui foi realizada, é a quantidade de máquinas físicas que estavam ligadas abrigando máquinas virtuais, ao final da evolução genética aplicada por cada abordagem.

No gráfico da Figura 5.10 estão plotados os dados dos números médios de máquinas físicas alocadas nas soluções obtidas, utilizando as abordagens VMPOS e iMaVMP. Como se pode observar, a abordagem VMPOS utiliza sempre um quantidade média menor de máquinas físicas, para alocar todas as VMs do ambiente, para todas as configurações de percentuais de VMs críticas e tamanhos de *datacenters*.

O principal motivo para alocar todas as VMs em menor número de máquinas físicas é que a VMPOS utiliza um indivíduo base como entrada, o que faz com que a abordagem tente encontrar uma solução aplicando uma quantidade de máquinas físicas igual ou menor do que aquela encontrada na representação do indivíduo base. Nesta pesquisa, os indivíduos base eram gerados pela abordagem *First Fit* para cada *datacenter* em que era aplicada a VMPOS.

Figura 5.10: Número médio de máquinas físicas alocadas.

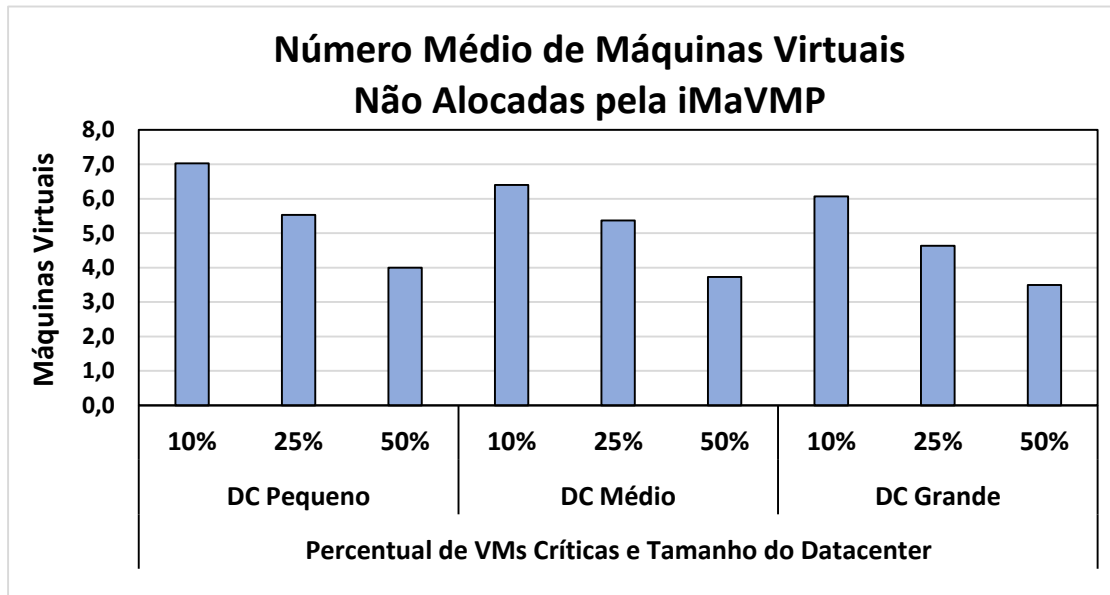


Fonte: Elaborada pelo autor.

Outro ponto positivo da abordagem VMPOS é que *em todos os experimentos*, com as possibilidades de níveis de comprometimento, percentuais de VMs críticas e tamanhos de *datacenters* aqui utilizados, todas as VMs existentes eram alocadas, sem exceção, em alguma máquina física na solução final de posicionamento. Apesar disso ser algo esperado, foi observado nos resultados da abordagem iMaVMP que nem todas as VMs não críticas estavam alocadas em alguma máquina física ao término da evolução ficando, portanto, de fora do posicionamento final obtido pela abordagem. Para a iMaVMP, as VMs que tinham o valor 1 (um) no parâmetro SLA tinham que ser alocadas obrigatoriamente, utilizando isso como restrição. E as que eram marcadas com SLA igual a 0 (zero) poderiam, a princípio, não serem alocadas. Justamente algumas VMs desse último tipo ficavam de fora da solução final de posicionamento.

A quantidade média de máquinas virtuais não alocadas pelo iMaVMP pode se vista na Figura 5.11.

Figura 5.11: Número médio de máquinas virtuais não alocadas pelo iMaVMP.



Fonte: Elaborada pelo autor.

Como se observa na Figura 5.11, há uma redução no número médio de VMs não alocadas com o aumento do percentual de VMs críticas no *datacenter*. Isso se deve justamente ao fato de que, ao se aumentar o percentual de VMs com SLA = 1 (que são as críticas no caso da VMPOS), as mesmas tem que necessariamente estarem alocadas, reduzindo, assim, o número de VMs que podem ficar de fora da solução final de posicionamento.

5.7 VMPOS x VMPOS

Nesta seção são apresentados os resultados obtidos pela abordagem VMPOS, favorecendo-se um dos objetivos de cada vez, sendo comparados com os dados, já apresentados, obtidos pela própria VMPOS quando os pesos foram fixados todos em 1 (um), conforme Seção 4.4.3.

Desejava-se verificar se um dos parâmetros de interesse poderia ser favorecido, de acordo com preferências do administrador do sistema. Isso foi levantado na questão de pesquisa QP3, lembrada aqui:

Questão de Pesquisa 3 (QP3): O favorecimento de uma das funções objetivo consegue melhorar o objetivo em questão e qual a influência nos demais objetivos?

Ficou comprovado que a abordagem VMPOS melhorou os resultados de posicionamento de VMs, com relação ao consumo de energia (contra *First Fit*, *Random Fit* e iMaVMP) e redução na dispersão (contra *First Fit* e *Random Fit*), de acordo com as conclusões das seções 5.6.2 e 5.6.3.

Os resultados obtidos nos experimentos, para responder a questão QP3, estão discutidos nas quatro seções a seguir.

5.7.1 Priorizando Redução no Consumo de Energia

Conforme descrito no Capítulo 3, o parâmetro Energia é um dos mais importantes a serem monitorados por seu impacto nos custos operacionais dos *datacenters*, sendo muito utilizado como indicador chave para avaliar a qualidade de estratégias para posicionamento de VMs.

Com o intuito de selecionar uma solução única, ao final do processo evolutivo através do algoritmo genético, e verificar a capacidade da abordagem VMPOS em conseguir melhorar seus resultados de acordo com preferências do administrador do ambiente, favorecendo um ou outro objetivo, foram realizadas comparações aplicando diferentes valores para os pesos normalizados.

A normalização foi feita dividindo-se cada valor obtido, de cada objetivo, pelo respectivo maior valor encontrado na fronteira 1 de Pareto. Para favorecer um dos itens da função objetivo, o valor 0,6 era atribuído ao parâmetro em questão, e para os outros dois atribuía-se o peso 0,2, de forma que a soma dos três fosse igual a 1 (um). Para nenhum favorecimento, era atribuído o valor 1 aos três pesos considerados.

Os valores de referência serão os resultados da VMPOS quando todos os pesos aplicados receberam o valor 1,0. Ou seja, o indivíduo escolhido como o melhor foi aquele com menor *fitness*, sem interferência de pesos.

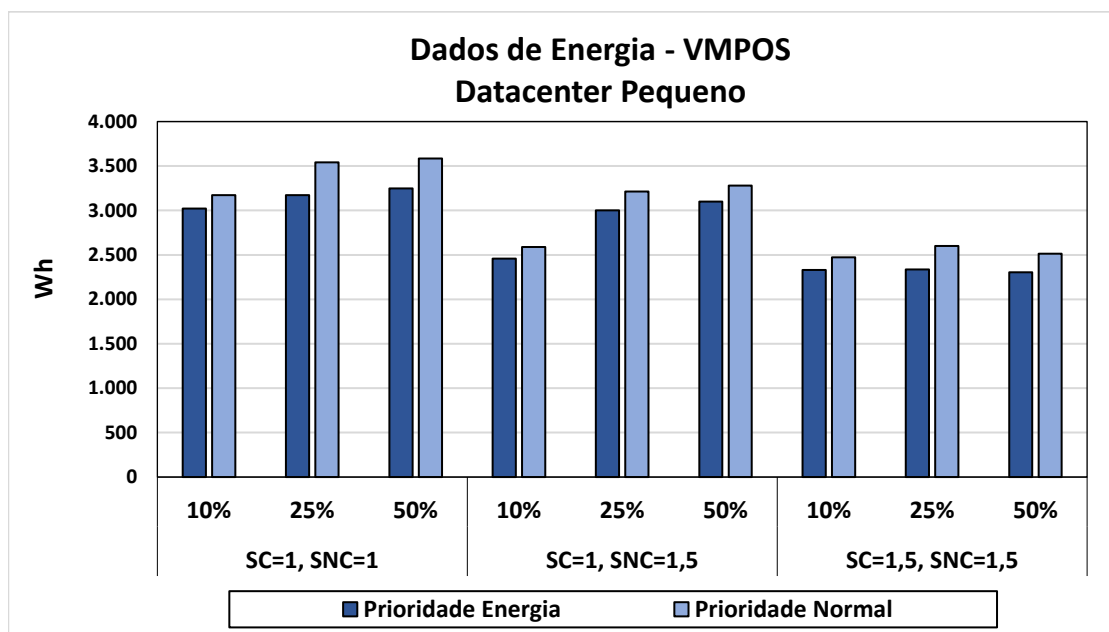
Os resultados obtidos foram avaliados com os testes estatísticos bilaterais para verificar se existiam diferenças significativas entre as amostras e, em caso positivo, utilizou-se os testes unilaterais para informar qual dos dois apresentavam melhores resultados.

Todos os testes bilaterais de Wilcoxon, para o *datacenter* Pequeno e para todas as possibilidades de níveis de comprometimento, deram resultados de p-valor menores que 0,05, indicando diferença estatística significativa entre todas as amostras comparadas.

Todos os outros testes bilaterais de Wilcoxon comparando a abordagem VMPOS com ela mesma, aplicando pesos diferentes contra os valores de referência, para todos os objetivos aqui utilizados, podem ser encontrados em tabelas no Apêndice G.

Um gráfico com os resultados médios das duas abordagens (sem pesos e priorizando redução no consumo de energia) se encontra na Figura 5.12. As amostras dos dados sem aplicação de pesos foram aqui chamadas de Prioridade Normal (PN), servindo como referência, e as amostras com atribuição de pesos que priorizavam os resultados com menor consumo de energia foram chamadas de Prioridade Energia (PE).

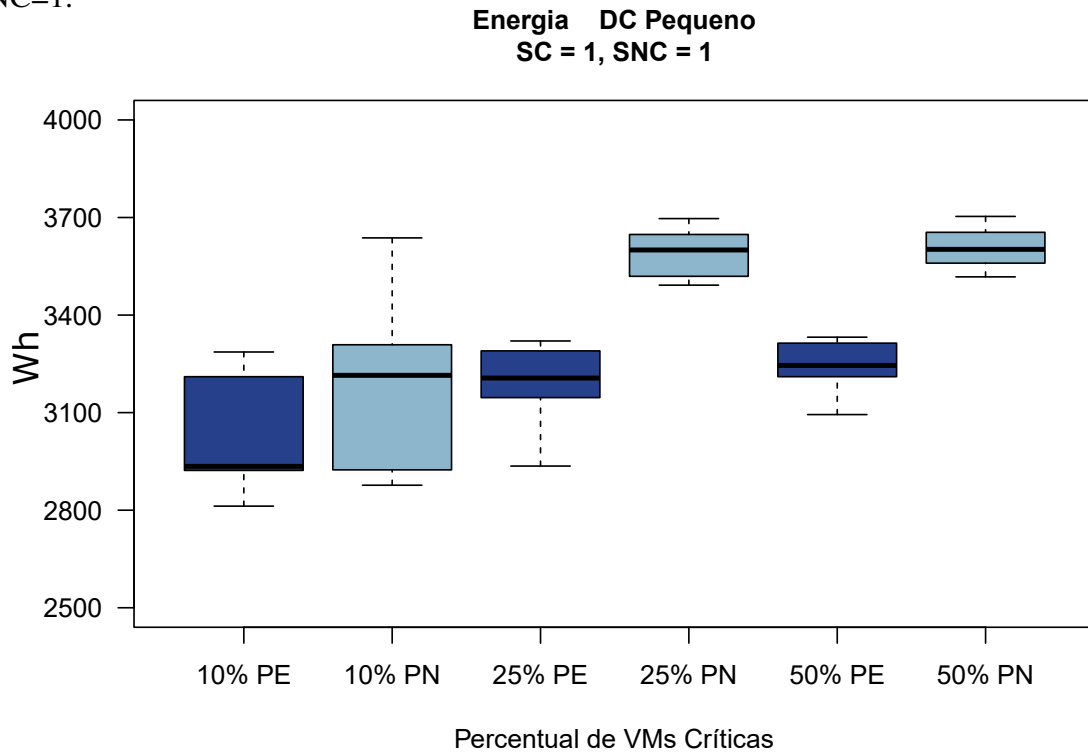
Figura 5.12: Dados médios do consumo de Energia, em *Datacenter* Pequeno, favorecendo o parâmetro Energia e sem aplicação de pesos.



Fonte: Elaborada pelo autor.

Olhando os gráficos de *boxplots*, Figura 5.13, é possível também confirmar as medianas bem inferiores nas amostras dos dados quando se priorizou o menor consumo de energia, aplicando os pesos para esse propósito. É possível notar que essa diferença aumenta com o crescimento no número de VMs críticas. Fato que ficou mais evidente com as amostras do *datacenter* pequeno.

Figura 5.13: *Boxplots* com as amostras de Energia para o *Datacenter* Pequeno, com SC=1 e SNC=1.



Na Tabela 5.7, encontram-se os valores médios e percentuais de redução no consumo de energia para o caso do *datacenter* pequeno.

De acordo com os dados listados na Tabela 5.7, foram registradas reduções médias de mais do que 10%, nos melhores casos, e de 4,7% no pior caso, quando aplicado o peso de priorização de redução do consumo de energia. São valores bastante significativos, principalmente considerando-se o elevado custo do fator energético em um *datacenter* de uma empresa, sendo um parâmetro que naturalmente já se busca minimizar.

Dando continuidade, foram realizados todos os testes unilaterais entre os pares de amostras, buscando confirmação estatística da melhora nos resultados de consumo de energia ao se aplicar os pesos favorecendo tal parâmetro, ratificando assim as suspeitas levantadas pelos gráficos apresentados para o *datacenter* pequeno.

Tabela 5.7: Valores médios e percentuais de reduções no consumo de Energia no *datacenter* Pequeno, aplicando-se peso com prioridade para Energia.

Nível de Comprometimento	Percentual de VMs críticas	Prioridade Energia	Prioridade Normal	Redução Observada
SC=1,0 SNC=1,0	10%	3022,55	3171,76	-4,7%
	25%	3171,40	3541,74	-10,5%
	50%	3249,09	3584,55	-9,4%
SC=1,0 SNC=1,5	10%	2458,10	2589,05	-5,1%
	25%	3001,16	3213,00	-6,6%
	50%	3100,48	3279,55	-5,5%
SC=1,5 SNC=1,5	10%	2331,60	2473,49	-5,7%
	25%	2334,70	2599,55	-10,2%
	50%	2304,45	2514,04	-8,3%

Fonte: Elaborada pelo autor.

Conforme observado na Tabela G.2 do Apêndice G, com comparações unilaterais dos dados de Energia, a hipótese alternativa de medianas menores para as amostras com aplicação de priorização não foi aceita para algumas configurações do ambiente. Nesses casos, não foi possível afirmar estatisticamente que existe diferença significativa entre as amostras. Especificamente para as amostras de consumo de energia, a hipótese nula não pôde ser refutada nos seguintes cenários: DC Médio com SC=1,5 e SNC=1,5, nos casos com 10% e 25% de VMs críticas (dois casos); E no DC Grande com SC=1 e SNC=1, para o caso com 10% de VMs críticas, e com SC=1 e SNC=1,5, para os casos de 10% e 25% de VMs críticas (três casos). Ou seja, de um total de 27 comparações, considerando todos os cenários, em cinco não foi possível afirmar, estatisticamente, que a aplicação de priorização foi melhor.

Apesar de não ser possível afirmar que a VMPOS foi melhor em todas as situações, no caso dos comparativos aplicando priorização do objetivo consumo de energia, ela conseguiu melhorar os resultados na maioria das situações e, de acordo com os dados da Tabela 5.7, foi melhor em todas as configurações para o *datacenter* Pequeno.

Nos cinco casos específicos em que não foi possível indicar melhora com a priorização do objetivo energia, constatou-se que as médias de consumo com prioridade para energia foram levemente inferiores aquelas obtidas quando não se aplicou nenhuma prioridade, com exceção em apenas uma das médias. Isso pode ser confirmado observando-se os dados da

Tabela 5.8 apenas com os casos em questão.

Tabela 5.8: Dados médios registrados em algumas amostras aplicando-se peso com prioridade para Energia em comparação com os dados de referência.

Tamanho do Datacenter	Nível de Comprometimento	Percentual de VMs críticas	Prioridade Energia	Prioridade Normal
Médio	SC=1,5	10%	11832,00	11890,00
	SNC=1,5	25%	11924,00	11955,00
Grande	SC=1,0	10%	33481,00	33457,00
	SNC=1,0			
	SC=1,0	25%	31918,00	31960,00
	SNC=1,5			

Fonte: Elaborada pelo autor.

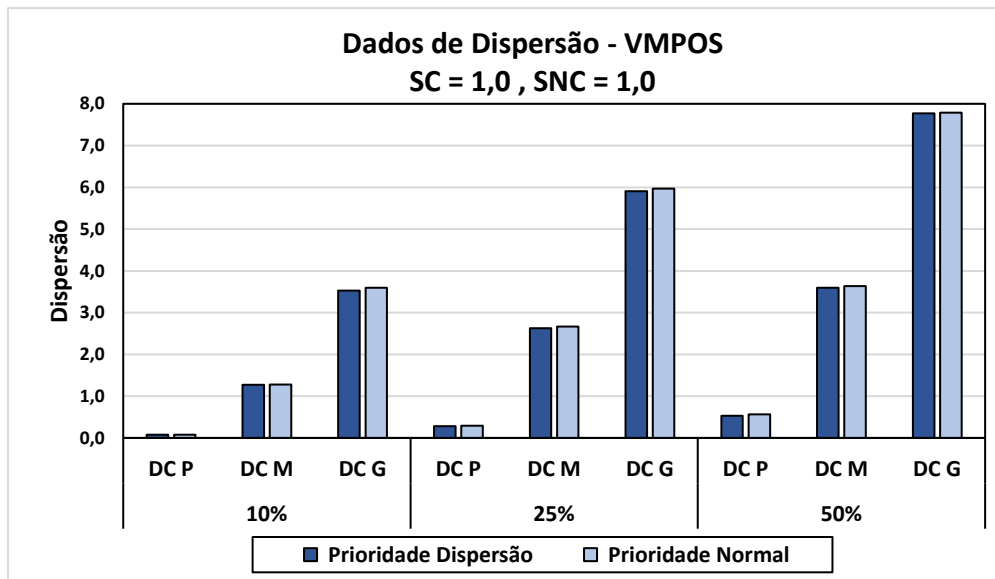
5.7.2 Priorizando Redução na Dispersão

Com relação ao objetivo Dispersão, os testes unilaterais não permitiram afirmar estatisticamente que a aplicação de prioridade para dispersão é inferior aos dados das amostras de referência. Os resultados desses testes estatísticos podem ser vistos na Tabela G.4 do Apêndice G.

Na sequência, são avaliados os resultados médios das amostras com prioridade para dispersão contra as médias das amostras de referência.

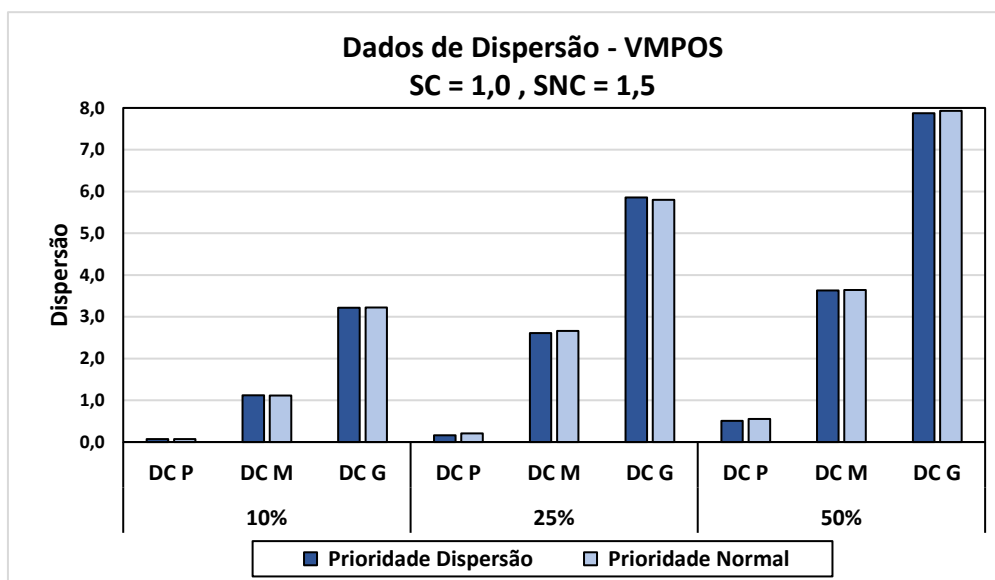
As Figuras 5.14, 5.15 e 5.16 são os gráficos dos dados de Dispersão, fixando os níveis de comprometimento, para todos os tamanhos de *datacenter* e percentuais de VMs críticas.

Figura 5.14: Resultados médios das amostras de Dispersão para os níveis de comprometimento SC=1,0 e SNC=1,0.



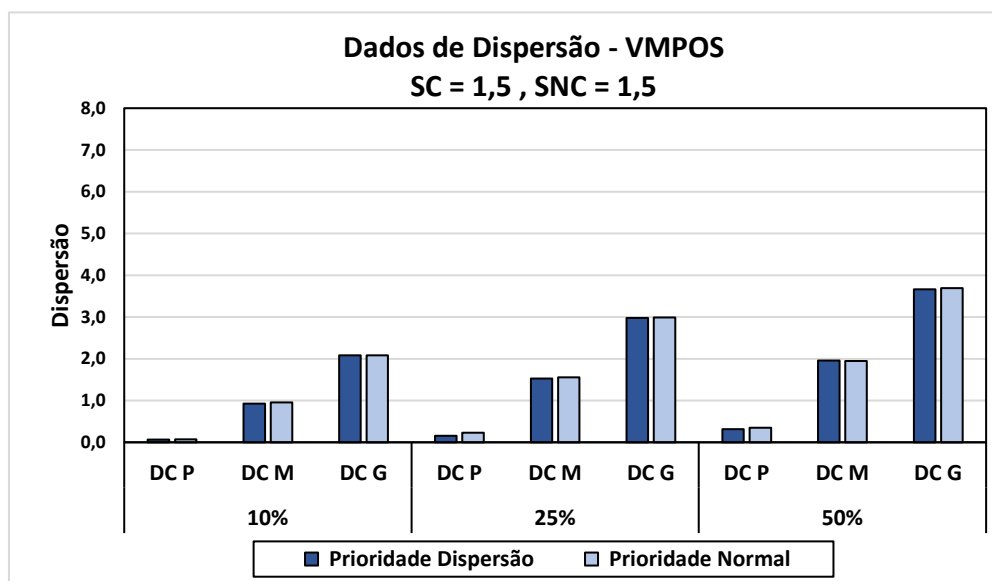
Fonte: Elaborada pelo autor.

Figura 5.15: Resultados médios das amostras de Dispersão para os níveis de comprometimento SC=1,0 e SNC=1,5.



Fonte: Elaborada pelo autor.

Figura 5.16: Resultados médios das amostras de Dispersão para os níveis de comprometimento SC=1,5 e SNC=1,5.



Fonte: Elaborada pelo autor.

De acordo com os resultados médios observados nas figuras, as diferenças registradas foram muito pequenas. Contudo, em algumas configurações percebe-se poucas ocorrências de aumento, embora pequenos, na média da dispersão dos dados com priorização em relação à média de referência. Os dados com destaques para essas situações estão na Tabela 5.9.

Na Tabela 5.9 estão registrados os percentuais de redução (ou aumento) nos valores de dispersão quando se compara as amostras de referência e as com priorização, assim como as médias calculadas. A coluna *Diferença Observada* registra a diferença percentual observada tomando-se a coluna *Prioridade Dispersão* em relação à coluna *Prioridade Normal*.

Para facilitar a consulta, os dados das linhas em destaque representam os casos em que se registrou aumento no valor da dispersão ao se aplicar o peso com prioridade para o objetivo em questão. Os dados das outras linhas indicam que houve redução da dispersão ao se aplicar a priorização.

Como a grandeza dos dados de dispersão é muito pequena, muitas vezes as diferenças são registradas nas casas dos décimos, centésimos, milésimos de decimais, ou ainda além. Variações como a observada de apenas 0,074 gerou a maior diferença percentual (de -32,2%) em favor da aplicação de peso para priorização.

Tabela 5.9: Dados médios registrados nos experimentos aplicando-se peso com prioridade para Dispersão em comparação com os dados de referência.

Tamanho do Datacenter	Nível de Comprometimento	Percentual de VMs críticas	Prioridade Dispersão	Prioridade Normal	Diferença Observada
Pequeno	SC=1 SNC=1	10%	0,0777	0,0772	0,5%
		25%	0,2789	0,2923	-4,6%
		50%	0,5304	0,5620	-5,6%
	SC=1 SNC=1,5	10%	0,0720	0,0710	1,4%
		25%	0,1631	0,2081	-21,6%
		50%	0,5061	0,5502	-8,0%
	SC=1,5 SNC=1,5	10%	0,0678	0,0688	-1,4%
		25%	0,1560	0,2300	-32,2%
		50%	0,3162	0,3475	-9,0%
Médio	SC=1 SNC=1	10%	1,2704	1,2777	-0,6%
		25%	2,6248	2,6636	-1,5%
		50%	3,5938	3,6329	-1,1%
	SC=1 SNC=1,5	10%	1,1183	1,1159	0,2%
		25%	2,6098	2,6579	-1,8%
		50%	3,6308	3,6435	-0,3%
	SC=1,5 SNC=1,5	10%	0,9260	0,9559	-3,1%
		25%	1,5292	1,5553	-1,7%
		50%	1,9599	1,9473	0,6%
Grande	SC=1 SNC=1	10%	3,5274	3,5956	-1,9%
		25%	5,9083	5,9714	-1,1%
		50%	7,7687	7,7888	-0,3%
	SC=1 SNC=1,5	10%	3,2174	3,2227	-0,2%
		25%	5,8545	5,8009	0,9%
		50%	7,8745	7,9300	-0,7%
	SC=1,5 SNC=1,5	10%	2,0821	2,0813	0,0%
		25%	2,9784	2,9884	-0,3%
		50%	3,6635	3,6935	-0,8%

Fonte: Elaborada pelo autor.

Conclui-se que, apesar da dificuldade em se capturar estatisticamente diferenças tão pequenas, a aplicação de priorização da dispersão ainda consegue, na grande maioria das vezes, reduzir um pouco mais as médias em relação aos dados de referência.

Por trás disso está a concentração dos dois tipos de VMs utilizados (críticas e não críticas) e uma redução, mesmo que pequena, pode significar uma ou mais máquinas físicas "desligadas".

É possível perceber que a medida em que se aumenta os níveis de comprometimento, a VMPOS aumenta o poder de concentração das VMs tanto nos resultados das amostras de referência como nos das amostras com priorização, em praticamente todos os casos. Refletindo a expectativa de que quanto mais espaço disponível por MF, maior o poder para concentração.

5.7.3 Priorizando Redução na Quantidade de Migração

Os resultados alcançados, relativos ao objetivo Migração, foram bons nos comparativos das médias obtidas aplicando-se priorização deste parâmetro, em relação aos dados médios de referência. Os resultados estatísticos, dos comparativos unilaterais de Wilcoxon, mostraram que em mais de 96% dos testes (26 de um total de 27) as medianas das amostras em que se aplicou pesos favorecendo a migração, são menores que as medianas das amostras de referência, para todos os ambientes e configurações aplicadas neste trabalho.

Os dados estão disponíveis no Apêndice G, na Tabela G.6. Logo, pode-se afirmar, com 95% de nível de confiança, que priorizar migração reduz os valores do fator de migração, que é um cálculo indicativo do número de migrações realizadas, com pesos diferentes de acordo com o tipo de VM (crítica ou não) sendo migrada.

As menores diferenças percentuais foram registradas nos experimentos com tamanhos de *datacenter* Médio e Grande, e as maiores diferenças ocorreram no Pequeno, conforme dados na Tabela 5.10, considerando as médias amostrais. A menor redução de todos os experimentos foi de apenas 0,2%, enquanto que registrou-se redução no fator de migração de até 15,7%, o que é benéfico ao *datacenter* visto que ao se reduzir o número de migrações, reduz-se a sobrecarga na rede, o consumo de energia, como também diminui-se a sobrecarga que ocorre durante o processo de migração nas aplicações das VMs.

Tabela 5.10: Dados médios registrados nos experimentos aplicando-se peso com prioridade para Migração em comparação com os dados de referência.

Tamanho do Datacenter	Nível de Comprometimento	Percentual de VMs críticas	Prioridade Migração	Prioridade Normal	Diferença Observada
Pequeno	SC=1 SNC=1	10%	42,73	47,37	-9,8%
		25%	48,83	55,30	-11,7%
		50%	56,47	65,10	-13,3%
	SC=1 SNC=1,5	10%	42,00	48,03	-12,6%
		25%	51,00	55,77	-8,5%
		50%	57,47	68,20	-15,7%
	SC=1,5 SNC=1,5	10%	40,80	43,47	-6,1%
		25%	49,23	53,13	-7,3%
		50%	53,73	61,60	-12,8%
Médio	SC=1 SNC=1	10%	314,10	319,57	-1,7%
		25%	352,27	360,03	-2,2%
		50%	426,93	431,33	-1,0%
	SC=1 SNC=1,5	10%	314,80	319,47	-1,5%
		25%	354,60	359,83	-1,5%
		50%	424,53	433,27	-2,0%
	SC=1,5 SNC=1,5	10%	306,87	310,93	-1,3%
		25%	346,87	354,13	-2,1%
		50%	414,57	423,70	-2,2%
Grande	SC=1 SNC=1	10%	644,97	647,30	-0,4%
		25%	733,87	736,97	-0,4%
		50%	876,47	883,37	-0,8%
	SC=1 SNC=1,5	10%	644,97	646,40	-0,2%
		25%	731,97	736,30	-0,6%
		50%	877,27	883,23	-0,7%
	SC=1,5 SNC=1,5	10%	637,47	642,83	-0,8%
		25%	724,40	728,03	-0,5%
		50%	866,00	876,47	-1,2%

Fonte: Elaborada pelo autor.

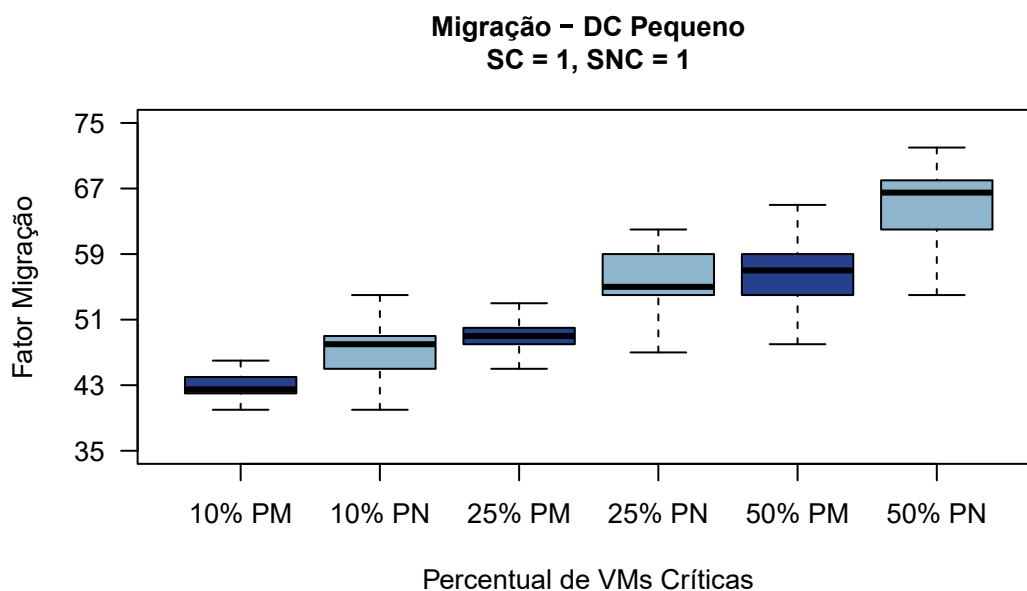
Para uma visão mais ampla dos dados, foi elaborado o gráfico de *boxplots* da Figura 5.17, para os dados do *datacenter* pequeno, que apresentou os melhores resultados. É fácil perceber menores medianas dos dados com aplicação de prioridade para migração. Mesmo

para os diferentes percentuais de VMs críticas no ambiente, a melhora é perceptível.

Como se pode ver, o aumento do número de VMs críticas no *datacenter* aumenta o número de migrações necessárias para se realizar uma mudança de um posicionamento base para outro posicionamento desejado. Isso se deve ao fato de que quanto mais VMs críticas e não críticas não agrupadas, maior será o número de migrações necessárias para agrupá-las. Porém, como consequência, o posicionamento final obtido registrará menor consumo energético e menor dispersão.

Um fato importante observado é que o crescimento do fator de migração, a medida em que se aumenta o número de VMs críticas, é menor para os dados com prioridade. Por exemplo, para o *datacenter* pequeno, o aumento do fator de migração, nos dados com prioridade quando se passa de 10% (média de: 42,73) para 50% (média de: 56,47) de VMs críticas, foi de **32,16%**. Já o aumento nos dados de referência (de: 47,37 a 65,10) foi de **37,43%**, um crescimento de mais que **16%**. Ou seja, além de apresentar médias menores, o crescimento do fator de migração com o aumento do número de VMs críticas também é menor.

Figura 5.17: *Boxplots* com as amostras de Migração para o *Datacenter* Pequeno, com SC=1 e SNC=1.



Fonte: Elaborada pelo autor.

5.7.4 Efeito da Priorização nos Demais Objetivos

Dar prioridade a um objetivo em particular em detrimento dos demais, permitiu melhorar ainda mais os resultados de cada objetivo utilizado, do que quando não se aplicou nenhuma priorização. Contudo, é importante ter conhecimento do que pode ocorrer com os outros dois objetivos ao se favorecer um terceiro.

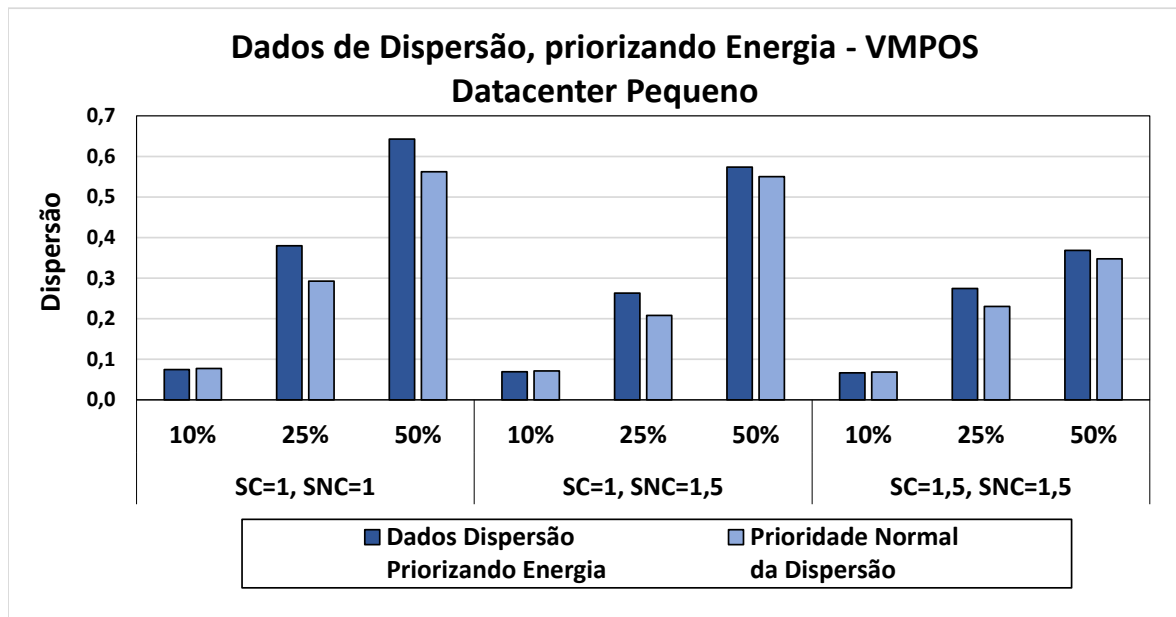
Com esse intuito, todos os resultados dos objetivos que não estavam sendo priorizados, nos experimentos com aplicação de alguma priorização, foram analisados e os resultados estão discutidos na sequência.

Como foram três os objetivos utilizados nesta pesquisa, as análises foram feitas considerando os seguintes eventos: *i*) ao se priorizar redução no consumo de Energia, o que aconteceu com os objetivos relacionados à Dispersão e Migração? *ii*) ao se priorizar redução na Dispersão, o que aconteceu com os objetivos relacionados à Energia e Migração? e, *iii*) ao se priorizar redução no número de Migrações, o que aconteceu com os objetivos relacionados à Energia e Dispersão?

