UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE ESTATÍSTICA
BACHARELADO EM ESTATÍSTICA

Análise de Adaptação de Estratégia Multiobjetivo por Enxame de Partículas para Problemas com Filas em Rede

Marco Antônio Baia Costa

Ouro Preto-MG
2023

Marco Antônio Baia Costa

Análise de Adaptação de Estratégia Multiobjetivo por Enxame de Partículas para Problemas com Filas em Rede

Monografia de Graduação apresentada ao Departamento de Estatística do Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de bacharel em Estatística.

Orientador: Anderson Ribeiro Duarte

Ouro Preto

2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C837a Costa, Marco Antonio Baia.
Análise de adaptação de estratégia multiobjetivo por enxame de partículas para problemas com filas em rede [manuscrito]: análise de adaptação de estratégia multiobjetivo. / Marco Antonio Baia Costa. - 2023.
55 f.: il.: color., gráf..

Orientador: Prof. Dr. Anderson Duarte.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Graduação em Estatística .

1. Algorítmos. 2. Heuristica. 3. Otimização. I. Duarte, Anderson. II. Universidade Federal de Ouro Preto. III. Título.

CDU 31

Bibliotecário(a) Responsável: Luciana De Oliveira - SIAPE: 1.937.800

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
COLEGIADO DO CURSO DE ESTATISTICA

FOLHA DE APROVAÇÃO

Marco Antônio Baia Costa

Análise de adaptação de estratégia multiobjetivo por enxame de partículas para problemas com filas em rede

Monografia apresentada ao Curso de Estatística da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Estatística

Aprovada em 30 agosto de 2023

Membros da banca

Dr. Anderson Ribeiro Duarte - Orientador (Universidade Federal de Ouro Preto)
Dr. Helgem de Souza Ribeiro Martins (Universidade Federal de Ouro Preto)
Ms. Gabriel Lima de Souza (Universidade Federal de Ouro Preto)

Professor Dr. Anderson Ribeiro Duarte, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 30/08/2023.

Documento assinado eletronicamente por **Anderson Ribeiro Duarte**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 04/09/2023, às 15:43, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0584862** e o código CRC **1744FE8A**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3559-1312 - www.ufop.br

Agradecimentos

Gostaria de agradecer todos os professores que tive a possibilidade de aprender, principalmente os do departamento de estatística, em especial os professores Anderson Duarte, Ivair, Rivert, Spencer, Ricardo, Eduardo, Carolina, Fernando e Thiago e também a todos os meu familiares que me apoiam e acompanham minha jornada até o momento e que continuam ao meu lado. Serei sempre eternamente grato pela oportunidade de estudar e formar em uma faculdade pública. Obrigado a todos.

Resumo

Os processos compostos por redes de filas interdependentes estão presentes em diversas áreas. Na fase de elaboração, estes projetos precisam de atenção especial na alocação de recursos nas áreas de espera (*buffers*). A alocação adequada desses recursos gera economia financeira e melhoria em níveis de qualidade do atendimento à demanda. Redes de filas com servidor único, topologias acíclicas arbitrárias com chegadas markovianas e serviços gerais são consideradas neste estudo. Uma abordagem para alocação ótima de buffers em uma rede de filas com topologia em divisão é apresentada. A metodologia utiliza uma estratégia heurística multiobjetivo através do algoritmo *Particle Swarm Optimization*. O interesse central está em verificar os impactos das variações na capacidade de atendimento dos servidores e também de alterações nas probabilidades de roteamento na divisão de tarefas entre as filas da rede. Estes impactos tem repercussão direta nas soluções produzidas pela heurística de otimização. É importante verificar se mesmo com estas variações as soluções fornecidas são condizentes com a estrutura do problema de otimização em estudo. Diversos resultados de experimentos computacionais confirmam a eficácia da metodologia.

Palavras-chave: Otimização, Heurística, Rede de Filas, Particle Swarm Optimization, Alocação de recursos.

Abstract

Processes composed of queueing networks with interdependent queues are present in several areas. In the design, these projects need special attention in allocating resources in the buffer areas. The proper allocation of these resources generates financial savings and improvement in the service quality to demand. Single server queueing networks, arbitrary acyclic topologies with Markov arrivals, and general services are considered in this study. An approach for optimal buffer allocation in a queueing network with split topology is presented. The methodology uses a multiobjective heuristic strategy through the Particle Swarm Optimization algorithm. The main interest is verifying the impacts of variations in the service capacity and changes in the routing probabilities in the merge topology between the network queueing. These impacts have direct repercussions on the solutions produced by the optimization heuristic. It is essential to check whether, even with these variations, the provided solutions remain consistent with the structure of the optimization problem under study. Several results of computational experiments confirm the effectiveness of the methodology.

Keywords: Optimization, Heuristics, Queuing Network, Particle Swarm Optimization, Resource Allocation.

Lista de ilustrações

Sumário

1 Introdução

As filas são uma realidade presente no cotidiano das pessoas em várias situações. Por exemplo, ao aguardar atendimento em uma agência bancária, um supermercado, um hospital ou até mesmo em atendimentos virtuais com um *call center*, as pessoas se deparam com filas. Essas filas podem causar problemas como tempo de espera prolongado, frustração, atrasos e ineficiência no atendimento. O problema das filas pode ser matematizado por meio de formulações específicas, por meio de conceitos de teoria das filas e probabilidade. Essas formulações consideram a taxa de chegada dos clientes, o tempo de serviço, o número de servidores disponíveis e o tamanho da área de espera.

A descrição de um sistema de filas pode ser proposta através de uma estratégia com a descrição de um modelo estocástico para a chegada dos clientes que buscam por um determinado serviço, e que em seguida se retiram desse sistema após terem sua demanda pelo serviço atendida. Além disso, a descrição de um modelo estocástico para delinear o tempo consumido para a prestação do referido serviço é de igual importância. Esse contexto pode ser observado em uma enorma gama de exemplos práticos. A dinâmica da chegada dos clientes, impactada por sua demanda e seu fluxo de atendimento geram um cenário de incerteza que somente podem ser abordados de uma forma estocástica. Um exemplo bastante recorrente pode ser encontrada na área de computação, uma possível lista de tarefas e processos que aguarda para ser executada em uma unidade central de processamento, conforme discutido por Ahmed e Ouyang (2007) [1], Chen, Hu e Ji (2010) [4] e Inzillo, Rango e Quintana (2019) [12].

Em investigações para os sistemas de filas, nas filas com áreas de espera limitadas (*buffers* finitos) para um determinado serviço ofertado, têm-se o que usualmente são denominadas por filas finitas. Para filas finitas com espaço total para k clientes ($k-1$ em espera e um em atendimento), P_k denota a probabilidade de encontrar k usuários no sistema, com a inclusão daqueles que já estão em atendimento. Dessa forma, a probabilidade de bloqueio da fila em estudo é representada por P_k. Quando ocorre a chegada de um cliente em busca por serviço e as posições de espera estão todas ocupadas, o referido cliente é bloqueado pelo sistema. Diante disso, Cruz, Duarte; van Woesel (2008) [8] afirmam que altas probabilidades de bloqueio implicam na ineficiência do sistema de filas. A Figura 1 ilustra uma esquematização para uma fila dessa forma, λ_i representa a taxa de chegadas, existem $k-1$ locais de espera, um único servidor que atende de acordo com uma taxa de serviço μ_i e a taxa de atendidos é dada pelo parâmetro θ_i.

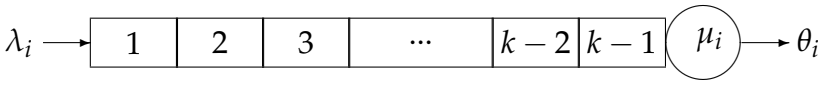


Figura 1 – Esquema ilustrativo de uma fila de servidor único com sua área de circulação.

Usualmente em um sistema de filas, as filas são interconectadas em rede e o fluxo de atendimento de uma determinada tarefa tem impacto relevante nas demais tarefas. Particularmente para sistemas de filas de servidor único, a alocação de recursos em áreas de espera é preponderante para o desempenho do sistema. Uma rede de filas pode ser representada com um grafo direcionado, em que as arestas interconectam as filas que são representadas pelos vértices do grafo. As entidades que percorrem a rede, transitam entre as filas para receber algum tipo de serviço. A Figura 2 exemplifica uma rede de filas que possui dezesseis nós e diversas possíveis rotas de acordo com o vetor de roteamentos $(p_1, p_2, p_3, p_4, p_5, p_6)$.

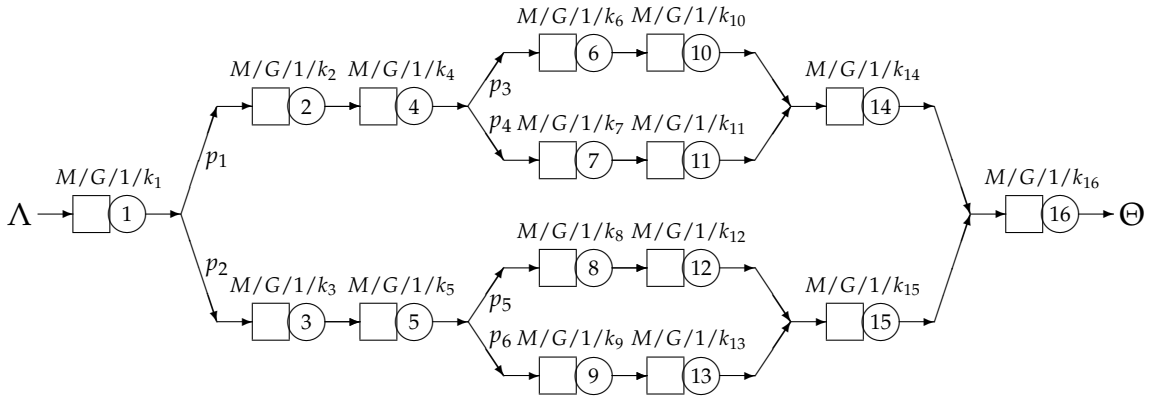


Figura 2 – Uma rede de filas, adaptada de MacGregor Smith e Cruz (2005) [16].

O problema de alocação de recursos para otimização de redes de filas é de grande interesse prático. Essa otimização diz respeito a diversos aspectos presentes na vida real. Esse tipo de estudo permite o auxília, a compreensão, e também inserção de melhorias em diversos sistemas presentes no cotidiano das pessoas, entre eles, processos industriais, sistemas de saúde, tráfego urbano, sistemas de comunicação, e outros [2,3,11,17,19,20].

Este estudo investiga a otimização de redes de filas finitas por meio de um método heurístico eficaz para maximizar simultaneamente o *throughput* (taxa de atendimento) da rede de filas e minimizar as capacidades em áreas de espera para redes acíclicas compostas por m filas $M/G/1/k$, de acordo com a clássica notação de Kendall [13], redes acíclicas de chegadas markovianas, tempos de atendimento geral, filas finitas de serviço único e o máximo de k clientes, com a inclusão daqueles em serviço.

Para obter o *throughput* máximo Θ e a capacidade total mínima $\sum_i k_i$, em uma dada topologia e sob uma taxa de chegada conhecida λ_i, o procedimento para o processo de otimização busca as coordenadas ótimas do vetor $\mathbf{k} = (k_1, k_2, \ldots, k_m)$, que determinem a configuração ideal para a rede de filas.

A estratégia heurísitca de otimização deve ser capaz de fornecer um conjunto Pareto de soluções sub-ótimas. A abordagem multiobjetivo também permite que o usuário eleve um objetivo (por exemplo, aumente a taxa de transferência) enquanto reduz outro objetivo (por exemplo, redução na alocação de *buffers*). Um algoritmo de otimização por enxame de partículas multiobjetivo (MOPSO) foi utilizado neste estudo. Versões preliminares deste algoritmo já haviam sido abordadas por Cruz, Duarte e Souza (2018) [23] e Souza et al. (2022) [24].

1.1 Motivação

Um problema de relevante interesse prático é avaliar o comportamento das soluções obtidos com respeito à variações na configuração da rede de filas sob estudo. O próposito central está na tarefa de obter uma melhor estratégia para alocar as áreas de circulação e verificar impactos decorrentes das capacidades dos servidores e das probabilidades de roteamento inerentes ao percurso da rede de filas.

O algoritmo de otimização aplicado nessa proposição, o *Particle Swarm Optimization* (PSO), é um algoritmo que replica uma analogia com a natureza várias espécies que optam por estratégias coletivas para extrair vantagens da sociabilidade. Exemplos clássicos podem ser vistos em: enxames de abelhas, colônia de formigas, revoada de pássaros entre diversos outros.

O algoritmo PSO foi proposto por Kennedy e Eberhart (1995) [14]. É um algoritmo bioinspirado, motivado pelo movimento de espécies de animais. O algoritmo reproduz o movimento de grupos e enxames na busca por alcançar o seu objetivo. Nessas situações, usualmente o alvo é a busca por melhores condições específicas para a espécie. Esse movimento é guiado pelo líder, mas também conta com a colaboração de cada um dos membros do enxame, a fim de guiar todo o enxame para uma melhor localização de bem estar da espécie.

O algoritmo PSO reproduz os movimentos de enxame por meio de equações matemáticas e faz com que um conjunto de partículas (soluções candidatas ao problema em estudo) se movimente em busca de um posicionamento ótimo. De maneira geral, as partículas são posicionadas inicialmente de maneira aleatória no espaço de soluções factíveis (viáveis). A cada iteração, sucessivos movimentos são realizados para esses pontos de acordo com as informações contidas nas próprias partículas, isso de acordo com as funções objetivo. Para movimentar cada ponto, basicamente calcula-se sua

direção e o tamanho de cada movimento. Essa operação é tratada como a velocidade da uma partícula desse enxame.

1.2 Objetivos

As informações anteriores são abrangentes para garantir justificativa e motivação desse estudo. Os objetivos gerais e específicos são apresentados em sequência.

1.2.1 Objetivos Gerais

i. discutir os aspectos de utilização da meta-heurística PSO;

ii. apresentar uma formulação matemática para o problema de otimização;

iii. comparar os impactos de variações na capacidade de serviço e roteamento da rede de filas.

1.2.2 Objetivos Específicos

- apresentar uma revisão bibliográfica na área de teoria de filas e problemas de otimização em filas;
- Uma formulação matemática multi-objetivo alternativa para o problema sob investigação;
- verificar eficácia do método heurístico utilizado para produzir soluções para o problema de rede de filas.

2 Fundamentação Teórica

O propósito central dessa investigação circunda os estudos associados aos problemas de otimização. Esse propósito conduz para a demanda de definir formamente os problemas de otimização e suas nuances.

2.1 Problemas de Otimização

De acordo com Yang (2010) [26], os estudos de otimização incluem extensa variabilidade de aplicações e contextos. Qualquer problema que exige a procura por algum nível de otimalidade se enquadra neste contexto. O cotidiano revela que a busca por soluções ótimas está presente nas mais variadas áreas. Por exemplo, todo negócio bem administrado busca maximizar os lucros, minimizar os gastos e até mesmo maximizar a satisfação dos envovlidos em seus processos. Existem formas de classificação para os mais diversos problemas de otimização. Uma classificação resumida pode ser observada na Figura 3 sugerida por Souza (2022) [22].

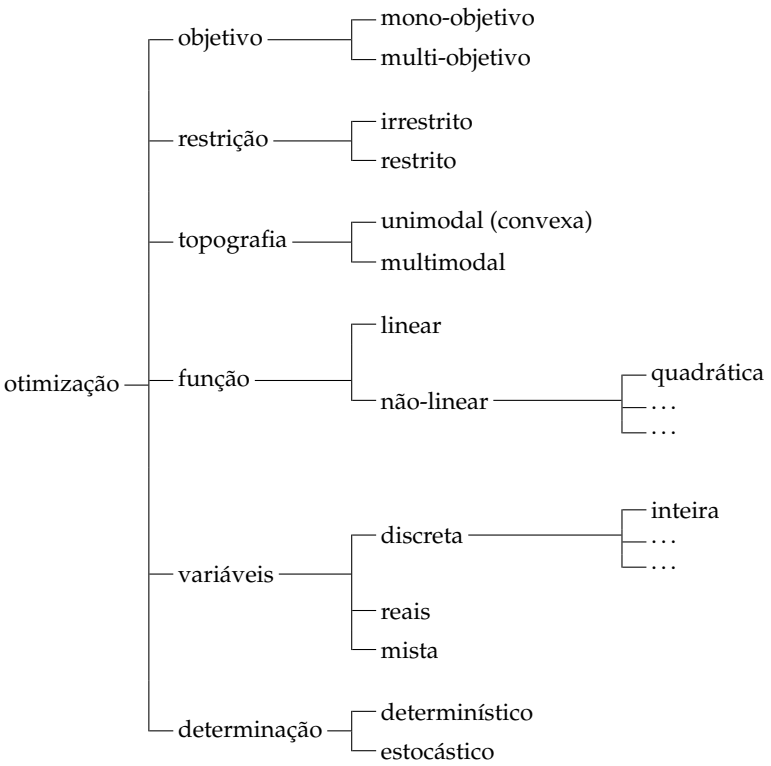


Figura 3 – Classificação dos problemas de otimização segundo Yang (2010) [26]

A discussão neste estudo recairá em um problema *multi-objetivo*, *restrito*, *unimodal*, *não-linear geral*, *inteiro* e *estocástico*. A literatura apresenta diversas técnicas para resolução

de problemas de otimização com essa estrutura. A escolha das técnicas é fortemente dependente da estrutura do problema. A complexidade de um problema de otimização depende, e muito, da sua função objetivo e do seu conjunto de restrições.

2.1.1 Otimização Multi-objetivo

Na maioria das situações, os problemas de otimização se movem na direção de otimizar funções com um ou mais de um objetivo. Os problemas chamados multi-objetivos, em geral, são os mais complexos e tem como solução não um valor único, mas sim um conjuto de soluções, uma família de vetores cuja dimensão é igual ao número de objetivos associados ao problema.

Resolver um problema de otimização multi-objetivo não é uma tarefa das mais simplístas. Determinar, de forma completa, o conjunto de soluções para estes problemas exige consumo de energia bastante representativo no procedimento de execução dessa tarefa. Souza (2022) [22] apresenta uma descrição dos problemas de otimização de forma bastante didática, este contexto apresentado foi a base central utilizada para a descrição que será colocada. Um problema de otimização multi-objetivo pode ser descrito da seguinte forma:

$$\text{minimizar } \big(f_1(x), f_2(x), \ldots, f_m(x)\big)^T$$

sujeito a:

$$x \in \mathcal{X} \subseteq \mathbb{R}^n$$

em que $f_i : \mathbb{R}^n \to \mathbb{R}, i \in \{1, 2, \ldots, m\}$ são funções objetivos, muitas vezes conflitantes. O vetor de decisão $x = (x_1, x_2, \ldots, x_n)^T$ pertence a região de soluções factíveis $\mathcal{X} \subseteq \mathbb{R}^n$.

A definição do conceito de solução ótima para um problema multi-objetivo é bastante específica e extremamente útil na abordagem de problemas dessa natureza, particularmente para problemas com diversos objetivos conflitantes. A finalidade da otimização multi-objetivo é encontrar as soluções que "minimizam" $f_i(x)$ no contexto de otimização. É importante ressaltar que o conceito de minimização ou maximização é análogo e não gera qualquer perda de generalidade na conceitualização do problema multi-objetivo. Para tanto, o conceito de dominância é bastante relevante. Dados os elementos $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, a relação de dominância $\prec$ é definida como:

$$\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}') \iff \mathbf{f}(\mathbf{x}) \leqslant \mathbf{f}(\mathbf{x}') \text{ e } \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}'),$$

em que $\mathbf{f}(\mathbf{x}) \leqslant \mathbf{f}(\mathbf{x}')$ se, e somente se, $f_i(x) \leqslant f_i(x')$, para todo $i = 1, \ldots, m$ e ainda, $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}')$ se, e somente se, existe algum $i \in \{1, \ldots, m\}$ tal que $f_i(x) \neq f_i(x')$.

Uma solução factível $\mathbf{x}^* \in \mathcal{X}$ é denominada solução Pareto-ótimo se existir $\mathbf{x} \in \mathcal{X}$ tal que $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}^*)$. Ao investigar um conjunto de soluções, o sub-conjunto $\mathcal{X}^*$ de $\mathcal{X}$, é possível determinar aquelas que são soluções Pareto-ótimo no que diz respeito apenas ao subconjunto $\mathcal{X}^*$ de soluções avaliadas. Estas soluções são denominadas soluções não-dominadas para o conjunto $\mathcal{X}^*$. O conjunto de todas as soluções não-dominadas, com respeito a um espaço investigado, é chamado usualmente de *fronteira*. Cruz et al. (2012) [9] ilustram de forma bastante didática a associação de dominância entre duas funções objetivos para minimização através da Figura 4.

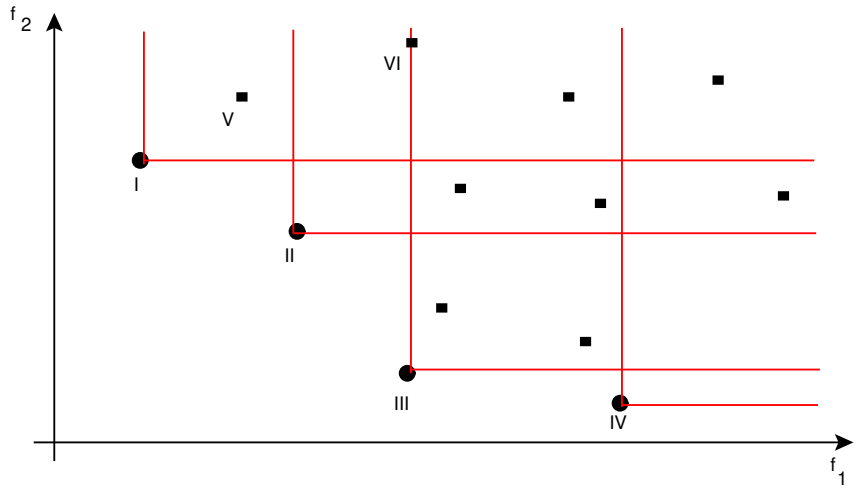


Figura 4 – Pontos dominados (▪) e não-dominados (•) retirados de Cruz et al. (2012) [9].

Observe que o ponto I domina os pontos V e VI, já o ponto II domina o ponto VI. Para concluir isso, verifique que o valor da função objetivo f_1 aplicada no ponto I é menor que quando aplicada aos pontos V e VI. Também ao avaliar a função f_2, o valor retornado para o ponto I é inferior que para os pontos V e VI. Por outro lado, para os pontos I, II, III e IV, não é possível definir uma relação de hierárquia, Isso porque para a função f_1 aplicada ao ponto I, retorna-se um valor menor que a mesma função aplicada ao ponto II, porém para a função f_2 aplicada ao ponto I o resultado é maior que em sua aplicação para o ponto II. O mesmo ocorre na comparação dos demais pontos III e IV. Portanto, os pontos I, II, III e IV são pontos não-dominados.

Para encontrar o conjunto de soluções não-dominadas no subconjunto das soluções sob investigação, diversas técnicas e estratégias podem ser utilizadas. Na maior parte dos problemas essa busca não é fácil e, muitas vezes, pode requerer um tempo de processamento computacional proibitivo.

2.2 Estratégia de Otimização

Usualmente denomina-se por heurística de otimização, a habilidade de buscar a solução de algum problema por alghum mecanismo de aproximação. O interesse está em fornecer uma solução que esteja suficientemente próxima da efetiva solução ótima para o problema em estudo. Em algumas situações, o processo de busca por tais soluções tende a ser extremamente custoso, diante disso requer alguma estratégia adequada para tal busca. Procedimentos que explicitam a mecânica desse processo de busca e também critérios de parada para tal busca são usualmente denominados estratégias heurísticos de otimização. Em particular, neste estudo, é abordada a estratégia *Particle Swarm Optimization Algorithm* (PSO) [14].

3 Abordagem do Problema

Para a investigação desse estudo, é relevante delimitar claramente a formulação matemática proposta para o problema em estudo e o algoritmo utilizado. O interesse deste estudo é fornecer soluções eficientes entre consumo de recursos e performance de atendimento. Estudos anteriores garantem a qualidade do algoritmo PSO no fornecimento dessas soluções, mas nenhum tipo de estudo acerca do padrão das soluções já foi apresentado na literatura

3.1 Formulação Mono-objetivo

O interesse deste estudo é verificar o comportamento do algoritmo PSO para otimizar (minimizar) a soma das capacidades $(\sum_i k_i)$ de uma rede acíclica de filas $M/G/1/k$, enquanto simultaneamente otimiza (maximiza) a taxa de atendimento (Θ). O algoritmo é muito dependente da formulação de programação matemática. Inicialmente é apresentada a formulação mais comum do problema de alocação de *buffers* (BAP).

O problema é definido através de um grafo $\mathcal{G}(V, A)$ em que V é um conjunto finito de m vértices (filas), e A é um conjunto finito de arestas (conexões entre os filas). O BAP, em sua formulação primal [16] tem a seguinte proposição:

$$\text{minimizar} \sum_{i=1}^{m} c_i k_i, \tag{3.1}$$

sujeito a:

$$\begin{aligned} &\Theta(\mathbf{K}) \geqslant \Theta_{\min}, \\ &k_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{3.2}$$

que minimiza o custo total de alocação de capacidade para a rede com m filas, sujeito a um limiar $\Theta_{\min}$ e capacidades totais inteiras k_i.

Nesta formulação mono objetivo, a taxa de atendimento é modelada como uma restrição, porém tal restrição precisa ser relaxada, o que caracteriza uma desvantagem para esta abordagem. Embora essa formulação BAP apresentada possa ser usada como auxílio para desenvolver algoritmos eficientes para resolver problemas de projeto de redes de filas, este estudo investiga um algoritmo baseado na formulação multiobjetivo apresentada em sequência.

3.2 Formulação Multiobjetivo

O problema de otimização de redes $M/G/1/k$, descrito na formulação anterior, pode ser reformulado em uma formulação de programação matemática multiobjetivo, que compreende a minimização das capacidades, simultaneamente com a maximização da taxa de atendimento. Este problema da rede de filas multiobjetivo pode ser expresso da seguinte forma :

$$\text{minimizar } F(\mathbf{K}, \boldsymbol{\mu}) = \Big[f_1(\mathbf{K}), f_2(\mathbf{K}) \Big], \tag{3.3}$$

sujeito a:

$$k_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \tag{3.4}$$

em que $f_1(\mathbf{K}) = \sum_{i=1}^{m} k_i$ representa as capacidades totais e $f_2(\mathbf{K}) = -\Theta(\mathbf{K})$ representa a taxa de atendimento. Observe o sinal negativo associado à taxa de atendimento, este formato faz com que o problema seja formulado apenas como um problema de maximização, isso sem perda de qualquer generalidade.

Ao considerar a formulação matemática aqui aventada para este problema, a sequência do procedimento exige calcular, para cada solução sob estudo, o valor das funções objetivos descritas nas Equações (3.3) e (3.4). O cálculo de $f_2(\mathbf{K})$ não é trivial e demanda a aplicação de mecanismos específicos de avaliação de performance. É necessária uma técnica eficaz de aproximação das probabilidades de bloqueio nas filas da rede. Assim como uma metodologia para obtenção da taxa de atendimento geral da rede de filas. O *método da expansão generalizado* (do inglês, *generalized expansion method*, GEM), desenvolvido por 15 é sabidamente eficaz para este propósito.

Investigações de problemas de redes de filas são abordadas de diversas perspectivas [5, 10, 21, 25]. Abordagens por meio de métodos de otimização são bastante comuns, por exemplo, o método de *Powell* [18], algoritmos genéticos [9], e *Simulated Annealing* [7] têm sido usados. Essas abordagens utilizam a taxa de atendimento (Θ), que geralmente é fornecida com o uso do GEM.