Devido ao grande número de situações, para o texto não ficar extenso, aqui decidiu-se abordar um evento de cada e os resultados dos demais estão disponíveis no Apêndice B.

Para o caso de priorização da redução no consumo de Energia, é possível observar na Figura 5.18 que houve aumento na Dispersão, em *datacenter* Pequeno com SC=1 e SNC=1, a medida em que se incrementa o percentual de VMs críticas no ambiente. Isso se atribui ao fato de que, aumentando-se o número de VMs críticas, cresce também a quantidade de máquinas físicas ocupadas por esse tipo de VM, refletindo diretamente no cálculo do parâmetro Dispersão. Pois, aumentando-se o número de MFs abrigando VMs críticas, aumenta-se o valor da Dispersão. A medida em que se amplia a capacidade de hospedagem das MFs, através de maiores níveis de comprometimento aplicados (SC=1 e SNC=1,5 / SC=1,5 e SNC=1,5), reduz-se mais a diferença em relação aos dados com prioridade normal da Dispersão.

Figura 5.18: Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em *Datacenter Pequeno*.



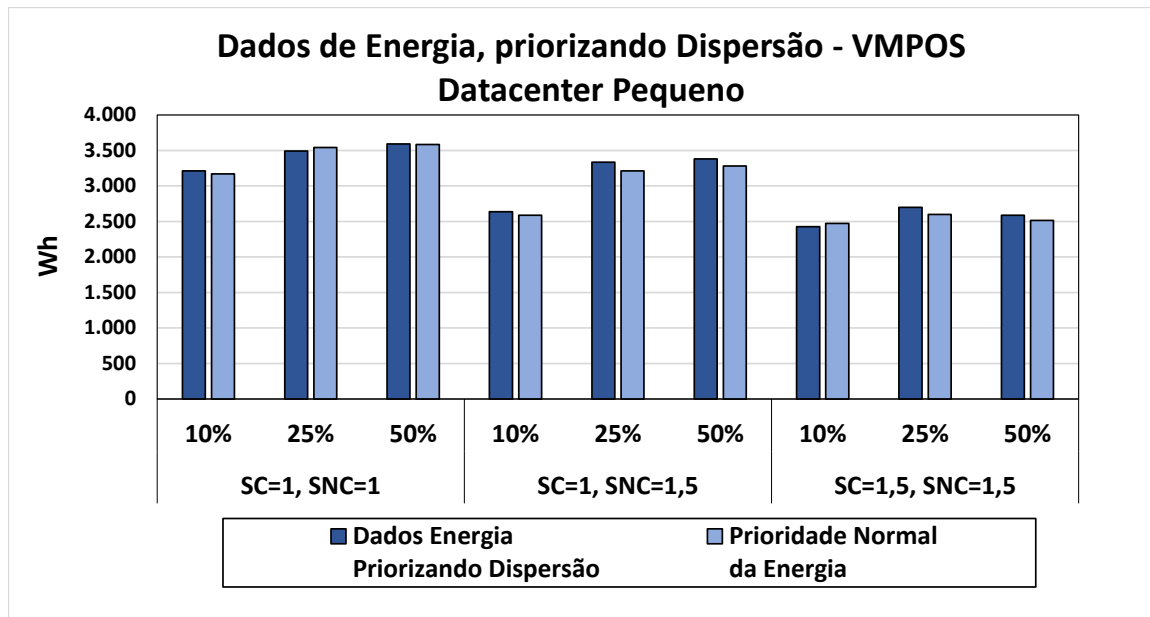
Fonte: Elaborada pelo autor.

Com os dados de Migração, quando se priorizava redução no consumo de energia, registrou-se aumento no fator de migração em relação aos dados com prioridade normal de migração, apenas para o *datacenter* pequeno, para todos os percentuais de VMs críticas e para todos os arranjos de níveis de comprometimento aplicados. O maior aumento médio registrado foi de 11,8% em relação à média dos dados com prioridade normal.

Para os *datacenters* médio e grande, as médias de dispersão e de migração, obtidos quando se priorizava a redução no consumo de energia, superaram os dados médios, com prioridade normal, com diferenças observadas menores. No caso da dispersão, a maior diferença atingiu 7,0% no *datacenter* médio e, 0,5% nos dados de migração, também para um dos cenários do *datacenter* médio.

Nos resultados com priorização do objetivo redução da Dispersão, as diferenças mais significativas foram, mais uma vez, observadas apenas no *datacenter* pequeno, considerando os dados de Energia. Com maior diferença alcançando 3,9% de aumento comparativamente aos dados com prioridade normal. Esse gráfico pode ser observado na Figura 5.19.

Figura 5.19: Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em *Datacenter* Pequeno.

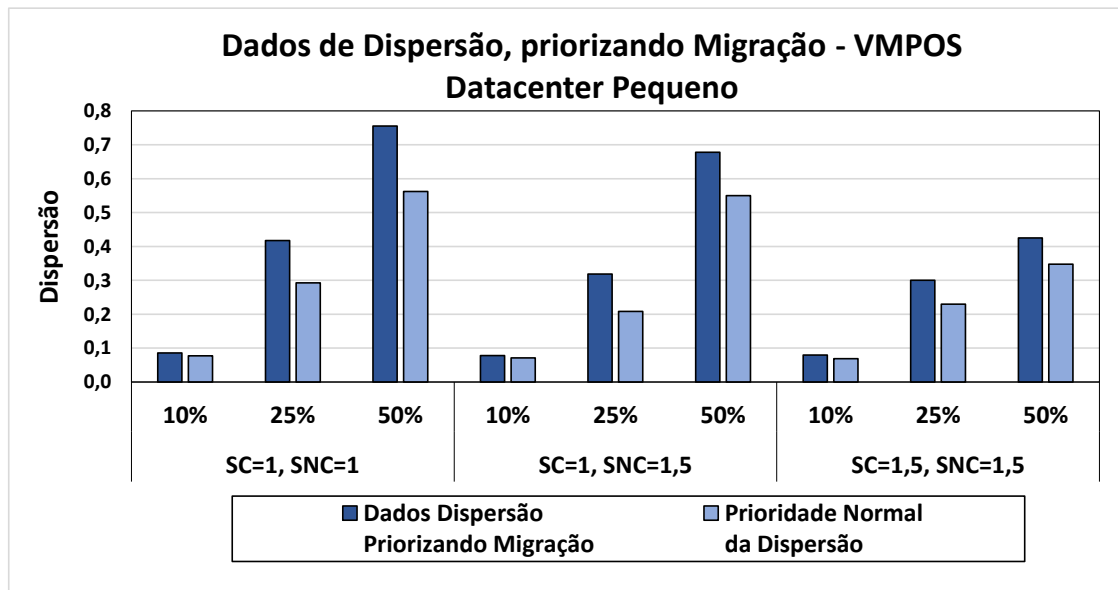


Fonte: Elaborada pelo autor.

Com os dados de Migração, quando se priorizava dispersão, a maior diferença observada foi de 10,7%, em *datacenter* pequeno. Para os *datacenters* médio e grande, as médias de consumo de energia e de migração, obtidos quando se priorizava a dispersão, não conseguiram superar os respectivos dados médios, com prioridade normal, em mais de 0,9%.

Os dados médios dos objetivos energia e dispersão, quando se priorizava o objetivo migração, também apresentaram os maiores aumentos, em relação aos respectivos dados com prioridade normal, em *datacenter* pequeno, e mais notadamente com os dados de dispersão conforme se observa no gráfico da Figura 5.20. Foi registrado aumento de 53,1% quando SC=1, SNC=1,5 e com 25% de VMs críticas.

Figura 5.20: Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em *Datacenter Pequeno*.



Fonte: Elaborada pelo autor.

Para os demais tamanhos de *datacenters* (médio e grande), quando se priorizava a redução na migração, os dados de energia apresentaram, como maior diferença registrada, o valor de 0,6% a mais que os dados médios com prioridade normal. Já com os dados de dispersão, as maiores diferenças foram mais significativas, atingindo diferenças de 6,5% no *datacenter* médio e de 3,9% no grande.

5.8 Análise do Fator Tempo

Uma abordagem responsável por realizar a decisão de posicionamento de VMs em MFs é uma das partes mais importantes no gerenciamento de um *datacenter*. É fundamental que essa etapa seja realizada em tempo hábil conseguindo contribuir, caso precise serem feitas alterações de posições de VMs, para a reordenação das máquinas virtuais no ambiente o quanto antes, conseqüentemente favorecendo para que os objetivos traçados sejam atingidos.

Com o intuito de dimensionar o impacto no tempo transcorrido durante o processo de decisão de posicionamento de VMs, pelas abordagens utilizadas neste trabalho, os tempos médios das abordagens foram analisados.

Nos casos do *First Fit* e *Random Fit*, por não utilizarem processo de evolução de população, essas duas abordagens foram favorecidas na questão tempo pela forma simples de realização do processo de posicionamento das VMs. Ter o benefício de ser rápido, mas não atender bem aos objetivos almejados, como observado nos resultados analisados principalmente para o RF, faz com que não seja boa escolha uma opção como essas. Considerou-se também que por não implementarem otimização de objetivos, não seria justa a comparação dessas abordagens com as que aplicam tal estratégia. Os tempos médios de execução, das abordagens FF e RF, para todos os tamanhos e configurações de *datacenter* utilizados nesta pesquisa, foram inferiores a 1 (um) segundo.

Uma estratégia desenvolvida e utilizada na VMPOS é a possibilidade de poder partir de um indivíduo base pré-existente. Tentar resolver o problema de alocação utilizando a mesma quantidade de MFs existente em um indivíduo base vai refletir, principalmente, em menor tempo de execução do processo de corrigir os indivíduos não viáveis, devido ao menor número de MFs a serem consideradas nas correções, e no processo de melhora local quando tenta-se reduzir o número de MFs utilizadas na solução. Por utilizar um método evolucionário para otimização de objetivos, a VMPOS é uma abordagem que demanda mais tempo em comparação com FF e RF. Os tempos médios obtidos pelas abordagens VMPOS e iMaVMP estão na Tabela 5.11.

Como descrito anteriormente, a abordagem iMaVMP não aplica sobrecarga de recursos, portanto, os tempos da abordagem VMPOS constantes na Tabela 5.11, também foram obtidos sem aplicação de sobrecarga de recursos (com $SC=1,0$ e $SNC=1,0$).

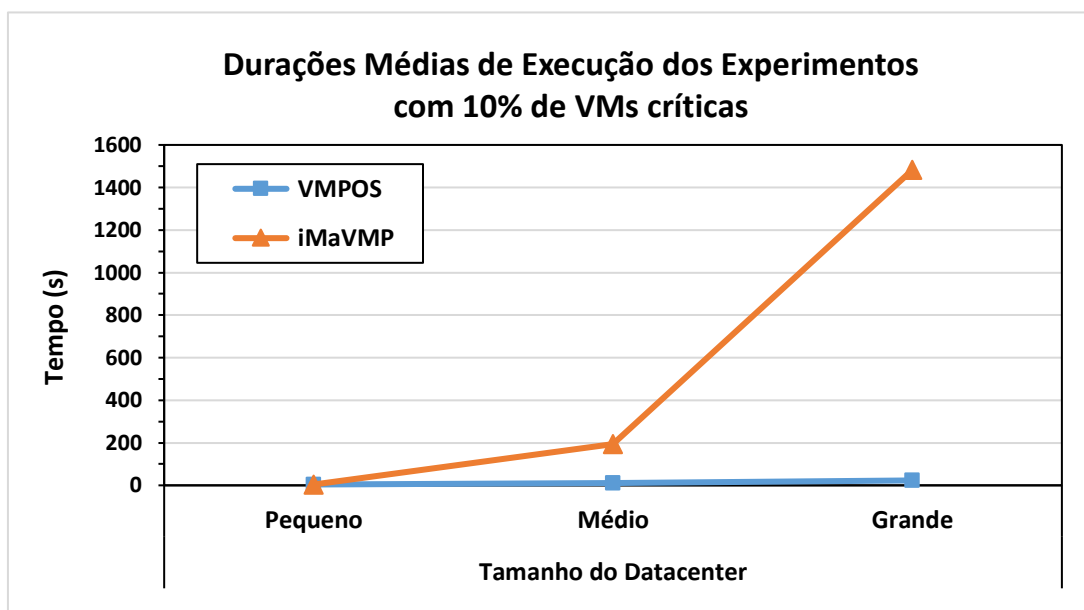
Tabela 5.11: Tempos médios, em segundos, registrados nos experimentos com as abordagens VMPOS e iMaVMP, com $SC=1,0$ e $SNC=1,0$.

Abordagens	Porcentagem de VMs críticas	Tamanho do <i>Datacenter</i>		
		Pequeno	Médio	Grande
VMPOS	10 %	4 s	12 s	23 s
iMaVMP		4 s	194 s	1.482 s
VMPOS	25 %	4 s	11 s	22 s
iMaVMP		5 s	193 s	1.521 s
VMPOS	50 %	4 s	10 s	20 s
iMaVMP		4 s	195 s	1.482 s

Fonte: Elaborada pelo autor.

O gráfico da Figura 5.21 ilustra o que acontece com os tempos médios de execução das abordagens VMPOS e iMaVMP, para diferentes tamanhos de *datacenter*. Nesse gráfico, foram utilizados os tempos de execução apenas para o *datacenter* Pequeno, embora o comportamento para os outros tamanhos de ambiente seja essencialmente o mesmo, haja vista os dados da Tabela 5.11.

Figura 5.21: Durações médias de execução dos experimentos, com $SC=1,0$ e $SNC=1,0$, e com 10% de VMs críticas.



Fonte: Elaborada pelo autor.

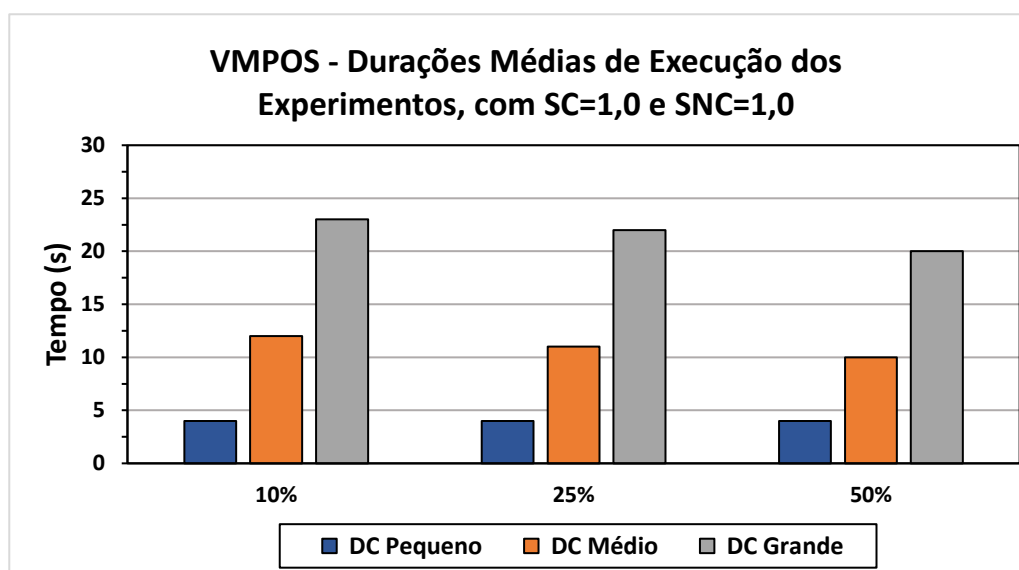
Os tempos médios foram bem inferiores na abordagem VMPOS, mesmo se registrando elevação na duração da execução a cada aumento de tamanho de *datacenter*.

No caso da abordagem iMaVMP, o aumento no tempo de execução se eleva acentuadamente. Se o tempo observado na execução com o *datacenter* Grande na VMPOS é em torno de 5 vezes o tempo médio registrado para o Pequeno, na iMaVMP esse tempo salta para mais de 285 vezes (considerando o tempo para o *datacenter* Pequeno igual a 5s e para o Grande igual a 1.482s). O que indica que a abordagem iMaVMP enfrenta muitas dificuldades, em termo de tempo de execução, com o aumento do tamanho do *datacenter* aplicado.

Ajustando a escala vertical do gráfico (utilizando 30 segundos), é possível perceber o aumento no tempo de execução na VMPOS com o aumento do tamanho do *datacenter*, girando em torno de pouco mais que o dobro do tempo registrado no tamanho de ambiente anterior, como se pode verificar na Figura 5.22. Contudo, percebe-se que esse aumento acontece de forma suavizada na abordagem VMPOS.

Os tempos de execução com a abordagem VMPOS, para todos as configurações de *datacenter*, foram considerados aceitáveis, haja vista os resultados dos experimentos realizados pelo autor, disponíveis em artigo publicado em 2019 [100], nos quais foram registradas durações de migrações de VMs em tempo real em torno de 200 segundos.

Figura 5.22: Durações médias de execução dos experimentos no VMPOS, com SC=1,0 e SNC=1,0.



Fonte: Elaborada pelo autor.

Nos experimentos de Petrônio Bezerra et al. [100] foram utilizadas VMs com serviços considerados críticos para empresas, como servidores *web* e gerenciadores de bancos de dados não relacionais, em máquinas reais, e com os serviços tendo altas demandas de acesso durante o processo de migração.

Claro que a duração da migração de uma máquina virtual, em tempo real, vai depender de vários fatores como: características da VM, largura de banda da rede, serviços em execução, etc. De acordo com [101], os tempos de migração conseguidos com o *hypervisor* KVM²⁸ ficaram na faixa de 7 a 400 segundos, e como o *hypervisor* Xen²⁹ os resultados foram de 6 a 363 segundos.

Diante dos resultados, acredita-se que realizar o processo de decisão de posicionamento das VMs de todo um *datacenter*, com duração em torno de 23 segundos para o maior dos ambientes avaliados, seja aceitável, considerando os benefícios que serão obtidos ao se atender os objetivos traçados.

No caso da *iMaVMP*, por essa abordagem realizar a geração da população inicial sempre utilizando a totalidade de máquinas físicas informadas no *datacenter* de entrada, naturalmente terá maior quantidade de MFs para manejar do que se observa na técnica empregada na abordagem VMPOS. Com isso, acaba necessitando de maior processamento para tentar desligar as MFs existentes, empenhando-se em alocar as VMs na menor quantidade de MFs possível, piorando o tempo total de resolução do problema de posicionamento. Além disso, em comparação com FF e RF, a *iMaVMP* é uma abordagem que naturalmente demanda mais tempo por utilizar um método evolucionário, o que exige um processo repetitivo em busca das melhores soluções para o problema.

Na abordagem *VMPOS* foram registrados tempos médios de execução muito inferiores aos obtidos pela abordagem *iMaVMP*. Atribui-se esses resultados principalmente à técnica de utilizar um indivíduo base como entrada. Como essa estratégia permite que seja fornecido um posicionamento prévio das VMs nas MFs, a *VMPOS* tenta alocar suas soluções em quantidade de MFs igual ou inferior àquela encontrada no indivíduo base que, nos experimentos realizados, foi gerado pela aplicação da abordagem *First Fit*. Isso significa que a *VMPOS* é “forçada” a conseguir alocar todas as VMs em quantidade possivelmente menor de MFs que

²⁸<https://www.linux-kvm.org/>

²⁹<https://xenproject.org/>

a quantidade informada no *datacenter* de entrada, já favorecendo a solução final.

5.9 Considerações Finais

Neste capítulo, descreveu-se os procedimentos estatísticos aplicados aos resultados dos experimentos realizados para coleta de dados. Foram apresentadas as características das amostras, indicando o que foi possível coletar em cada uma das abordagens trabalhadas nesta tese, assim como que comparações foram possíveis de serem realizadas entre a VMPOS e cada uma delas.

Durante a elaboração dos objetivos de pesquisa (ver Seção 1.1.2) três questões foram produzidas e transformadas em hipóteses a serem respondidas através dos dados coletados segundo a aplicação de testes estatísticos. Foi possível constatar dos resultados que a abordagem VMPOS se mostrou muito eficiente em melhorar o posicionamento de VMs nas MFs do *datacenter*, atendendo aos diversos objetivos, como também se sobressaindo no atendimento de cada objetivo apresentado em relação as demais abordagens.

Na técnica aplicada para se favorecer um dos objetivos em detrimento dos demais, observou-se que a VMPOS conseguiu melhorar ainda mais os objetivos considerados, comparando-se com a própria abordagem sem aplicação de favorecimento, mais notadamente para os objetivos de redução no consumo de energia e na diminuição no número de migrações.

Na análise dos tempos de execução das abordagens que utilizam método evolucionário na otimização dos objetivos, foi possível confirmar que a abordagem VMPOS realiza a tarefa de decidir o posicionamento das VMs nas MFs, atendendo aos objetivos considerados, em tempo muito inferior aos da abordagem iMaVMP (considerando os tamanhos de *datacenter* Médio e Grande).

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho foi apresentada uma nova abordagem para o posicionamento de VMs nas MFs de um *datacenter* privado, denotada por VMPOS. A abordagem utiliza várias técnicas para que, partindo da configuração inicial, seja possível se obter o melhor posicionamento das VMs nas MFs, buscando atender três objetivos simultaneamente através de algoritmo memético.

Baseado em conceitos da Computação Evolucionária, desenvolveu-se um algoritmo que evolui uma população de indivíduos (posicionamentos), chegando no melhor posicionamento possível baseado em um conjunto de múltiplos objetivos. Nesse trabalho foram considerados como objetivos a serem alcançados: redução no consumo de energia do *datacenter*; redução no número de migrações necessárias de serem realizadas para, partindo do indivíduo base, chegar no indivíduo final (solução); e redução da dispersão de VMs dos tipos críticas e não críticas dentre as máquinas físicas. Além disso, a complexidade foi ampliada ao inserir a possibilidade de aplicar sobrecarga dos recursos físicos, feita de forma dinâmica, o que favorece a alocação de um maior número de VMs do tipo não crítica, ao mesmo tempo que permite melhorar o desempenho dos serviços implantados em VMs críticas.

Algumas questões de pesquisa foram levantadas, demarcando o escopo desta tese e que são lembradas aqui:

- *Questão de Pesquisa 1 (QP1): Até que ponto usar a abordagem VMPOS para o posicionamento de VMs fornece melhores resultados que as outras abordagens avaliadas para alocação de VMs em um Datacenter?*

- *Questão de Pesquisa 2 (QP2): Até que ponto usar a abordagem VMPOS ao invés de abordagens que representam o estado da arte, para otimizar a alocação de VMs considerando múltiplos objetivos através de algoritmos genéticos, permite melhorar os resultados em termos das funções objetivos consideradas na pesquisa?*
- *Questão de Pesquisa 3 (QP3): O favorecimento de uma das funções objetivo consegue melhorar o objetivo em questão e qual a influência nos demais objetivos?*

Foi necessário realizar uma gama de experimentos para responder a essas perguntas os quais, através dos resultados, mostraram que a abordagem VMPOS foi, de forma significativa, estatisticamente superior em relação às três abordagens de posicionamento de VMs utilizadas para comparação.

Na questão QP1, foi comprovado através de testes estatísticos que a abordagem VMPOS foi superior às abordagens *First Fit* e *Random Fit*, tanto no objetivo de redução do consumo de energia quanto no de dispersão, principalmente quando se aplicou sobrecarga de recursos. Os únicos comparativos em que não possível atribuir melhores resultados da VMPOS foram registrados em relação ao *First Fit*, no objetivo redução do consumo de energia, apenas nos *datacenters* Médio e Grande e nos casos em que nenhum nível de sobrecarga era aplicado. Para todos os outros cenários, abordagens e níveis de comprometimento utilizados, a abordagem VMPOS foi estatisticamente melhor.

Para responder à questão QP2, foi utilizado um modelo atual na literatura e similar ao aqui desenvolvido, o iMaVMP, que também utiliza evolução de objetivos simultaneamente com algoritmo memético e aplicação da fronteira de Pareto. Os trabalhos abordam objetivos diferentes, sendo a redução do consumo de energia o único objetivo comum. Além disso, a iMaVMP não aplica sobrecarga de recursos. Por esse último motivo, as comparações mais justas entre essas duas abordagens ocorreram sem a aplicação dessa técnica na VMPOS. Mesmo assim, os resultados estatísticos comprovaram que a VMPOS é melhor que a iMaVMP no objetivo de reduzir o consumo de energia no *datacenter*. A maior redução média observada chegou a 34,06% no ambiente com tamanho Grande e com 50% de VMs críticas.

No fator tempo de execução, as abordagens VMPOS e iMaVMP obtiveram resultados similares quando aplicadas ao *datacenter* Pequeno. Contudo, para os ambientes Médio e

Grande, as diferenças nos tempos de execução foram muito grandes, sendo favoráveis à VMPOS que conseguiu manter um crescimento pequeno no tempo de execução a medida em que se aumentava o tamanho do DC (em torno de duas vezes o tempo observado para o DC anterior, a cada aumento de tamanho). Já na iMaVMP o crescimento foi de mais de 285 vezes quando se comparou o tempo de execução para o DC Grande em relação ao DC Pequeno.

Portanto, também nos comparativos com a iMaVMP foi possível afirmar estatisticamente que a abordagem VMPOS conseguiu obter soluções de posicionamento de VMs com menor custo energético do que as conseguidas com a abordagem iMaVMP, para todos os tamanhos de *datacenters* e percentuais de VMs críticas aqui utilizados. Mesmo sem aplicação de sobrecarga de recursos na VMPOS. Assim, ressalta-se ainda mais as qualidades da abordagem aqui desenvolvida, visto que, além de ser melhor em atender aos objetivos traçados, o fez em tempo significativamente menor para *datacenters* maiores.

Com relação à questão QP3, desejava-se fornecer ao tomador de decisão da nuvem privada a opção de favorecer um dos três objetivos utilizados na abordagem. Ou seja, que fosse possível por exemplo, aplicar ao *datacenter* a solução em que o fator de migração obteve o menor valor, independentemente do custo energético ou do nível de dispersão das VMs críticas e não críticas. Acredita-se que essa seja uma possibilidade poderosa nas mãos do administrador do ambiente por possibilitar aplicar, de forma fácil, o favorecimento de um dos objetivos. Testes estatísticos separados para cada um dos três objetivos mostraram que conseguiu-se reduzir ainda mais, na grande maioria dos casos, os resultados obtidos sem a aplicação de favorecimento, comparando a abordagem VMPOS contra ela mesma com a aplicação de favorecimento. Fato confirmado principalmente para os dados de Energia e Migração.

Com os dados de Dispersão, para a maioria dos casos não foi possível melhorar os resultados já obtidos pela VMPOS. Logo, não fez diferença aplicar priorização de tal objetivo. Por ser um objetivo que está relacionado ao número de VMs alocadas nas MFs, não é possível obter grandes diferenças entre a execução em que não há priorização para a que há priorização desse objetivo visto que todas as VMs terão que estar posicionadas ao final da evolução e escolha de indivíduo único. Ou seja, conseguir resultado de posicionamento com uma MF a mais ou a menos, afeta mais o consumo de energia e número de migrações do que

a dispersão.

Esta tese acrescentou à literatura contribuições relacionadas ao problema de posicionamento de VMs em MFs, em nuvem privada, através da aplicação de abordagem evolutiva, atendendo a múltiplos objetivos sendo evoluídos de forma simultânea, com aplicação de sobrecarga de recursos às máquinas físicas. Além de fornecer um método para avaliação do nível de dispersão das máquinas com e sem serviços críticos.

6.1 Publicações Científicas

As etapas desenvolvidas até a finalização desta pesquisa culminaram com as seguintes publicações científicas:

(a) *Comparing performance overhead of Virtual Machine Monitors*, nos *Proceedings* da 11th *International Conference on Applied Computing*, Porto, Portugal, 2014. (ISSN: 1646-3692) [31];

(b) *Evaluating VM live migration overhead with Xen hypervisor*, nos *Proceedings* da 15th *International Conference WWW/Internet*, Mannheim, Alemanha, 2016. (ISSN: 1645-7641) [30];

(c) *Evaluating Live Virtual Machine Migration Overhead on Client's Application Perspective*, nos *Proceedings* da 31th *ICOIN (International Conference on Information Networking)*, Da Nang, Vietnam, 2017. (Extrato B1, DOI: 10.1109/ICOIN.2017.7899536) [24];

(d) *An Experimental Study on Virtual Machine Live Migration Impact on Services Performance*, no periódico *Computing and Informatics*, 38(2):1001–1030, 2019. (Extrato B1, ISSN: 2585-8807) [100].

6.2 Trabalhos Futuros

Mesmo com comprovada qualidade da abordagem VMPOS nos experimentos realizados, acredita-se que alguns aspectos podem contribuir com a evolução da pesquisa em trabalhos

futuros, os quais são discutidos a seguir:

- Devido à complexidade da criação de uma ferramenta completa de gerenciamento de nuvem privada e a dificuldade em integrá-la a um orquestrador real para melhorar a alocação, não foi possível expandir o trabalho inserindo os componentes macros do gerenciador do ambiente, a exemplo do Escalonador e Gerenciador de recursos, que fazem interface com a abordagem VMPOS. Com a continuidade do trabalho, será possível desenvolver uma ferramenta de gerenciamento de nuvem privada mais completa;
- Aplicar diferentes custos de migração aos dois diferentes tipos de máquinas virtuais, com e sem serviços críticos, pode trazer situações diferentes das encontradas nos experimentos apresentados. Nos experimentos descritos aqui foram mantidos os custos de migração fixos em 1,0 e 2,0 para VMs com e sem serviços críticos, respectivamente;
- Explorar outros níveis de comprometimento aplicados às máquinas físicas, além dos que aqui foram utilizados, pode melhorar ainda mais os resultados obtidos visto que favorecerá muito os objetivos de redução de custo energético e redução da dispersão;
- A abordagem VMPOS foi desenvolvida com foco em nuvem privada. Sugere-se que a mesma seja testada e ajustada para funcionar com ambientes maiores, em nuvens híbridas e públicas, onde será necessário incluir alguns fatores relacionados aos clientes como os contratos de níveis de serviços estabelecidos entre provedor e cliente;
- Realizar comparações entre a VMPOS, que utiliza evolução de múltiplos objetivos de forma pura, com alguma abordagem mono objetivo que aplica pesos para reduzir os múltiplos objetivos a um objetivo único;
- Avaliar a abordagem VMPOS ao longo do tempo para verificar o comportamento do *datacenter* após sucessivas trocas de posicionamento.

Bibliografia

- [1] Qiao Yan, F Richard Yu, Qingxiang Gong, and Jianqiang Li. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys & Tutorials*, 18(1):602–622, 2016.
- [2] Jiangtao Zhang, Zhixiang He, Hejiao Huang, Xuan Wang, Chonglin Gu, and Lingmin Zhang. SLA aware cost efficient virtual machines placement in cloud computing. In *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International*, pages 1–8. IEEE, 2014.
- [3] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08*, pages 1–10. Ieee, 2008.
- [4] Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andrew Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, and Ion Stoica. Above the clouds: A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, 28(13):2009, 2009.
- [5] Chunlin Li and Layuan Li. Efficient resource allocation for optimizing objectives of cloud users, IaaS provider and saas provider in cloud environment. *The Journal of Supercomputing*, 65(2):866–885, 2013.
- [6] Sukhpal Singh and Inderveer Chana. A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing*, 14(2):217–264, 2016.
- [7] Chandu Thota, Gunasekaran Manogaran, Daphne Lopez, and V Vijayakumar. Big

- data security framework for distributed cloud data centers. In *Cybersecurity Breaches and Issues Surrounding Online Threat Protection*, pages 288–310. IGI Global, 2017.
- [8] Ivan I Ivanov. Utility computing: Reality and beyond. In *International Conference on E-Business and Telecommunications*, pages 16–29. Springer, 2007.
- [9] Rajeev Kumar Gupta and RK Pateriya. Survey on virtual machine placement techniques in cloud computing environment. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, 4(4):1–7, 2014.
- [10] Sunilkumar S Manvi and Gopal Krishna Shyam. Resource management for infrastructure as a service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications*, 41:424–440, 2014.
- [11] Mohammad Masdari, Sayyid Shahab Nabavi, and Vafa Ahmadi. An overview of virtual machine placement schemes in cloud computing. *Journal of Network and Computer Applications*, 66:106–127, 2016.
- [12] Jiangtao Zhang, Hejiao Huang, and Xuan Wang. Resource provision algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, 64:23–42, 2016.
- [13] Ahmad Amjad Gawanmeh, Alain April, et al. Optimizing resource allocation scheduling in cloud computing services. *Journal of Theoretical & Applied Information Technology*, 95(1), 2017.
- [14] Pooja S Kshirsagar and Anita M Pujar. Resource allocation strategy with lease policy and dynamic load balancing. *International Journal of Modern Education & Computer Science*, 9(2), 2017.
- [15] CN Sahoo and Veena Goswami. Performance evaluation of cloud centers with high degree of virtualization to provide mapreduce as service. *International Journal of Advances in Soft Computing & Its Applications*, 8(3), 2016.
- [16] Madjid Tavana, Saleh Shahdi-Pashaki, Ehsan Teymourian, Francisco J Santos-Arteaga, and Mohammad Komaki. A discrete cuckoo optimization algorithm for con-

- solidation in cloud computing. *Computers & Industrial Engineering*, 115:495–511, 2018.
- [17] Oshin Sharma and Hemraj Saini. Energy and SLA efficient virtual machine placement in cloud environment using non-dominated sorting genetic algorithm. *International Journal of Information Security and Privacy (IJISP)*, 13(1):1–16, 2019.
- [18] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56:684–700, 2016.
- [19] Ligang He, Deqing Zou, Zhang Zhang, Chao Chen, Hai Jin, and Stephen A Jarvis. Developing resource consolidation frameworks for moldable virtual machines in clouds. *Future Generation Computer Systems*, 32:69–81, 2014.
- [20] In Kee Kim, Sai Zeng, Christopher Young, Jinho Hwang, and Marty Humphrey. A supervised learning model for identifying inactive VMs in private cloud data centers. In *Proceedings of the Industrial Track of the 17th International Middleware Conference*, page 2. ACM, 2016.
- [21] Robayet Nasim and Andreas J Kassler. Deploying openstack: Virtual infrastructure or dedicated hardware. In *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*, pages 84–89. IEEE, 2014.
- [22] Glauco Estácio Gonçalves, Patrícia Takako Endo, Thiago Damasceno Cordeiro, AVA Palhares, Djamel Sadok, Judith Kelner, Bob Melander, and J Mangs. Resource allocation in clouds: concepts, tools and research challenges. *XXIX SBRC-Gramado-RS*, 2011.
- [23] Omnia AbdElRahem, Ayman M Bahaa-Eldin, and Ayman Taha. Virtualization security: A survey. In *Computer Engineering & Systems (ICCES), 2016 11th International Conference on*, pages 32–40. IEEE, 2016.
- [24] Petrônio Bezerra, Gustavo Martins, Reinaldo Gomes, Felype Cavalcante, and Anderson Costa. Evaluating live virtual machine migration overhead on client’s application

- perspective. In *Information Networking (ICOIN), 2017 International Conference on*, pages 503–508. IEEE, 2017.
- [25] Anum Masood, Muhammad Sharif, Mussarat Yasmin, and Mudassar Raza. Virtualization tools and techniques: Survey. *Nepal Journal of Science and Technology*, 15(2):141–150, 2015.
- [26] Yefu Wang and Xiaorui Wang. Performance-controlled server consolidation for virtualized data centers with multi-tier applications. *Sustainable Computing: Informatics and Systems*, 4(1):52–65, 2014.
- [27] Pradeep Padala, Xiaoyun Zhu, Zhikui Wang, Sharad Singhal, Kang G Shin, et al. Performance evaluation of virtualization technologies for server consolidation. *HP Labs Tec. Report*, 2007.
- [28] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5):755–768, 2012.
- [29] Raja Benali, Hana Teyeb, Ali Balma, Samir Tata, and Nejib Ben Hadj-Alouane. Evaluation of traffic-aware VM placement policies in distributed cloud using cloudsims. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2016 IEEE 25th International Conference on*, pages 95–100. IEEE, 2016.
- [30] Petrônio Bezerra, Gustavo Martins, César Rocha, Reinaldo Gomes, and Fellype Cavalcante. Evaluating VM live migration overhead with xen hypervisor. In *Proceedings of the 15th International Conference WWW/Internet 2016*, pages 274–278, Mannheim, Germany, October 2016.
- [31] Petrônio Carlos Bezerra and Reinaldo Morais Gomes. Comparing performance overhead of virtual machine monitors. In *Proceedings of the 11th International Conference on Applied Computing 2014*, pages 206–213, Porto, Portugal, October 2014.
- [32] Gustavo Martins, Petrônio Bezerra, Reinaldo Gomes, Fellype Albuquerque, and Anderson Costa. Evaluating performance degradation in nosql databases generated by

- virtualization. In *Network Operations and Management Symposium (LANOMS), 2015 Latin American*, pages 84–91. IEEE, 2015.
- [33] Wei Wei, Xuanzhong Wei, Tao Chen, Xiaofeng Gao, and Guihai Chen. Dynamic correlative VM placement for quality-assured cloud service. In *Communications (ICC), 2013 IEEE International Conference on*, pages 2573–2577. IEEE, 2013.
- [34] Ahmed Sallam and Kenli Li. A multi-objective virtual machine migration policy in cloud systems. *The Computer Journal*, 57(2):195–204, 2014.
- [35] Shao-Heng Wang, Patrick P-W Huang, Charles H-P Wen, and Li-Chun Wang. EQVMP: Energy-efficient and QoS-aware virtual machine placement for software defined datacenter networks. In *Information Networking (ICOIN), 2014 International Conference on*, pages 220–225. IEEE, 2014.
- [36] Xianghua Xu, Feng Zhou, Jian Wan, and Yucheng Jiang. Quantifying performance properties of virtual machine. In *2008 International Symposium on Information Science and Engineering*, pages 24–28. IEEE, 2008.
- [37] Seyed Ebrahim Dashti and Amir Masoud Rahmani. Dynamic VMs placement for energy efficiency by PSO in cloud computing. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(1-2):97–112, 2016.
- [38] Mohamed Almorsy, John Grundy, and Ingo Müller. An analysis of the cloud computing security problem. *arXiv preprint arXiv:1609.01107*, 2016.
- [39] Alireza Sadeghi Milani and Nima Jafari Navimipour. Load balancing mechanisms and techniques in the cloud environments: systematic literature review and future trends. *Journal of Network and Computer Applications*, 71:86–98, 2016.
- [40] Cisco Visual Networking. Cisco global cloud index: Forecast and methodology 2015-2020 (white paper). *Cisco Public, San Jose*, 2016.
- [41] Sukhpal Singh and Inderveer Chana. Cloud resource provisioning: survey, status and future research directions. *Knowledge and Information Systems*, 49(3):1005–1069, 2016.