O GEM é um algoritmo que já foi usado com sucesso para estimar o desempenho de redes acíclicas de fila finita configuradas arbitrariamente em diversos estudos anteriores [7–9] entre outros. No processo de estimação, o GEM atualiza as medidas de desempenho do sistema em tentativas repetidas. O método considera o efeito de atraso gerado por diversos possíveis bloqueios ocorridos no fluxo de clientes ao longo da rede de filas. O GEM resolve um conjunto de equações não lineares simultâneas por meio de procedimentos iterativos. Isso leva a uma melhoria considerável na precisão da estimativa das medidas de desempenho da rede de filas. O método é uma combinação de decomposição *nó a nó* e tentativas repetidas, nas quais cada fila é analisada separada-

mente e as correções são feitas para levar em conta os efeitos inter-relacionados entre as filas da rede. Não é interesse deste estudo detalhar o funcionamento do GEM, uma descrição bastante detalhada pode ser encontrada em Kerbache e MacGregor Smith (1987) 15.

3.3 Algoritmo de Otimização por Enxame de Partículas

Um algoritmo de otimização de enxame de partículas multiobjetivo (MOPSO) é utilizado para resolver a formulação multiobjetivo desse estudo. Trata-se de uma adaptação da versão proposta inicialmente por Kennedy e Eberhart em 1995 [14]. No MOPSO em estudo, cada partícula deve representar uma possível solução para a alocação de recursos (capacidades k_i) que otimize a rede de filas finitas em estudo. Assim, nesta formulação particular, cada partícula pode ser representada pelas variáveis $(x_1, \ldots, x_m) = (k_1, k_2, \ldots, k_m)$.

Deve ser ressaltado que o problema de otimização multiobjetivo investigado é um problema com variáveis de decisão inteiras. Diante disso, uma estratégia de adaptação de partículas é necessária. De fato, as alterações nas capacidades são realizadas e, em seguida, os valores inteiros são usados, pois $k_i \geqslant 1$ é sempre respeitado. Da mesma forma, as restrições associadas às taxas de serviço também são respeitadas, pois é necessário garantir que $\rho < 1$. De outra forma, a taxa de chegada da fila deve ser estritamente menor que a taxa de serviço μ. Essas considerações garantem a viabilidade das soluções investigadas.

3.3.1 Detalhamento do algoritmo por enxame de partículas PSO

Considere s como o tamanho da população de partículas (tamanho do enxame), então cada partícula i, com $1 \leqslant i \leqslant s$ possui os seguintes atributos de definição:

- Posição das partículas $\mathrm{x_i} = (x_{1i}, x_{2i}, \ldots, x_{mi})$;
- Velocidade das partículas $\mathrm{v_i} = (v_{1i}, v_{2i}, \ldots, v_{mi})$;
- Melhor posição pessoal $(pbest)$ p_i;
- Melhor posição global$(gbest)$ g_i.

O interese do algoritmo PSO reside em reproduzir o movimento de partículas que trabalham de forma colaborativa. Para tanto, a posição e velocidade de deslocamento das particulas é monitorada ao longo do procedimento. Na formulação multi-objetivo, a posição da i-ésima partícula no espaço m-dimensional de busca é representada por $x_i = (x_{1i}, x_{2i}, \ldots, x_{mi})$. Já a velocidade da referida partícula é representada pelo vetor

$v_i = (v_{1i}, v_{2i}, \ldots, v_{mi})$. A melhor posição da i-ésima partícula durante as buscas é dada por $p_i = (p_{1i}, p_{1i}, \ldots, p_{mi})$. A velocidade e a posição das partículas são atualizadas da iteração t para a iteração $t+1$ conforme as equações:

$$v_i^{t+1} = w^t + r_1(p_i - x_i^t) + r_2(g_i - x_i^t), \tag{3.5}$$
$$x_i^{t+1} = x_i^t + v_i^{t+1}; \tag{3.6}$$

A abordagem MOPSO proposta para otimização da rede de filas pode ser descrita pela execução do seguinte algoritmo apresentado na Figura 5

```
algoritmo
  /* gera o enxame de partículas inicial */
  X ← GeraPopulaçãoInicial(swarmSize)
  P ← X
  /* encontre fronteiras não-dominadas F = (F_1, F_2, ...) */
  F ← OrdenaçãoNãoDominante(X)
  g ← random (F)
  para t = 1 até numIter faça
    para i = 0 até swarmSize faça
      v_i^{t+1} ← Velocidade(x_i^t, p_i, g)
      x_i^{t+1} ← NovaPosição(x_i^t, v_i)
      se x_i^{t+1} domina p_i então p_i ← x_i^{t+1}
        senão
          se p_i domina x_i^{t+1} então p_i ← p_i
            senão p_i ← random (x_i^{t+1}, p_i)
          fim se
      fim se
    fim para
    F ← OrdenaçãoNãoDominante(X)
    g ← random (F)
  fim para
  escreva F
fim algoritmo
```

Figura 5 – Algoritmo PSO multi-objetivo

A escolha da melhor posição da i-ésima partícula (p_i) é feita a cada iteração, da seguinte forma: se a nova posição é superior (em termos de dominância no conceito multi-objetivo) à posição p_i, a mesma é atualizada pela nova posição x_i^{t+1}. Se a posição atual é inferior (dominada) pela posição p_i, a posição p_i é mantida. Caso p_i não seja

superior ou inferior (pertencer a mesma classe em termos de dominância no conceito multi-objetivo) à posição atual x_i^{t+1}, a escolha é feita de maneira aleatória entre p_i e x_i^{t+1}. A melhor posição global (g_i) é escolhida aleatoriamente a cada iteração entre as partículas não dominadas.

Os parâmetros do algoritmo MOPSO foram definidos da seguinte forma: r_1 e r_2 são números aleatórios positivos com distribuição uniforme pertencente ao intervalo $[0, 1]$, $w(t)$ é o peso da inércia. O peso da inércia foi definido $w(t) = 0{,}4$. O MOPSO aqui descrito, é uma adaptação da implementação clássica apresentada por Coello-Coello & Lechunga [6].

4 Resultados Experimentais

O algoritmo de otimização discutido anteriormente foi implementado em FORTRAN. O ambiente de execução para realização dos experimentos computacionais foi um AMD FX(tm)-6300 Six-Core Processor 3.50 GHz, com sistema operacional Windows 10 Pro 64 bits, com 8,00 GB de memória RAM.

A rede de filas, apresentada na Figura 6, foi adaptada de MacGregor Smith & Cruz (2005) [16] e analisada com o método proposto. Esta rede é bastante adequada para os experimentos em estudo, isso porque inclui uma situação topológica específica de divisão.

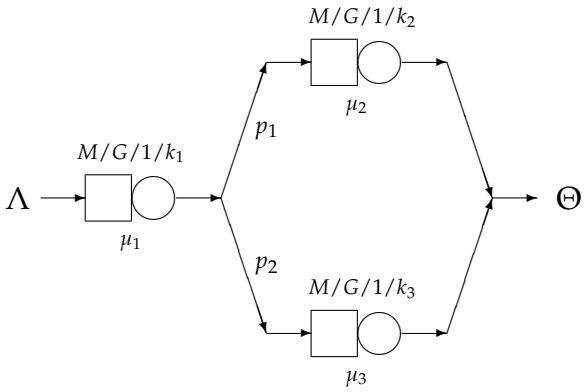


Figura 6 – Rede de filas com topologia divisão (adaptada de MacGregor Smith & Cruz [16]).

O interesse central é verificar se variações nas taxas de atendimento e probabilidades de roteamento afetam as soluções produzidas de uma forma padronizada. Configurações assimétricas de roteamento deveriam produzir estratégias de alocação de recursos assimétricas. Um padrão de variação nesta assimétria deveria produzir uma variação na estratégia de alocação de recursos padronizada. Isso corroboraria a confiabilidade nas soluções produzidas pelo algoritmo PSO.

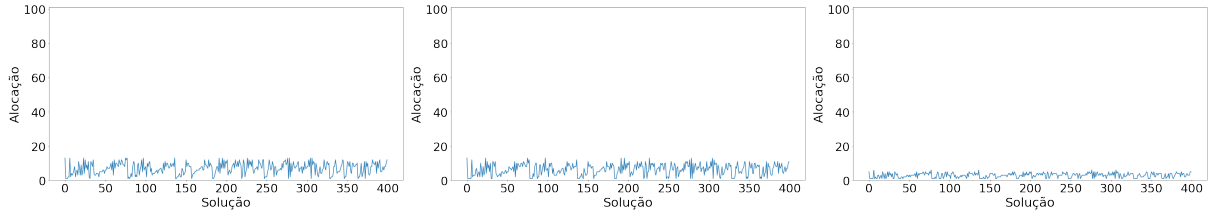
Foram analisados três valores distintos para os quadrados do coeficiente de variação $s^2 = \{0{,}5; 1{,}0; 1{,}5\}$ para caracterizar sistemas que são hipoexponenciais, exponenciais (markovianos) e hiperexponenciais, respectivamente. A taxa de chegada no sistema de filas foi sempre fixada em $\Lambda = 5{,}0$. Um conjunto de variações no vetor de roteamentos (p_1, p_2) e no vetor de taxas de atendimento (μ_2, μ_3) foi estabelecido afim de identificar padrões específicos de alocação e das taxas de atendimento produzidas.

A Tabela 1 apresenta as configurações utilizadas.

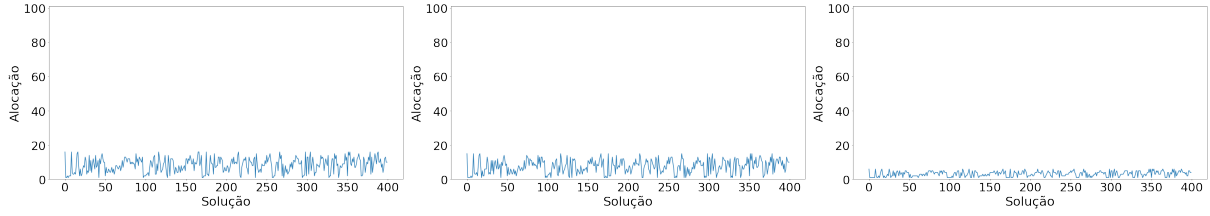
Tabela 1 – Configurações avaliadas para a rede de filas da Figura 6.

(p_1, p_2)	(μ_2, μ_3)	(p_1, p_2)	(μ_2, μ_3)	(p_1, p_2)	(μ_2, μ_3)
(0,5 0,5)	(5 15)	(0,6 0,4)	(5 15)	(0,7 0,3)	(5 15)
(0,5 0,5)	(10 10)	(0,6 0,4)	(10 10)	(0,7 0,3)	(10 10)
(0,5 0,5)	(15 5)	(0,6 0,4)	(15 5)	(0,7 0,3)	(15 5)
(0,8 0,2)	(5 15)	(0,9 0,1)	(5 15)		
(0,8 0,2)	(10 10)	(0,9 0,1)	(10 10)		
(0,8 0,2)	(15 5)	(0,9 0,1)	(15 5)		

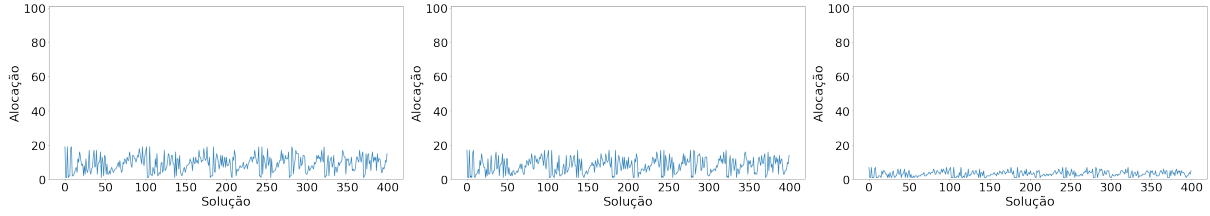
Para cada uma das configurações em estudo, são apresentados o comportamento da alocação nas três filas da rede. Ainda para cada configuração são apresentados os resultados para cada um dos quadrados do coeficiente de variação $s^2 = \{0{,}5; 1{,}0; 1{,}5\}$ para caracterizar sistemas que são hipoexponenciais, exponenciais (markovianos) e hiperexponenciais. As Figuras 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35 apresentam as alocações para cada uma das filas. Além disso, as Figuras 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36 apresentam o pareto-solução fornecido para analise das taxas de atendimento alcançadas.