- [42] Jie Chen, Chun Cao, Ying Zhang, Xiaoxing Ma, Haiwei Zhou, and Chengwei Yang. Improving cluster resource efficiency with oversubscription. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pages 144–153. IEEE, 2018.
- [43] Zoha Usmani and Shailendra Singh. A survey of virtual machine placement techniques in a cloud data center. *Procedia Computer Science*, 78:491–498, 2016.
- [44] Fabio Lopez Pires and Benjamín Barán. A many-objective optimization framework for virtualized datacenters. In *CLOSER*, pages 439–450, 2015.
- [45] Aisha Fatima, Nadeem Javaid, Ayesha Anjum Butt, Tanzeela Sultana, Waqar Hussain, Muhammad Bilal, Mariam Akbar, Manzoor Ilahi, et al. An enhanced multi-objective gray wolf optimization for virtual machine placement in cloud data centers. *Electronics*, 8(2):218, 2019.
- [46] Takfarinas Saber, Joao Marques-Silva, James Thorburn, and Anthony Ventresque. Exact and hybrid solutions for the multi-objective VM reassignment problem. *International Journal on Artificial Intelligence Tools*, 26(01):1760004, 2017.
- [47] Guangyi Cao. Topology-aware multi-objective virtual machine dynamic consolidation for cloud datacenter. *Sustainable Computing: Informatics and Systems*, 21:179–188, 2019.
- [48] Peter Mell and Tim Grance. The nist definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50, 2011.
- [49] Minxian Xu, Wenhong Tian, and Rajkumar Buyya. A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurrency and Computation: Practice and Experience*, 29(12):e4123, 2017.
- [50] Christian von Lüken, Benjamín Barán, and Carlos Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational optimization and applications*, 58(3):707–756, 2014.
- [51] Robert P Goldberg. Survey of virtual machine research. *Computer*, 7(6):34–45, 1974.

- [52] Alexandre Carissimi. Virtualização: da teoria a soluções. *Minicursos do Simpósio Brasileiro de Redes de Computadores–SBRC*, 2008:173–207, 2008.
- [53] Mendel Rosenblum and Tal Garfinkel. Virtual machine monitors: Current technology and future trends. *Computer*, 38(5):39–47, 2005.
- [54] Keith Adams and Ole Agesen. A comparison of software and hardware techniques for x86 virtualization. *ACM SIGOPS Operating Systems Review*, 40(5):2–13, 2006.
- [55] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *ACM SIGOPS operating systems review*, volume 37, pages 164–177. ACM, 2003.
- [56] VMware. Understanding full virtualization, paravirtualization, and hardware assist. *White Paper*, pages 1–14, 2007.
- [57] Marcos Laureano. *Máquinas virtuais e emuladores: conceitos, técnicas e aplicações*. Novatec Editora, 2006.
- [58] William Von Hagen. *Professional xen virtualization*. John Wiley & Sons, 2008.
- [59] Khoa Huynh and Stefan Hajnoczi. KVM/QEMU storage stack performance discussion. In *Linux Plumbers Conference*. IBM, 2010.
- [60] William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya. Cost of virtual machine live migration in clouds: A performance evaluation. In *IEEE International Conference on Cloud Computing*, pages 254–265. Springer, 2009.
- [61] Antonios Gkortzis, Stamatia Rizou, and Diomidis Spinellis. An empirical analysis of vulnerabilities in virtualization technologies. In *Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on*, pages 533–538. IEEE, 2016.
- [62] Sogand Shirinbab, Lars Lundberg, and Emiliano Casalicchio. Performance evaluation of container and virtual machine running cassandra workload. In *Cloud Computing Technologies and Applications (CloudTech), 2017 3rd International Conference of*, pages 1–8. IEEE, 2017.

- [63] Roberto Morabito, Jimmy Kjällman, and Miika Komu. Hypervisors vs. lightweight virtualization: a performance comparison. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pages 386–393. IEEE, 2015.
- [64] Zhiyong Li, Dong Chen, Ahmed Sallam, and Li Zhao. Novel multi-objective genetic algorithm based on static bayesian game strategy. *Advances in Swarm Intelligence*, pages 612–619, 2010.
- [65] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.
- [66] Rogério Ishibashi. *Extração de conhecimentos com interpretabilidade aumentada utilizando modelagem fuzzy e otimização multi-objetivo*. PhD thesis, Instituto Tecnológico de Aeronáutica, 2013.
- [67] Jing Xu and Jose AB Fortes. Multi-objective virtual machine placement in virtualized data center environments. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pages 179–188. IEEE Computer Society, 2010.
- [68] Dan Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
- [69] Sambit Kumar Mishra, Deepak Puthal, Bibhudatta Sahoo, Prem Prakash Jayaraman, Song Jun, Albert Y Zomaya, and Rajiv Ranjan. Energy-efficient VM-placement in cloud data center. *Sustainable Computing: Informatics and Systems*, 2018.
- [70] JA Vasconcelos, Jaime Arturo Ramirez, RHC Takahashi, and RR Saldanha. Improvements in genetic algorithms. *IEEE Transactions on magnetics*, 37(5):3414–3417, 2001.
- [71] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [72] Pablo Moscato and Carlos Cotta. A modern introduction to memetic algorithms. In *Handbook of metaheuristics*, pages 141–183. Springer, 2010.

- [73] Richard Dawkins. *The selfish gene, revised edn.* Oxford University Press, Oxford, 1989.
- [74] Xingsi Xue and Yuping Wang. Using memetic algorithm for instance coreference resolution. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):580–591, 2016.
- [75] Syed Hamid Hussain Madni, Muhammad Shafie Abd Latiff, Yahaya Coulibaly, et al. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities. *Journal of Network and Computer Applications*, 68:173–200, 2016.
- [76] Violeta Medina and Juan Manuel García. A survey of migration mechanisms of virtual machines. *ACM Computing Surveys (CSUR)*, 46(3):30, 2014.
- [77] Rafael Weingärtner, Gabriel Beims Bräscher, and Carlos Becker Westphall. Cloud resource management: A survey on forecasting and profiling models. *Journal of Network and Computer Applications*, 47:99–106, 2015.
- [78] Weijia Song, Zhen Xiao, Qi Chen, and Haipeng Luo. Adaptive resource provisioning for the cloud using online bin packing. *IEEE Transactions on Computers*, 63(11):2647–2660, 2014.
- [79] Hui Wang and Huaglory Tianfield. Energy-aware dynamic virtual machine consolidation for cloud datacenters. *IEEE Access*, 6:15259–15273, 2018.
- [80] Qi Zhang, Quanyan Zhu, and Raouf Boutaba. Dynamic resource allocation for spot markets in cloud computing environments. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, pages 178–185. IEEE, 2011.
- [81] Lingfang Zeng, Bharadwaj Veeravalli, and Qingsong Wei. Space4time: Optimization latency-sensitive content service in cloud. *Journal of Network and Computer Applications*, 41:358–368, 2014.
- [82] Adnan Ashraf and Ivan Porres. Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system. *International Journal of Parallel, Emergent and Distributed Systems*, 33(1):103–120, 2018.

- [83] Fabio López Pires and Benjamín Barán. Multi-objective virtual machine placement with service level agreement: A memetic algorithm approach. In *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, pages 203–210. IEEE Computer Society, 2013.
- [84] Fabio López-Pires and Benjamín Barán. Many-objective virtual machine placement. *Journal of Grid Computing*, 15(2):161–176, Jun 2017.
- [85] Qingqing Xiong, Emre Ates, Martin C Herbordt, and Ayse K Coskun. Tangram: Colocating hpc applications with oversubscription. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2018.
- [86] Rachel Householder, Scott Arnold, and Robert Green. On cloud-based oversubscription. *arXiv preprint arXiv:1402.4758*, 2014.
- [87] Kangkang Li, Huanyang Zheng, Jie Wu, and Xiaojiang Du. Virtual machine placement in cloud systems through migration process. *International Journal of Parallel, Emergent and Distributed Systems*, 30(5):393–410, 2015.
- [88] Alexandru-Florian Antonescu, Philip Robinson, and Torsten Braun. Dynamic SLA management with forecasting using multi-objective optimization. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 457–463. IEEE, 2013.
- [89] Glauco Estácio Gonçalves, André Vitor de Almeida Palhares, Marcelo Anderson Batista dos Santos, Patricia Takako Endo, Judith Kelner, and Djamel Sadok. Balancing the load across virtual links from virtual machine requests in distributed clouds. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 506–512. IEEE, 2013.
- [90] Xiangming Dai, Jason Min Wang, and Brahim Bensaou. Energy-efficient virtual machine placement in data centers with heterogeneous requirements. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 161–166. IEEE, 2014.

- [91] James F Kurose and Keith W Ross. *Computer networking: a top-down approach*, 2013.
- [92] Tiago C Ferreto, Marco AS Netto, Rodrigo N Calheiros, and César AF De Rose. Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, 27(8):1027–1034, 2011.
- [93] Richard E Korf. A new algorithm for optimal bin packing. In *AAAI/IAAI*, pages 731–736, 2002.
- [94] Fabio López Pires, Elias Melgarejo, and Benjamín Barán. Virtual machine placement. a multi-objective approach. In *Computing Conference (CLEI), 2013 XXXIX Latin American*, pages 1–8. IEEE, 2013.
- [95] Herbert SCHILDT and C Completo. *Total*. Makron Books, 1997.
- [96] Raj K Jain. *Art of Computer Systems Performance Analysis: Techniques for Experimental Design Measurements... Simulation and Modeling*. John Wiley, 2015.
- [97] Thorsten Pohlert and Maintainer Thorsten Pohlert. Package ‘pmcmr’. *R package*, 2018.
- [98] Peter Nemenyi. Distribution-free multiple comparisons. In *Biometrics*, volume 18, page 263. INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1962.
- [99] Thorsten Pohlert. The pairwise multiple comparison of mean ranks package (pmcmr). *R package*, 27, 2014.
- [100] Petrônio Bezerra, Marcela Santos, Edlane Alves, Fellype Albuquerque, Gustavo Martins, Reinaldo Gomes, and Anderson Costa. An experimental study on virtual machine live migration impact on services performance. *Computing and Informatics*, 38(2):1001–1030, 2019.
- [101] Senthil Nathan, Umesh Bellur, and Purushottam Kulkarni. Towards a comprehensive performance model of virtual machine live migration. In *Proceedings of the Sixth ACM Symposium on Cloud Computing*, pages 288–301. ACM, 2015.

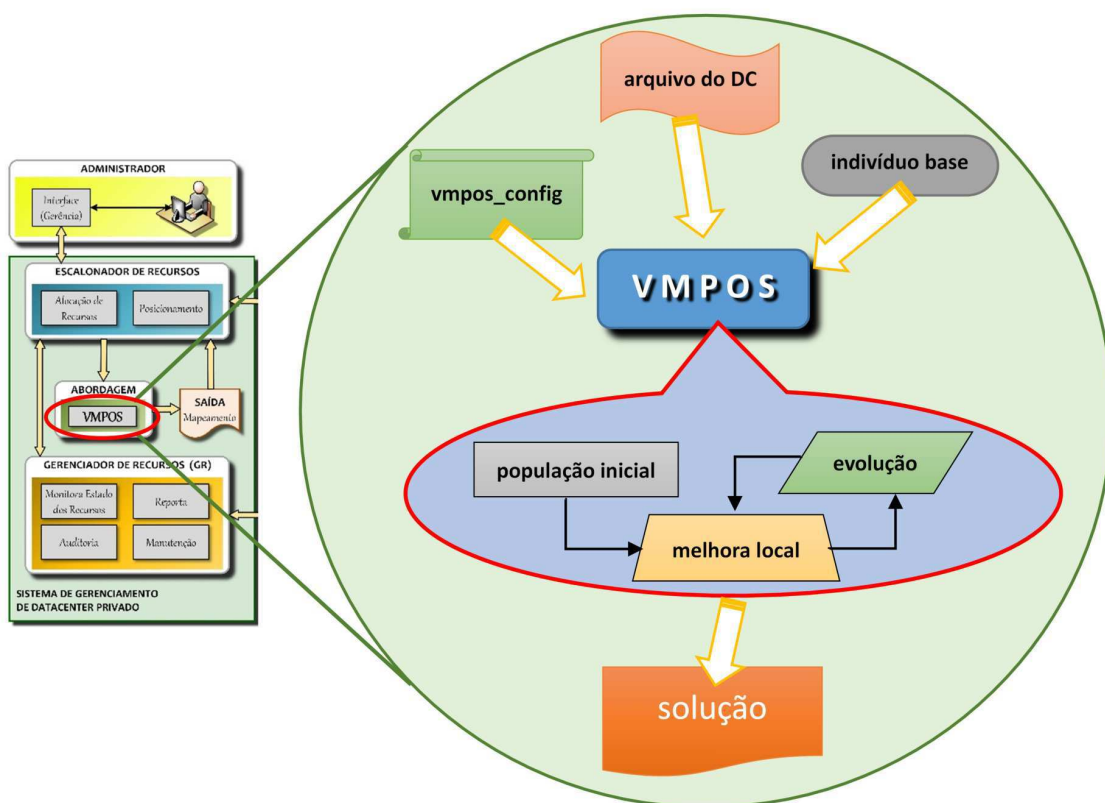
Apêndices

Apêndice A

Configurações da VMPOS

As informações de entrada para a abordagem VMPOS podem ser vistas na Figura A.1.

Figura A.1: Ampliação com detalhes da abordagem VMPOS.



Fonte: Elaborada pelo autor.

A.1 Arquivo de Configuração

Para facilitar as alterações nas variáveis da VMPOS foi criado um arquivo de configuração, chamado *vmpos_config.vmp*, no qual é possível ajustar facilmente os parâmetros necessários para a execução da abordagem. Na Figura A.2 encontra-se uma representação da aplicação das configurações desse arquivo, para uma melhor visualização.

Figura A.2: Arquivo *vmpos_config.vmp* com exemplo de configuração da VMPOS.

```
POPULATION
5      200

OBJECTIVES WEIGHTS
1.0    1.0    1.0

COMMITMENT LEVELS
1.0    2.0

MIGRATION COSTS
2.0    1.0
```

Fonte: Elaborada pelo autor.

No arquivo de configuração apresentado, todos os números que aparecem estão separados por tabulação. Na linha logo abaixo da palavra *POPULATION*, existem dois números, o primeiro identifica quantos indivíduos se deseja utilizar na população, e o segundo número informa por quantas gerações a população deve ser evoluída. Com a configuração do exemplo, a abordagem VMPOS utilizaria 5 (cinco) indivíduos, sendo evoluídos por 200 (duzentas) gerações.

Na linha logo abaixo de *OBJECTIVES WEIGHTS*, existem três números, os quais indicam os pesos que se deseja aplicar para cada um dos três objetivos a serem otimizados, sendo atribuídos, respectivamente a: energia, dispersão e migração. Quando não se desejar favorecer nenhum dos objetivos, aplica-se o valor 1,0 a cada um desses pesos, como no exemplo. Dessa forma, não se influenciará nos resultados tendendo a um dos objetivos, mas, mesmo assim, será escolhida uma única solução em que a soma normalizada dos três objetivos seja

a menor dentre as soluções que pertencem a fronteira de Pareto. Caso se deseje favorecer um ou outro objetivo, atribui-se pesos com valores entre 0 e 1 de tal forma que a soma dos três pesos seja igual a 1. No Capítulo 5, com os resultados dos experimentos realizados, contempla-se a aplicação das duas situações.

Na linha logo abaixo de *COMMITMENT LEVELS* existem mais dois números os quais estão associados aos níveis de comprometimento que se deseja aplicar às máquinas físicas do *datacenter*. O primeiro valor será o nível de comprometimento das MFs que abrigam pelo menos uma VM crítica (Serviços Críticos, ou SC), e o segundo valor o nível de comprometimento das MFs que *não* abrigam nenhuma VM crítica (Serviços Não Críticos, ou SNC). No exemplo apresentado, tem-se: SC = 1,0 e SNC = 2,0.

Perceba que fica muito fácil realizar alterações nesses últimos parâmetros que terão grande impacto no resultado da abordagem. Tanto é possível deixar as máquinas físicas do *datacenter* sem aplicar nenhum tipo de nível de sobrecarga de recursos, basta apenas aplicar os valores 1,0 e 1,0 nesse arquivo, assim como é exequível que qualquer arranjo de valores seja aplicado. Por exemplo, se desejar que até as MFs que abrigam VMs críticas possam ter nível de sobrecarga de recursos aumentado, basta alterar os valores no arquivo como por exemplo, 1,5 e 2,5, em que a capacidade teórica de recursos das MFs que abrigam pelo menos uma VM crítica estaria configurada para uma vez e meia a quantidade de recursos reais, e para as que não abrigam nenhuma VM crítica, para duas vezes e meia.

Por último, abaixo da linha *MIGRATION COSTS*, existem os números que estão associados aos custos aplicados à migração de VMs críticas (primeiro valor) e as não críticas (segundo valor). No exemplo apresentado, tem-se: MC_SC = 2,0 e MC_SNC = 1,0. O objetivo aqui é fazer com que as soluções escolhidas, para serem aplicadas, sejam as que migrem o menor número possível de VMs com serviços críticos.

Os valores dos custos aplicados foram decididos de forma aleatória, visando não dificultar a migração de VMs não críticas, e evitar ao máximo a migração das que executam serviços críticos tentando, assim, evitar o *downtime* observado no processo de migração em tempo real.

A.2 Simulador Gerador de *Datacenter*

O simulador sorteia aleatoriamente os valores a serem preenchidos de capacidades das MFs, assim como dos requisitos das VMs, lidos de uma gama de possíveis valores previamente inseridos no arquivo de entrada para o simulador. Mais detalhes sobre o funcionamento do Simulador elaborado encontram-se na página do autor no GitHub³⁰.

O arquivo de entrada utilizado pelo simulador tem, por exemplo, uma estrutura como a da Figura A.3.

Figura A.3: Arquivo de entrada do Simulador.

```
PHYSICAL MACHINES
4 8 16 24 32
8 16 32 48 64
100 120 140 160 200
410 410 420 430 440

VIRTUAL MACHINES
1 2 4
2 4 8
4 6 10
50 0 0
```

Fonte: Elaborada pelo autor.

As quatro linhas existentes na parte de Máquinas Físicas (em inglês, *Physical Machines*) indicam os valores possíveis, que serão atribuídos às MFs do *datacenter*, sendo tomados através da posição na matriz de dados, usando-se um índice que é gerado de forma aleatória. A sequência dessas linhas representa, respectivamente: Número de Processadores (no exemplo, 4 8 16 24 32), Quantidade de Memória (no exemplo, 8 16 32 48 64) em GB, Capacidade de Disco (no exemplo, 100 120 140 160 200) em GB e Consumo Energético Máximo quando o servidor está sendo completamente utilizado (no exemplo, 410 410 420 430 440) em *Watt-hora*.

Os dados são tratados como uma matriz, no caso de MFs de 4x5 valores, e um valor aleatório é gerado para representar o índice e assim montar os recursos da máquina física.

³⁰<https://github.com/petroniocg/simulator>

Suponha que o índice 2 foi sorteado, logo, os valores a serem colocados em uma das MFs do DC seriam: Número de Processadores (16), Quantidade de Memória (32), Capacidade de Disco (140) e Consumo Energético (420). O que corresponde aos valores da posição da coluna 2 em cada linha, começando a contar do zero.

As 4 linhas existentes na parte de Máquinas Virtuais (em inglês, *Virtual Machines*) indicam respectivamente os valores possíveis, para cada uma das VMs do *datacenter*. E a sequência destas 4 linhas representa, respectivamente: Número de Processadores Virtuais (no exemplo, 1 2 4), Quantidade de Memória (no exemplo, 2 4 8), Capacidade de Disco (no exemplo, 4 6 10) e Percentual de VMs Críticas desejado para o *datacenter* (no exemplo, 50 0 0), o que indica que serão gerados 50% de VMs Críticas. No caso dessa linha, apenas o primeiro valor será utilizado.

Se desejar colocar um valor fixo, por exemplo, de memória ou qualquer outro parâmetro (exceto o percentual de VMs críticas), basta repetir o mesmo valor desejado em todas as colunas. Isso tanto para MFs como para VMs. Por exemplo, digamos que se deseje que todas as MFs tenham capacidade de memória fixa de 32 GB. Então a segunda linha, das configurações de MFs, deve ser preenchida como: 32 32 32 32 32.

Para se ter ideia do resultado de uma execução do simulador, considere que se deseja gerar o *datacenter* pequeno que é composto de 12x60 (MFs x VMs) com 50% de VMs críticas. Via linha de comando, os valores 12 e 60 são passados nesta sequência, indicando número de máquinas físicas e VMs, respectivamente. E o percentual de VMs críticas deve estar preenchido no arquivo de entrada mostrado anteriormente. Para esse cenário, o simulador geraria um arquivo como o da Figura A.4 (mostrado parcialmente por questão de espaço):

Figura A.4: Arquivo de saída do Simulador.

PHYSICAL MACHINES			
24	48	160	430
8	16	120	410
16	32	140	420
16	32	140	420
8	16	120	410
32	64	200	440
4	8	100	410
16	32	140	420
16	32	140	420
4	8	100	410
4	8	100	410
24	48	160	430
VIRTUAL MACHINES			
4	8	10	1
2	4	6	0
2	4	6	1
4	8	10	0
4	8	10	0
1	2	4	0
4	8	10	0
1	2	4	1
1	2	4	0
1	2	4	1
.			
.			
.			

Fonte: Elaborada pelo autor.

Um arquivo como o ilustrado na Figura A.4 é utilizado como entrada para a abordagem VMPOS. Cada uma das linhas representa ou os recursos de uma máquina física ou os requisitos de uma máquina virtual. As quatro colunas em uma mesma linha, no caso das máquinas físicas, representam: quantidade de processadores, quantidade de memória, capacidade do disco e consumo energético máximo. Para a parte de VMs, as informações são: quantidade de processadores virtuais, quantidade de memória, capacidade do disco e indicação se a VM

executa (1) ou não (0) serviços críticos.

Portanto, o simulador foi utilizado para gerar os arquivos que serviram de entrada para a abordagem aqui desenvolvida. Para cada tamanho de *datacenter* (P, M e G) foram criados três arquivos, cada um com uma quantidade diferente de VMs críticas. As quantidades utilizadas foram 10%, 25% e 50%. Com isso, foram gerados com o simulador nove arquivos de *datacenter* ao todo: P_10%, P_25%, P_50%, M_10%, M_25%, M_50%, G_10%, G_25% e G_50%. Esses arquivos foram a base para todos os experimentos realizados com todas as abordagens utilizadas.

As características das VMs, com relação aos requisitos de recursos, tomaram por base valores aplicados atualmente em alguns tipos de instâncias utilizadas na nuvem pública da *Amazon Elastic Compute Cloud (EC2)*³¹, como por exemplo, os tipos *t2.small*, *t2.medium* e *t2.large*. Segundo informações da Amazon: "*As instâncias T2 são uma boa opção para diversas cargas de trabalho de uso geral, incluindo microsserviços; aplicativos interativos de baixa latência; bancos de dados de pequeno e médio portes; desktops virtuais; ambientes de desenvolvimento, compilação e preparação; repositórios de código e protótipos de produtos*". Acredita-se que estas características estão bem dimensionadas ao tipo de *datacenter* privado que este trabalho se propõe a atender.

A.3 Número de Indivíduos Utilizados

A abordagem desenvolvida utiliza uma população a qual é evoluída para encontrar indivíduos que pertençam ou se aproximem ao máximo da fronteira de Pareto. Na abordagem, o tamanho dessa população é facilmente inserido através de um número inteiro no arquivo de configuração do sistema, como visto na seção anterior.

Antes de realizar as simulações da abordagem, comparando-a a outras metodologias de posicionamento de VMs existentes, decidiu-se perfazer uma série de testes objetivando obter um valor representativo do número de indivíduos a ser aplicado nas simulações. Para isso, foram utilizados os três arquivos referentes ao *datacenter* pequeno, com as seguintes definições: *i)* não foi utilizado indivíduo base, ou seja, a abordagem gerava um indivíduo base aleatoriamente a cada nova execução; *ii)* os pesos para os três objetivos foram mantidos fixos

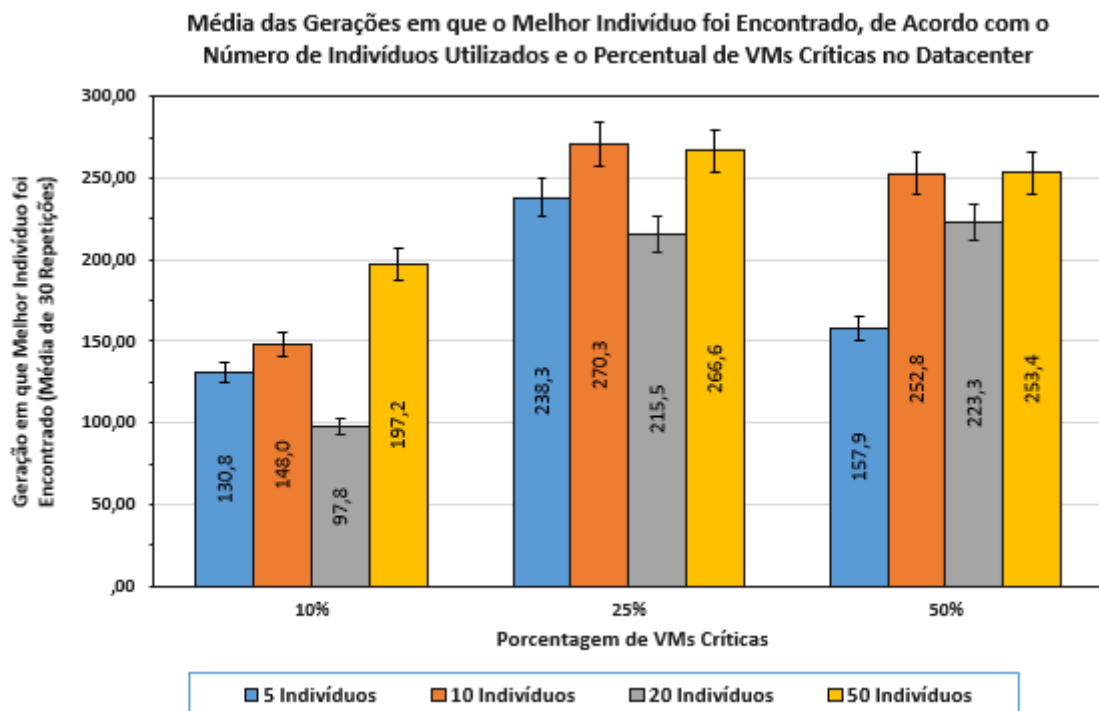
³¹<https://aws.amazon.com/pt/ec2/instance-types/>

em 1,0; e *iii*) os níveis de comprometimento foram fixados em 1,0 para máquinas físicas que abrigam pelo menos uma VM crítica, e 2,0 para máquinas físicas que abrigam apenas VMs não críticas.

Foram realizadas simulações com 5, 10, 20 e 50 indivíduos como população. Para cada uma dessas quantidades, foram executadas evolução da população por 500 gerações. Os testes foram repetidos 30 vezes, perfazendo um total geral de 360 rodadas, e as médias foram utilizadas para as análises.

Para cada uma das 30 execuções, com evolução por 500 gerações, registrava-se a geração em que o melhor indivíduo foi encontrado, sendo esse valor utilizado posteriormente no cálculo da média das 30 execuções. No gráfico da Figura A.5 têm-se a média das gerações em que o melhor indivíduo foi encontrado.

Figura A.5: Escolha do número de indivíduos a ser utilizado nos experimentos.



Fonte: Elaborada pelo autor.

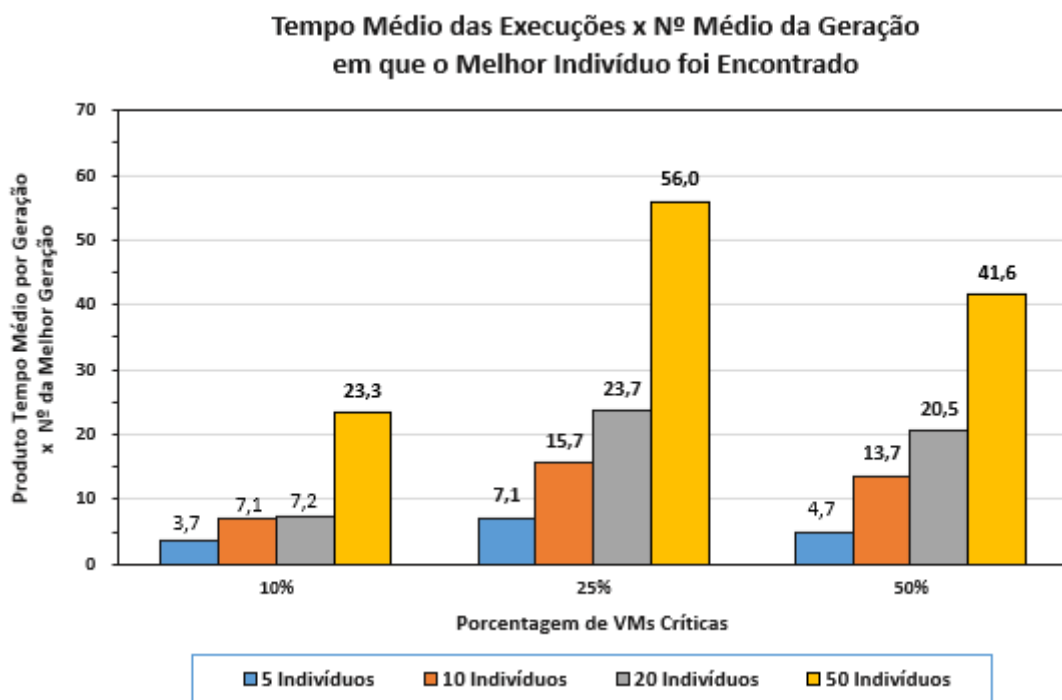
Como se pode observar, o melhor indivíduo foi encontrado, em média, em gerações anteriores quando se utilizava população com 20 indivíduos, para os casos com 10% e 25% de VMs críticas. Com 50% de VMs críticas, o melhor indivíduo foi encontrado em gerações anteriores com 5 indivíduos na população. Porém, as simulações com 20 indivíduos ainda

foram melhores do que os casos com 10 e 50 indivíduos, quando existiam 50% de VMs críticas.

Com o objetivo de realizar uma avaliação dos resultados inserindo o fator tempo, foi utilizada a média do tempo total de cada uma das 30 repetições, dividindo esta média pelo total de gerações de cada rodada, que eram 500 gerações, dando uma ideia da duração de tempo médio para finalizar uma geração. Este resultado foi então multiplicado pela média da geração em que foi encontrado o melhor indivíduo.

Como se pode perceber na Figura A.6, existe um crescimento que acontece com o aumento do percentual de VMs críticas no *datacenter*, assim como com o número de indivíduos. O resultado é melhor para 5 indivíduos em todos os casos, mas, é necessário lembrar que o melhor indivíduo é encontrado mais tardiamente para as configurações com 10% e 25% de VMs críticas utilizando 5 indivíduos do que com 20 indivíduos (ver Figura A.5).

Figura A.6: Fator média de tempo vezes geração do melhor indivíduo.



Fonte: Elaborada pelo autor.

Com 20 indivíduos, utilizando 10% de VMs críticas, o resultado obtido foi tão bom como aquele com 10 indivíduos, e permanece em torno do mesmo patamar quando a quantidade de VMs críticas presentes é de 25% ou 50%, algo na faixa de 20,5 a 23,7. Os resultados utilizando 5 ou 10 indivíduos foram melhores, contudo, as simulações mostraram que o

melhor indivíduo era encontrado mais tardiamente quando utilizando 5 ou 10 indivíduos do que com 20 indivíduos (exceção com 5 indivíduos, com 50% de VMs críticas).

Apesar de com 20 indivíduos demorar um pouco mais para encontrar a geração com melhor indivíduo do que com 5 ou 10 indivíduos, acredita-se que um tempo inferior a 1 minuto (que foi a média encontrada) seja aceitável. Além do mais, aumenta-se a possibilidade de melhor explorar o espaço de soluções com 20 indivíduos do que com 10 ou 5, e com um custo computacional menor do que com 50 indivíduos.

Diante dos resultados, foi decidido sempre utilizar 20 indivíduos nas simulações para comparação da abordagem desenvolvida, assim como utilizar 200 gerações a serem evoluídas, que é um valor aproximado das maiores médias encontradas (Figura A.5).

Apêndice B

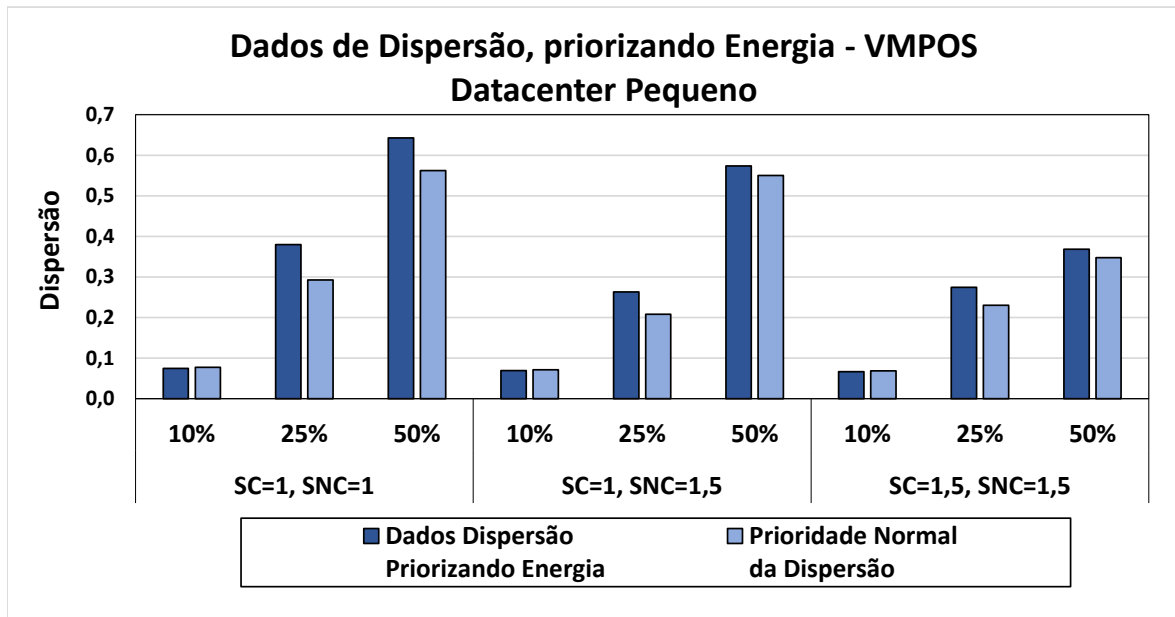
Priorização de Objetivo

Os gráficos presentes neste Apêndice foram obtidos a partir dos dados em que se desejava priorizar um dos objetivos em detrimento dos demais. Nos resultados dos gráficos, foram plotados os valores dos outros dois objetivos, quando se priorizava um terceiro. Assim, é possível entender o que acontece com cada objetivo quando um outro está sendo priorizado.