(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

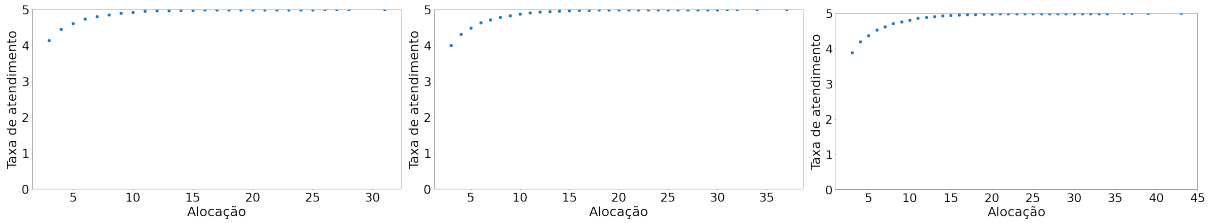


(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)



(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 7 – Resultados para configuração $p_1 = 0{,}5$, $p_2 = 0{,}5$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

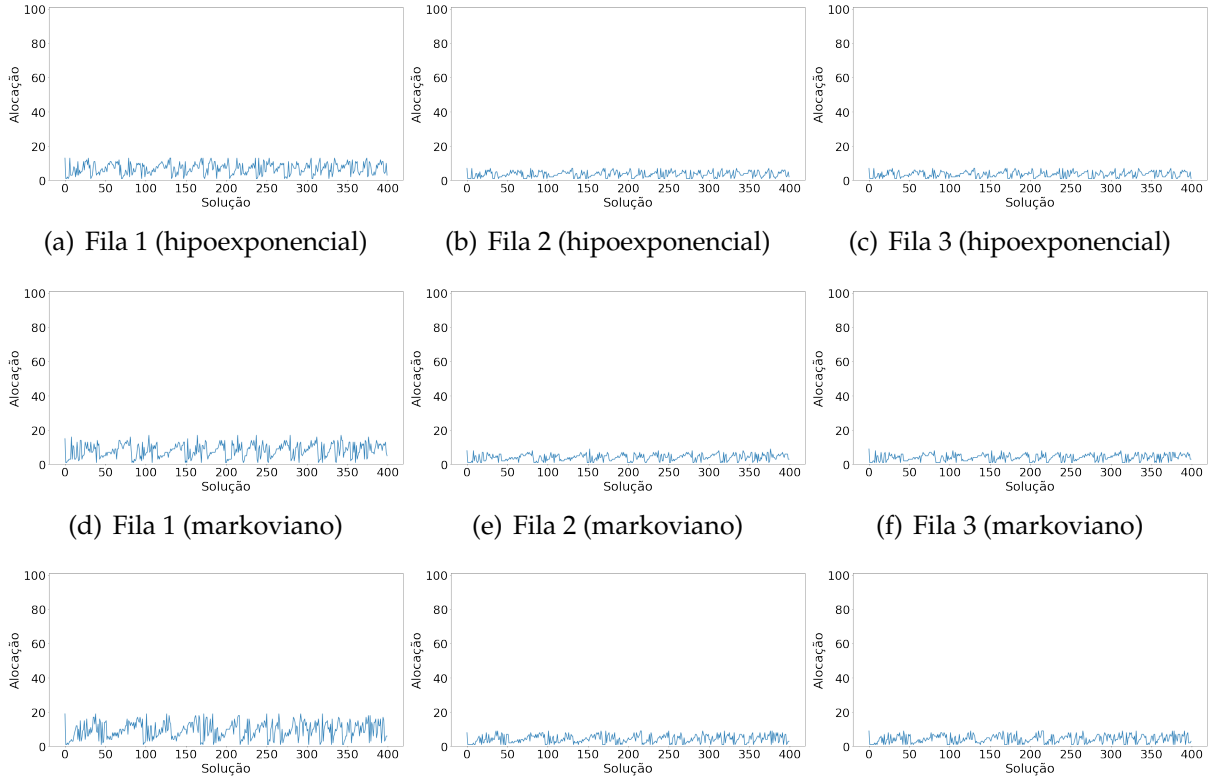
Figura 8 – Pareto-solução para a configuração $p_1 = 0{,}5$, $p_2 = 0{,}5$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

Para situação em que as probabilidades de roteamento são iguais, o grande impacto na estratégia de alocação de recursos nas soluções produzidas recai nas taxas de atendimento das filas 2 e 3. A Figura 7 ilustra a situação com $\mu_2 = 5$ e $\mu_3 = 15$. As soluções representam exatamente o formato previsto. Dada a menor capacidade de atendimento da fila 2, as soluções produzidas alocam mais *buffers* na fila 2. Este efeito ocorre tanto para sistemas hipoexponenciais, markovianos e hiperexponenciais. Entretanto, é visível que à medida que a variabilidade aumenta (hiperexponencial tem mais variabilidade que markoviano que tem mais variabilidade que hipoexponencial) as alocação para as filas 1 e 2 aumentam.

A Figura 8 confirma este efeito, o pareto-solução apresentado em cada situação mostra que o algoritmo é capaz de fornecer uma boa cobertura do espaço de soluções. Para as três situações a taxa de atendimento fica bem próxima da taxa total para alocações totais entre 15 e 20 unidades de buffers. Entretanto, as soluções de menor alocação total, ou seja, soluções mais baratas resultam em taxas de atendimentos menores à medida que a variabilidade no sistema de atendimento cresce.

Na sequência, variações nas taxas de serviço serão apresentadas para a mesma configuração das Figuras 7 e 8. A Figura 9 apresenta a situação em que as taxas se igualam, com taxas $\mu_2 = 10$ e $\mu_3 = 10$. A configuração hiperexponencial possui maior variabilidade com maior quantidade de *buffers* alocados na fila 1 e diminui esta quantidade com as filas subsequentes. A configuração hipoexponencial utiliza uma menor quantidade de *buffers* para as filas. Para as três configurações, a quantidade de *buffers* alocados decai de acordo com a próxima fila. Comparativamente com a figura 25 em que $p_1 = 0{,}6$ e $p_2 = 0{,}4$ e $\mu_2 = 10$ e $\mu_3 = 10$, os resultados demostram maior variabilidade nas soluções e quantidade *buffers* alocados na fila 3 é reduzida.

A Figura 10 apresenta leve superioridade ao ser comparada com o pareto fornecido na Figura 8. Para as soluções de baixa alocação, as taxas de atendimento são ligeiramente superiores. É possivel que o equilíbrio entre os servidores das filas 2 e 3 seja responsável por essa melhoria.

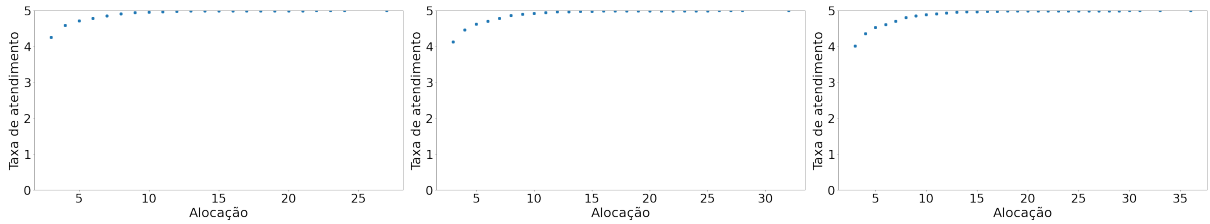


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

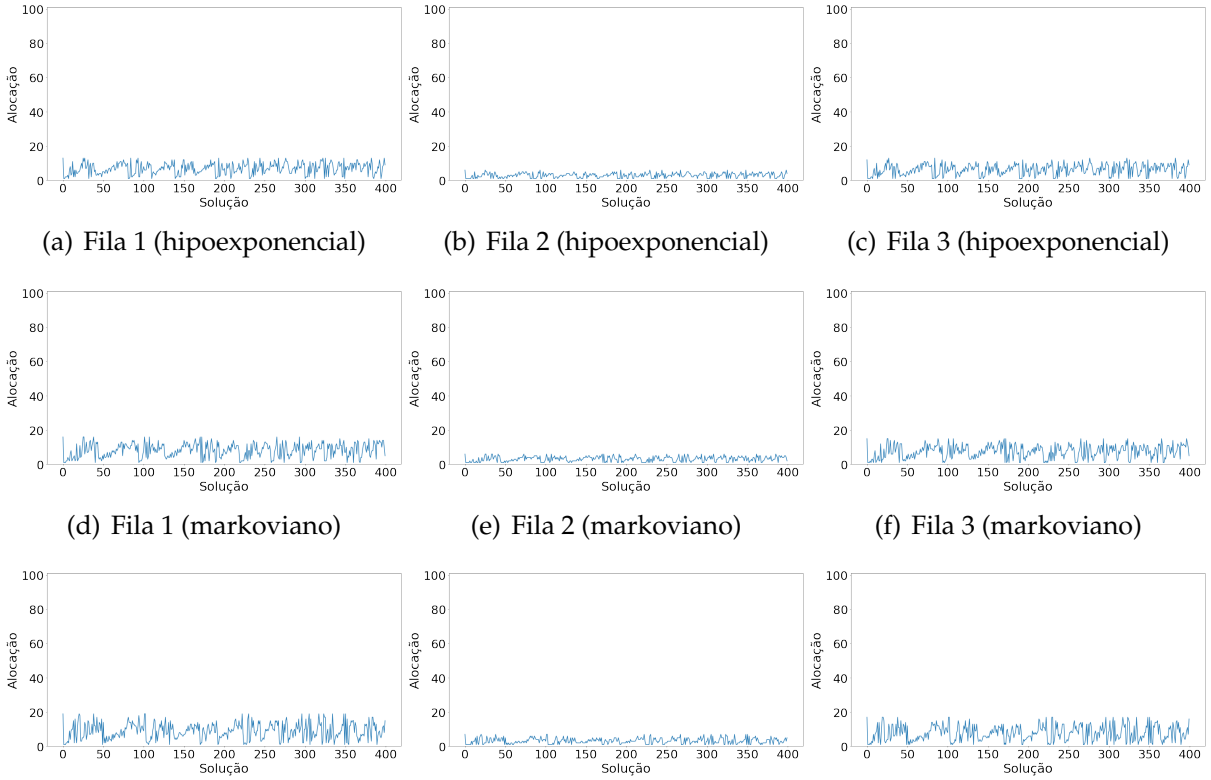
Figura 9 – Resultados para configuração $p_1 = 0{,}5$, $p_2 = 0{,}5$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 10 – Pareto-solução para a configuração $p_1 = 0{,}5$, $p_2 = 0{,}5$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

A seguir, o efeito das taxas de serviço ($\mu_2 = 5$ e $\mu_3 = 15$), das Figuras 7 e 8 é invertido. Para garantir a adequabilidade do algoritmo proposto ao problema, seria previsível encontrar soluções similares com a inversao dos efeitos das filas 2 e 3. As Figuras 11 e 12 relatam o cenário para $\mu_2 = 15$ e $\mu_3 = 5$.

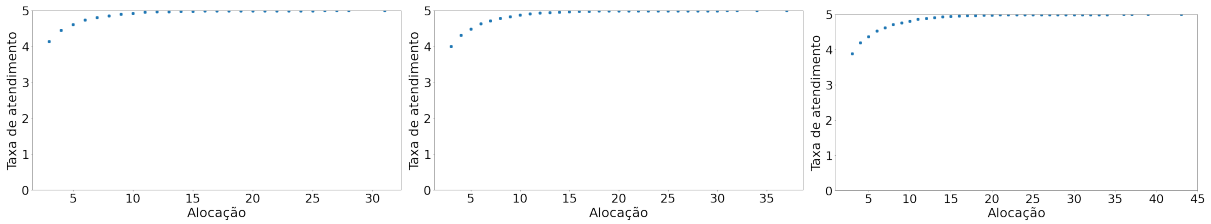


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 11 – Resultados para configuração $p_1 = 0{,}5$, $p_2 = 0{,}5$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).

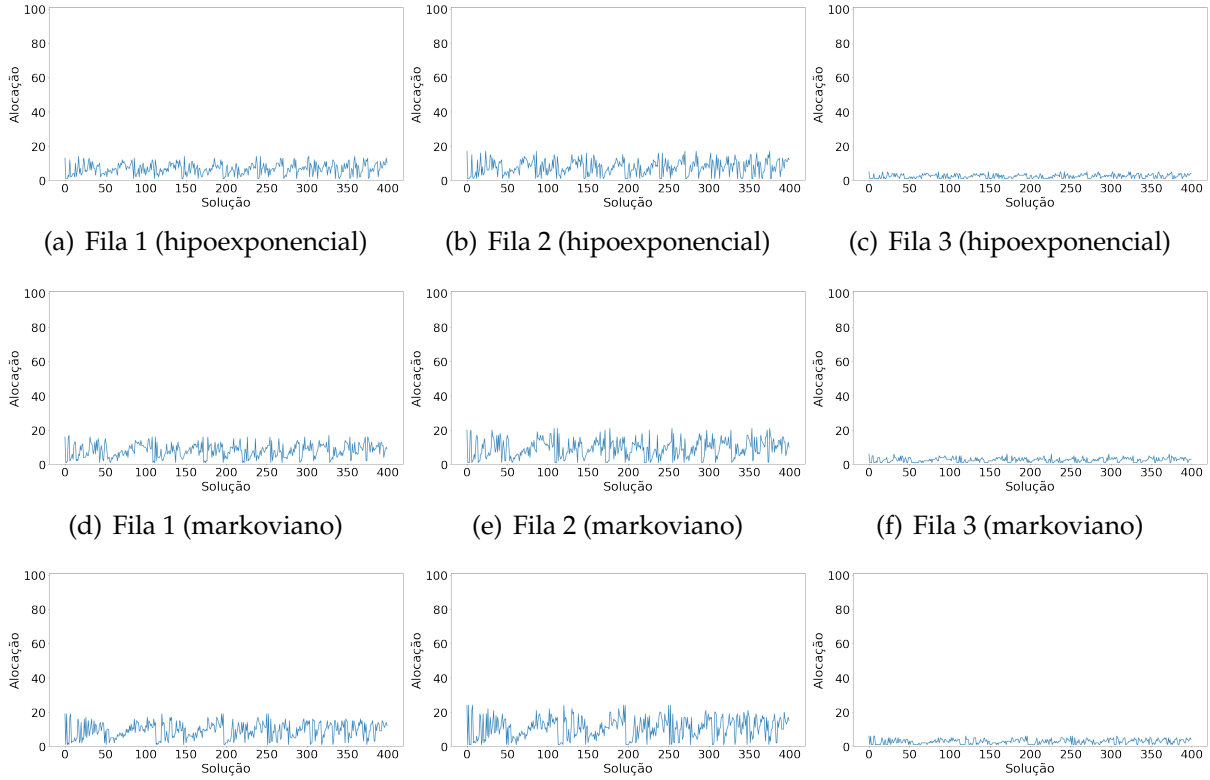


(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 12 – Pareto-solução para a configuração $p_1 = 0{,}5$, $p_2 = 0{,}5$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

De fato, o efeito de inversão pode ser claramente observado. Ocorre um clara inversão da estratégia de alocação de *buffers* entre as filas 2 e 3.

Baseado na investigação até aqui, variações nas estratégias de roteamento são propostas, mas com as relações anteriores nas taxas de serviço preservadas. As Figuras com as diferentes relações de roteamento serão apresentadas em sequência e uma análise mais abrangente será apresentada posteriormente.