B.1 Quando foi priorizado o objetivo Energia

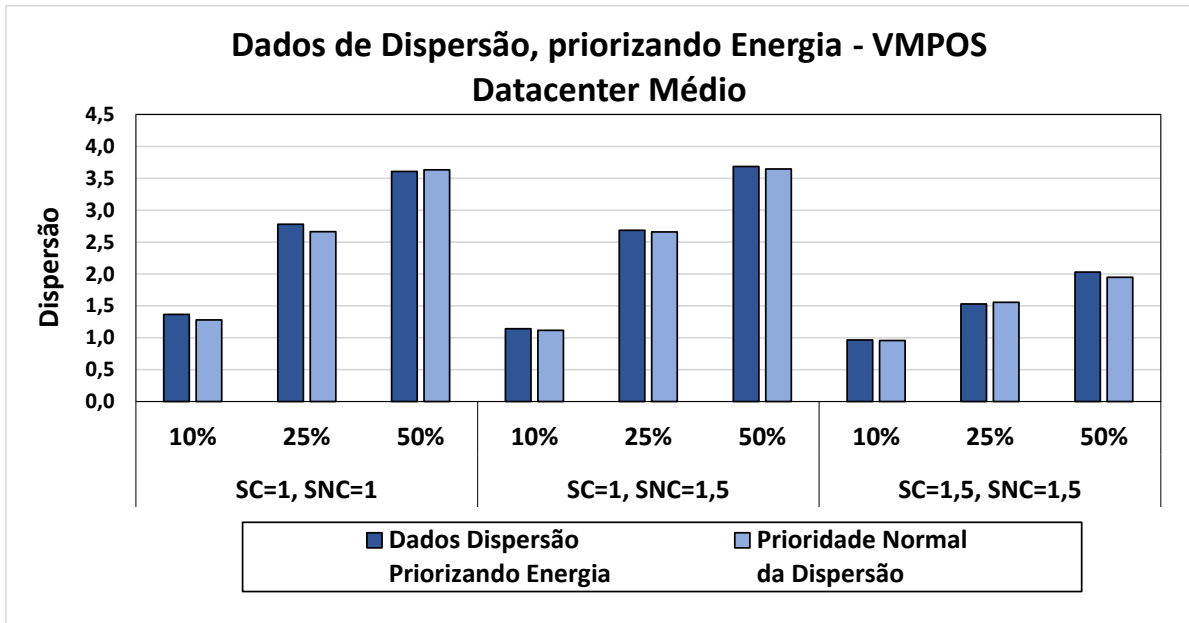
B.1.1 Efeito no objetivo Dispersão

Figura B.1: Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em *Datacenter Pequeno*.



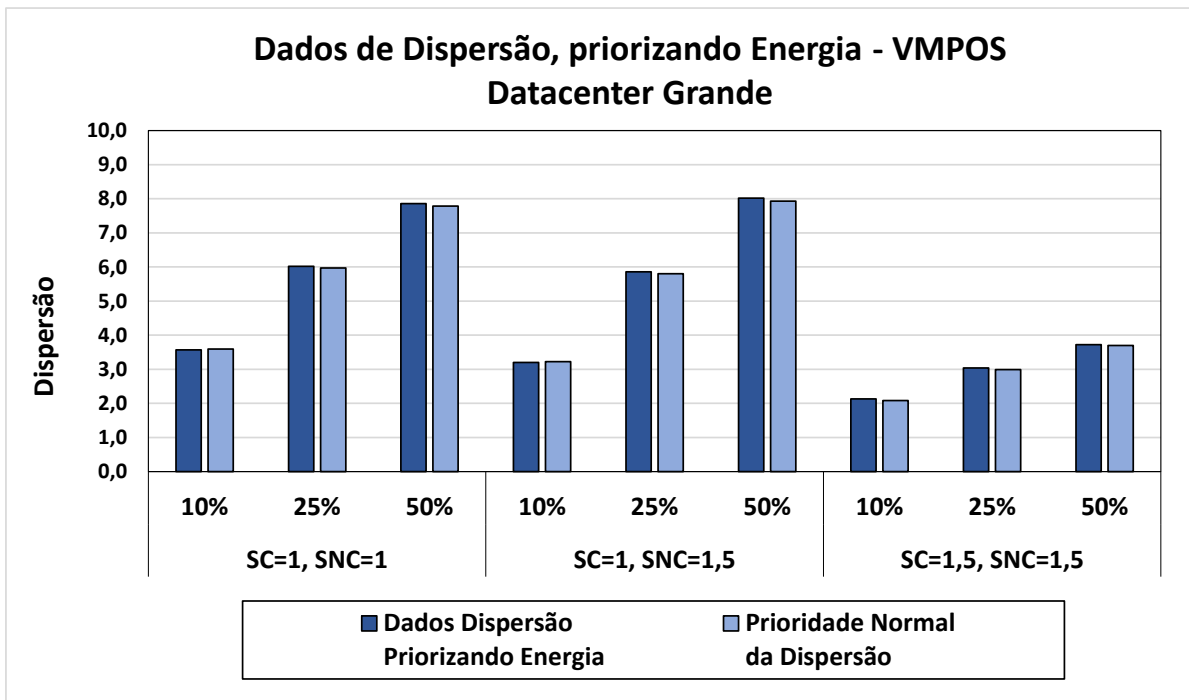
Fonte: Elaborada pelo autor.

Figura B.2: Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em *Datacenter Médio*.



Fonte: Elaborada pelo autor.

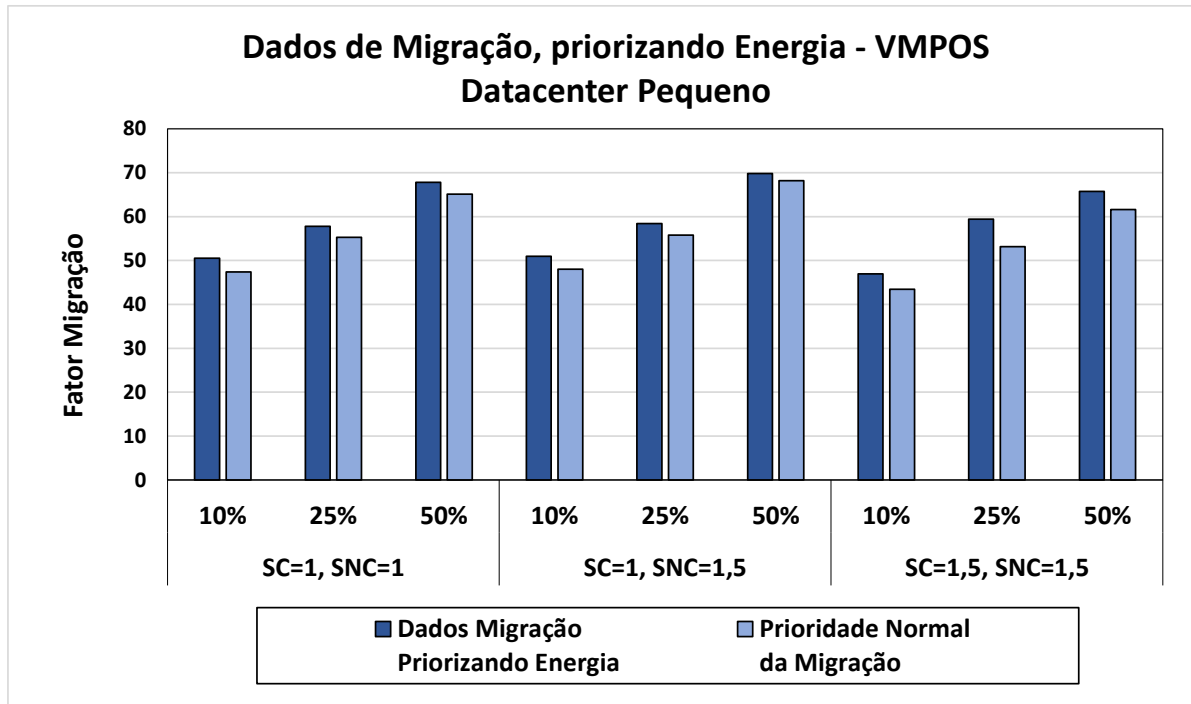
Figura B.3: Resultado da Dispersão quando se aplicou prioridade para o objetivo Energia, em *Datacenter Grande*.



Fonte: Elaborada pelo autor.

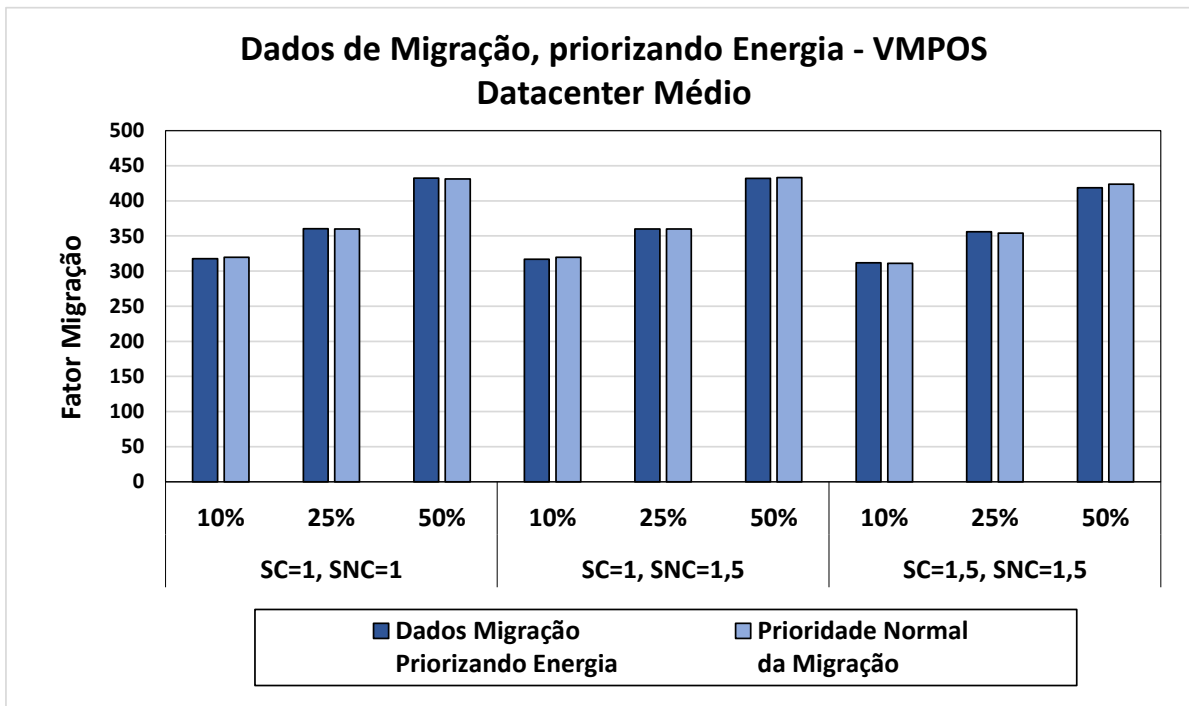
B.1.2 Efeito no objetivo Migração

Figura B.4: Resultado da Migração quando se aplicou prioridade para o objetivo Energia, em *Datacenter Pequeno*.



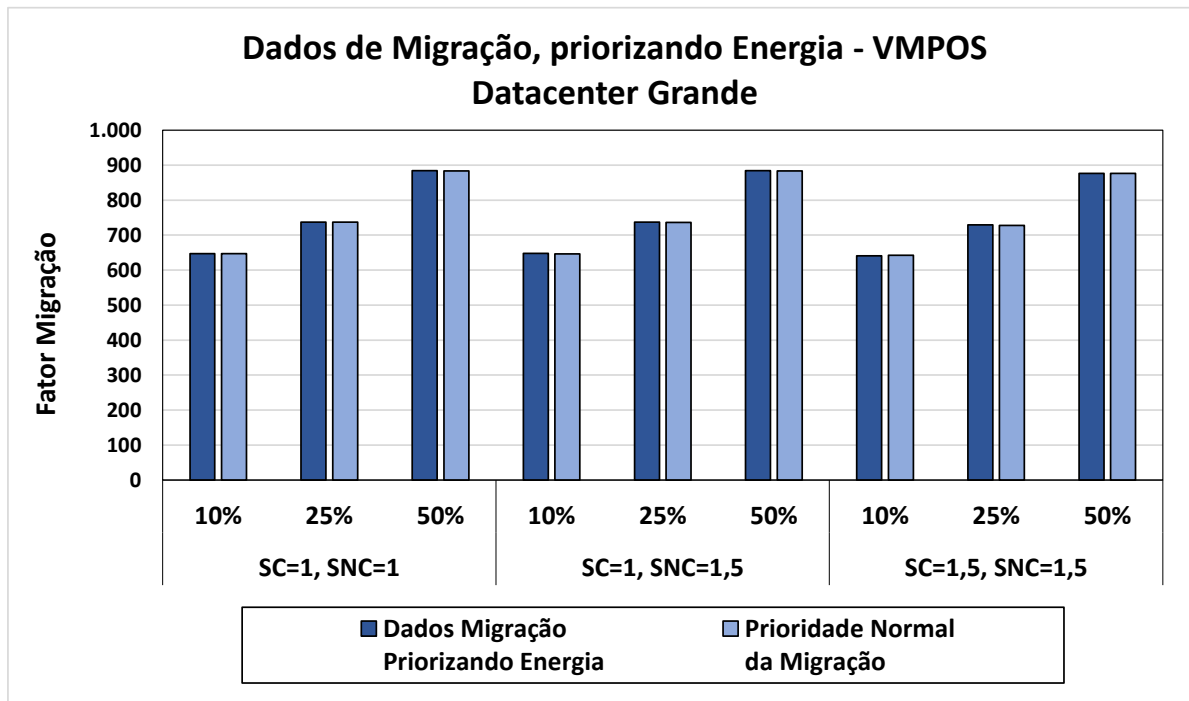
Fonte: Elaborada pelo autor.

Figura B.5: Resultado da Migração quando se aplicou prioridade para o objetivo Energia, em *Datacenter Médio*.



Fonte: Elaborada pelo autor.

Figura B.6: Resultado da Migração quando se aplicou prioridade para o objetivo Energia, em *Datacenter Grande*.

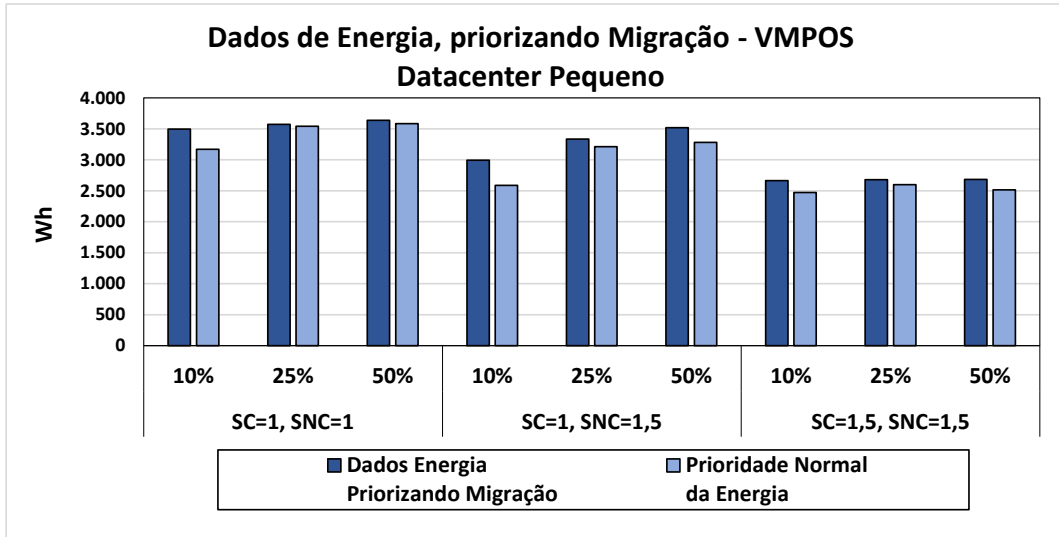


Fonte: Elaborada pelo autor.

B.2 Quando foi priorizado o objetivo Migração

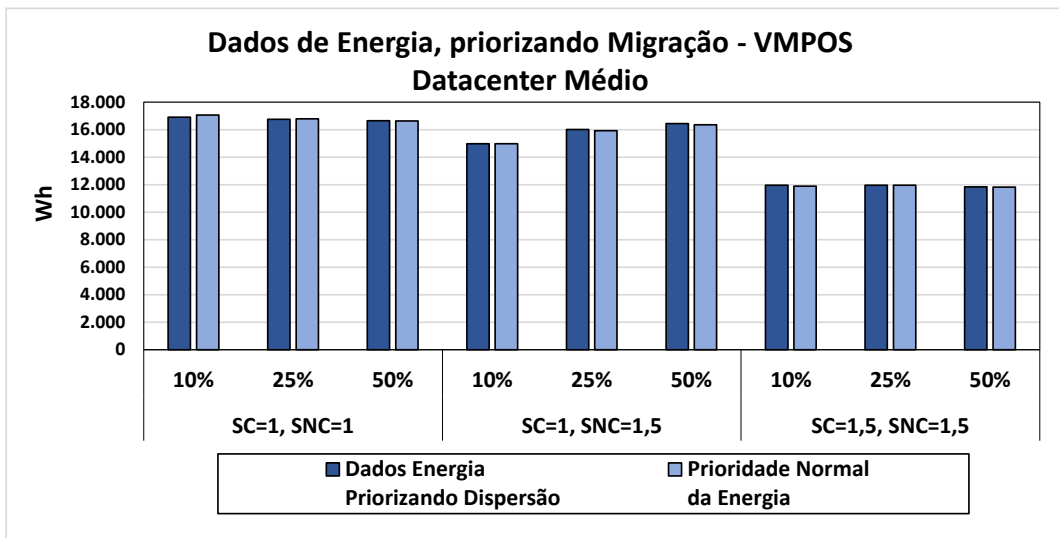
B.2.1 Efeito no objetivo Energia

Figura B.7: Resultado da Energia quando se aplicou prioridade para o objetivo Migração, em *Datacenter Pequeno*.



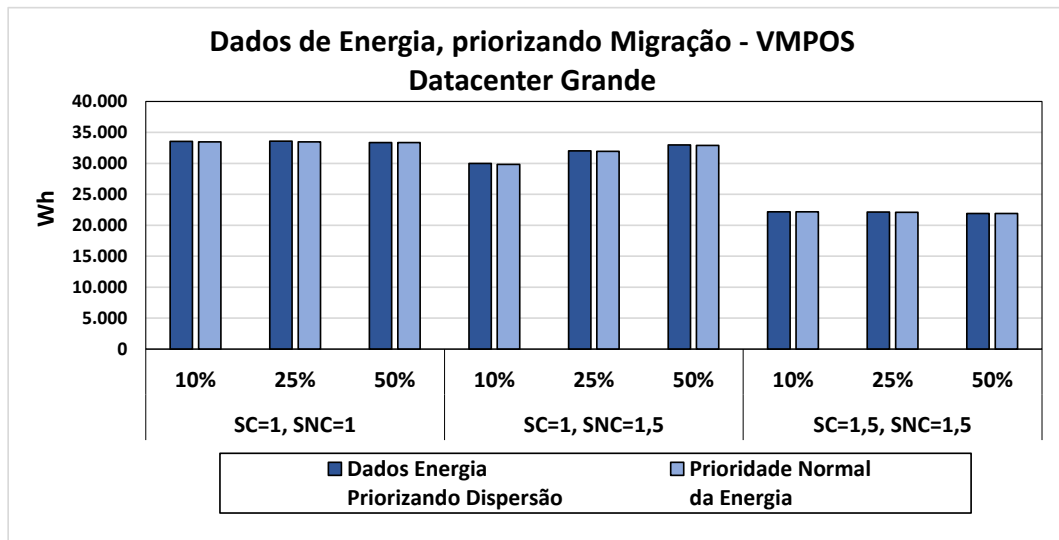
Fonte: Elaborada pelo autor.

Figura B.8: Resultado da Energia quando se aplicou prioridade para o objetivo Migração, em *Datacenter Médio*.



Fonte: Elaborada pelo autor.

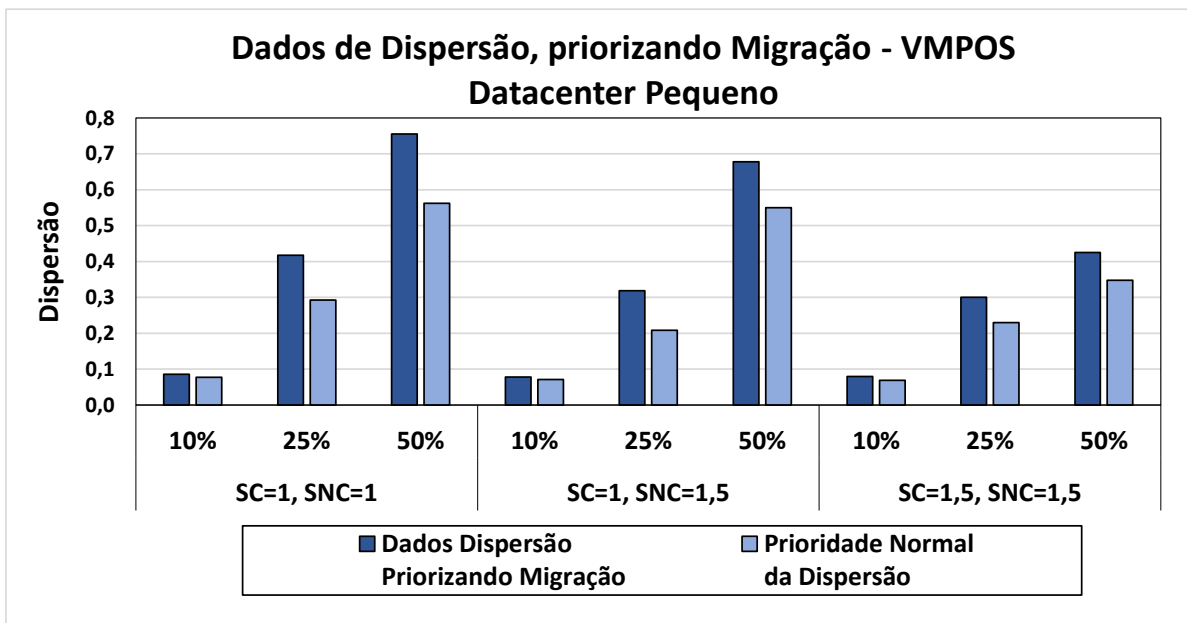
Figura B.9: Resultado da Energia quando se aplicou prioridade para o objetivo Migração, em *Datacenter Grande*.



Fonte: Elaborada pelo autor.

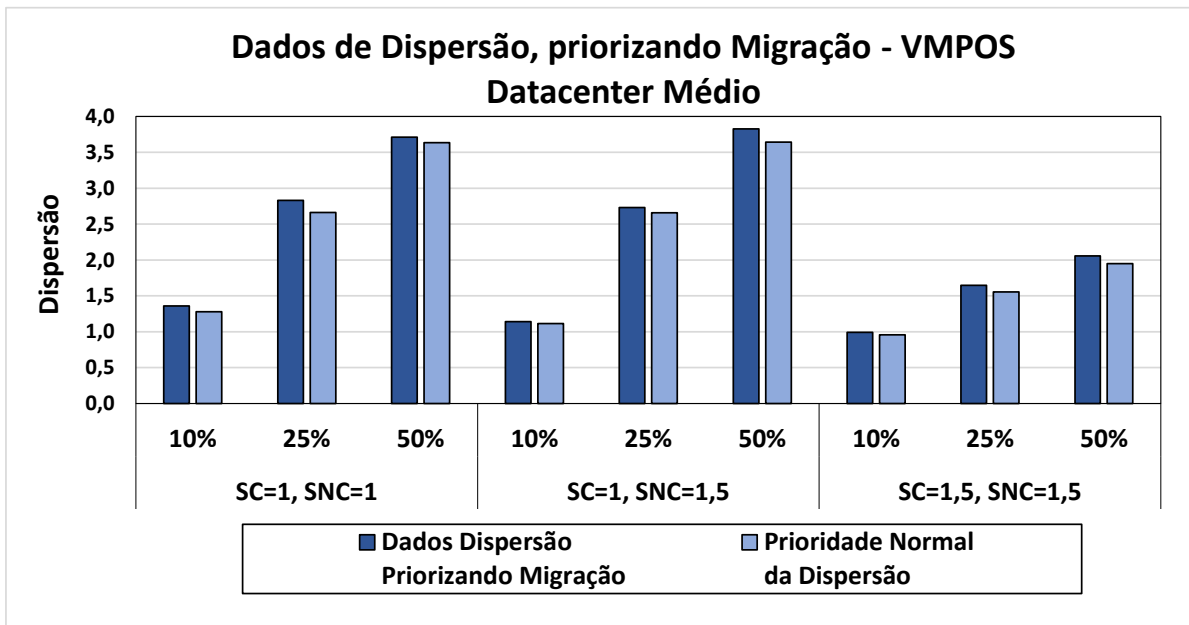
B.2.2 Efeito no objetivo Dispersão

Figura B.10: Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em *Datacenter Pequeno*.



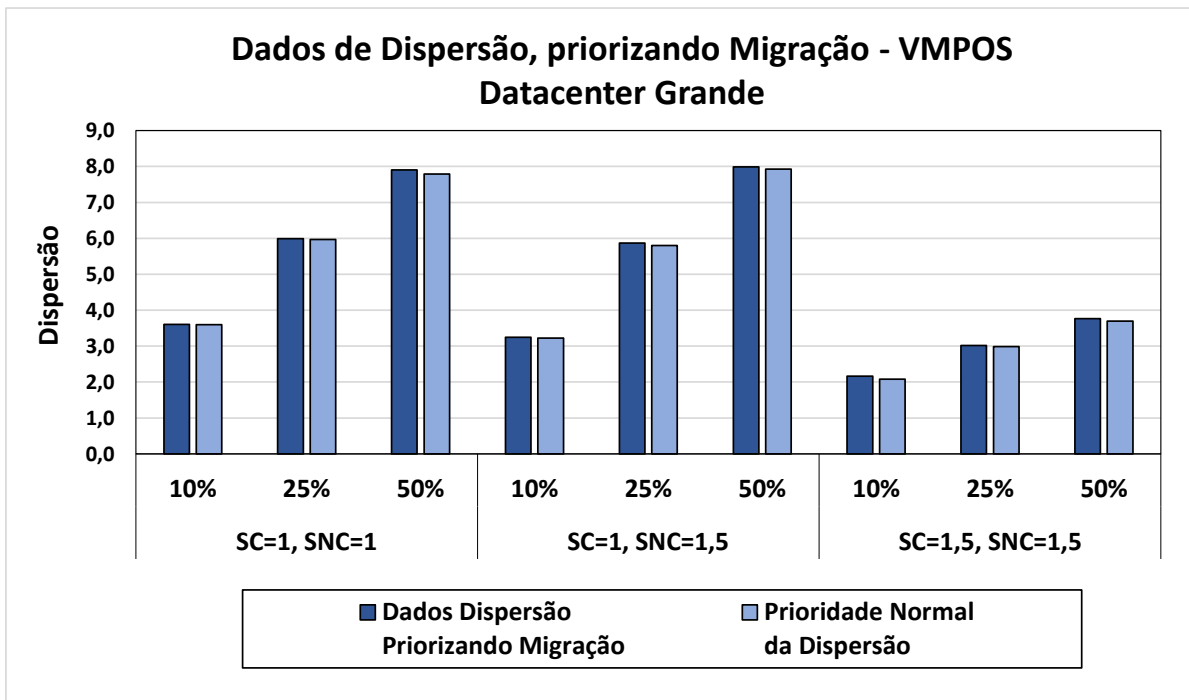
Fonte: Elaborada pelo autor.

Figura B.11: Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em *Datacenter Médio*.



Fonte: Elaborada pelo autor.

Figura B.12: Resultado da Dispersão quando se aplicou prioridade para o objetivo Migração, em *Datacenter Grande*.

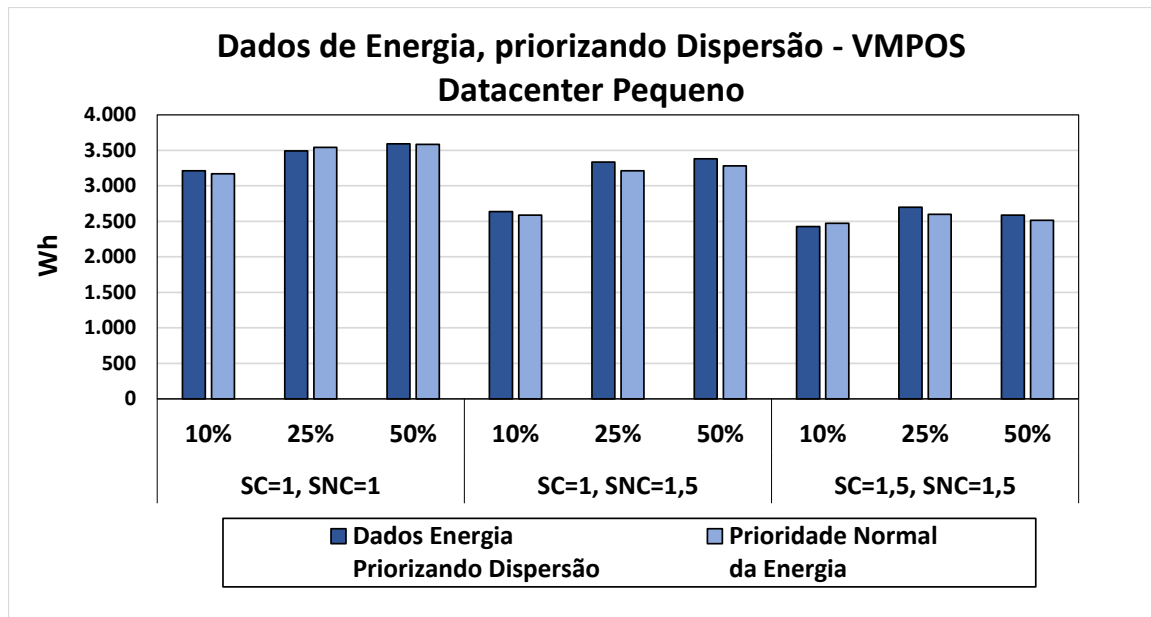


Fonte: Elaborada pelo autor.

B.3 Quando foi priorizado o objetivo Dispersão

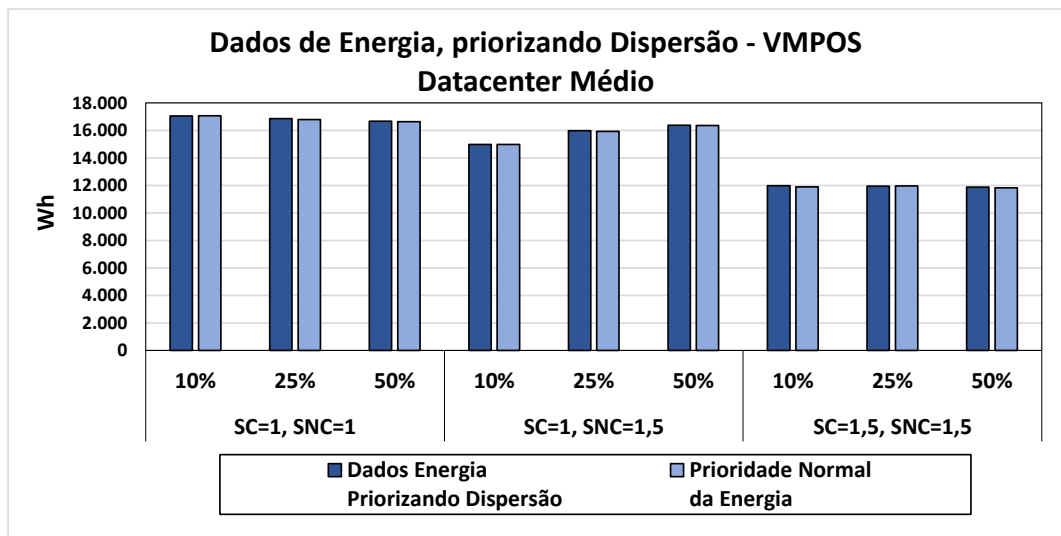
B.3.1 Efeito no objetivo Energia

Figura B.13: Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em *Datacenter Pequeno*.



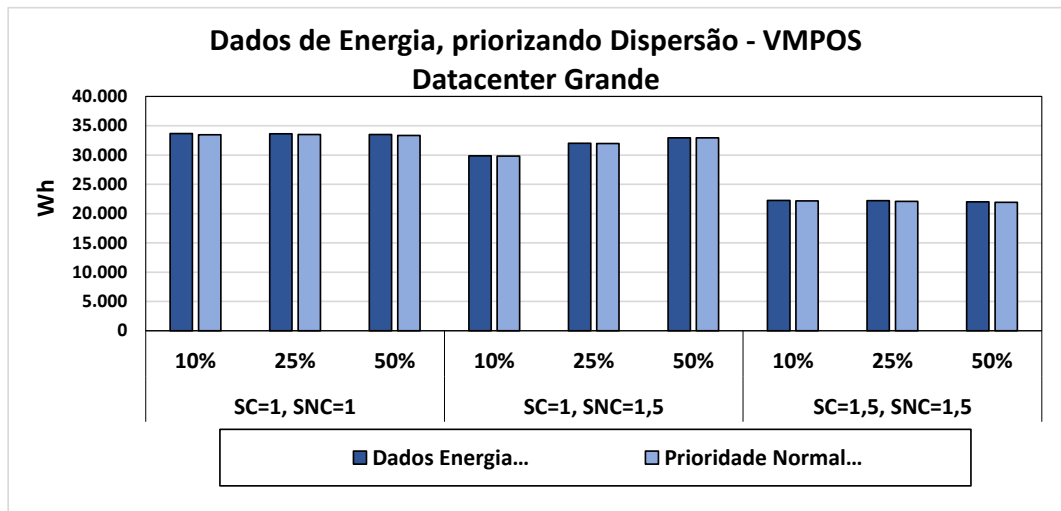
Fonte: Elaborada pelo autor.

Figura B.14: Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em *Datacenter Médio*.



Fonte: Elaborada pelo autor.

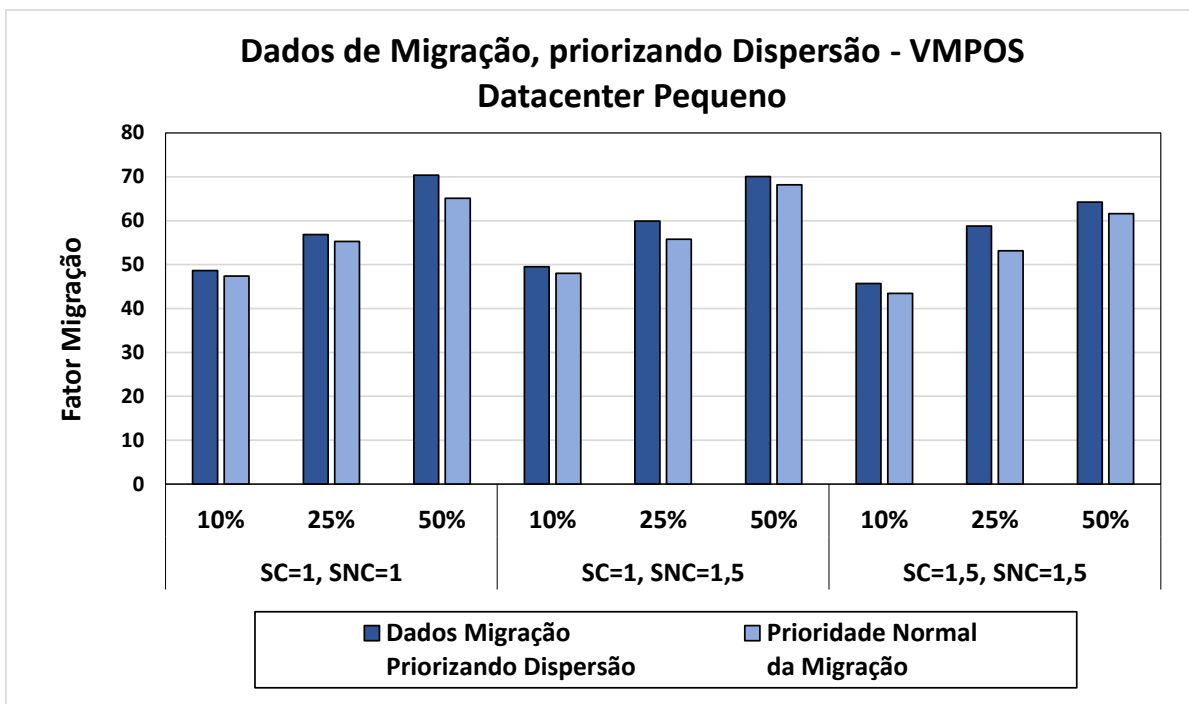
Figura B.15: Resultado da Energia quando se aplicou prioridade para o objetivo Dispersão, em *Datacenter Grande*.



Fonte: Elaborada pelo autor.

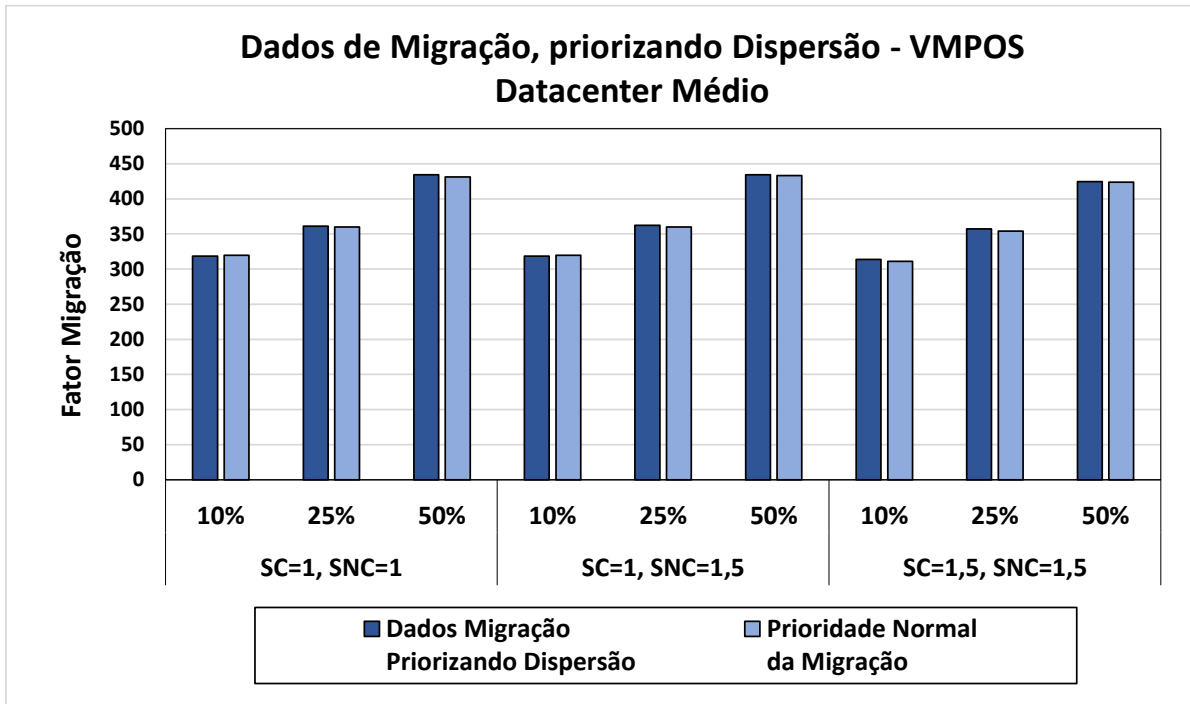
B.3.2 Efeito no objetivo Migração

Figura B.16: Resultado da Migração quando se aplicou prioridade para o objetivo Dispersão, em *Datacenter Pequeno*.



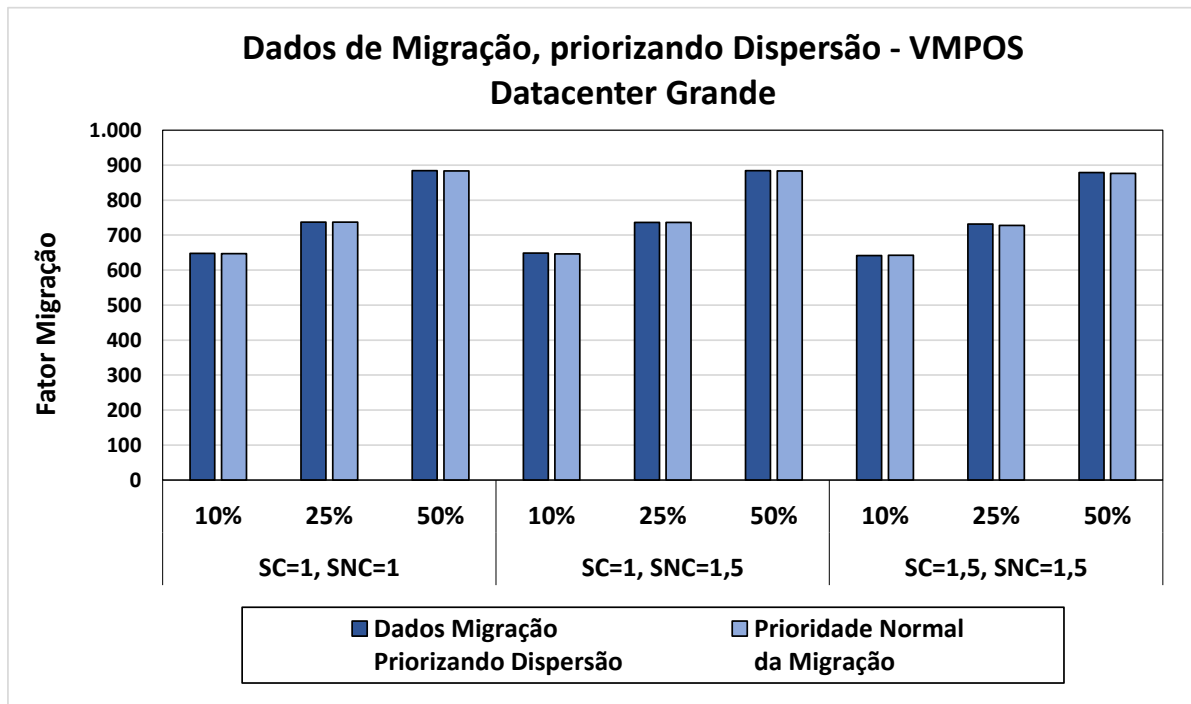
Fonte: Elaborada pelo autor.

Figura B.17: Resultado da Migração quando se aplicou prioridade para o objetivo Dispersão, em *Datacenter Médio*.



Fonte: Elaborada pelo autor.

Figura B.18: Resultado da Migração quando se aplicou prioridade para o objetivo Dispersão, em *Datacenter Grande*.



Fonte: Elaborada pelo autor.

Apêndice C

Testes de Normalidade

Para verificar se seria possível aplicar os testes estatísticos paramétricos, foi necessário testar se as amostras provinham de populações com distribuições normais.

Para isto, foi utilizado a estatística de *Shapiro-Wilk*, chamada de W , para todas as amostras coletadas. Com p -valor menor do que o nível de significância de 0,05 (5%) e com o parâmetro $W < 1$, se rejeita a hipótese nula. A hipótese nula atesta que a amostra provém de uma população Normal, com a hipótese alternativa indicando que não.

Os resultados de todas as configurações possíveis de *datacenter* (percentual de VMs críticas, diferentes níveis de comprometimento e diferentes tamanhos de *datacenter*) estão distribuídos nas tabelas a seguir.

C.1 Testes com os dados de Dispersão

Tabela C.1: Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1$, com os dados de Dispersão.

Tam / Modelo		Porcentagem VMs Críticas	W	p-valor	Conclusão
Pequeno	VMPOS	10%	0,76328	1,504e-05 < 0,05	Rejeita-se H_0
		25%	0,51276	7,328e-09 < 0,05	Rejeita-se H_0
		50%	0,66117	4,453e-07 < 0,05	Rejeita-se H_0
	RF	10%	0,81364	0,0001174 < 0,05	Rejeita-se H_0
		25%	0,94099	0,09669 > 0,05	Não rejeita-se H_0
		50%	0,96954	0,5268 > 0,05	Não rejeita-se H_0
Médio	VMPOS	10%	0,92272	0,03157 < 0,05	Rejeita-se H_0
		25%	0,96673	0,454 > 0,05	Não rejeita-se H_0
		50%	0,90733	0,01275 < 0,05	Rejeita-se H_0
	RF	10%	0,96517	0,4167 > 0,05	Não rejeita-se H_0
		25%	0,96386	0,3872 > 0,05	Não rejeita-se H_0
		50%	0,94248	0,1061 > 0,05	Não rejeita-se H_0
Grande	VMPOS	10%	0,97646	0,7256 > 0,05	Não rejeita-se H_0
		25%	0,94936	0,1625 > 0,05	Não rejeita-se H_0
		50%	0,96864	0,5027 > 0,05	Não rejeita-se H_0
	RF	10%	0,9593	0,2979 > 0,05	Não rejeita-se H_0
		25%	0,94471	0,1219 > 0,05	Não rejeita-se H_0
		50%	0,96186	0,3452 > 0,05	Não rejeita-se H_0

Tabela C.2: Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1,5$, com os dados de Dispersão.

Tam / Modelo		Porcentagem VMs Críticas	W	p-valor	Conclusão
Pequeno	VMPOS	10%	0,74897	$8,763e-06 < 0,05$	Rejeita-se H_0
		25%	0,7296	$4,335e-06 < 0,05$	Rejeita-se H_0
		50%	0,85871	$0,0009489 < 0,05$	Rejeita-se H_0
	RF	10%	0,88255	$0,003228 < 0,05$	Rejeita-se H_0
		25%	0,95164	$0,187 > 0,05$	Não rejeita-se H_0
		50%	0,96232	$0,3547 > 0,05$	Não rejeita-se H_0
Médio	VMPOS	10%	0,88312	$0,003327 < 0,05$	Rejeita-se H_0
		25%	0,93993	$0,09059 > 0,05$	Não rejeita-se H_0
		50%	0,95794	$0,2741 > 0,05$	Não rejeita-se H_0
	RF	10%	0,97497	$0,6817 > 0,05$	Não rejeita-se H_0
		25%	0,9381	$0,08085 > 0,05$	Não rejeita-se H_0
		50%	0,96202	$0,3485 > 0,05$	Não rejeita-se H_0
Grande	VMPOS	10%	0,97302	$0,6248 > 0,05$	Não rejeita-se H_0
		25%	0,96158	$0,3398 > 0,05$	Não rejeita-se H_0
		50%	0,93575	$0,06994 > 0,05$	Não rejeita-se H_0
	RF	10%	0,91414	$0,01895 < 0,05$	Rejeita-se H_0
		25%	0,90372	$0,01037 < 0,05$	Rejeita-se H_0
		50%	0,94707	$0,1411 > 0,05$	Não rejeita-se H_0

Tabela C.3: Resultados dos testes de normalidade, usando $SC = 1,5$ e $SNC = 1,5$, com os dados de Dispersão.

Tam / Modelo		Porcentagem VMs Críticas	W	p-valor	Conclusão
Pequeno	VMPOS	10%	0,59663	$6,64e-08 < 0,05$	Rejeita-se H_0
		25%	0,80171	$7,046e-05 < 0,05$	Rejeita-se H_0
		50%	0,90724	$0,01269 < 0,05$	Rejeita-se H_0
	RF	10%	0,87845	$0,002598 < 0,05$	Rejeita-se H_0
		25%	0,93918	$0,08644 > 0,05$	Não rejeita-se H_0
		50%	0,91004	$0,01492 < 0,05$	Rejeita-se H_0
Médio	VMPOS	10%	0,91621	$0,02142 < 0,05$	Rejeita-se H_0
		25%	0,90277	$0,009824 < 0,05$	Rejeita-se H_0
		50%	0,92153	$0,0294 < 0,05$	Rejeita-se H_0
	RF	10%	0,93913	$0,08617 > 0,05$	Não rejeita-se H_0
		25%	0,95496	$0,2291 > 0,05$	Não rejeita-se H_0
		50%	0,95864	$0,2859 > 0,05$	Não rejeita-se H_0
Grande	VMPOS	10%	0,95484	$0,2274 > 0,05$	Não rejeita-se H_0
		25%	0,94459	$0,121 > 0,05$	Não rejeita-se H_0
		50%	0,94505	$0,1244 > 0,05$	Não rejeita-se H_0
	RF	10%	0,93459	$0,0651 > 0,05$	Não rejeita-se H_0
		25%	0,93437	$0,06423 > 0,05$	Não rejeita-se H_0
		50%	0,87277	$0,001932 < 0,05$	Rejeita-se H_0

C.2 Testes com os dados de Energia

Tabela C.4: Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1$, com os dados de Energia.

Tam / Modelo		Porcentagem VMs Críticas	W	p-valor	Conclusão
Pequeno	VMPOS	10%	0,90232	0,009573 < 0,05	Rejeita-se H_0
		25%	0,79744	5,889e-05 < 0,05	Rejeita-se H_0
		50%	0,82399	0,0001853 < 0,05	Rejeita-se H_0
	RF	10%	0,95447	0,2224 > 0,05	Não rejeita-se H_0
		25%	0,78155	3,077e-05 < 0,05	Rejeita-se H_0
		50%	0,76455	1,579e-05 < 0,05	Rejeita-se H_0
	iMaVMP	10%	0,88259	0,003235 < 0,05	Rejeita-se H_0
		25%	0,82737	0,0002157 < 0,05	Rejeita-se H_0
		50%	0,84305	0,000445 < 0,05	Rejeita-se H_0
Médio	VMPOS	10%	0,85304	0,0007184 < 0,05	Rejeita-se H_0
		25%	0,95086	0,1783 > 0,05	Não rejeita-se H_0
		50%	0,89708	0,007131 < 0,05	Rejeita-se H_0
	RF	10%	0,8671	0,001445 < 0,05	Rejeita-se H_0
		25%	0,98813	0,978 > 0,05	Não rejeita-se H_0
		50%	0,97784	0,7656 > 0,05	Não rejeita-se H_0
	iMaVMP	10%	0,88127	0,003016 < 0,05	Rejeita-se H_0
		25%	0,87585	0,002267 < 0,05	Rejeita-se H_0
		50%	0,82072	0,0001602 < 0,05	Rejeita-se H_0
Grande	VMPOS	10%	0,95083	0,178 > 0,05	Não rejeita-se H_0
		25%	0,97746	0,7547 > 0,05	Não rejeita-se H_0
		50%	0,96228	0,3538 > 0,05	Não rejeita-se H_0
	RF	10%	0,86575	0,00135 < 0,05	Rejeita-se H_0
		25%	0,96299	0,3685 > 0,05	Não rejeita-se H_0
		50%	0,93427	0,06383 > 0,05	Não rejeita-se H_0
	iMaVMP	10%	0,815	0,0001246 < 0,05	Rejeita-se H_0
		25%	0,86966	0,001647 < 0,05	Rejeita-se H_0
		50%	0,69994	1,557e-06 < 0,05	Rejeita-se H_0

Tabela C.5: Resultados dos testes de normalidade, usando $SC = 1$ e $SNC = 1,5$, com os dados de Energia.

Tam / Modelo	Porcentagem VMs Críticas	W	p-valor	Conclusão	
Pequeno	VMPOS	10%	0,8843	$0,003545 < 0,05$	Rejeita-se H_0
		25%	0,79322	$4,946e-05 < 0,05$	Rejeita-se H_0
		50%	0,87476	$0,002142 < 0,05$	Rejeita-se H_0
	RF	10%	0,86024	$0,001023 < 0,05$	Rejeita-se H_0
		25%	0,97638	$0,7232 > 0,05$	Não rejeita-se H_0
		50%	0,96919	$0,5173 > 0,05$	Não rejeita-se H_0
Médio	VMPOS	10%	0,95397	$0,2157 > 0,05$	Não rejeita-se H_0
		25%	0,96213	$0,3507 > 0,05$	Não rejeita-se H_0
		50%	0,98201	$0,876 > 0,05$	Não rejeita-se H_0
	RF	10%	0,98423	$0,9233 > 0,05$	Não rejeita-se H_0
		25%	0,91031	$0,01515 < 0,05$	Rejeita-se H_0
		50%	0,97941	$0,8095 > 0,05$	Não rejeita-se H_0
Grande	VMPOS	10%	0,96642	$0,4464 > 0,05$	Não rejeita-se H_0
		25%	0,97492	$0,6803 > 0,05$	Não rejeita-se H_0
		50%	0,94602	$0,1322 > 0,05$	Não rejeita-se H_0
	RF	10%	0,96783	$0,4817 > 0,05$	Não rejeita-se H_0
		25%	0,92517	$0,0366 < 0,05$	Rejeita-se H_0
		50%	0,91049	$0,01531 < 0,05$	Rejeita-se H_0

Tabela C.6: Resultados dos testes de normalidade, usando $SC = 1,5$ e $SNC = 1,5$, com os dados de Energia.

Tam / Modelo		Porcentagem VMs Críticas	W	p-valor	Conclusão
Pequeno	VMPOS	10%	0,76113	$1,385e-05 < 0,05$	Rejeita-se H_0
		25%	0,77511	$2,382e-05 < 0,05$	Rejeita-se H_0
		50%	0,85898	$0,0009618 < 0,05$	Rejeita-se H_0
	RF	10%	0,97452	$0,6686 > 0,05$	Não rejeita-se H_0
		25%	0,86802	$0,86802 > 0,05$	Não rejeita-se H_0
		50%	0,91237	$0,01708 < 0,05$	Rejeita-se H_0
Médio	VMPOS	10%	0,90283	$0,009858 < 0,05$	Rejeita-se H_0
		25%	0,96601	$0,4365 > 0,05$	Não rejeita-se H_0
		50%	0,84542	$0,0004979 < 0,05$	Rejeita-se H_0
	RF	10%	0,96666	$0,4523 > 0,05$	Não rejeita-se H_0
		25%	0,9051	$0,01122 < 0,05$	Rejeita-se H_0
		50%	0,8977	$0,007382 < 0,05$	Rejeita-se H_0
Grande	VMPOS	10%	0,9283	$0,04427 < 0,05$	Rejeita-se H_0
		25%	0,93169	$0,05444 > 0,05$	Não rejeita-se H_0
		50%	0,90445	$0,01081 < 0,05$	Rejeita-se H_0
	RF	10%	0,96512	$0,4156 > 0,05$	Não rejeita-se H_0
		25%	0,97067	$0,97067 > 0,05$	Não rejeita-se H_0
		50%	0,98809	$0,9777 > 0,05$	Não rejeita-se H_0

Apêndice D

Testes de *Kruskal*

D.1 Testes com os dados de Energia

A Tabela D.1 contém os resultados dos testes estatísticos de Kruskal em busca de diferenças entre as quatro abordagens utilizadas neste trabalho. Um p-valor menor que 0,05 indica que pelo uma das abordagens provêm de população diferente das demais.

Tabela D.1: Resultados dos testes de Kruskal com os dados de Energia considerando todas as abordagens utilizadas.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	χ^2 (df)	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	$\chi^2(3) = 103,71$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 105,62$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,87$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	$\chi^2(3) = 102,52$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 102,05$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,67$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$\chi^2(3) = 102,41$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 102,09$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,71$	$2,2e-16 < 0,05$	Rejeita-se H_0
Médio	SC = 1,0 SNC = 1,0	10%	$\chi^2(3) = 111,88$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 109,26$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(2) = 109,26$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	$\chi^2(3) = 102,09$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 102,25$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 100,33$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$\chi^2(3) = 103,12$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 102,02$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,01$	$2,2e-16 < 0,05$	Rejeita-se H_0
Grande	SC = 1,0 SNC = 1,0	10%	$\chi^2(3) = 113,34$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 109,26$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 111,88$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	$\chi^2(3) = 102,57$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 103,24$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,04$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$\chi^2(3) = 103,02$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 102,15$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,02$	$2,2e-16 < 0,05$	Rejeita-se H_0

D.2 Testes com os dados de Dispersão

Tabela D.2: Resultados dos testes de Kruskal com os dados de Dispersão considerando todas as abordagens utilizadas.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	χ^2 (df)	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	$\chi^2(3) = 107,01$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 104,31$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 105,29$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	$\chi^2(3) = 102,99$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 105,03$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,82$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$\chi^2(3) = 102,14$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 104,9$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 103,07$	$2,2e-16 < 0,05$	Rejeita-se H_0
Médio	SC = 1,0 SNC = 1,0	10%	$\chi^2(3) = 105,19$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 104,7$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 104,04$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	$\chi^2(3) = 105,49$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 103,2$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 105,55$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$\chi^2(3) = 105,02$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 103,65$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 104,93$	$2,2e-16 < 0,05$	Rejeita-se H_0
Grande	SC = 1,0 SNC = 1,0	10%	$\chi^2(3) = 102,95$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 106,07$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 102,73$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	$\chi^2(3) = 103,74$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 88,501$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 96,675$	$2,2e-16 < 0,05$	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$\chi^2(3) = 106,19$	$2,2e-16 < 0,05$	Rejeita-se H_0
		25%	$\chi^2(3) = 103,54$	$2,2e-16 < 0,05$	Rejeita-se H_0
		50%	$\chi^2(3) = 103,02$	$2,2e-16 < 0,05$	Rejeita-se H_0

Apêndice E

Testes de *Nemenyi*

E.1 Testes com os dados de Energia

Tabela E.1: Resultados do *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *data-center* Pequeno, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas								
	10%			25%			50%		
	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit
iMaVMP	4.1e-07	-	-	1.3e-08	-	-	3.6e-09	-	-
Random Fit	0.00023	0.63812	-	0.00099	0.16666	-	0.00038	0.17644	-
VMPOS	0.01005	< 2e-16	5.0e-13	0.01967	< 2e-16	3.8e-11	0.06422	< 2e-16	1.6e-10

Tabela E.2: Resultados do *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *data-center* Médio, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas								
	10%			25%			50%		
	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit
iMaVMP	< 2e-16	-	-	< 2e-16	-	-	< 2e-16	-	-
Random Fit	1.6e-09	0.010	-	6.8e-09	0.0100	-	6.8e-09	0.0100	-
VMPOS	0.020	3.7e-10	0.007	0.0642	8.1e-11	0.0033	0.0642	8.1e-11	0.0033

Tabela E.3: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *datacenter* Grande, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas									
	10%			25%			50%		
	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit	First Fit	iMaVMP	Random Fit
iMaVMP	< 2e-16	-	-	< 2e-16	-	-	< 2e-16	-	-
Random Fit	7.8e-10	0.01	-	6.8e-09	0.0100	-	1.6e-09	0.010	-
VMPOS	0.01	7.8e-10	0.01	0.0642	8.0e-11	0.0033	0.020	3.7e-10	0.007

Tabela E.4: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *datacenter* Pequeno, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01	-	0.01	-	0.01	-
VMPOS	2.0e-06	< 2e-16	7.2e-06	1.2e-15	1.5e-06	< 2e-16

Tabela E.5: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *datacenter* Médio, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01	-	0.01	-	0.00704	-
VMPOS	6.0e-06	8.9e-16	3.6e-06	3.3e-16	4.4e-06	2.2e-16

Tabela E.6: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *datacenter* Grande, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
10%		25%		50%		
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01	-	0.01005	-	0.01	-
VMPOS	1.8e-06	< 2e-16	7.0e-07	< 2e-16	1.8e-05	5.8e-15

Tabela E.7: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *datacenter* Pequeno, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
10%		25%		50%		
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01	-	0.01	-	0.01	-
VMPOS	2.5e-06	2.2e-16	6.0e-06	8.9e-16	1.4e-06	< 2e-16

Tabela E.8: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *datacenter* Médio, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
10%		25%		50%		
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01005	-	0.01	-	0.01	-
VMPOS	8.1e-07	< 2e-16	1.6e-05	4.3e-15	1.1e-05	2.6e-15

Tabela E.9: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Energia, no *datacenter* Grande, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas					
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01005	-	0.01	-	0.01	-
VMPOS	0.00012	1.6e-13	4.8e-06	5.6e-16	9.6e-06	1.9e-15

E.2 Testes com os dados de Dispersão

Tabela E.10: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Pequeno, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas					
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.00864	-	0.032	-	0.00930	-
VMPOS	1.9e-07	< 2e-16	6.4e-08	< 2e-16	3.1e-07	< 2e-16

Tabela E.11: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Médio, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas					
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.00996	-	0.00998	-	0.01001	-
VMPOS	1.2e-07	< 2e-16	1.8e-07	< 2e-16	3.1e-07	< 2e-16

Tabela E.12: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Grande, com SC=1 e SNC=1, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
10%		25%		50%		
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01	-	0.00999	-	< 2e-16	-
VMPOS	1.2e-07	< 2e-16	5.7e-08	< 2e-16	0.00492	1.3e-08

Tabela E.13: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Pequeno, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
10%		25%		50%		
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.032	-	0.00964	-	0.0099	-
VMPOS	3.4e-07	2.2e-16	2.2e-07	< 2e-16	1.8e-06	< 2e-16

Tabela E.14: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Médio, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas						
10%		25%		50%		
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.00998	-	0.01002	-	0.01001	-
VMPOS	9.0e-08	< 2e-16	7.8e-07	< 2e-16	8.3e-08	< 2e-16

Tabela E.15: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Grande, com SC=1 e SNC=1,5, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas					
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.01003	-	3.5e-05	-	0.0015	-
VMPOS	4.1e-07	< 2e-16	0.00022	< 2e-16	3.0e-06	< 2e-16

Tabela E.16: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Pequeno, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas					
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.057	-	0.0089	-	0.0097	-
VMPOS	1.1e-07	3.3e-16	1.2e-06	< 2e-16	2.1e-06	< 2e-16

Tabela E.17: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Médio, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.

	Percentual de VMs críticas					
	10%		25%		50%	
	First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit	0.00948	-	0.0095	-	0.00997	-
VMPOS	2.9e-07	< 2e-16	1.9e-06	< 2e-16	1.4e-07	< 2e-16

Tabela E.18: Resultados da função estatística *posthoc.kruskal.nemenyi.test*, com os dados de Dispersão, no *datacenter* Grande, com SC=1,5 e SNC=1,5, para diferentes percentuais de VMs críticas.

Percentual de VMs críticas							
		10%		25%		50%	
		First Fit	Random Fit	First Fit	Random Fit	First Fit	Random Fit
Random Fit		0.00992	-	0.00992	-	0.0099	-
VMPOS		5.7e-08	< 2e-16	6.3e-07	< 2e-16	1.2e-06	< 2e-16

Apêndice F

Testes Estatísticos das Diferenças

Nas tabelas abaixo temos os testes em busca de diferenças significativas entre as amostras sendo comparadas. Foi aplicado o *teste unilateral* de Wilcoxon.

F.1 Testes unilaterais com os dados de Energia

Tabela F.1: Resultados dos testes unilaterais de Wilcoxon, com os dados de Energia, entre o VMPOS e o *First Fit*,

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	900	6,044e-13 < 0,05	VMPOS é melhor
		25%	870	1,676e-11 < 0,05	VMPOS é melhor
		50%	810	6,653e-09 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	6,049e-13 < 0,05	VMPOS é melhor
		25%	900	6,059e-13 < 0,05	VMPOS é melhor
		50%	900	6,049e-13 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	5,97e-13 < 0,05	VMPOS é melhor
		25%	900	6,029e-13 < 0,05	VMPOS é melhor
		50%	900	6,034e-13 < 0,05	VMPOS é melhor
Médio	SC = 1,0 SNC = 1,0	10%	30	1 > 0,05	ND
		25%	90	1 > 0,05	ND
		50%	90	1 > 0,05	ND
	SC = 1,0 SNC = 1,5	10%	900	6,059e-13 < 0,05	VMPOS é melhor
		25%	900	6,059e-13 < 0,05	VMPOS é melhor
		50%	900	6,059e-13 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	6,054e-13 < 0,05	VMPOS é melhor
		25%	900	6,059e-13 < 0,05	VMPOS é melhor
		50%	900	6,059e-13 < 0,05	VMPOS é melhor
Grande	SC = 1,0 SNC = 1,0	10%	0	1 > 0,05	ND
		25%	90	1 > 0,05	ND
		50%	30	1 > 0,05	ND
	SC = 1,0 SNC = 1,5	10%	900	6,059e-13 < 0,05	VMPOS é melhor
		25%	900	6,059e-13 < 0,05	VMPOS é melhor
		50%	900	6,059e-13 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	6,054e-13 < 0,05	VMPOS é melhor
		25%	900	6,054e-13 < 0,05	VMPOS é melhor
		50%	900	6,049e-13 < 0,05	VMPOS é melhor

Tabela F.2: Resultados dos testes unilaterais de Wilcoxon, com os dados de Energia, entre o VMPOS e o *Random Fit*.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	900	1,489e-11 < 0,05	VMPOS é melhor
		25%	900	1,498e-11 < 0,05	VMPOS é melhor
		50%	900	1,495e-11 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	1,503e-11 < 0,05	VMPOS é melhor
		25%	900	1,507e-11 < 0,05	VMPOS é melhor
		50%	900	1,505e-11 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	1,49e-11 < 0,05	VMPOS é melhor
		25%	900	1,501e-11 < 0,05	VMPOS é melhor
		50%	900	1,502e-11 < 0,05	VMPOS é melhor
Médio	SC = 1,0 SNC = 1,0	10%	900	1,505e-11 < 0,05	VMPOS é melhor
		25%	900	1,504e-11 < 0,05	VMPOS é melhor
		50%	900	1,506e-11 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	1,509e-11 < 0,05	VMPOS é melhor
		25%	900	1,507e-11 < 0,05	VMPOS é melhor
		50%	900	1,509e-11 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	1,506e-11 < 0,05	VMPOS é melhor
		25%	900	1,508e-11 < 0,05	VMPOS é melhor
		50%	900	1,505e-11 < 0,05	VMPOS é melhor
Grande	SC = 1,0 SNC = 1,0	10%	900	1,501e-11 < 0,05	VMPOS é melhor
		25%	900	1,498e-11 < 0,05	VMPOS é melhor
		50%	900	1,498e-11 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	1,502e-11 < 0,05	VMPOS é melhor
		25%	900	1,487e-11 < 0,05	VMPOS é melhor
		50%	900	1,491e-11 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	1,5e-11 < 0,05	VMPOS é melhor
		25%	900	1,504e-11 < 0,05	VMPOS é melhor
		50%	900	1,502e-11 < 0,05	VMPOS é melhor

Tabela F.3: Resultados dos testes unilaterais de Wilcoxon, com os dados de Energia, entre o VMPOS e o *iMaVMP*.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	900	1,507e-11 < 0,05	VMPOS é melhor
		25%	900	1,507e-11 < 0,05	VMPOS é melhor
		50%	900	1,504e-11 < 0,05	VMPOS é melhor
Médio	SC = 1,0 SNC = 1,0	10%	900	1,509e-11 < 0,05	VMPOS é melhor
		25%	900	1,507e-11 < 0,05	VMPOS é melhor
		50%	900	1,51e-11 < 0,05	VMPOS é melhor
Grande	SC = 1,0 SNC = 1,0	10%	900	1,51e-11 < 0,05	VMPOS é melhor
		25%	900	1,509e-11 < 0,05	VMPOS é melhor
		50%	900	1,51e-11 < 0,05	VMPOS é melhor

F.2 Testes unilaterais com os dados de Dispersão

Tabela F.4: Resultados dos testes unilaterais de Wilcoxon, com os dados de Dispersão, entre o VMPOS e o *First Fit*.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	900	2,404e-13 < 0,05	VMPOS é melhor
		25%	900	1,028e-13 < 0,05	VMPOS é melhor
		50%	900	2,982e-13 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	2,525e-13 < 0,05	VMPOS é melhor
		25%	900	4,739e-13 < 0,05	VMPOS é melhor
		50%	900	5,252e-13 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	1,683e-13 < 0,05	VMPOS é melhor
		25%	900	4,084e-13 < 0,05	VMPOS é melhor
		50%	900	5,248e-13 < 0,05	VMPOS é melhor
Médio	SC = 1,0 SNC = 1,0	10%	900	5,601e-13 < 0,05	VMPOS é melhor
		25%	900	5,592e-13 < 0,05	VMPOS é melhor
		50%	900	5,371e-13 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	5,684e-13 < 0,05	VMPOS é melhor
		25%	900	5,874e-13 < 0,05	VMPOS é melhor
		50%	900	5,698e-13 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	4,767e-13 < 0,05	VMPOS é melhor
		25%	900	4,437e-13 < 0,05	VMPOS é melhor
		50%	900	5,624e-13 < 0,05	VMPOS é melhor
Grande	SC = 1,0 SNC = 1,0	10%	900	5,722e-13 < 0,05	VMPOS é melhor
		25%	900	5,55e-13 < 0,05	VMPOS é melhor
		50%	60	1 > 0,05	ND
	SC = 1,0 SNC = 1,5	10%	900	5,736e-13 < 0,05	VMPOS é melhor
		25%	870	1,624e-11 < 0,05	VMPOS é melhor
		50%	900	5,874e-13 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	5,455e-13 < 0,05	VMPOS é melhor
		25%	900	5,624e-13 < 0,05	VMPOS é melhor
		50%	900	5,628e-13 < 0,05	VMPOS é melhor

Tabela F.5: Resultados dos testes unilaterais de Wilcoxon, com os dados de Dispersão, entre o VMPOS e o *Random Fit*.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	900	4,876e-12 < 0,05	VMPOS é melhor
		25%	900	3,721e-12 < 0,05	VMPOS é melhor
		50%	900	8,301e-12 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	6,206e-12 < 0,05	VMPOS é melhor
		25%	900	1,161e-11 < 0,05	VMPOS é melhor
		50%	900	1,319e-11 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	4,474e-12 < 0,05	VMPOS é melhor
		25%	900	9,646e-12 < 0,05	VMPOS é melhor
		50%	900	1,306e-11 < 0,05	VMPOS é melhor
Médio	SC = 1,0 SNC = 1,0	10%	900	1,398e-11 < 0,05	VMPOS é melhor
		25%	900	1,409e-11 < 0,05	VMPOS é melhor
		50%	900	1,368e-11 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	1,422e-11 < 0,05	VMPOS é melhor
		25%	900	1,449e-11 < 0,05	VMPOS é melhor
		50%	900	1,437e-11 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	1,22e-11 < 0,05	VMPOS é melhor
		25%	900	1,184e-11 < 0,05	VMPOS é melhor
		50%	900	1,415e-11 < 0,05	VMPOS é melhor
Grande	SC = 1,0 SNC = 1,0	10%	900	1,43e-11 < 0,05	VMPOS é melhor
		25%	900	1,401e-11 < 0,05	VMPOS é melhor
		50%	900	1,465e-11 < 0,05	VMPOS é melhor
	SC = 1,0 SNC = 1,5	10%	900	1,423e-11 < 0,05	VMPOS é melhor
		25%	900	1,427e-11 < 0,05	VMPOS é melhor
		50%	900	1,44e-11 < 0,05	VMPOS é melhor
	SC = 1,5 SNC = 1,5	10%	900	1,386e-11 < 0,05	VMPOS é melhor
		25%	900	1,419e-11 < 0,05	VMPOS é melhor
		50%	900	1,421e-11 < 0,05	VMPOS é melhor

Apêndice G

Estatísticas VMPOS x VMPOS

Este capítulo contém as tabelas das comparativas dos experimentos do VMPOS x VMPOS, verificando a existência de diferenças entre as amostras, usando o *teste bilateral*, e depois aplicando os *testes unilaterais* quando cada um dos três objetivos utilizados era favorecido individualmente, visando apontar qual das abordagens apresentou melhor resultados.

G.1 Testes bilaterais com os dados de Energia

Tabela G.1: Resultados dos testes estatísticos bilaterais de Wilcoxon, com os dados de Energia, nos comparativos VMPOS x VMPOS, amostras com prioridade para Energia contra as amostras sem atribuir prioridade alguma.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	p-valor	Intervalo de Confiança (95%)	Resultado
Pequeno	SC = 1 SNC = 1	10%	$0,01 < 0,05$	(-289,84 ; -18,83)	Rejeita-se H_0
		25%	$0,5e-8 < 0,05$	(-426,8 ; -328,7)	Rejeita-se H_0
		50%	$0,7e-10 < 0,05$	(-386,3 ; -304,8)	Rejeita-se H_0
	SC = 1 SNC = 1,5	10%	$0,02 < 0,05$	(-297,61 ; -9,23)	Rejeita-se H_0
		25%	$0,0002 < 0,05$	(-325,2 ; -182,0)	Rejeita-se H_0
		50%	$0,0004 < 0,05$	(-320,67 ; -70,67)	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$0,3e-5 < 0,05$	(-105,78 ; -41,44)	Rejeita-se H_0
		25%	$0,00002 < 0,05$	(-359,9 ; -284,4)	Rejeita-se H_0
		50%	$0,00004 < 0,05$	(-332,89 ; -56,22)	Rejeita-se H_0
Médio	SC = 1 SNC = 1	10%	$0,9e-7 < 0,05$	(-552,0 ; -324,2)	Rejeita-se H_0
		25%	$0,0001 < 0,05$	(-334,8 ; -101,0)	Rejeita-se H_0
		50%	$0,0003 < 0,05$	(-145,9 ; -47,5)	Rejeita-se H_0
	SC = 1 SNC = 1,5	10%	$0,00007 < 0,05$	(-188,5 ; -72,5)	Rejeita-se H_0
		25%	$0,007 < 0,05$	(-200,2 ; -32,7)	Rejeita-se H_0
		50%	$0,01 < 0,05$	(-129,5 ; -22,0)	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$0,1 > 0,05$	(-119,4 ; 8,1)	Não rejeita-se H_0
		25%	$0,1 > 0,05$	(-68,5 ; 5,9)	Não rejeita-se H_0
		50%	$0,00003 < 0,05$	(-217,0 ; -37,3)	Rejeita-se H_0
Grande	SC = 1 SNC = 1	10%	$0,9 > 0,05$	(-87,7 ; 88,8)	Não rejeita-se H_0
		25%	$0,08 > 0,05$	(-160,7 ; 8,9)	Não rejeita-se H_0
		50%	$0,01 < 0,05$	(-199,8 ; -26,0)	Rejeita-se H_0
	SC = 1 SNC = 1,5	10%	$0,2 > 0,05$	(-223,8 ; 45,2)	Não rejeita-se H_0
		25%	$0,5 > 0,05$	(-124,3 ; 53,3)	Não rejeita-se H_0
		50%	$0,001 < 0,05$	(-266,0 ; -65,5)	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	$0,008 < 0,05$	(-314,6 ; -31,2)	Rejeita-se H_0
		25%	$0,0005 < 0,05$	(-311,5 ; -58,0)	Rejeita-se H_0
		50%	$0,00009 < 0,05$	(-258,7 ; -61,8)	Rejeita-se H_0

G.2 Testes unilaterais com os dados de Energia

Tabela G.2: Resultados dos testes estatísticos unilaterais de Wilcoxon, com os dados de Energia, nos comparativos VMPOS x VMPOS, amostras com prioridade para Energia contra as amostras sem atribuir prioridade alguma.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	283,5	0,007 < 0,05	PE é melhor
		25%	53,5	2,379e-09 < 0,05	PE é melhor
		50%	8,5	3,494e-11 < 0,05	PE é melhor
	SC = 1,0 SNC = 1,5	10%	289	0,0088 < 0,05	PE é melhor
		25%	201	0,0001 < 0,05	PE é melhor
		50%	209	0,0001 < 0,05	PE é melhor
	SC = 1,5 SNC = 1,5	10%	132	1,323e-06 < 0,05	PE é melhor
		25%	163,5	1,172e-05 < 0,05	PE é melhor
		50%	173,5	2,241e-05 < 0,05	PE é melhor
Médio	SC = 1,0 SNC = 1,0	10%	87,5	4,347e-08 < 0,05	PE é melhor
		25%	186,5	5,044e-05 < 0,05	PE é melhor
		50%	207,5	0,0001 < 0,05	PE é melhor
	SC = 1,0 SNC = 1,5	10%	181	3,599e-05 < 0,05	PE é melhor
		25%	266	0,0033 < 0,05	PE é melhor
		50%	275,5	0,005 < 0,05	PE é melhor
	SC = 1,5 SNC = 1,5	10%	348	0,066 > 0,05	ND
		25%	344	0,059 > 0,05	ND
		50%	167	1,477e-05 < 0,05	PE é melhor
Grande	SC = 1,0 SNC = 1,0	10%	442	0,45 > 0,05	ND
		25%	332,5	0,041 < 0,05	PE é melhor
		50%	282,5	0,0067 < 0,05	PE é melhor
	SC = 1,0 SNC = 1,5	10%	355	0,081 > 0,05	ND
		25%	407	0,26 > 0,05	ND
		50%	231	0,0006 < 0,05	PE é melhor
	SC = 1,5 SNC = 1,5	10%	269	0,0038 < 0,05	PE é melhor
		25%	214,5	0,0002 < 0,05	PE é melhor
		50%	185	4,603e-05 < 0,05	PE é melhor

Na coluna de Resultados, a sigla ND corresponde a *Não Diferentes*, ou seja, não se rejeita a hipótese nula de que as amostras com prioridade para energia apresentaram medianas inferiores as das amostras de referência, ao nível de 5% de significância.