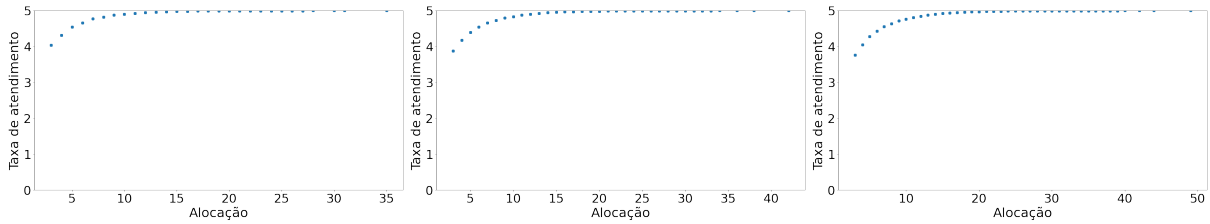


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

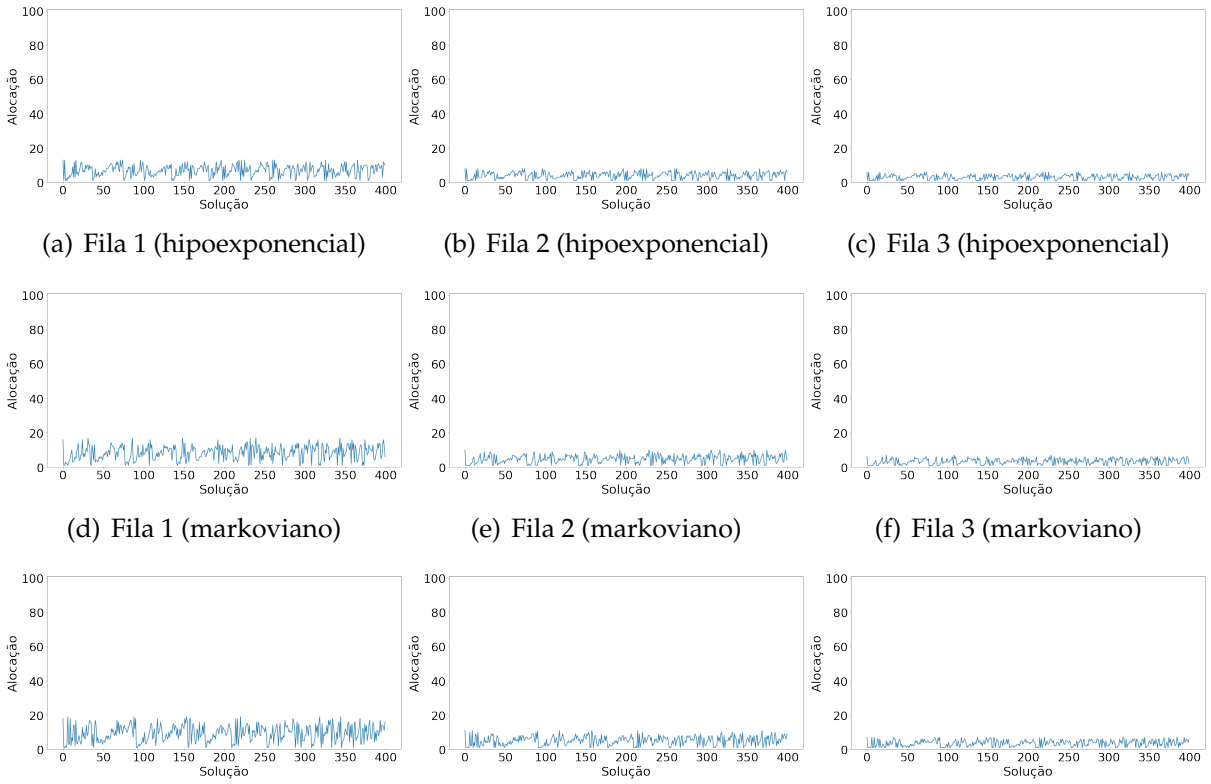
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 13 – Resultados para configuração $p_1 = 0{,}6$, $p_2 = 0{,}4$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 14 – Pareto-solução para a configuração $p_1 = 0{,}6$, $p_2 = 0{,}4$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

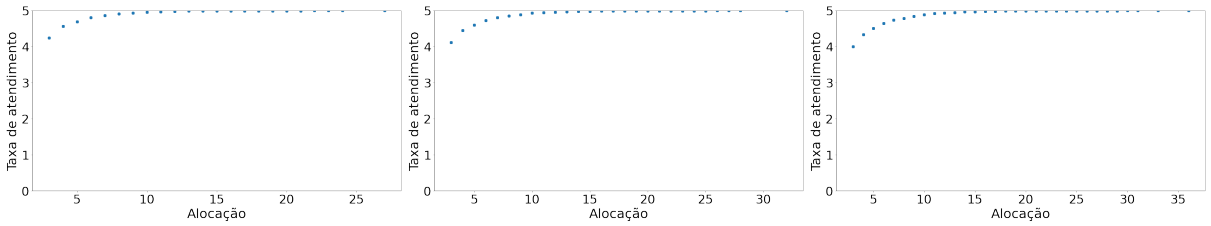


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

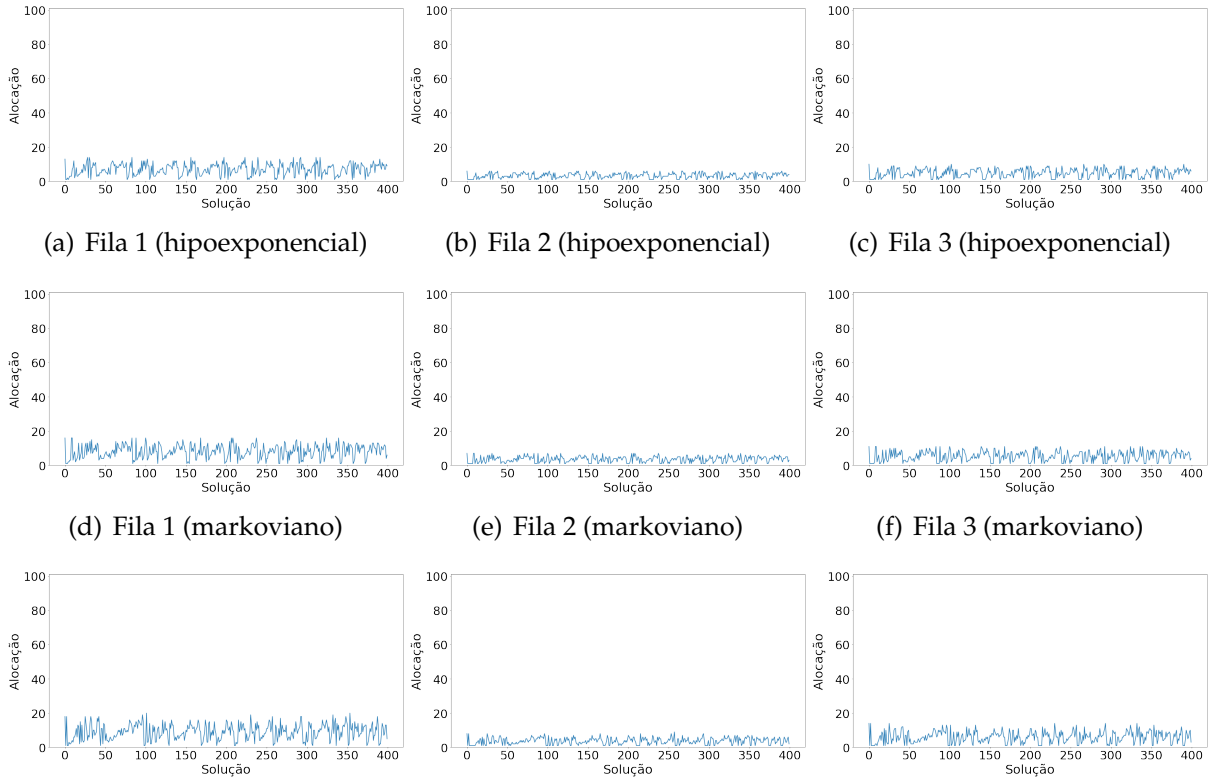
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 15 – Resultados para configuração $p_1 = 0{,}6$, $p_2 = 0{,}4$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 16 – Pareto-solução para a configuração $p_1 = 0{,}6$, $p_2 = 0{,}4$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

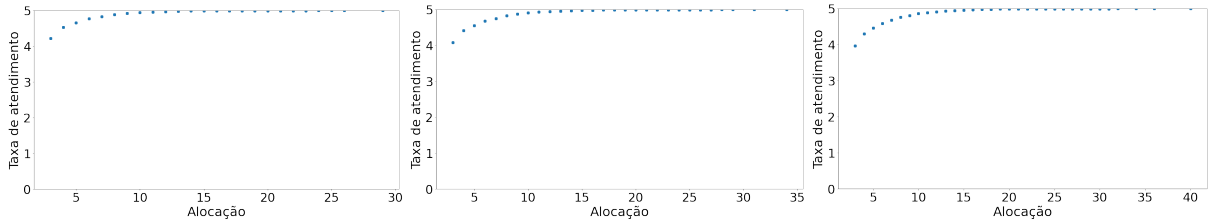


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

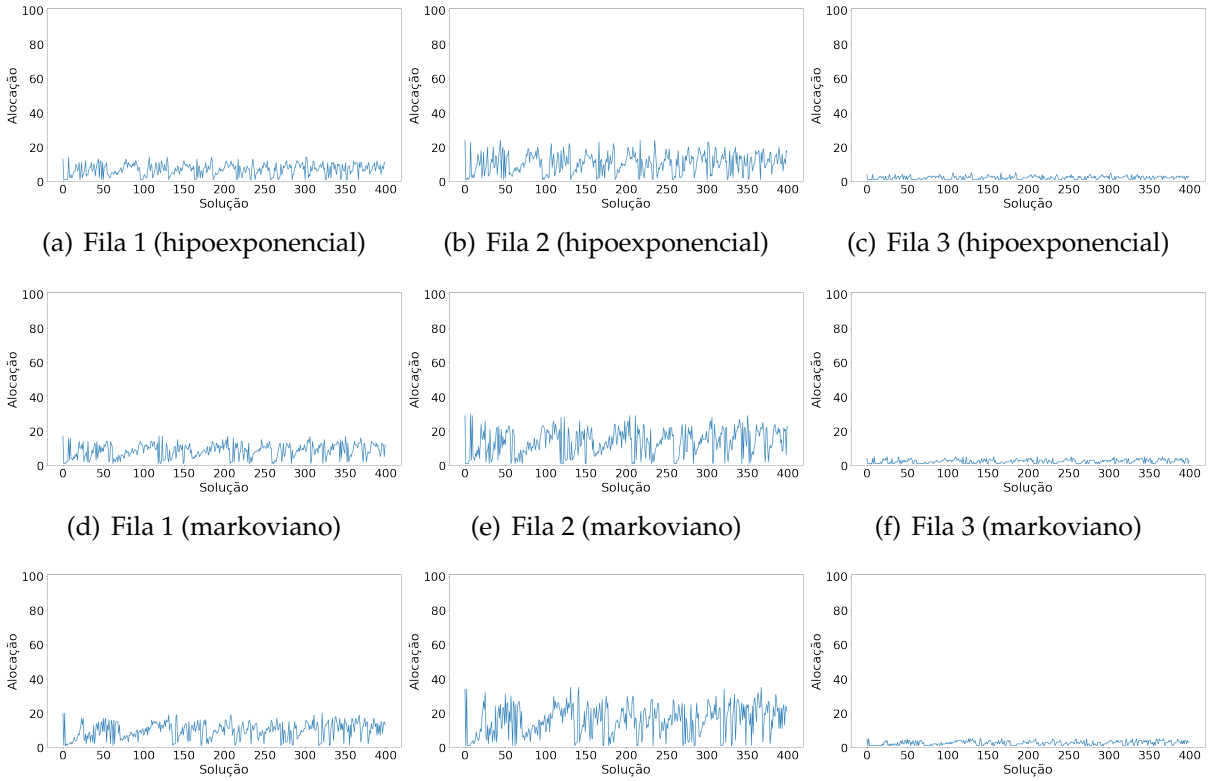
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 17 – Resultados para configuração $p_1 = 0{,}6$, $p_2 = 0{,}4$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 18 – Pareto-solução para a configuração $p_1 = 0{,}6$, $p_2 = 0{,}4$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

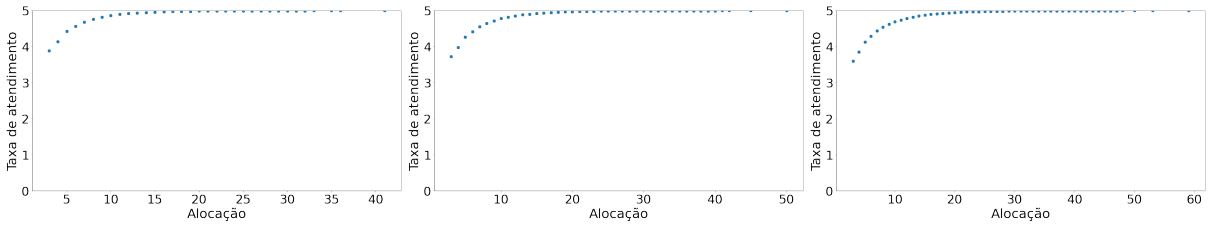


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

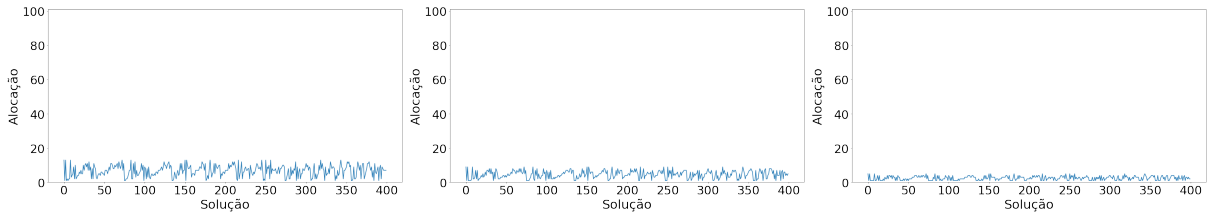
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 19 – Resultados para configuração $p_1 = 0{,}7$, $p_2 = 0{,}3$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).