G.3 Testes bilaterais com os dados de Dispersão

Tabela G.3: Resultados dos testes estatísticos bilaterais de Wilcoxon, com os dados de Dispersão, nos comparativos VMPOS x VMPOS, amostras com prioridade para Dispersão contra as amostras sem atribuir prioridade alguma.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	p-valor	Intervalo de Confiança (95%)	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	7,18e-05 < 0,05	(1,24e-05 ; 6,62e-05)	Rejeita-se H_0
		25%	0,0127 < 0,05	(-1,76e-02 ; 3,76e-05)	Rejeita-se H_0
		50%	0,02027 < 0,05	(-5,51e-06 ; 4,95e-05)	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	0,00821 < 0,05	(-4,39e-05 ; 1,07e-04)	Rejeita-se H_0
		25%	0,04255 < 0,05	(-1,02e-01 ; 1,04e-05)	Rejeita-se H_0
		50%	0,4355 > 0,05	(-8,11e-02 ; 6,87e-05)	Não Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	0,02224 < 0,05	(-7,72e-05 ; 2,56e-05)	Rejeita-se H_0
		25%	1,32e-07 < 0,05	(-0,118 ; -0,022)	Rejeita-se H_0
		50%	0,00833 < 0,05	(-5,03e-02 ; -5,26e-05)	Rejeita-se H_0
Médio	SC = 1,0 SNC = 1,0	10%	0,8703 > 0,05	(-0,046 ; 0,040)	Não rejeita-se H_0
		25%	0,2389 > 0,05	(-0,104 ; 0,043)	Não rejeita-se H_0
		50%	0,256 > 0,05	(-9,18e-02 ; 6,91e-05)	Não rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	0,3909 > 0,05	(-0,021 ; 0,048)	Não rejeita-se H_0
		25%	0,1237 > 0,05	(-0,128 ; 0,004)	Não rejeita-se H_0
		50%	0,5831 > 0,05	(-0,051 ; 0,044)	Não rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	0,615 > 0,05	(-7,44e-02 ; 4,87e-05)	Não rejeita-se H_0
		25%	0,2963 > 0,5	(-0,0739 ; 0,00004)	Não rejeita-se H_0
		50%	0,3231 > 0,05	(-0,030 ; 0,036)	Não rejeita-se H_0
Grande	SC = 1,0 SNC = 1,0	10%	0,06623 > 0,05	(-1,41e-01 ; 3,22e-05)	Não rejeita-se H_0
		25%	0,1565 > 0,05	(-1,10e-01 ; 5,84e-05)	Não rejeita-se H_0
		50%	0,3867 > 0,05	(-0,093 ; 0,044)	Não rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	0,8416 > 0,05	(-0,094 ; 0,091)	Não rejeita-se H_0
		25%	0,2221 > 0,05	(-0,047 ; 0,150)	Não rejeita-se H_0
		50%	0,2032 > 0,05	(-0,106 ; 0,023)	Não rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	0,5502 > 0,05	(-0,059 ; 0,031)	Não rejeita-se H_0
		25%	0,2748 > 0,05	(-0,064 ; 0,037)	Não rejeita-se H_0
		50%	0,04958 < 0,05	(-6,90e-02 ; 1,02e-05)	Rejeita-se H_0

G.4 Testes unilaterais com os dados de Dispersão

Tabela G.4: Resultados dos testes estatísticos unilaterais de Wilcoxon, com os dados de Dispersão, nos comparativos VMPOS x VMPOS, amostras com prioridade para Dispersão contra as amostras sem atribuir prioridade alguma.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	708	1 > 0,05	ND
		25%	306,5	0,006 < 0,05	PD é melhor
		50%	601	0,99 > 0,05	ND
	SC = 1,0 SNC = 1,5	10%	624	0,99 > 0,05	ND
		25%	310	0,021 < 0,05	PD é melhor
		50%	397	0,21 > 0,05	ND
	SC = 1,5 SNC = 1,5	10%	300	0,01 < 0,05	PD é melhor
		25%	102	6,576e-08 < 0,05	PD é melhor
		50%	277,5	0,004 < 0,05	PD é melhor
Médio	SC = 1,0 SNC = 1,0	10%	438,5	0,43 > 0,05	ND
		25%	370	0,11 > 0,05	ND
		50%	373	0,12 > 0,05	ND
	SC = 1,0 SNC = 1,5	10%	508	0,8 > 0,05	ND
		25%	345,5	0,061 > 0,05	ND
		50%	412,5	0,29 > 0,05	ND
	SC = 1,5 SNC = 1,5	10%	416	0,30 > 0,05	ND
		25%	381	0,14 > 0,05	ND
		50%	517	0,84 > 0,05	ND
Grande	SC = 1,0 SNC = 1,0	10%	325,5	0,033 > 0,05	ND
		25%	354	0,078 > 0,05	ND
		50%	391	0,19 > 0,05	ND
	SC = 1,0 SNC = 1,5	10%	463	0,42 > 0,05	ND
		25%	533	0,89 > 0,05	ND
		50%	363,5	0,1 > 0,05	ND
	SC = 1,5 SNC = 1,5	10%	490,5	0,73 > 0,05	ND
		25%	367	0,13 > 0,05	ND
		50%	317	0,024 < 0,05	PD é melhor

G.5 Testes bilaterais com os dados de Migração

Tabela G.5: Resultados dos testes estatísticos bilaterais de Wilcoxon, com os dados de Migração, nos comparativos VMPOS x VMPOS, amostras com prioridade para Migração contra as amostras sem atribuir prioridade alguma.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	p-valor	Intervalo de Confiança (95%)	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	7,84e-07 < 0,05	(-6,00 ; -3,00)	Rejeita-se H_0
		25%	1,02e-07 < 0,05	(-8,00 ; -5,00)	Rejeita-se H_0
		50%	3,36e-07 < 0,05	(-11,00 ; -7,00)	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	2,86e-08 < 0,05	(-7,99 ; -4,00)	Rejeita-se H_0
		25%	0,00022 < 0,05	(-7,00 ; -2,00)	Rejeita-se H_0
		50%	7,33e-10 < 0,05	(-12,99 ; -8,00)	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	0,00062 < 0,05	(-3,99 ; -1,00)	Rejeita-se H_0
		25%	0,00014 < 0,05	(-6,00 ; -1,99)	Rejeita-se H_0
		50%	4,41e-07 < 0,05	(-10,00 ; -5,99)	Rejeita-se H_0
Médio	SC = 1,0 SNC = 1,0	10%	4,38e-06 < 0,05	(-7,00 ; -3,00)	Rejeita-se H_0
		25%	2,14e-06 < 0,05	(-10,99 ; -5,00)	Rejeita-se H_0
		50%	0,0025 < 0,05	(-6,99 ; -1,99)	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	0,00011 < 0,05	(-6,99 ; -2,00)	Rejeita-se H_0
		25%	0,0015 < 0,05	(-8,00 ; -2,00)	Rejeita-se H_0
		50%	2,48e-07 < 0,05	(-11,00 ; -5,99)	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	0,0053 < 0,05	(-6,00 ; -1,00)	Rejeita-se H_0
		25%	3,77e-07 < 0,5	(-9,99 ; -5,00)	Rejeita-se H_0
		50%	3,46e-06 < 0,05	(-12,99 ; -5,99)	Rejeita-se H_0
Grande	SC = 1,0 SNC = 1,0	10%	0,01869 < 0,05	(-4,00 ; -0,99)	Rejeita-se H_0
		25%	0,0026 < 0,05	(-5,00 ; -1,00)	Rejeita-se H_0
		50%	5,55e-05 < 0,05	(-10,00 ; -4,00)	Rejeita-se H_0
	SC = 1,0 SNC = 1,5	10%	0,1597 > 0,05	(-3,99 ; 4,76)	Não rejeita-se H_0
		25%	0,0054 < 0,05	(-7,00 ; -1,00)	Rejeita-se H_0
		50%	0,00098 < 0,05	(-9,00 ; -2,00)	Rejeita-se H_0
	SC = 1,5 SNC = 1,5	10%	3,42e-05 < 0,05	(-7,99 ; -3,00)	Rejeita-se H_0
		25%	0,0355 < 0,05	(-7,00 ; -2,402e-05)	Rejeita-se H_0
		50%	2,95e-06 < 0,05	(-13,99 ; -5,00)	Rejeita-se H_0

G.6 Testes unilaterais com os dados de Migração

Tabela G.6: Resultados dos testes estatísticos unilaterais de Wilcoxon, com os dados de Migração, nos comparativos VMPOS x VMPOS, amostras com prioridade para Migração contra as amostras sem atribuir prioridade alguma.

Tamanho do Datacenter	Nível de Comprometimento	Porcentagem VMs Críticas	W	p-valor	Resultado
Pequeno	SC = 1,0 SNC = 1,0	10%	117	3,918e-07 < 0,05	PM é melhor
		25%	90,5	5,089e-08 < 0,05	PM é melhor
		50%	105	1,678e-07 < 0,05	PM é melhor
	SC = 1,0 SNC = 1,5	10%	75,5	1,432e-08 < 0,05	PM é melhor
		25%	200,5	0,0001 < 0,05	PM é melhor
		50%	33,5	3,665e-10 < 0,05	PM é melhor
	SC = 1,5 SNC = 1,5	10%	220,5	0,0003 < 0,05	PM é melhor
		25%	194	7,345e-05 < 0,05	PM é melhor
		50%	108,5	2,203e-07 < 0,05	PM é melhor
Médio	SC = 1,0 SNC = 1,0	10%	140	2,191e-06 < 0,05	PM é melhor
		25%	129,5	1,069e-06 < 0,05	PM é melhor
		50%	246	0,001 < 0,05	PM é melhor
	SC = 1,0 SNC = 1,5	10%	190	5,947e-05 < 0,05	PM é melhor
		25%	236	0,0007 < 0,05	PM é melhor
		50%	101	1,238e-07 < 0,05	PM é melhor
	SC = 1,5 SNC = 1,5	10%	261,5	0,0026 < 0,05	PM é melhor
		25%	106,5	1,884e-07 < 0,05	PM é melhor
		50%	136	1,728e-06 < 0,05	PM é melhor
Grande	SC = 1,0 SNC = 1,0	10%	291	0,0093 < 0,05	PM é melhor
		25%	247	0,0013 < 0,05	PM é melhor
		50%	177,5	2,774e-05 < 0,05	PM é melhor
	SC = 1,0 SNC = 1,5	10%	355	0,079 > 0,05	ND
		25%	262	0,0027 < 0,05	PM é melhor
		50%	227	0,0004 < 0,05	PM é melhor
	SC = 1,5 SNC = 1,5	10%	170	1,712e-05 < 0,05	PM é melhor
		25%	307,5	0,017 < 0,05	PM é melhor
		50%	134	1,476e-06 < 0,05	PM é melhor

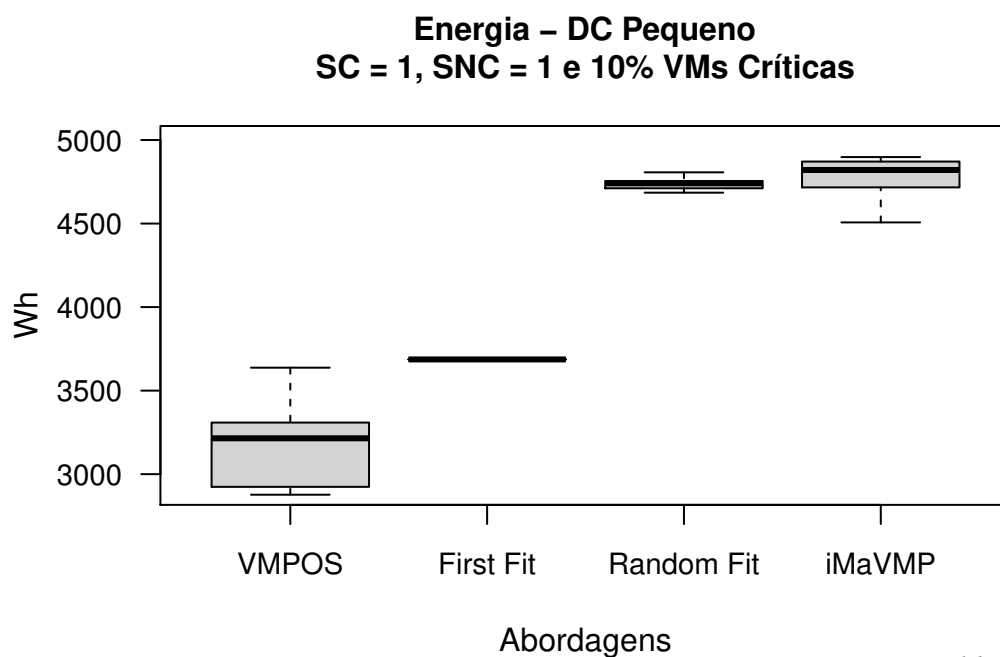
Apêndice H

Gráficos

Neste capítulo dos anexos foram inseridos diversos gráficos dos cenários testados nos experimentos.

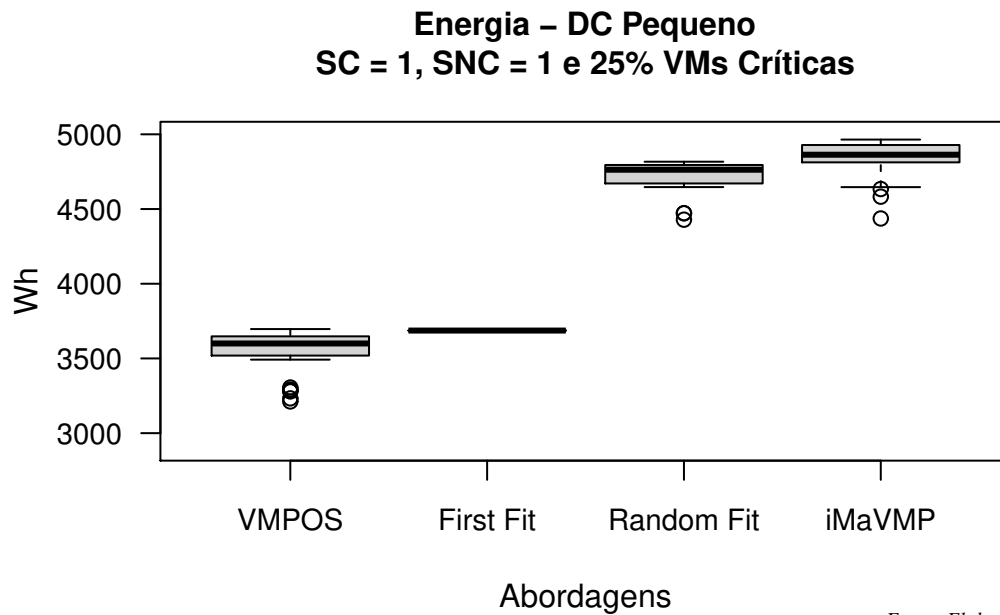
H.1 Dados de Energia, SC=1,0 e SNC=1,0

Figura H.1: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com SC=1,0 e SNC=1,0.



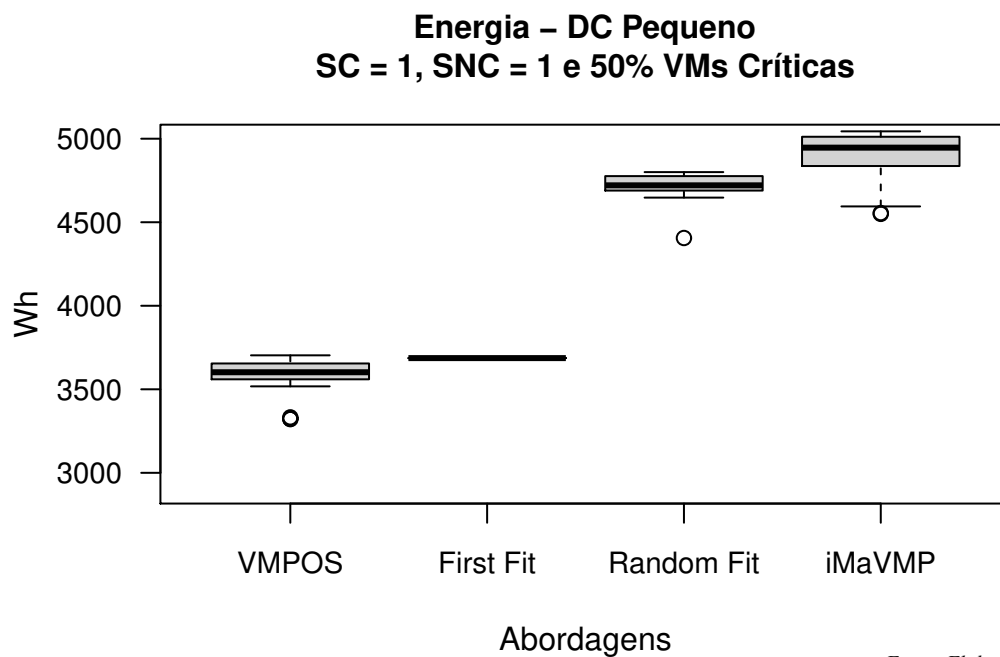
Fonte: Elaborada pelo autor.

Figura H.2: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

Figura H.3: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

Figura H.4: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.

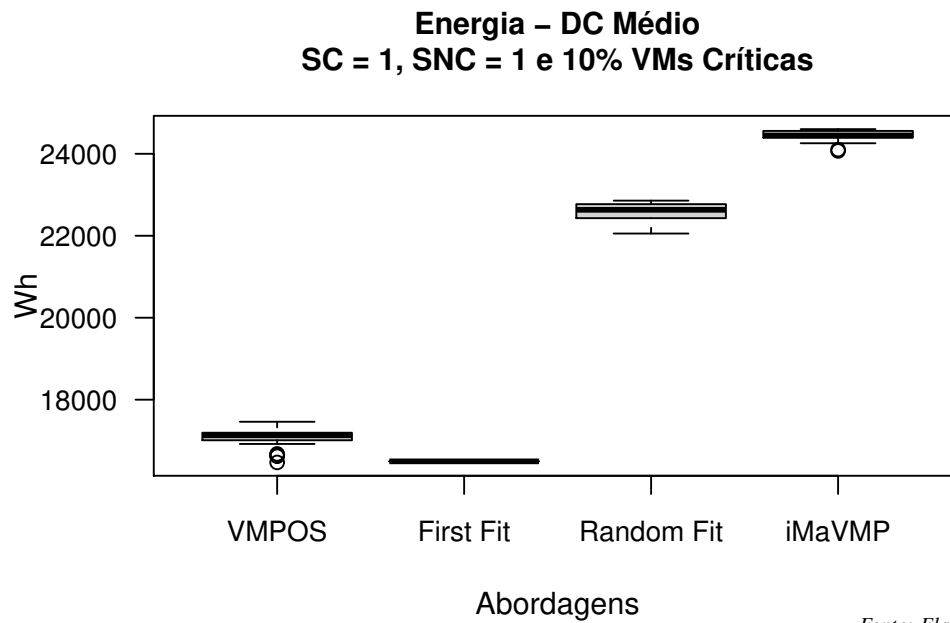


Figura H.5: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.

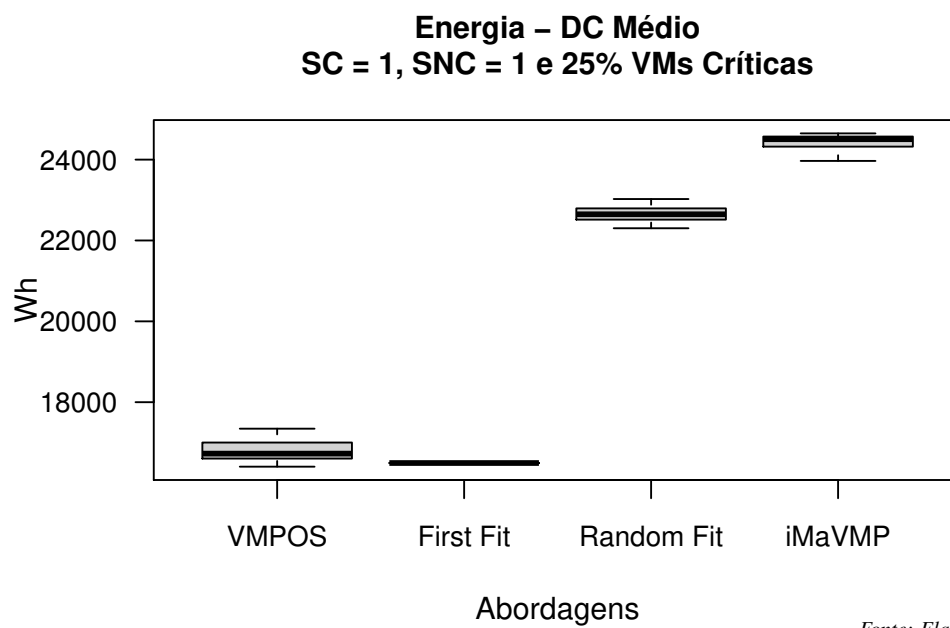


Figura H.6: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.

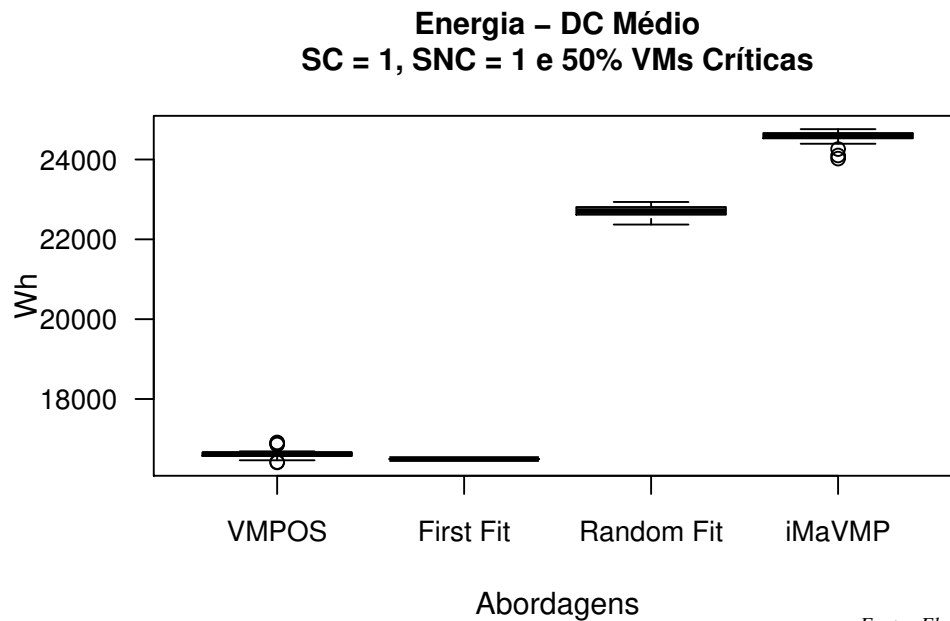


Figura H.7: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.

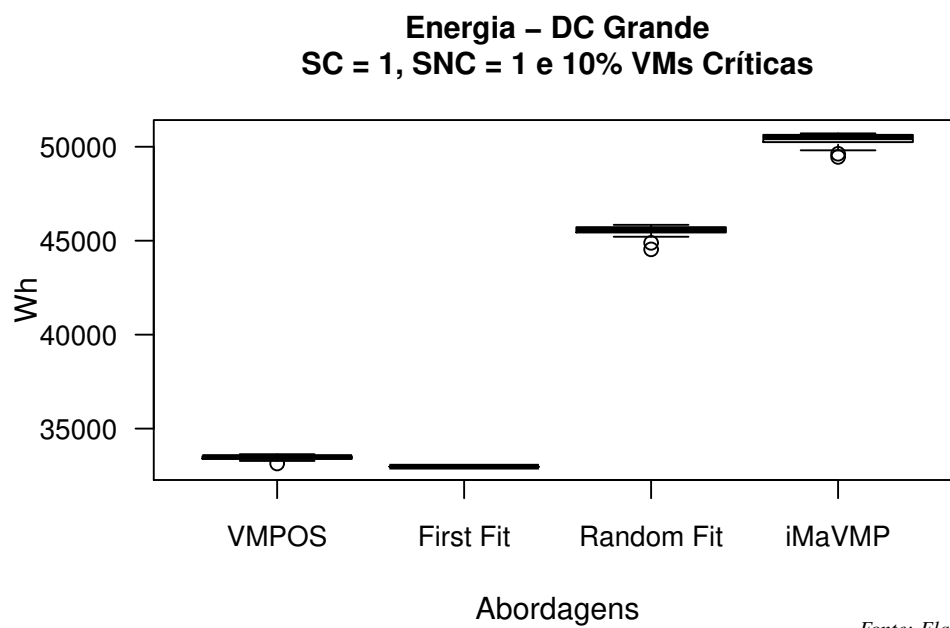
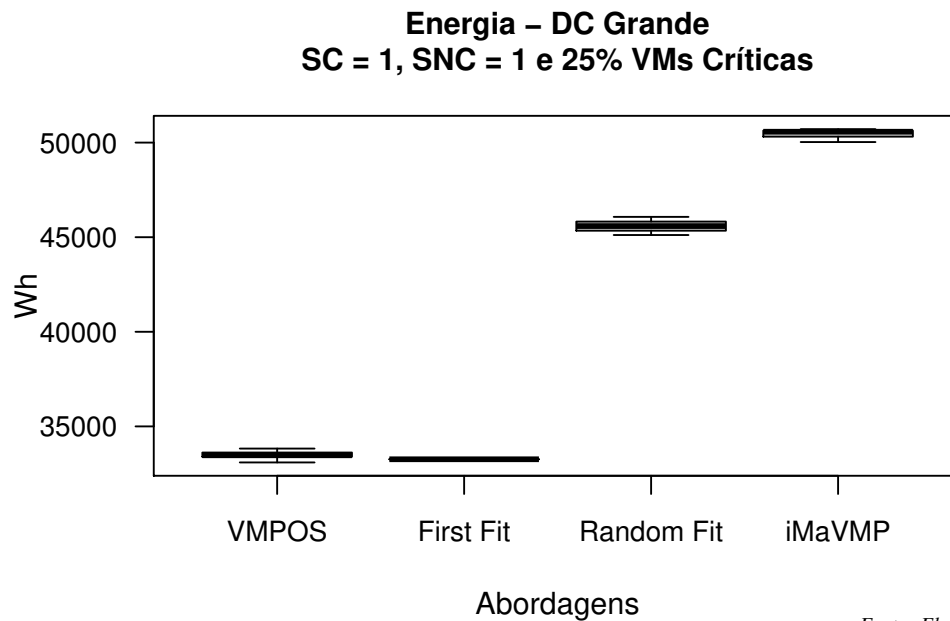
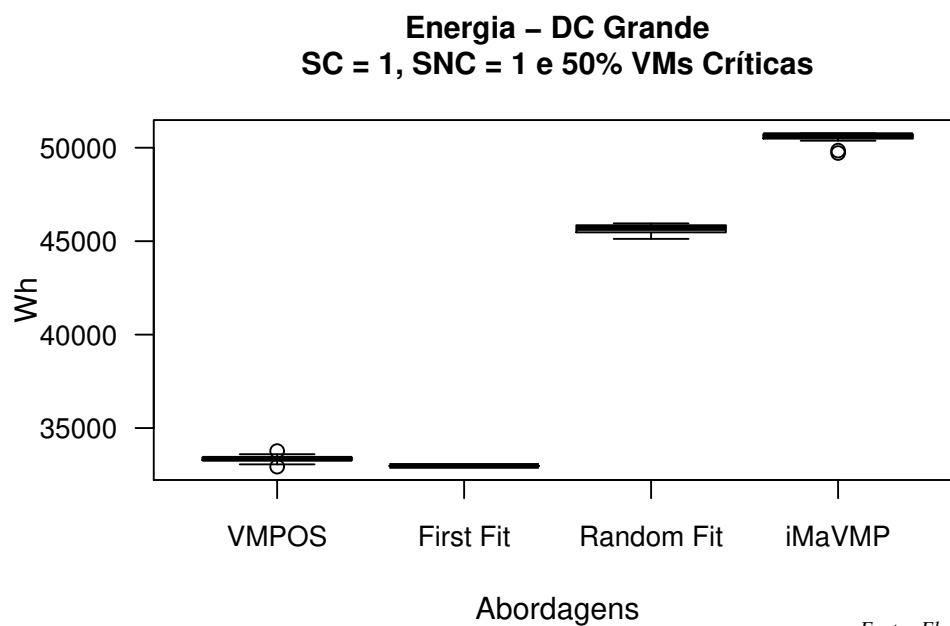


Figura H.8: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

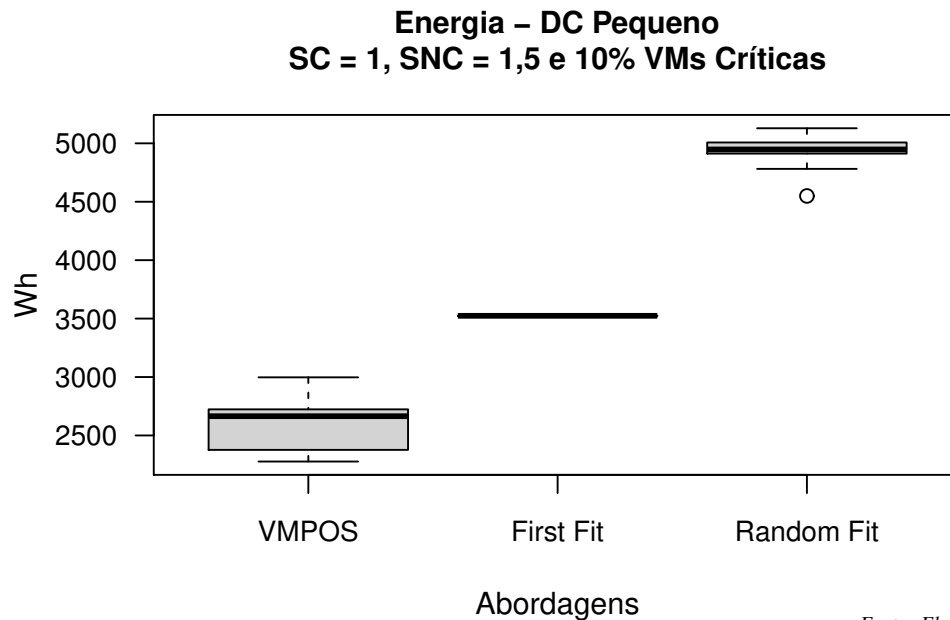
Figura H.9: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

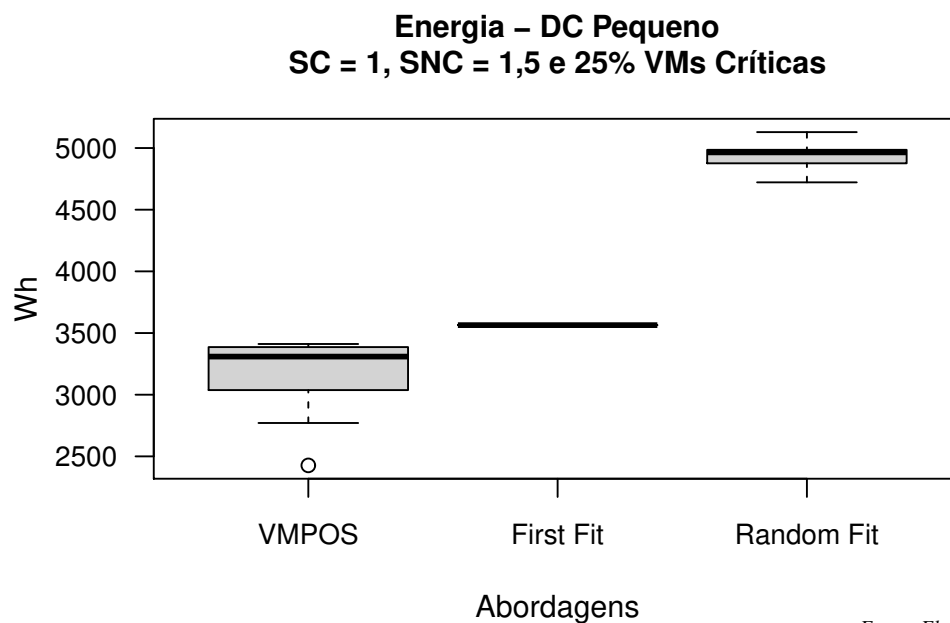
H.2 Dados de Energia, $SC=1,0$ e $SNC=1,5$

Figura H.10: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

Figura H.11: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

Figura H.12: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.

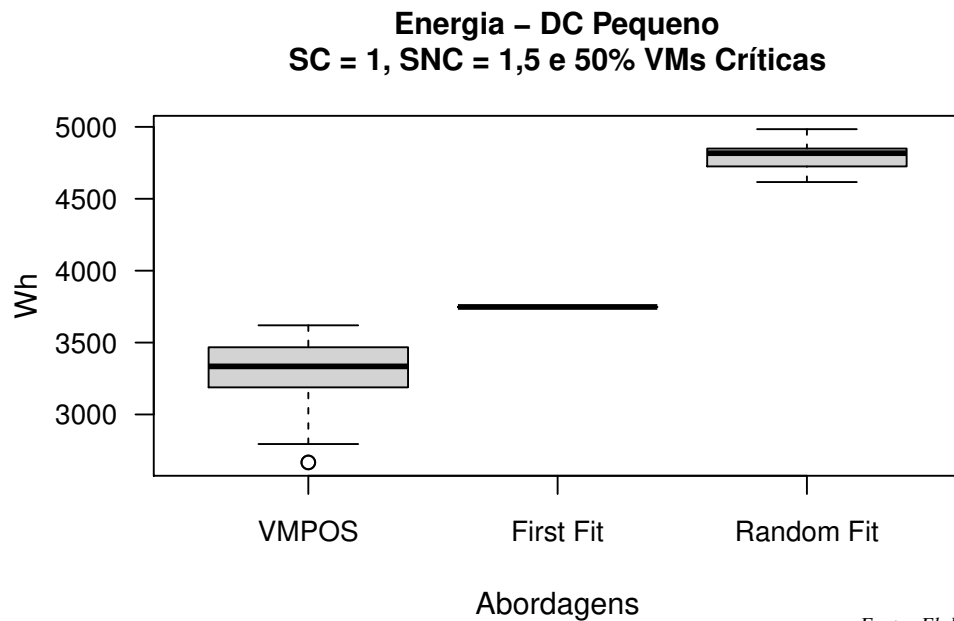


Figura H.13: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.

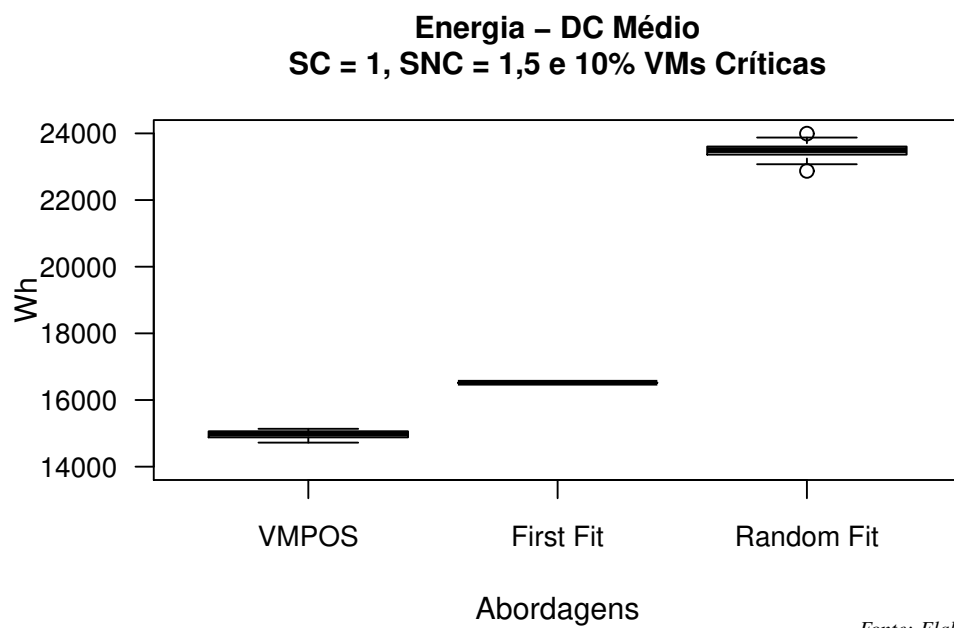
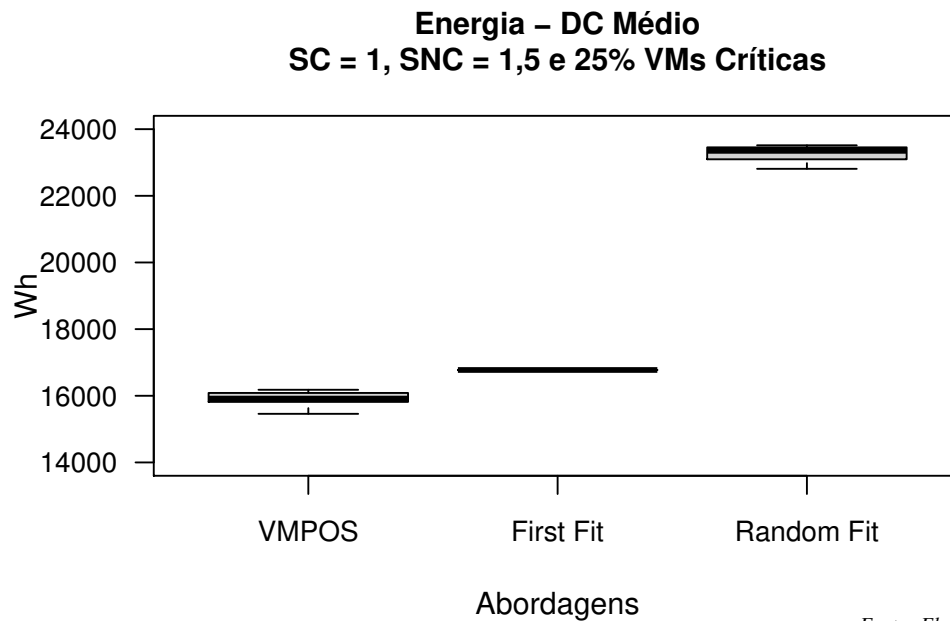
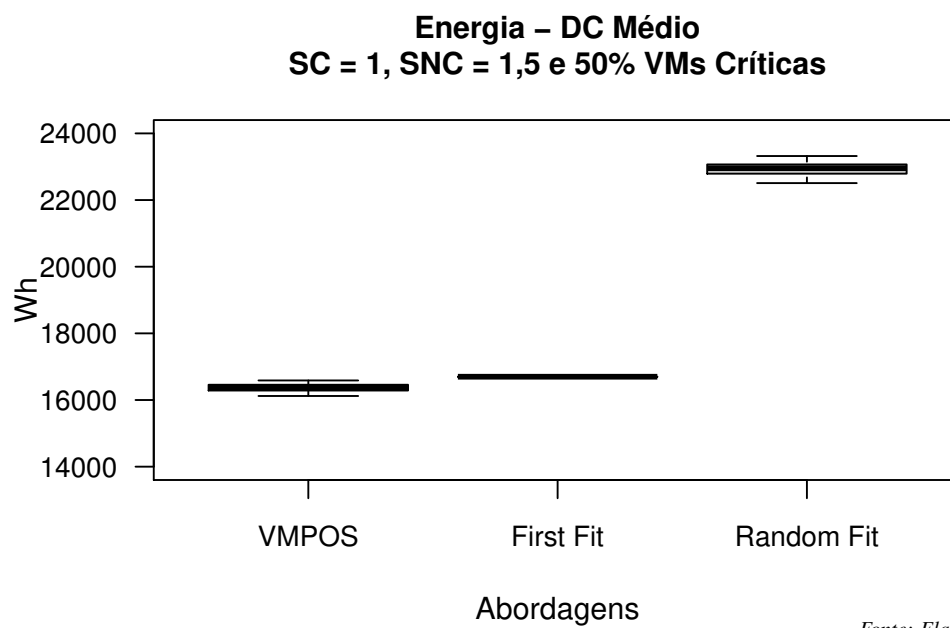


Figura H.14: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



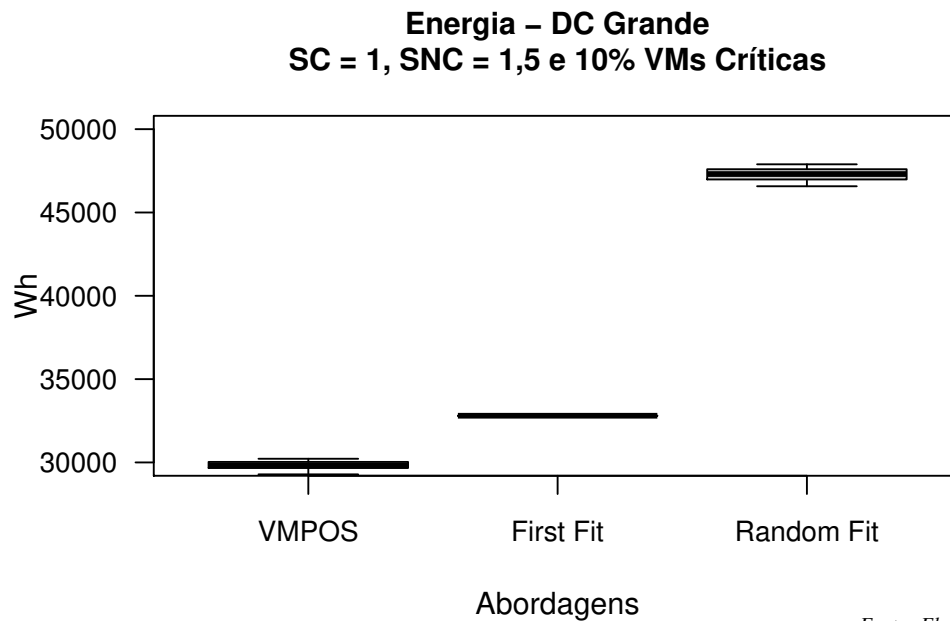
Fonte: Elaborada pelo autor.

Figura H.15: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



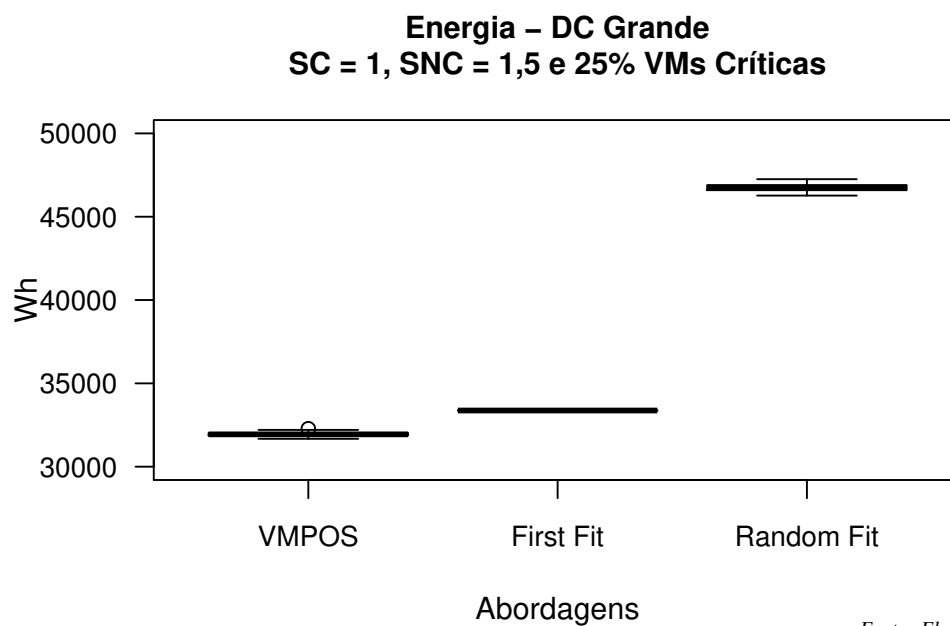
Fonte: Elaborada pelo autor.

Figura H.16: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



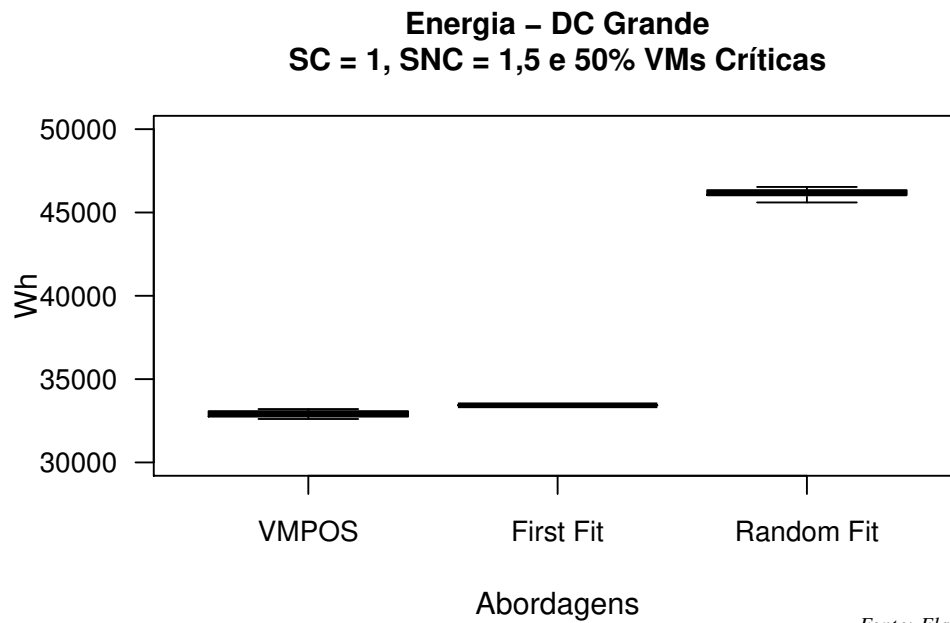
Fonte: Elaborada pelo autor.

Figura H.17: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

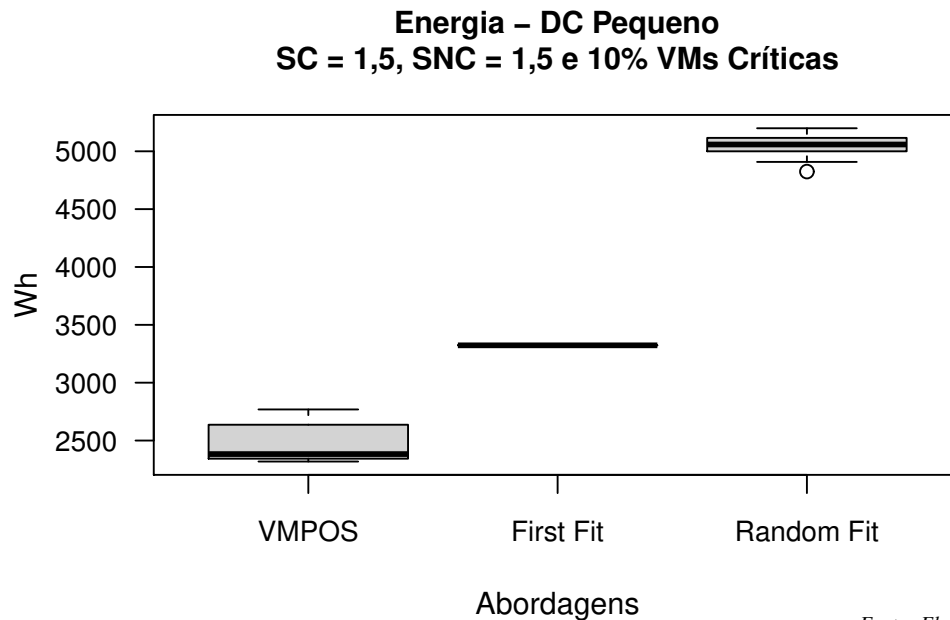
Figura H.18: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

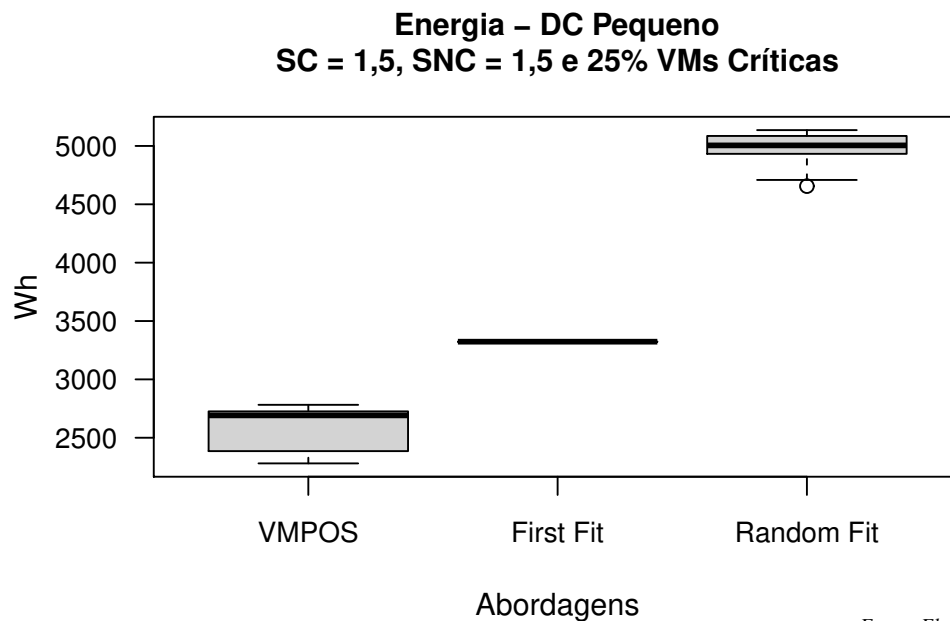
H.3 Dados de Energia, $SC=1,5$ e $SNC=1,5$

Figura H.19: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



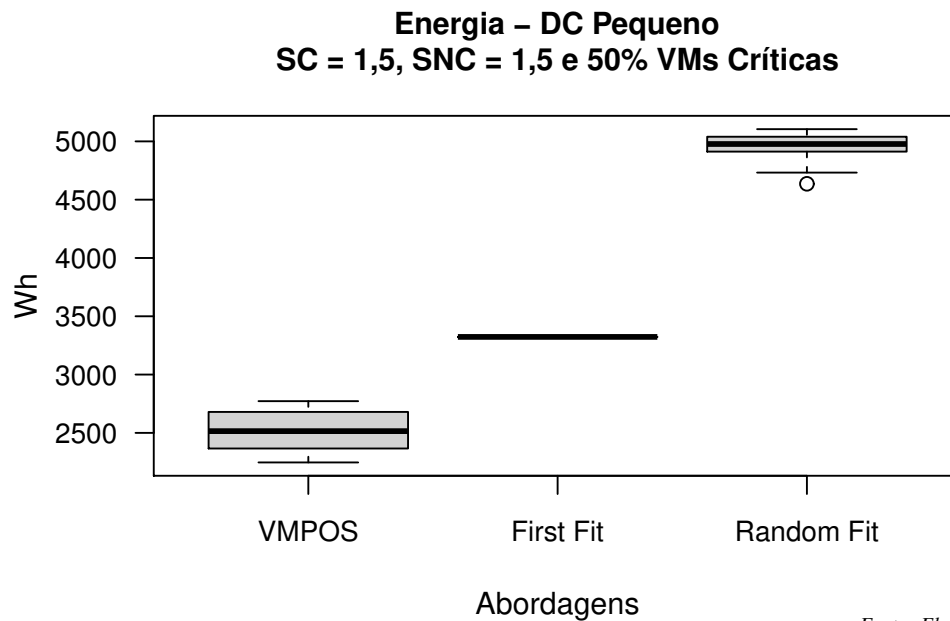
Fonte: Elaborada pelo autor.

Figura H.20: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



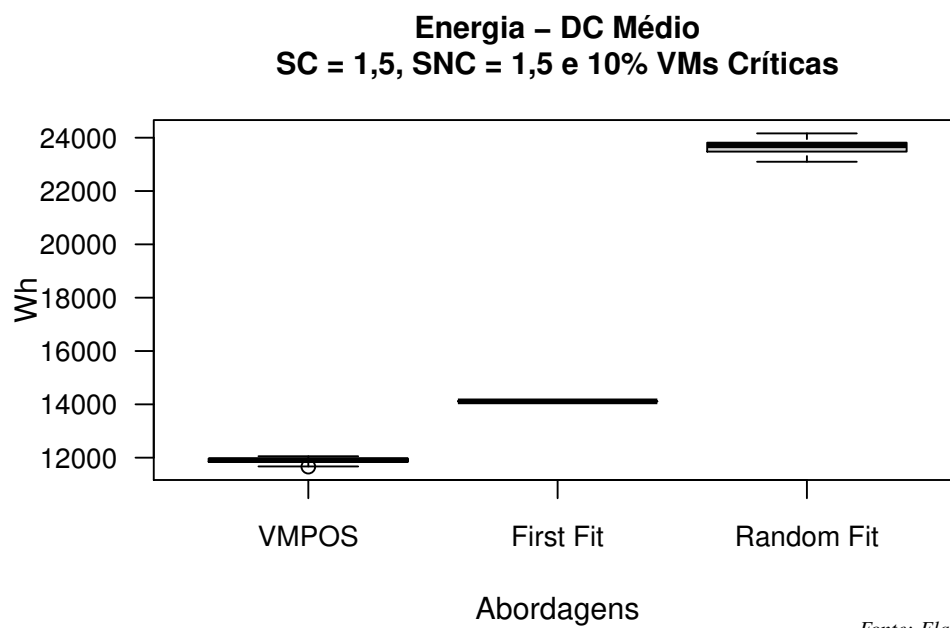
Fonte: Elaborada pelo autor.

Figura H.21: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



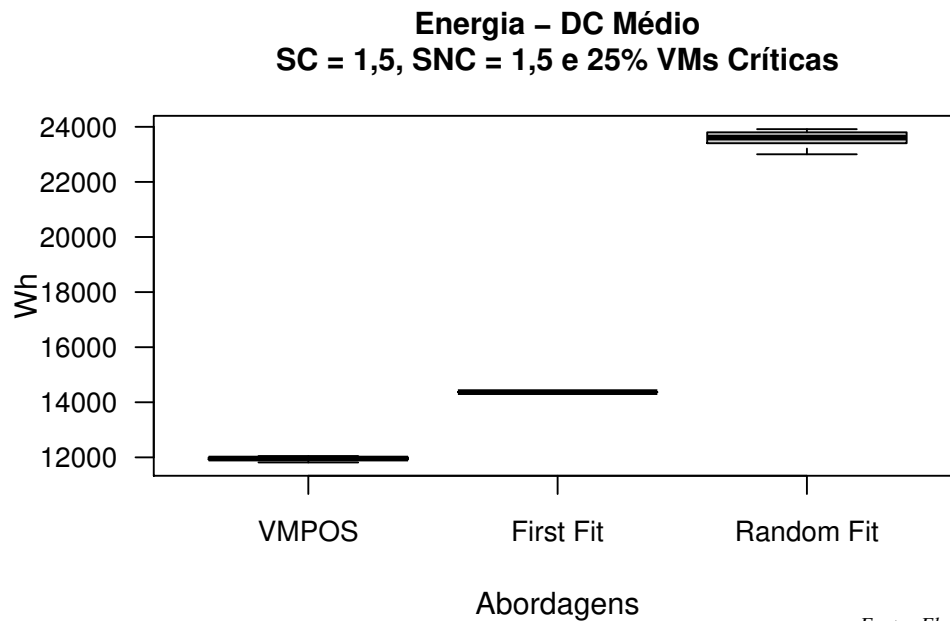
Fonte: Elaborada pelo autor.

Figura H.22: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



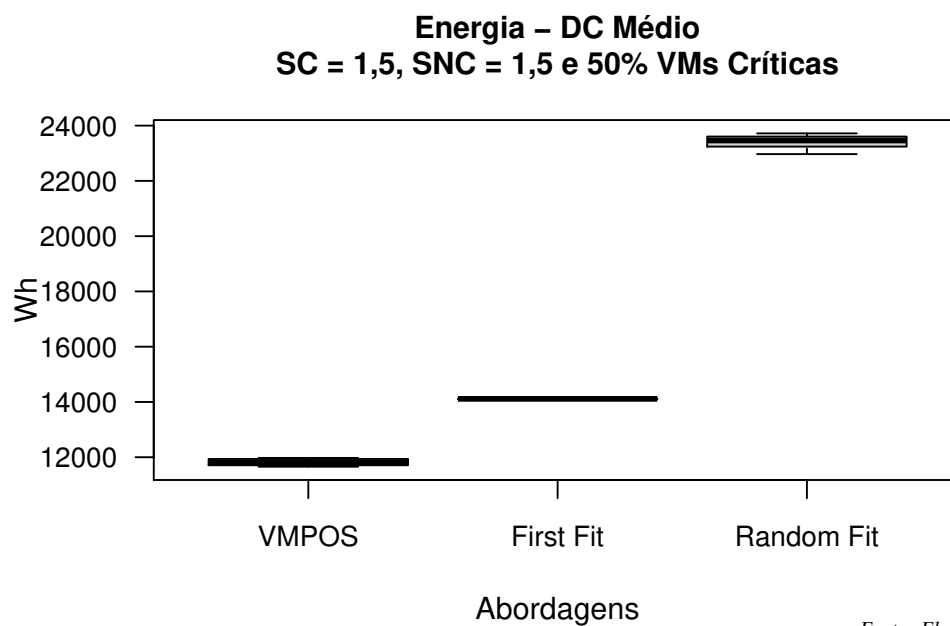
Fonte: Elaborada pelo autor.

Figura H.23: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



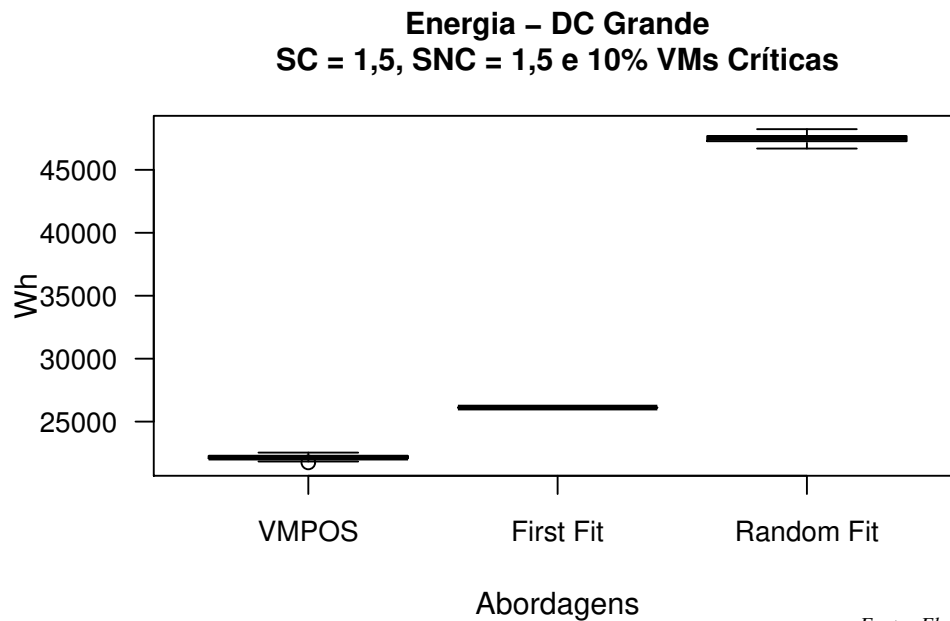
Fonte: Elaborada pelo autor.

Figura H.24: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



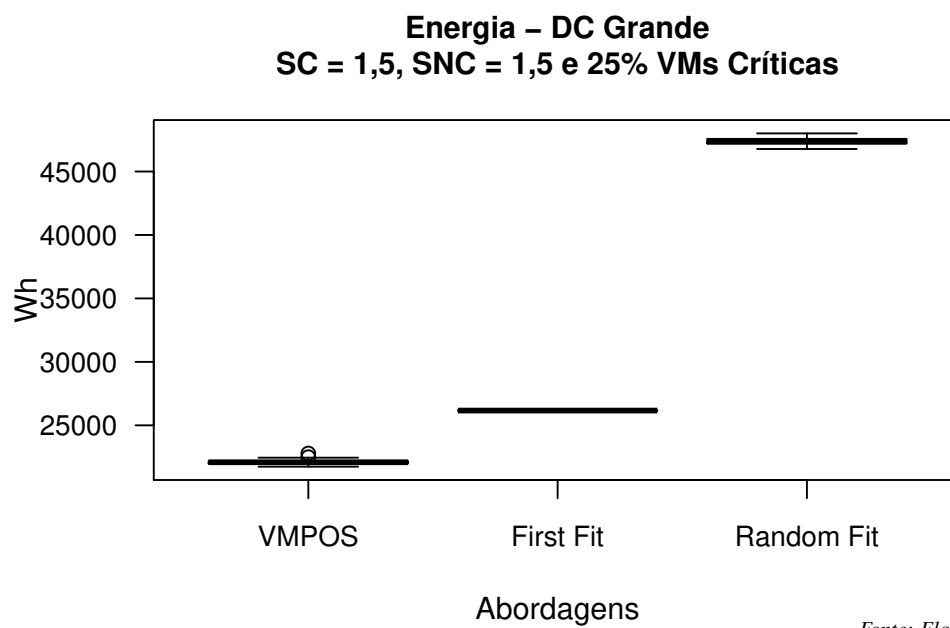
Fonte: Elaborada pelo autor.

Figura H.25: *Boxplots* com as amostras de Energia com 10% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



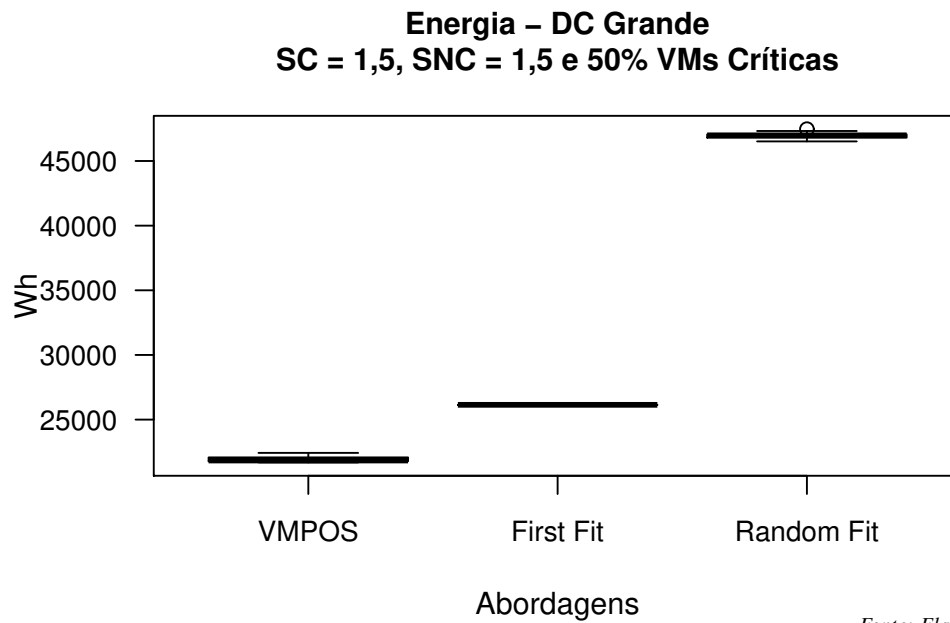
Fonte: Elaborada pelo autor.

Figura H.26: *Boxplots* com as amostras de Energia com 25% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

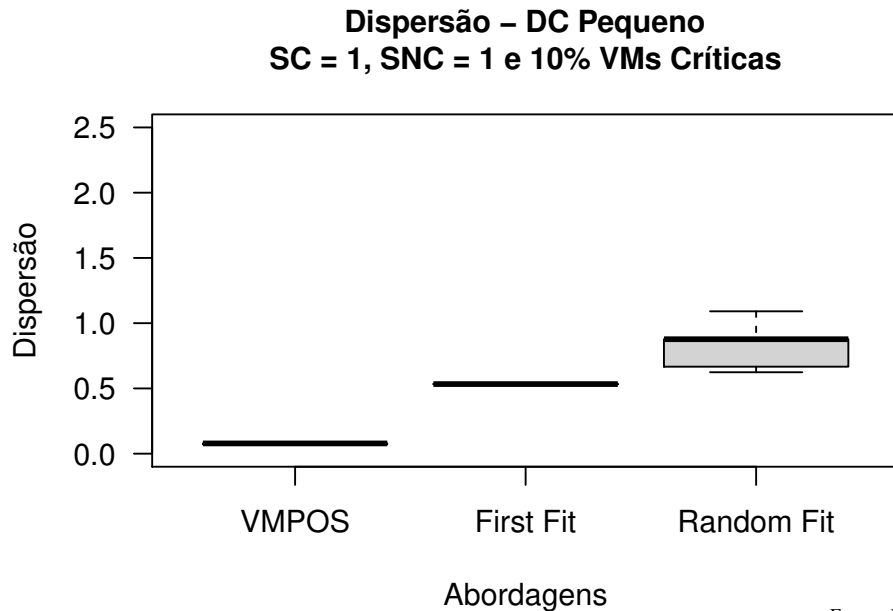
Figura H.27: *Boxplots* com as amostras de Energia com 50% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

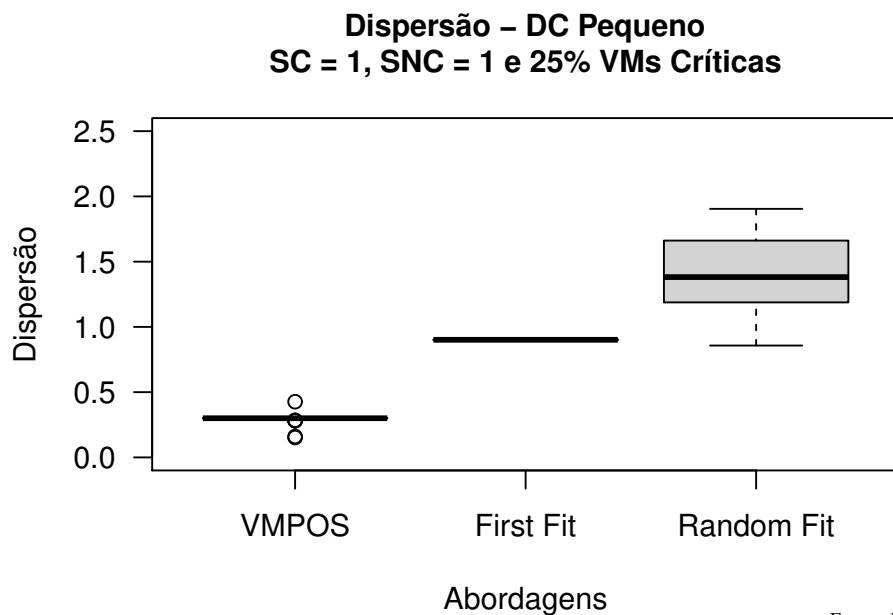
H.4 Dados de Dispersão, $SC=1,0$ e $SNC=1,0$

Figura H.28: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



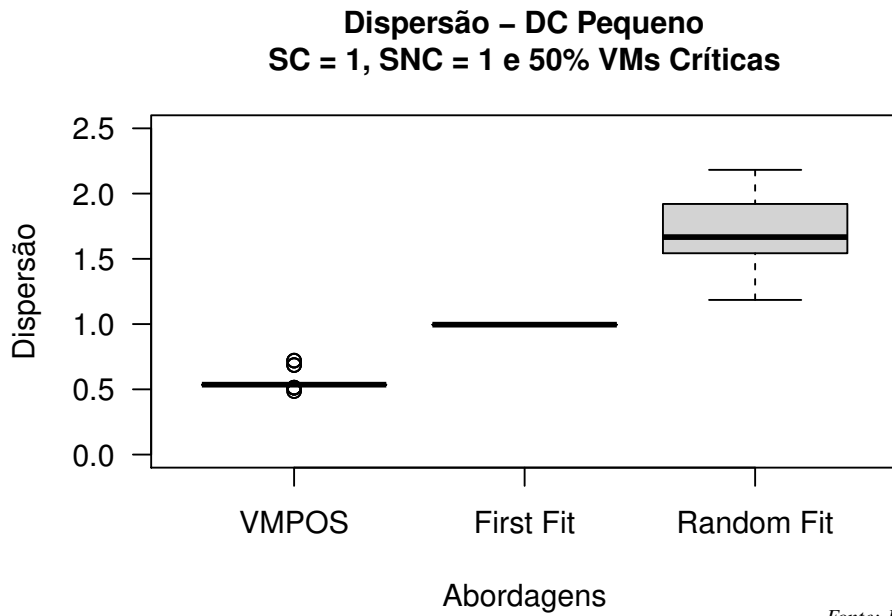
Fonte: Elaborada pelo autor.

Figura H.29: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



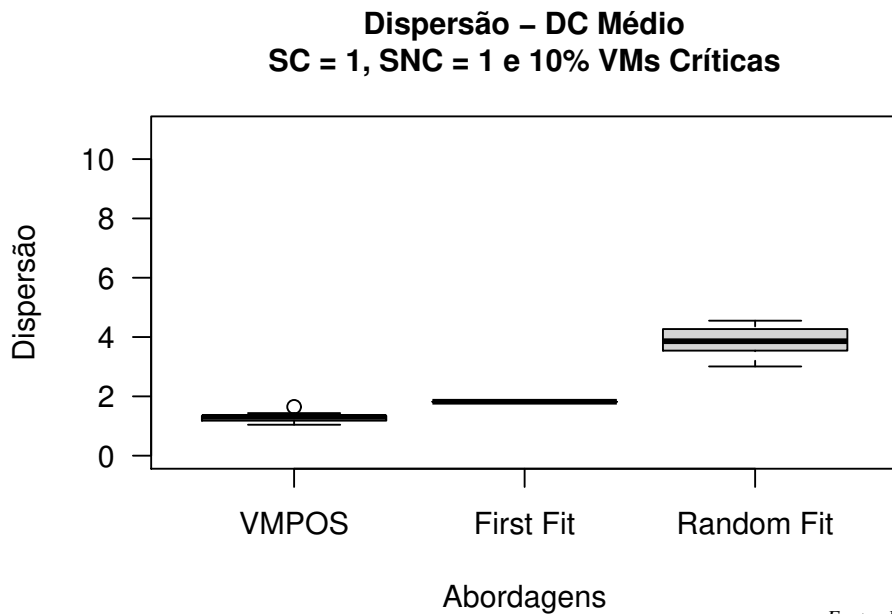
Fonte: Elaborada pelo autor.

Figura H.30: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

Figura H.31: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

Figura H.32: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.

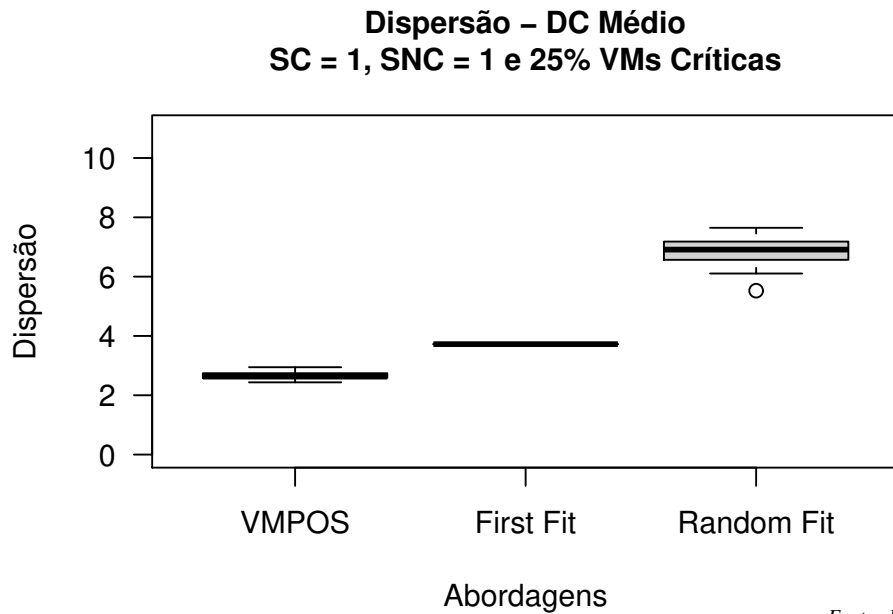


Figura H.33: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.