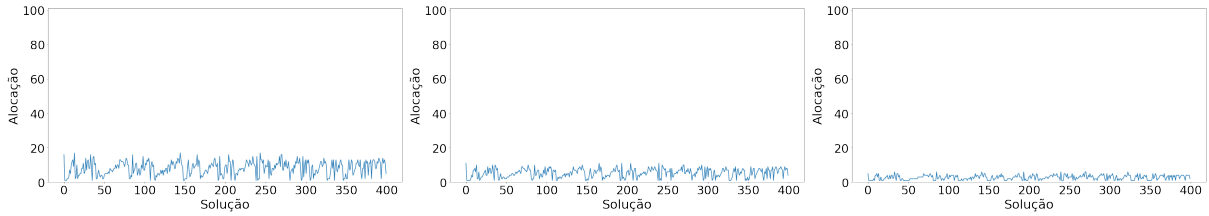


(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

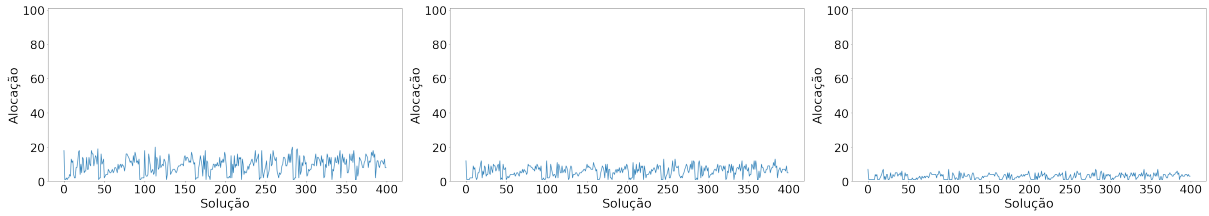
Figura 20 – Pareto-solução para a configuração $p_1 = 0{,}7$, $p_2 = 0{,}3$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial, markoviano e hiperexponencial.



(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

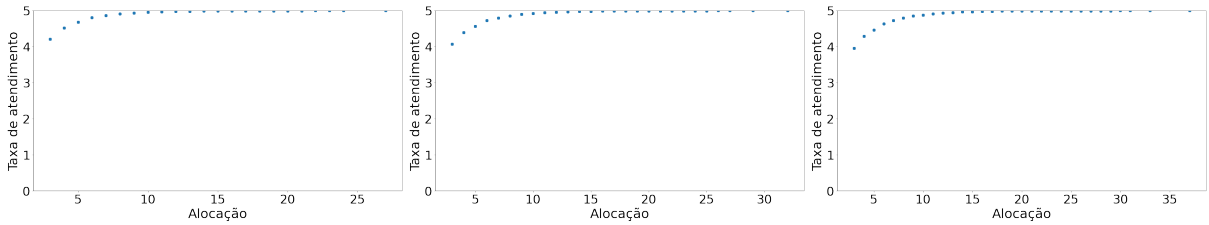


(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)



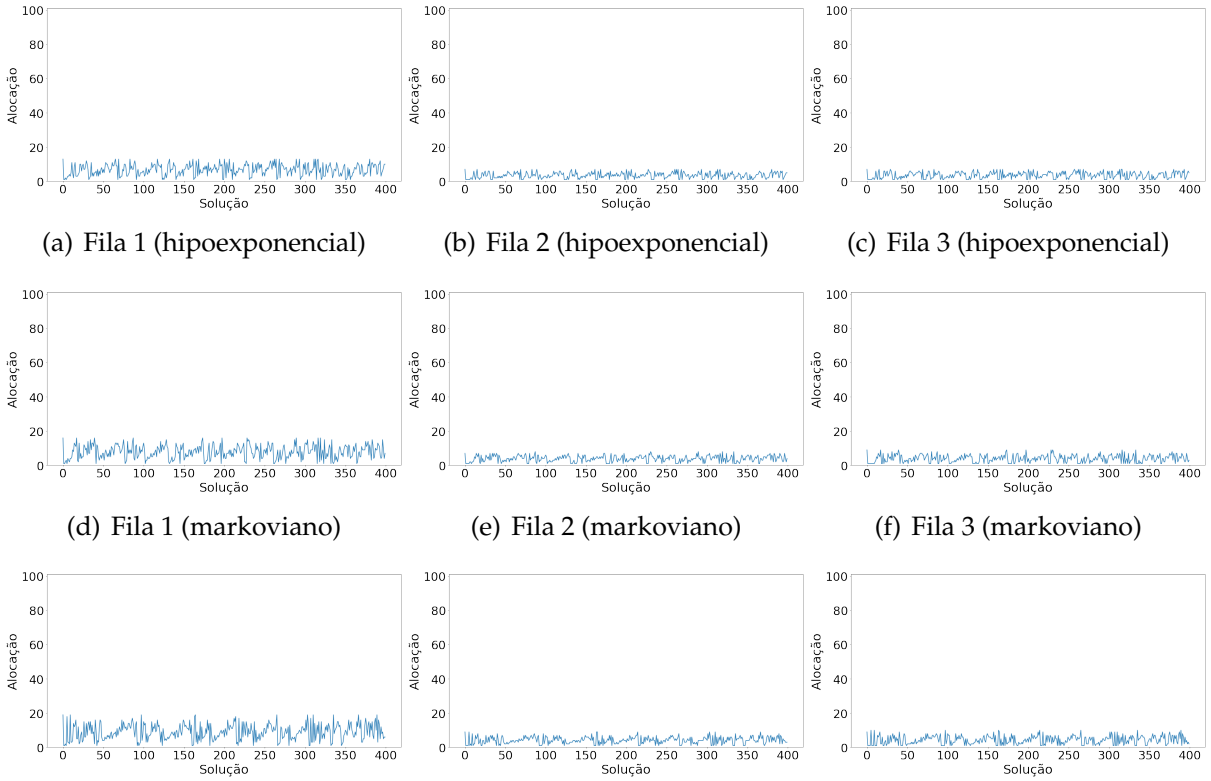
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 21 – Resultados para configuração $p_1 = 0{,}7$, $p_2 = 0{,}3$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 22 – Pareto-solução para a configuração $p_1 = 0{,}7$, $p_2 = 0{,}3$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

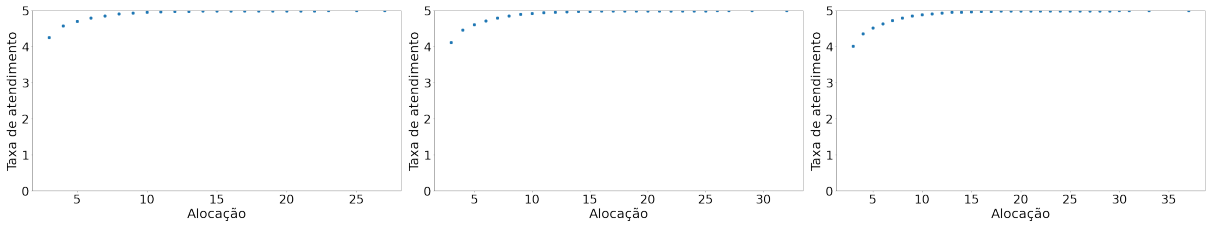


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

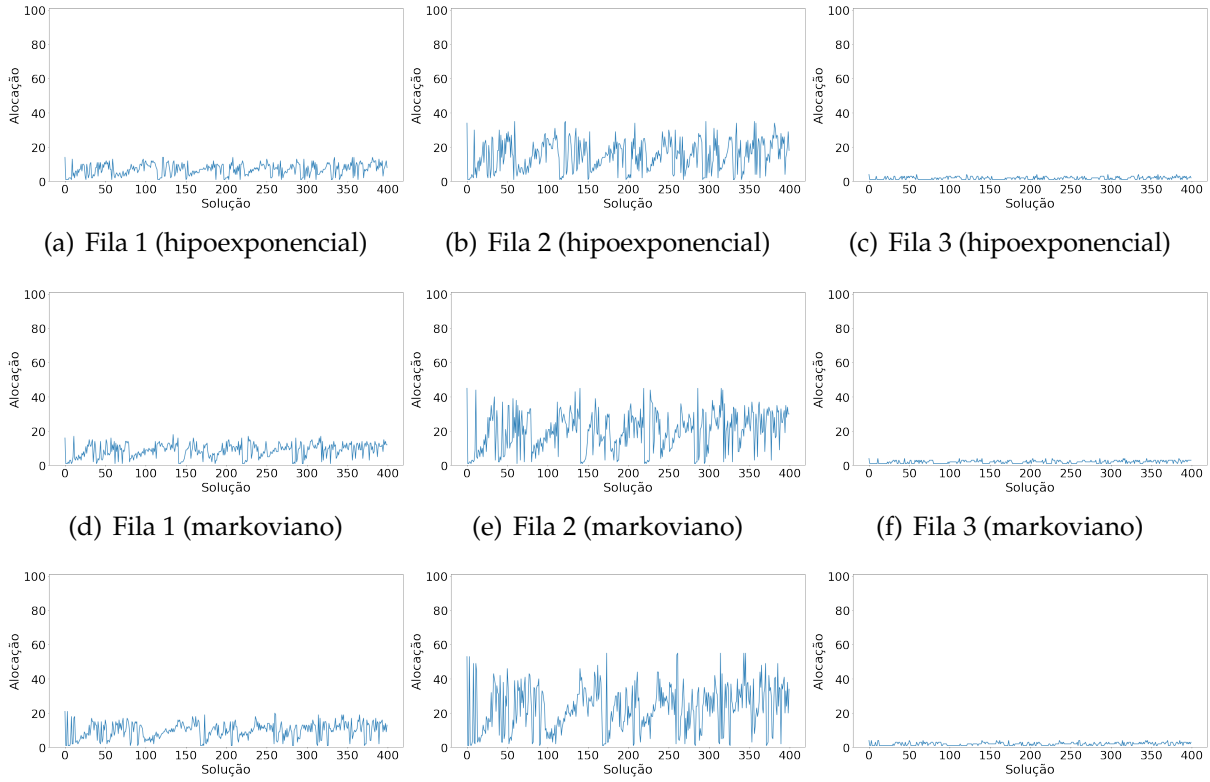
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 23 – Resultados para configuração $p_1 = 0{,}7$, $p_2 = 0{,}3$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 24 – Pareto-solução para a configuração $p_1 = 0{,}7$, $p_2 = 0{,}3$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

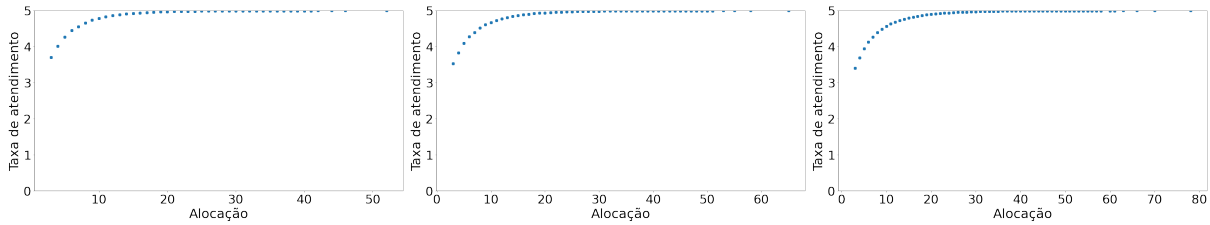


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

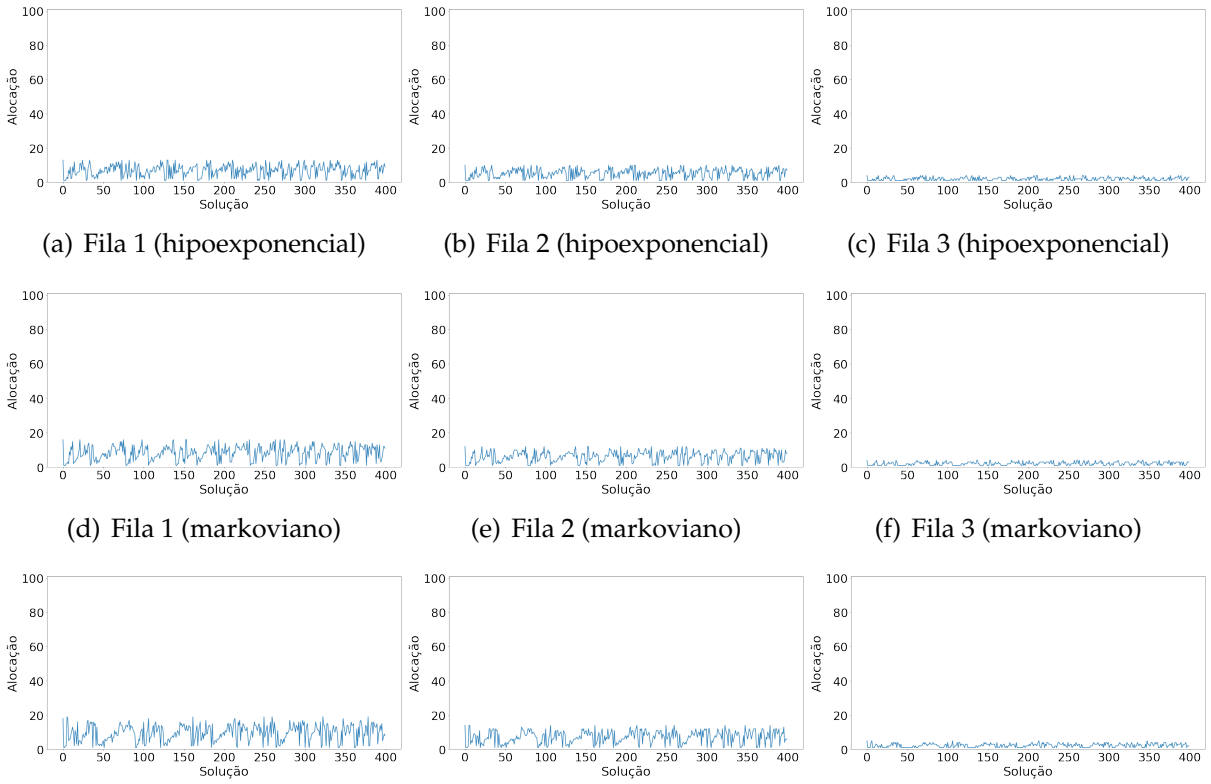
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 25 – Resultados para configuração $p_1 = 0{,}8$, $p_2 = 0{,}2$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 26 – Pareto-solução para a configuração $p_1 = 0{,}8$, $p_2 = 0{,}2$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

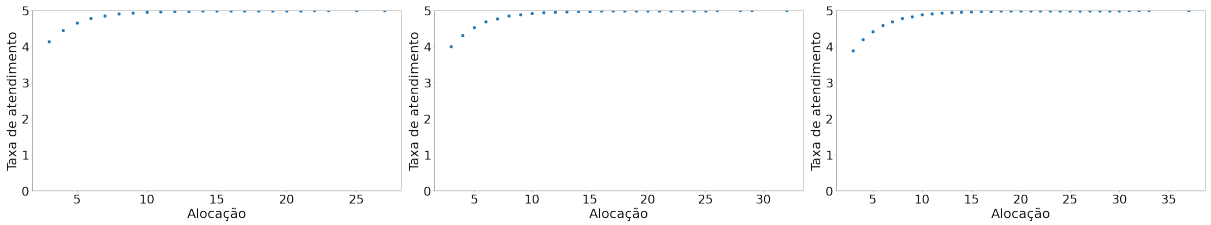


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

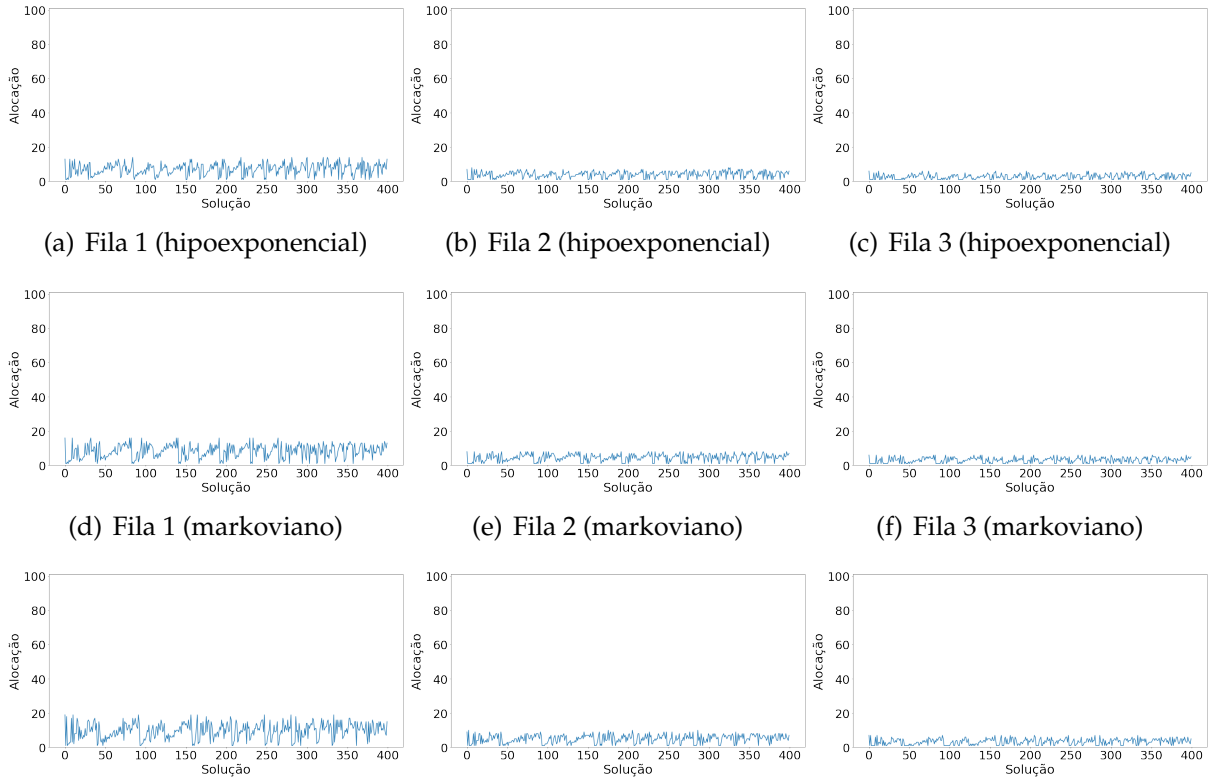
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 27 – Resultados para configuração $p_1 = 0{,}8$, $p_2 = 0{,}2$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 28 – Pareto-solução para a configuração $p_1 = 0{,}8$, $p_2 = 0{,}2$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

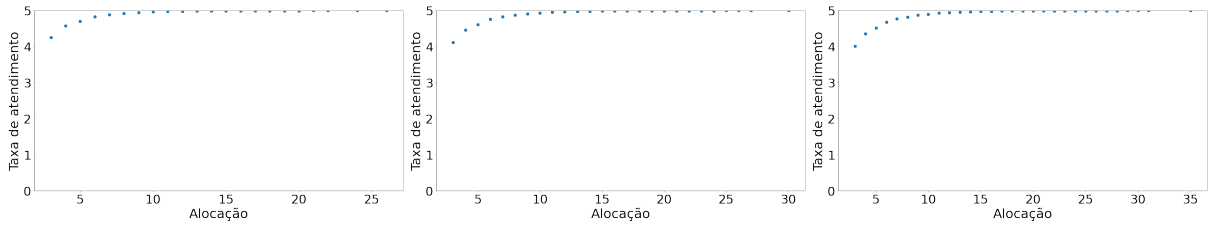


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

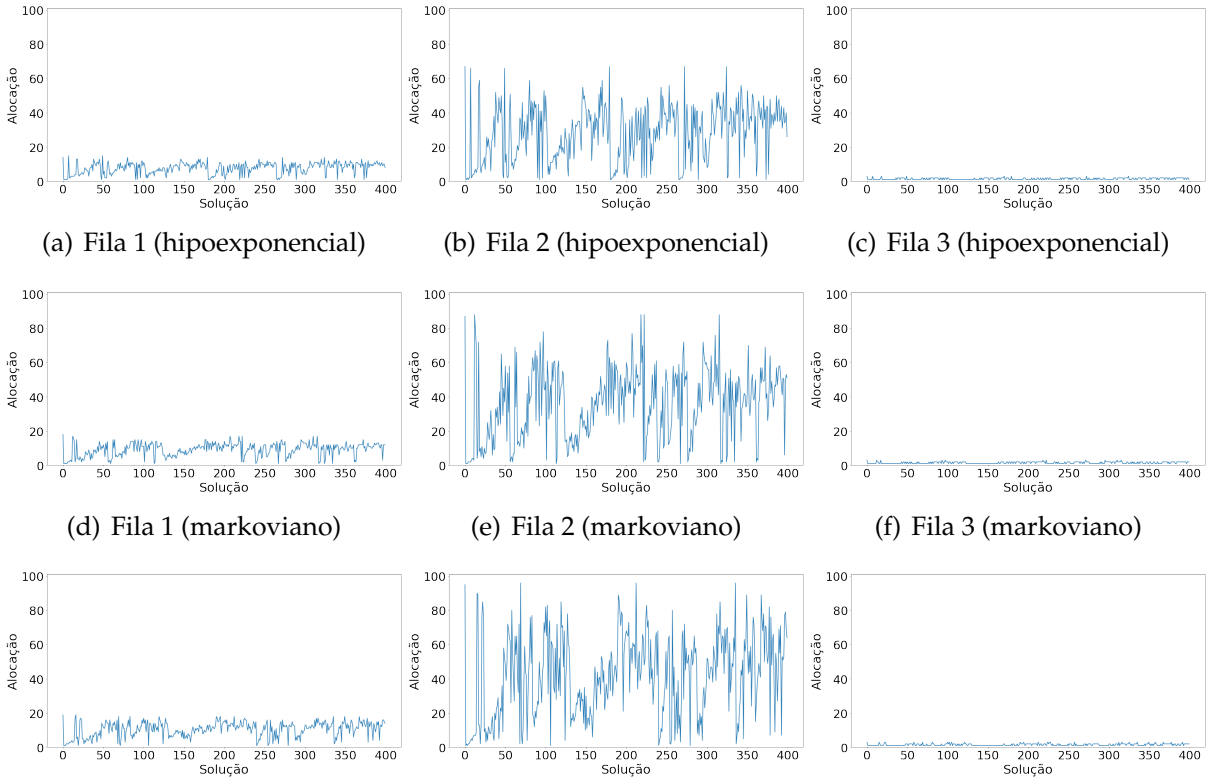
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 29 – Resultados para configuração $p_1 = 0{,}8$, $p_2 = 0{,}2$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 30 – Pareto-solução para a configuração $p_1 = 0{,}8$, $p_2 = 0{,}2$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

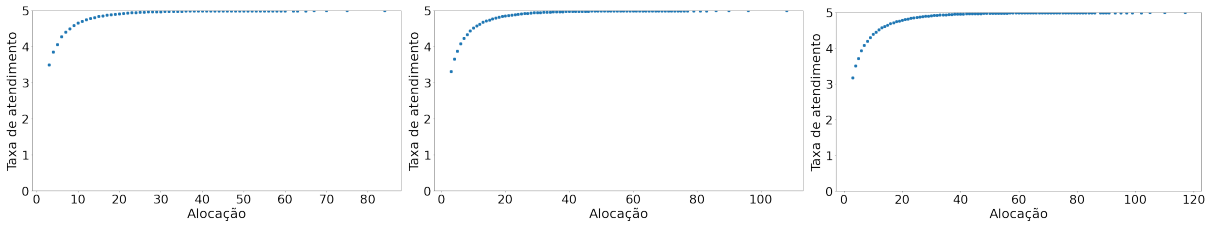


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

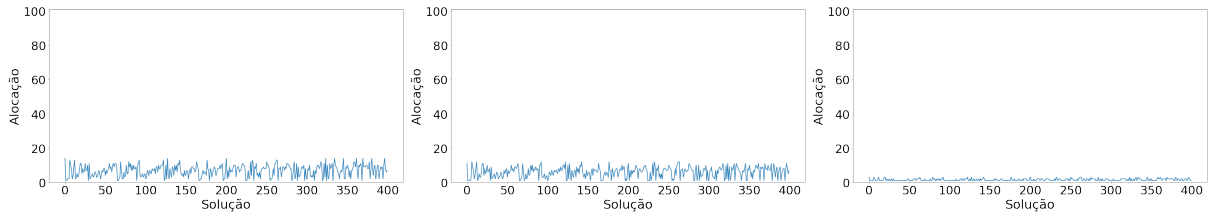
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 31 – Resultados para configuração $p_1 = 0{,}9$, $p_2 = 0{,}1$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).

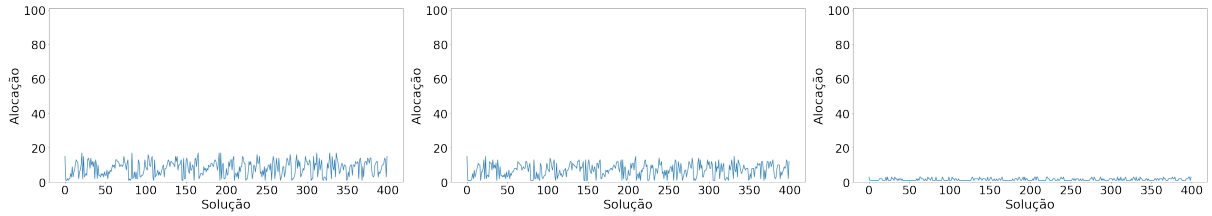


(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

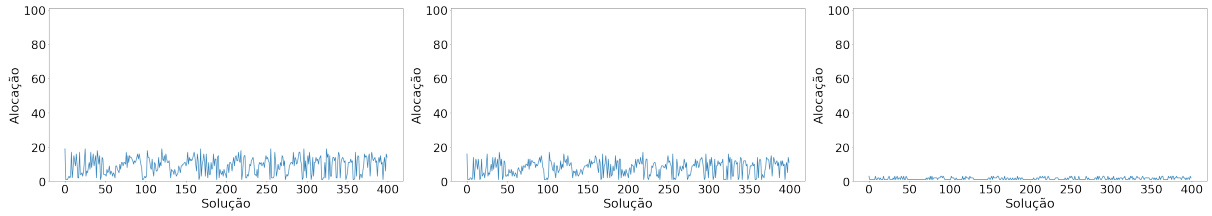
Figura 32 – Pareto-solução para a configuração $p_1 = 0{,}9$, $p_2 = 0{,}1$, $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 15$ com atendimentos hipoexponencial, markoviano e hiperexponencial.