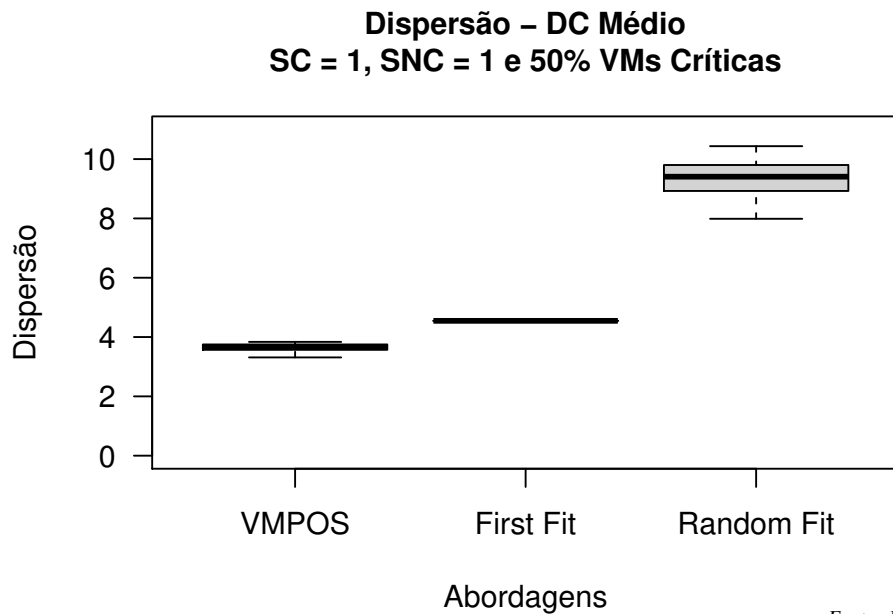
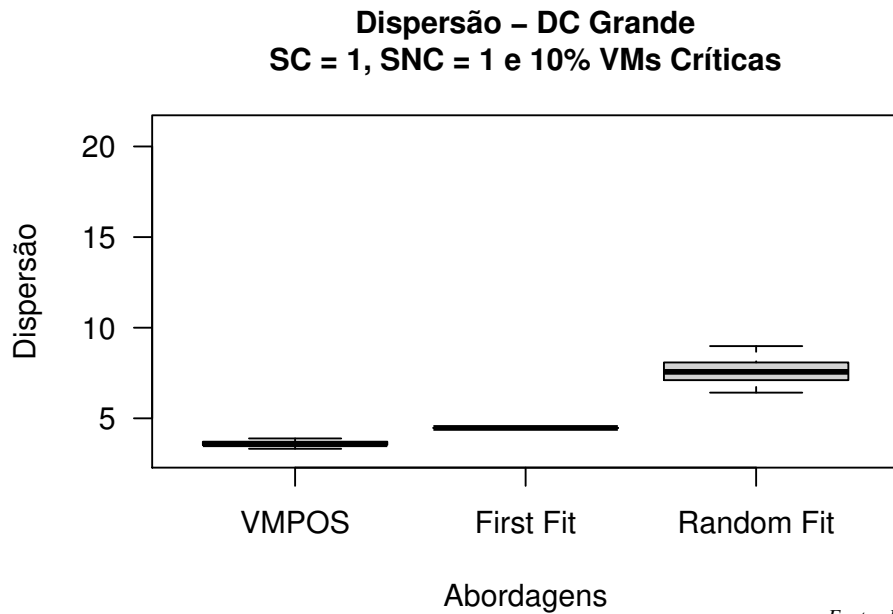
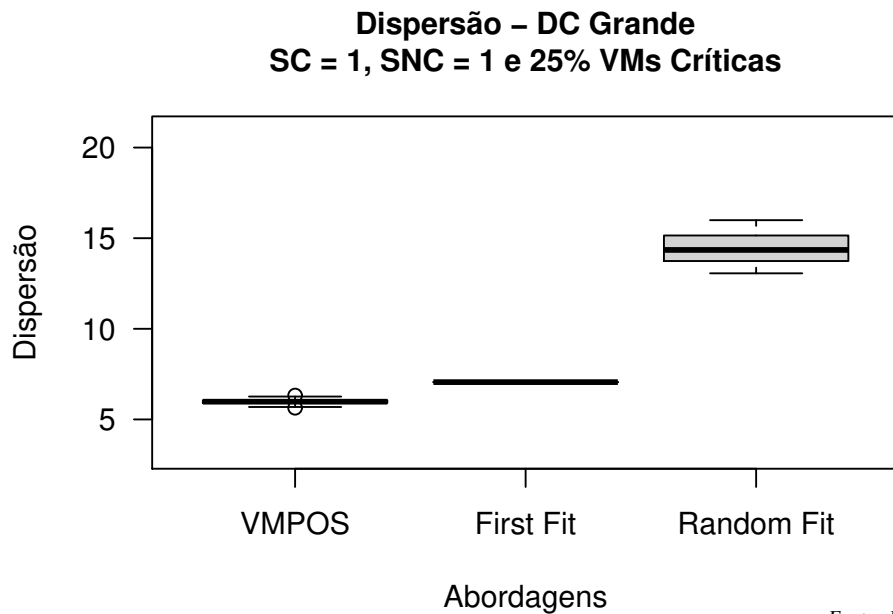


Figura H.34: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



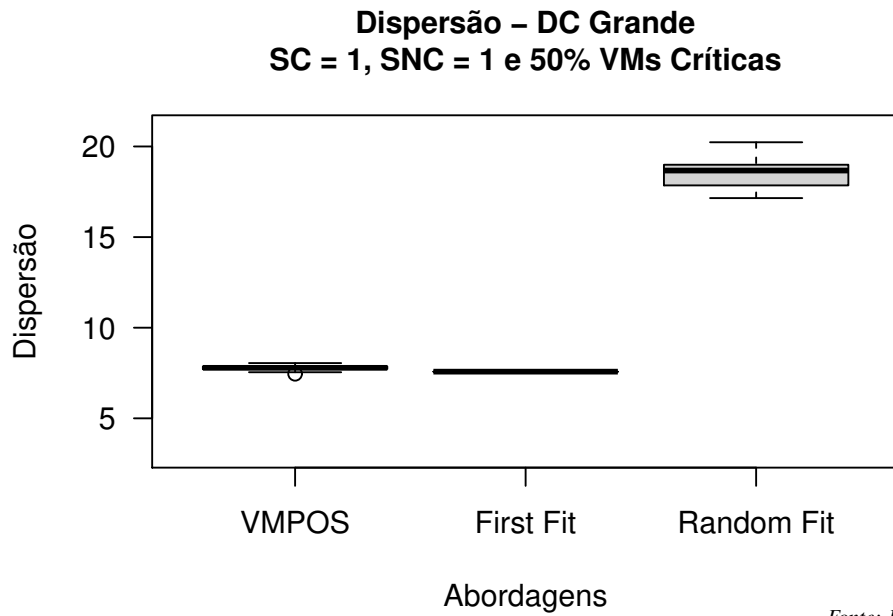
Fonte: Elaborada pelo autor.

Figura H.35: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

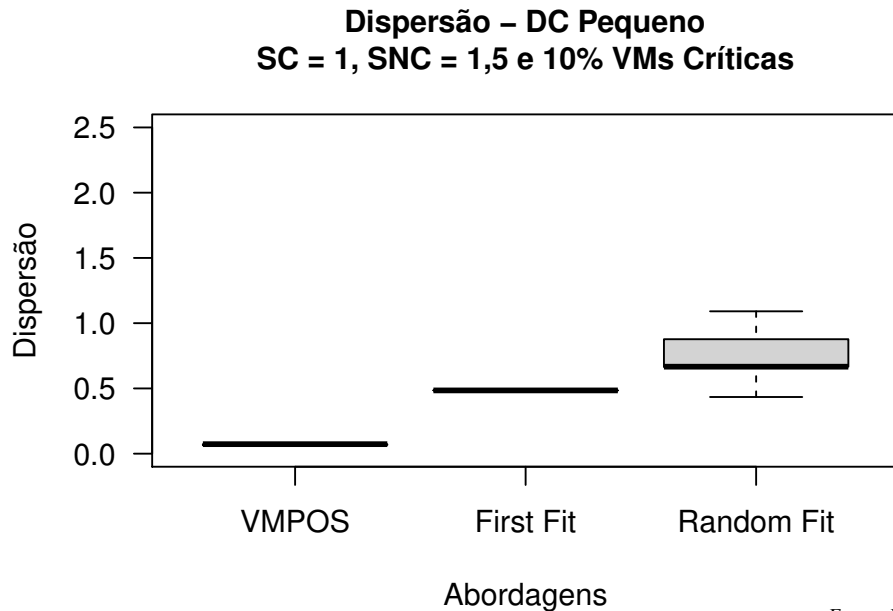
Figura H.36: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,0$.



Fonte: Elaborada pelo autor.

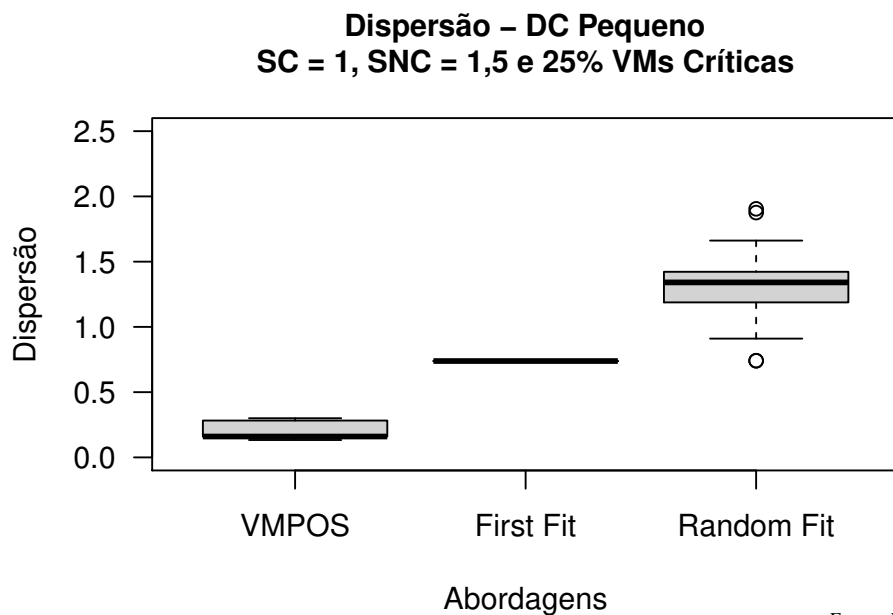
H.5 Dados de Dispersão, SC=1,0 e SNC=1,5

Figura H.37: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,0 e SNC=1,5.



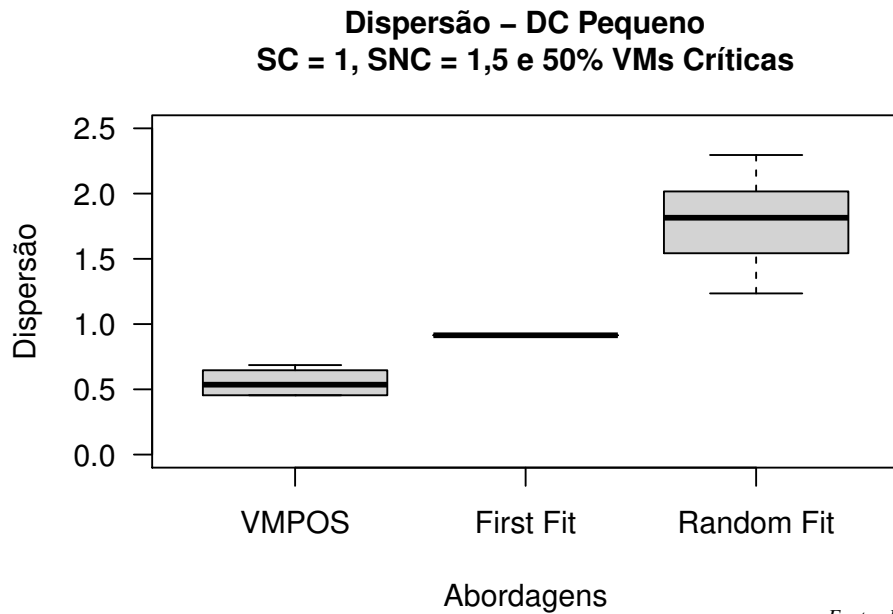
Fonte: Elaborada pelo autor.

Figura H.38: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,0 e SNC=1,5.



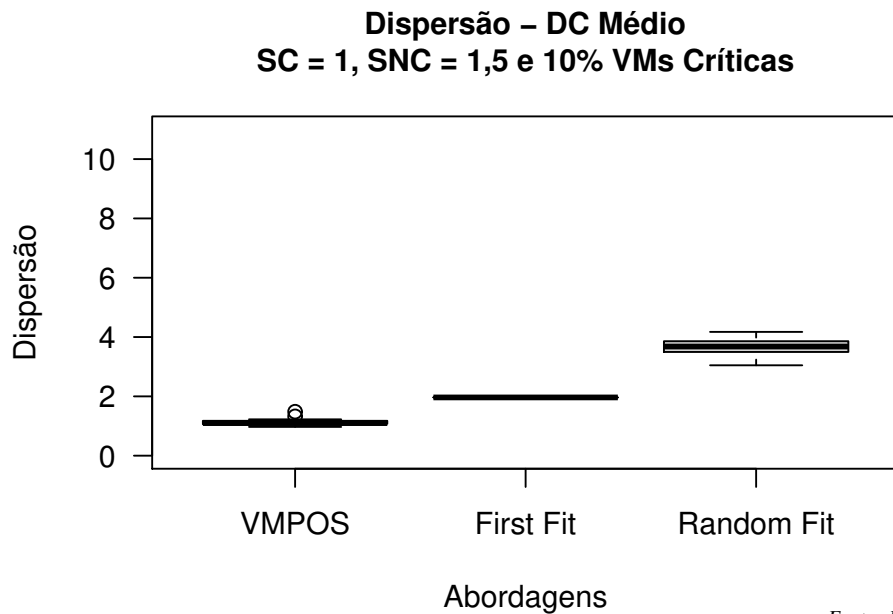
Fonte: Elaborada pelo autor.

Figura H.39: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

Figura H.40: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

Figura H.41: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.

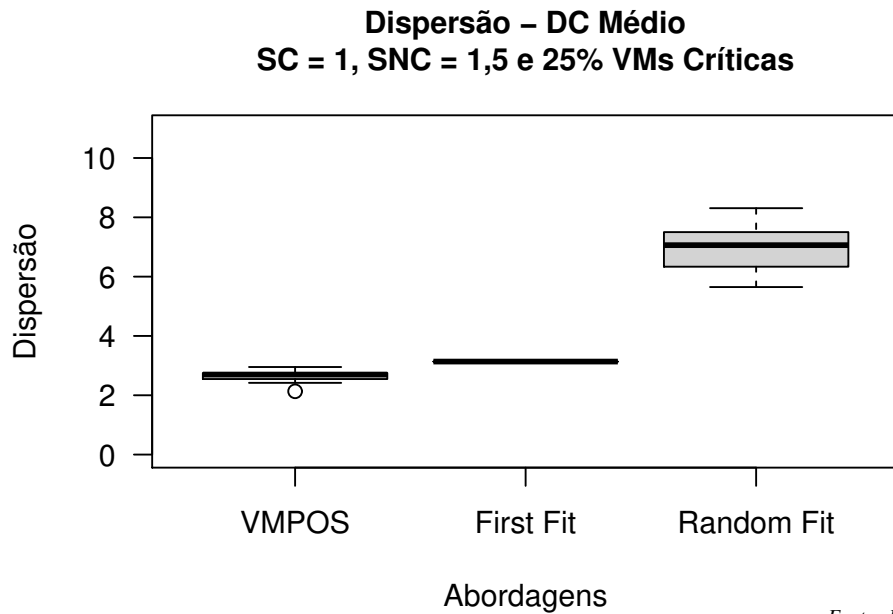


Figura H.42: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.

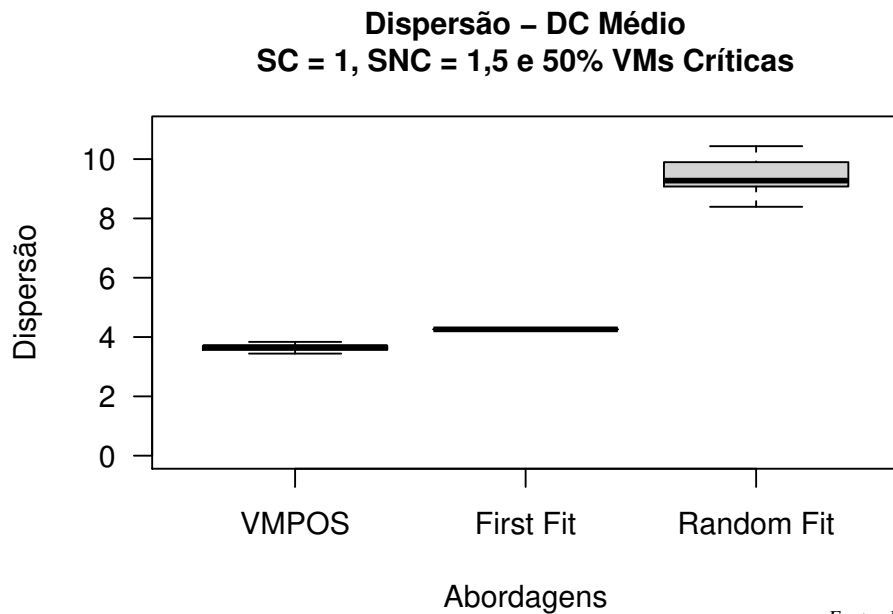


Figura H.43: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.

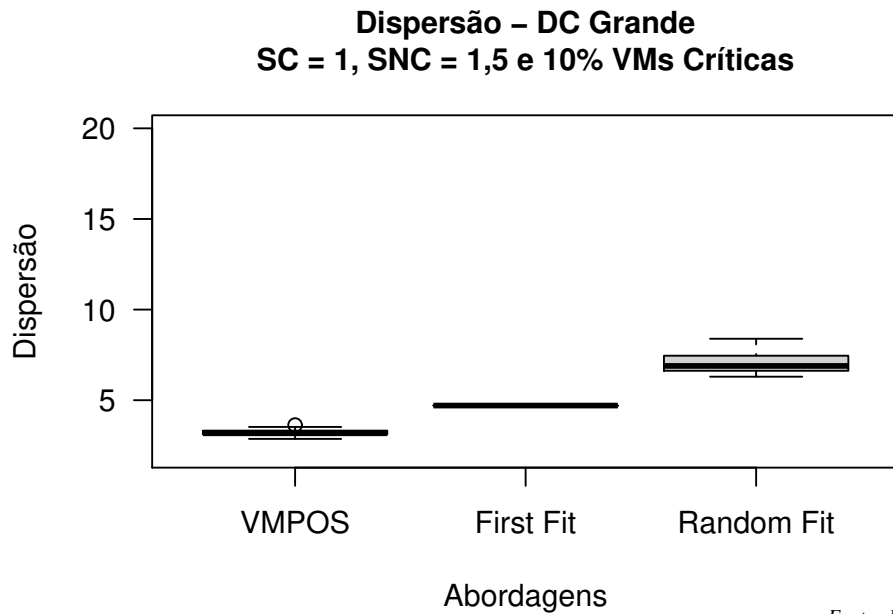


Figura H.44: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.

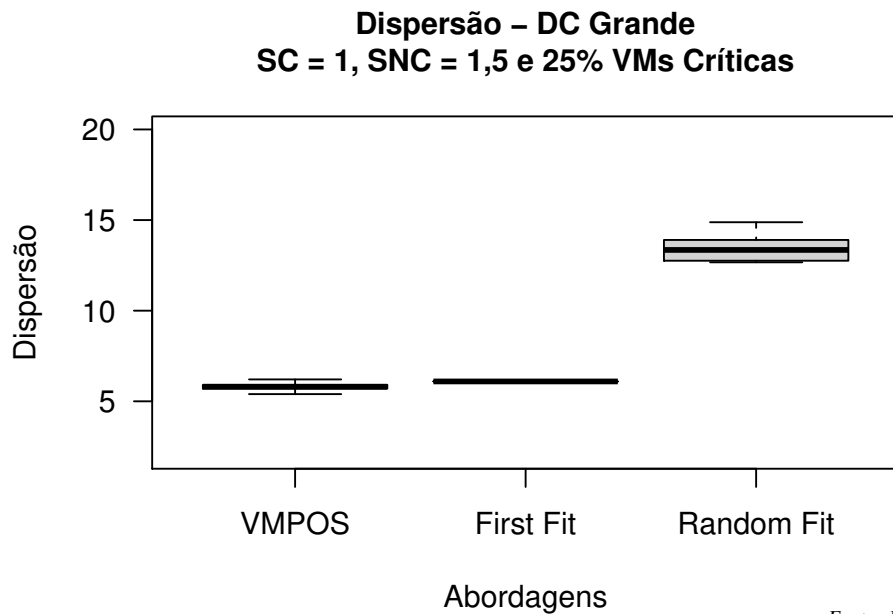
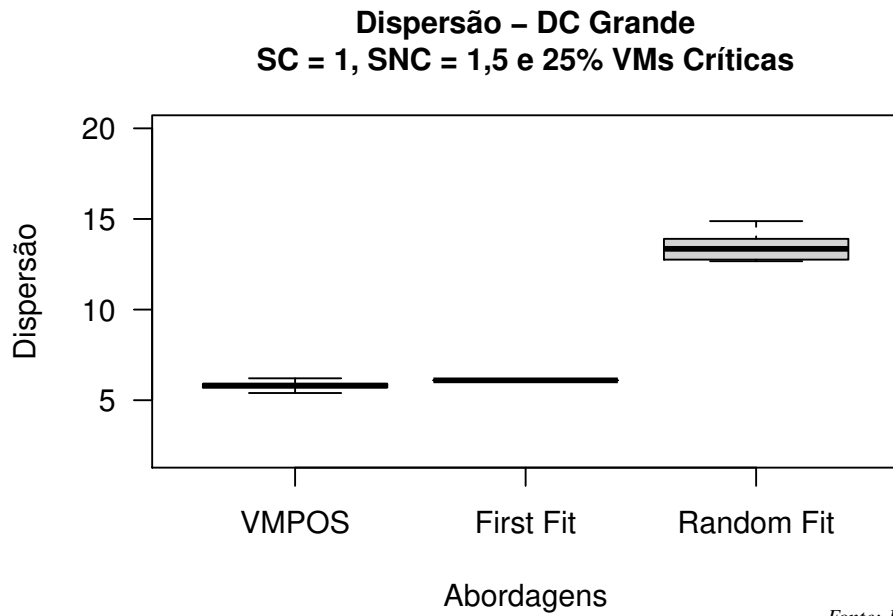


Figura H.45: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,0$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.

H.6 Dados de Dispersão, SC=1,5 e SNC=1,5

Figura H.46: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com SC=1,5 e SNC=1,5.

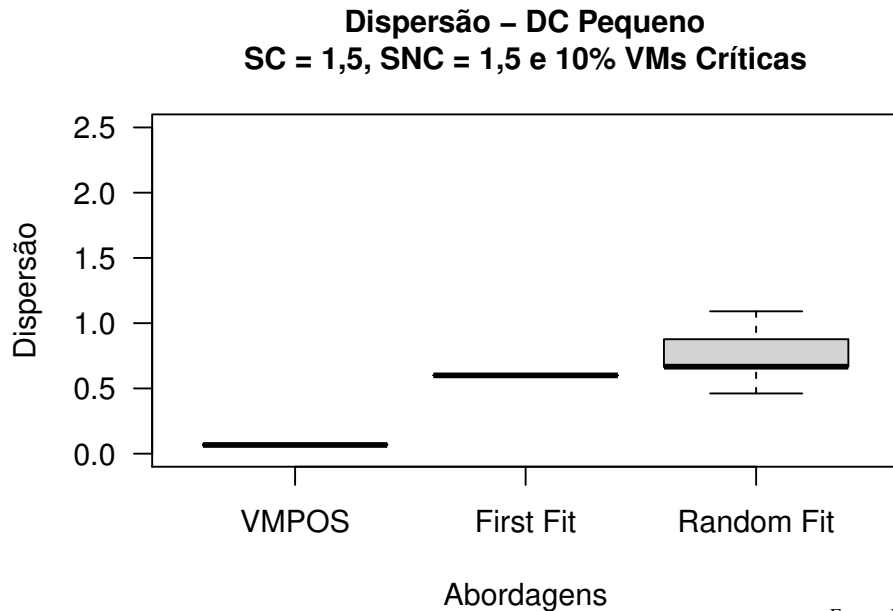


Figura H.47: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com SC=1,5 e SNC=1,5.

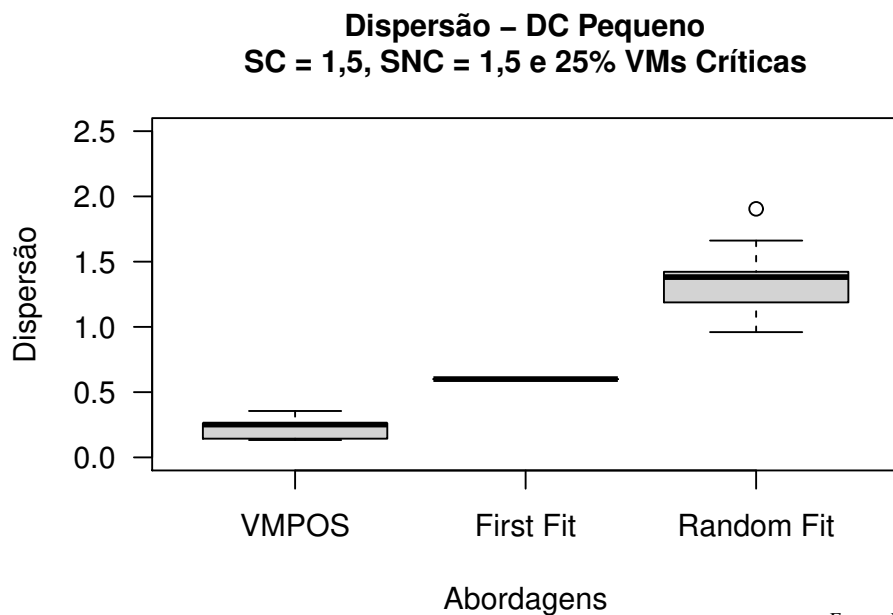


Figura H.48: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.

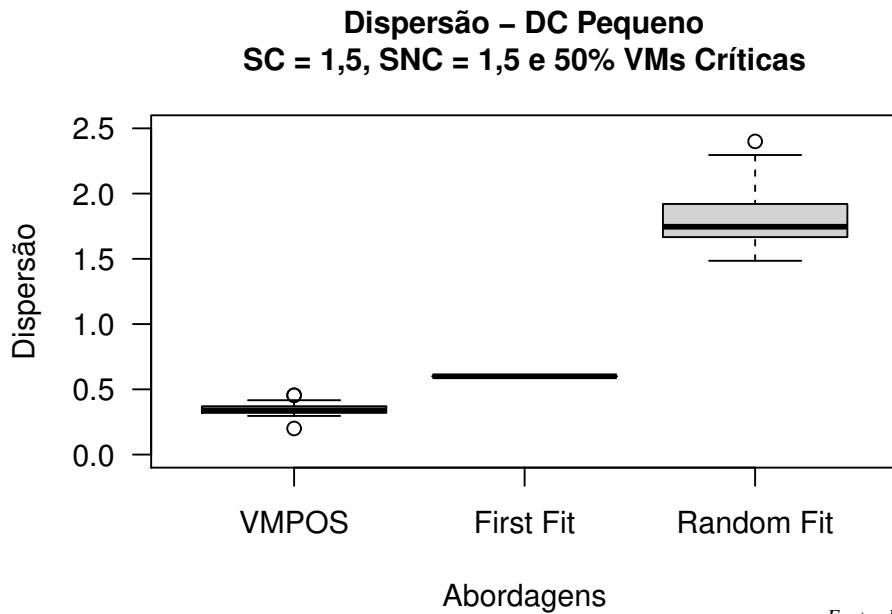


Figura H.49: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.

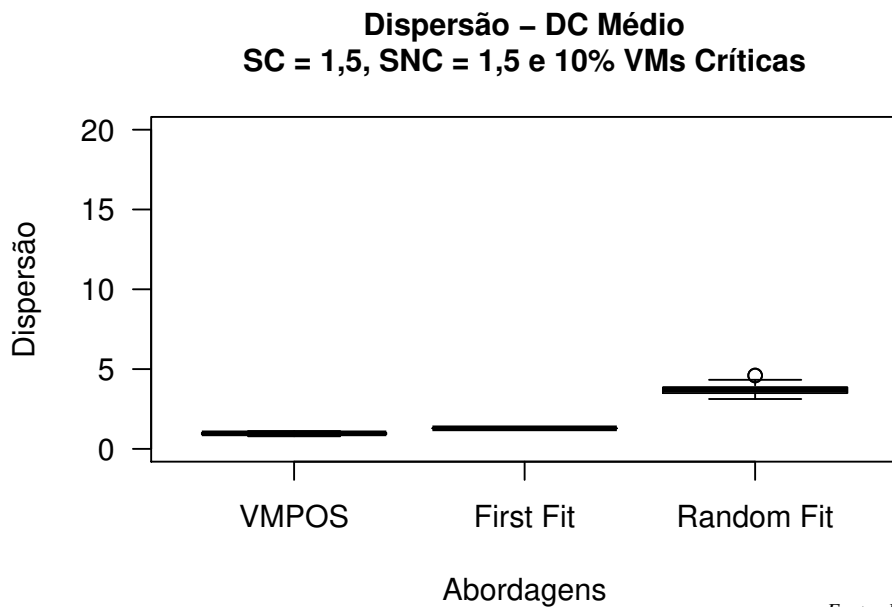


Figura H.50: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.

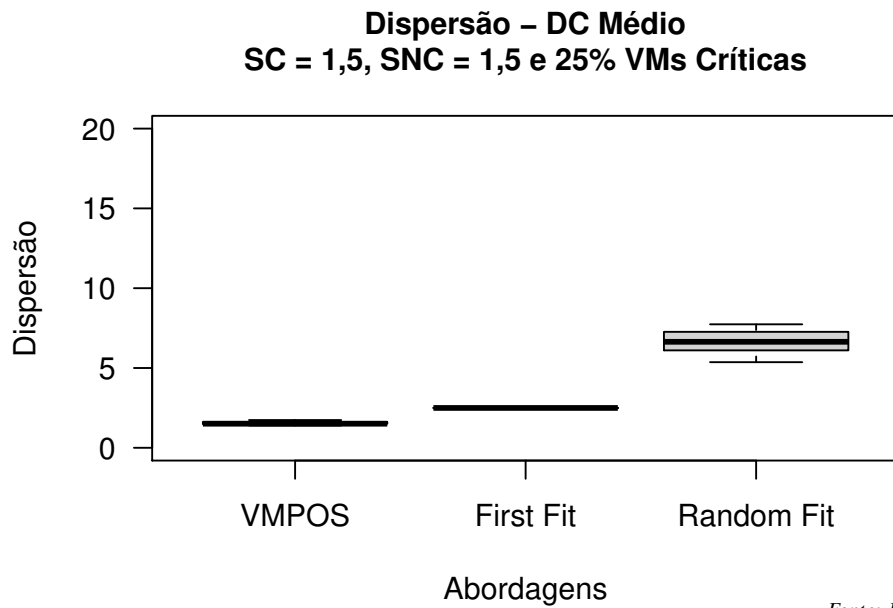


Figura H.51: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.

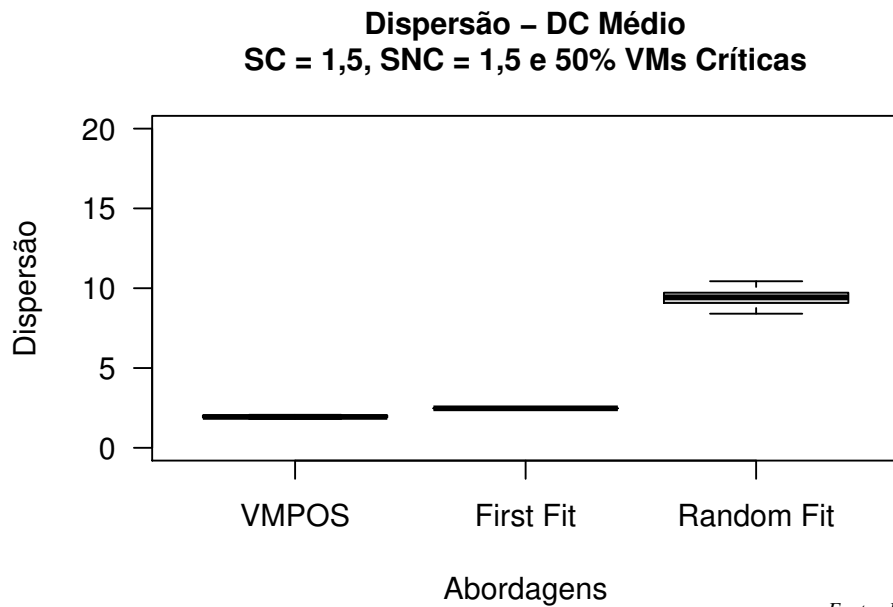


Figura H.52: *Boxplots* com as amostras de Dispersão, com 10% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.

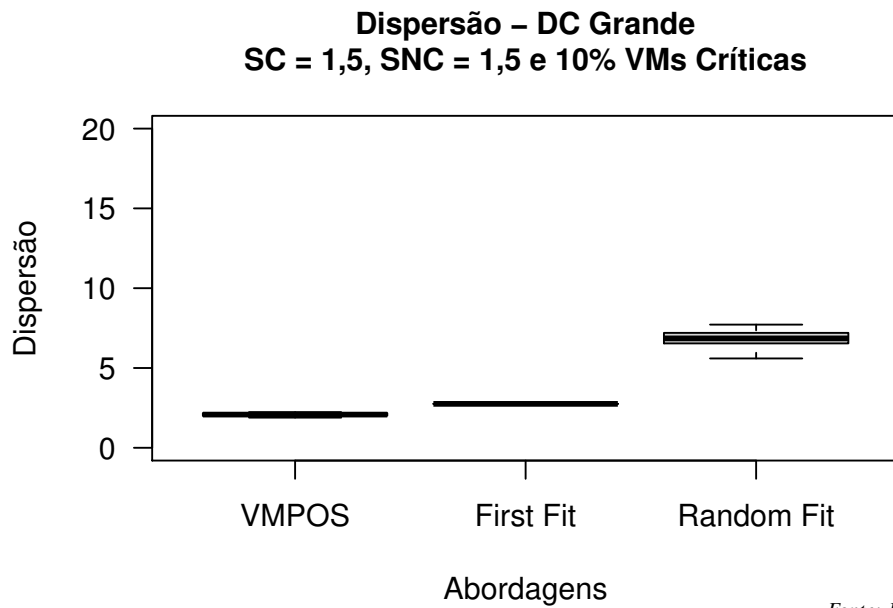


Figura H.53: *Boxplots* com as amostras de Dispersão, com 25% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.

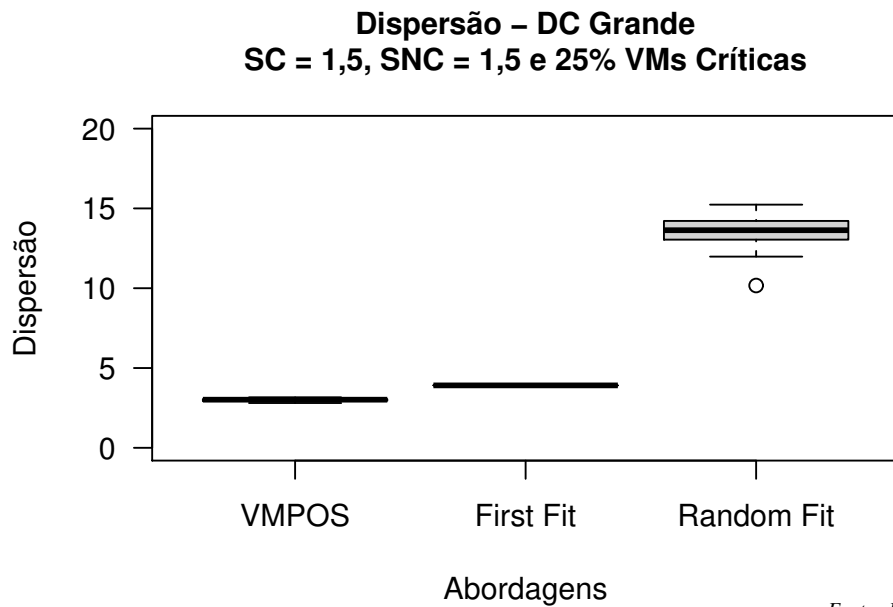
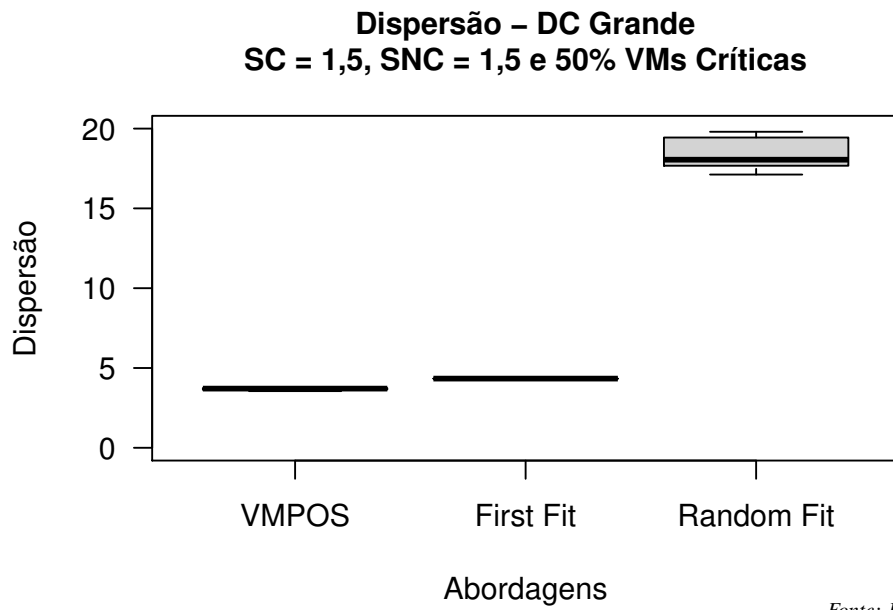


Figura H.54: *Boxplots* com as amostras de Dispersão, com 50% de VMs críticas e com $SC=1,5$ e $SNC=1,5$.



Fonte: Elaborada pelo autor.