(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

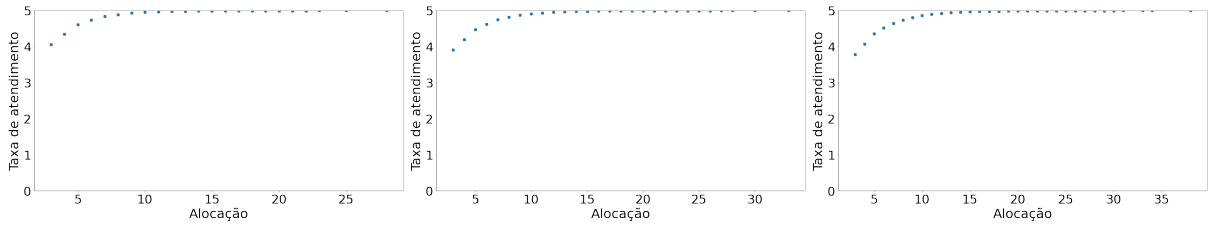


(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)



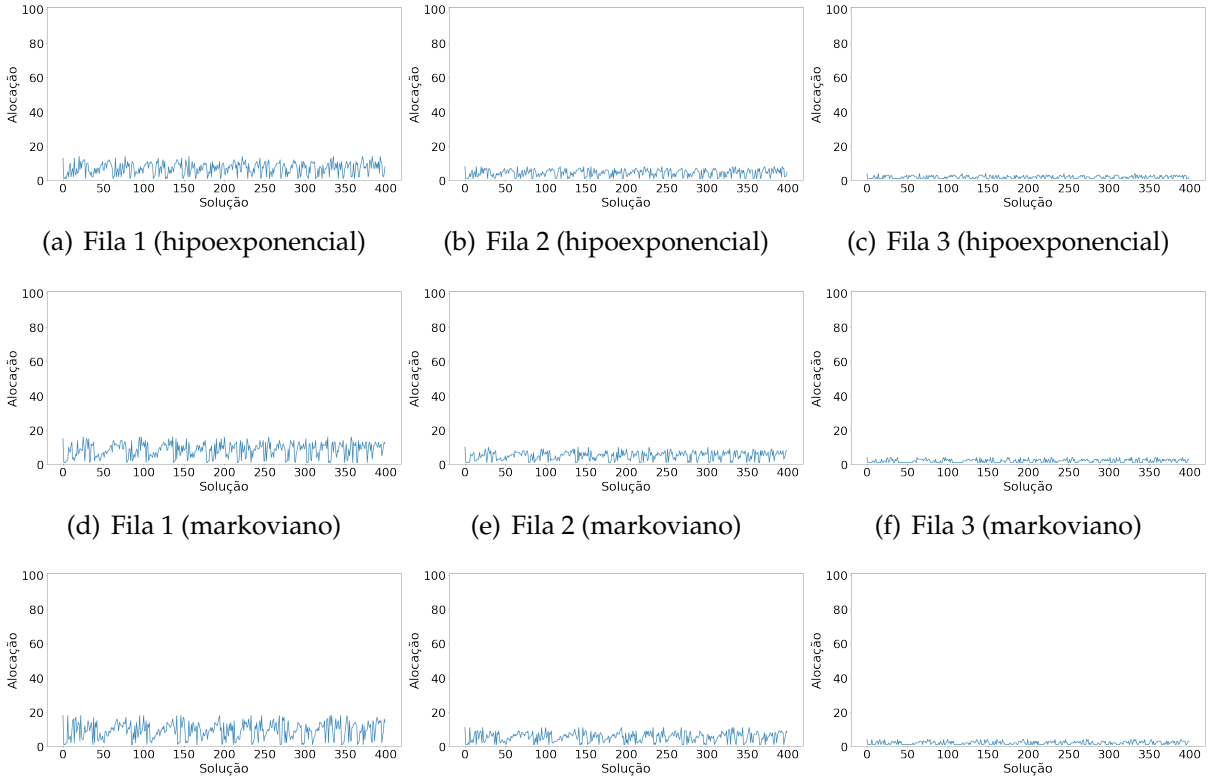
(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 33 – Resultados para configuração $p_1 = 0{,}9$, $p_2 = 0{,}1$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 34 – Pareto-solução para a configuração $p_1 = 0{,}9$, $p_2 = 0{,}1$, $\mu_1 = 10$, $\mu_2 = 10$, $\mu_3 = 10$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

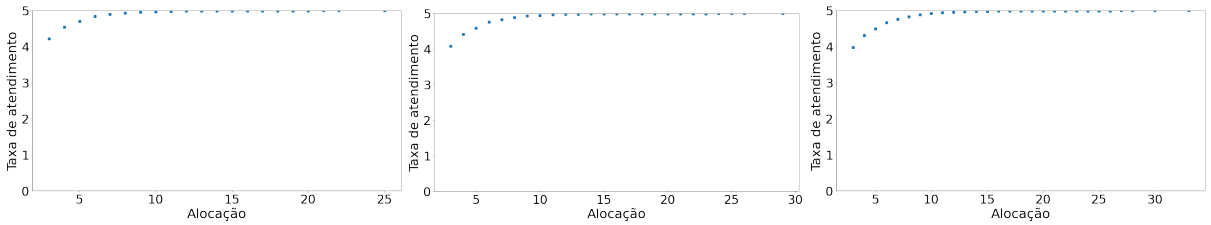


(a) Fila 1 (hipoexponencial) (b) Fila 2 (hipoexponencial) (c) Fila 3 (hipoexponencial)

(d) Fila 1 (markoviano) (e) Fila 2 (markoviano) (f) Fila 3 (markoviano)

(g) Fila 1 (hiperexponencial) (h) Fila 2 (hiperexponencial) (i) Fila 3 (hiperexponencial)

Figura 35 – Resultados para configuração $p_1 = 0{,}9$, $p_2 = 0{,}1$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial ($s^2 = 0{,}5$), markoviano ($s^2 = 1{,}0$) e hiperexponencial ($s^2 = 1{,}5$).



(a) Hipoexponencial (b) Markoviano (c) Hiperexponencial

Figura 36 – Pareto-solução para a configuração $p_1 = 0{,}9$, $p_2 = 0{,}1$, $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 5$ com atendimentos hipoexponencial, markoviano e hiperexponencial.

Para a Figura 13, o roteamento direciona para a fila 2 60% dos clientes, mas a taxa de serviço da fila 2 é três vezes menor que a taxa de seviço da fila 3. A alocação média de áreas de espera na fila 2 é maior nas três configurações (hipoexponencial, markoviano e hiperexponencial). Isso reflete nos Paretos-solução verificados na Figura 14 em que as soluções de alocação baixa possuem taxas de atendimento até mesmo menores que 80% das taxas de entrada da rede de filas.

O efeito anterior desaparece nas figuras 15 e 16 em virtude da igualdade das taxas de serviço. A alocação de na fila 2 ainda é superior devido ao roteamento de 60%,

mas este efeito é menos danoso. Os Paretos-solução na Figura 16 confirmam este fato.

As figuras 17 e 18 invertem a relação entre as taxas de serviço das filas 2 e 3. A alocação na fila 3 é maior, mas este efeito é amenizado com o reoteamento de 60% para a fila 2. O padrão de qualidade das soluções no Pareto-solução é bastante semelhante ao verificado na Figura 16.

As Figuras 19 e 20 apresentam um padrão que remete ao verificado na Figura 13, porém o efeito é mais significativo, as alocações para fila 2 são ainda maiores. O cenário roteia 70% para fila 2 e a fila 3 tem capacidade três vezes maior de serviço que a fila 2. Este fato remete a uma perda de capacidade de atendimento para alocações menores, como pode ser visto nas soluções de menor alocação. A quantidade é ainda menor em capacidade de atendimento que a verificada na Figura 14.

Novamente o efeito é amenizado na figura 21 em decorrência da igualdade da capacidade de serviço entre as filas 2 e 3. A alocação na fila dois ainda é superior devido ao roteamento de 70%, mas este é sutil. Os Paretos-solução na Figura 22 revelam uma capacidade de atendimento da rede de filas maior que a verificada no Pareto-solução da Figura 14.

Na figura 23, com roteamento de 70% para a fila 2 e a fila 2 com capacidade três vezes maior de atendimento que a fila 3, um equilíbrio de alocação entre as filas 2 e 3 fica ainda mais evidente. Quanto mais equilibrada essa alocação, maior qualidade de atendimento é verificada no Pareto-solução, principalmente para as soluções menos onerosas, de custo inferior em alocação, a Figura 24 retrata bem este efeito.

A Figura 25 apresenta um padrão que remete ao verificado nas Figuras 13 e 19, porém o efeito é mais evidente, as alocações para fila 2 são ainda mais elevadas. O roteamento para fila 2 está em 80% e a fila 3 tem novamente a capacidade três vezes maior de serviço que a fila 2. Aqui a perda de capacidade de atendimento para alocações menores é ainda mais notória, como mostrado nas soluções de menor alocação, através da Figura 26. A qualidade das alocações menos onerosas é ainda menor em capacidade de atendimento que a verificada nas Figuras 14 e 20.

O aumento de desequilibrio verificado nas Figuras 25 e 26 é atenuado na Figura 27, mas dada a discrepância no roteamento, a igualdade entre as taxa é insuficiente para reequilibrar completamente as alocações entre as filas 2 e 3. Esse desequilíbrio ainda pode ser notado na Figura 28 que apresenta algumas soluções com taxa de atendimento menores que 4. O efeito é menor para sistemas hipoexponenciais, mas se agrava a medida que a variabilidade cresce em sistemas hiperexponenciais.

A Figura 29 já retrata uma situação de quase equilíbrio, pois a capacidade de atendimento da fila 2 é três vezes maior que a capacidade de atendimento da fila 3. O equilibrio não é o mais adequado, pois o roteamento para fila 2 é 4 vezes maior que

para a fila 3. essas condições permitem um Pareto-solução com taxas de atendimento bem elevadas como verificado na Figura 30.

Dentre as situações investigadas, a de maior desequilíbrio é apresentada na Figura 31. O padrão é o mesmo das Figuras 13, 19 e 25 porém ainda mais forte. A fila 3 tem capacidade de serviço três vezes maior que a fila 2, porém o roteamento para fila 2 é nove vezes maior que o roteamento para a fila 3. Nessa situação mais extrema que é verificada, a pior condição para a taxa de atendimento das estratégias de alocação de menor custo. A soluções de alocação mais baixa apresentam taxas de atendimento muito pouco superiores à 60% da taxa de chegada na rede de filas, ou seja, quase 40% não seria atendido neste tipo de alocação proposta, estas constatações podem ser verificadas na Figura 32.

O maior desequilibrio visualizado nas Figuras 31 e 32 é reduzido na Figura 33, porém, em decorrência da discrepância no roteamento, mesmo a igualdade entre as taxa é incapaz de reequilibrar completamente as alocações entre as filas 2 e 3. Esse desequilíbrio ainda é latente na Figura 34, que apresenta algumas soluções com taxa de atendimento menor que 4. Novamente o efeito piora com o aumento da variabilidade nos tempos de serviço.

Por fim, a Figura 35 apresenta uma situação mais próxima de equilíbrio, novamente a maior capacidade de serviço da fila 2 ameniza o forte desequilíbrio do roteamento. O equilibrio ainda não é o mais adequado, dada a diferença grande no roteamento. Essas condições permitem um Pareto-solução com taxas de atendimento bem elevadas, mesmo em alocações menos custosas, como verificado na Figura 36.

Como era previsível em nenhuma situação dentre as investigadas, o algoritmo propõe alterações na estratégia de alocação para a fila 1. Existe obviamente alguma flutuação estocástica, mas o padrão é o mesmo para todos os roteamentos analisados. Por outra lado, a variação no roteamento cria um impacto significativo na alocação *buffers* nas filas 2 e 3.

Para situações em que as probabilidades de roteamento direcionam mais serviço para a fila 2 que para a fila 3 ocorrem dois cenários distintos. Em um destes cenários, a fila 2 recebe mais serviços, mas sua capacidade de atendimento é maior. Por outro lado, existem situações em que a capacidade de atendimento da fila 2 é inferior à capacidade de atendimento da fila 3. Para tanto, o algoritmo mostra uma clara tendência em compensar essa incapacidade com o aumento expressivo na alocação de *buffers*. Quanto mais desequilibrado é o roteamento, mas evidente este efeito se torna. O caso mais extremo pode ser observado na Figura 31.

Este desequilíbrio nas alocações de *buffers* para uma das filas da rede gera um claro impacto nas taxas de atendimento da rede de filas. Para as alocações menores, as

taxas de atendimento se tornam notoriamente inferiores. Um situação específica pode ser visualizada na Figura 32.

Em uma análise mais ampla, para sistemas com menor capacidade financeira para investir em atendimento, as alocações são inferiores e para situações com desequilíbrio no vetor de roteamento o algoritmo apresenta dificuldades em prover soluções eficazes e de baixo investimento. Trata-se de um contexto específico, mas é uma fragilidade das soluções fornecidas por essa estratégia de algoritmo para o problema de alocação de *buffers* em redes de filas de servidor único.

5 Considerações Finais

Este estudo propôs uma investigação que trata da formulação e análise do *Problema de Alocação de Buffers* para redes de filas de atendimento geral. Nesse trabalho todos os servidores possuem a capacidade de atendimento pré-fixada. Este estudo apresentou capacidade de agregar resultados tanto ao meio acadêmico quanto ao viés de aplicabilidade comercial. A busca por alocação ótima é um assunto de interesse em plantas industriais, e canais de escomento de produtos e muitos outros propósitos.

O estudo apresentou um levantamento da bibliografia com estudos recentes na área de alocação em redes das filas. Principalmente para problemas correlatos aos descritos aqui, no que tange a formulação matemática e também estratégia de otimização que foi utilizada.

Para testar o método proposto, foi experimentada uma topologia específica de rede e os impactos causados por variações nos parâmetros dessa mesma topologia foram investigados. Para essa rede, dentro de suas limitações, foram variadas as especificações pontuais para obter resultados bastante abrangentes. As variações na taxa de atendimento dos servidores envolvidos e também no vetor de roteamento foi importante para mostrar a capacidade de adaptabilidade do algoritmo proposto para diferentes situações do problema em estudo. O método proposto conseguiu fornecer soluções eficientes para o problema proposto em diversas situações.

Observa-se que, dos efeitos decorrentes das variações nos parâmetros da rede de filas apresentam resultados com algum padrão específico para os diversos casos investigados. Estes efeitos foram explorados de forma clara através da análise de resultados das diversas configurações investigadas.

Investigações futuras também incluem a avaliação da qualidade na estimação de outras medidas de desempenho das filas da rede. Outras investigações com filas de estruturas distintas, tais como filas markovianas multi-servidoras finitas, M/G/c/k. Estes são apenas alguns tópicos para trabalhos futuros nesta instigante linha de pesquisa.

Referências

[1] N. Ahmed and X. Ouyang. Suboptimal red feedback control for buffered tcp flow dynamics in computer network. *Mathematical Problems in Engineering*, 2007, 2007. Citado na página 1.

[2] F. S. Q. Alves, H. C. Yehia, L. A. C. Pedrosa, F. R. B. Cruz, and L. Kerbache. Upper bounds on performance measures of heterogeneous $M/M/c$ queues. *Mathematical Problems in Engineering*, 2011(Article ID 702834):18 pages, 2011. Citado na página 2.

[3] K. Chaudhuri, A. Kothari, R. Pendavingh, R. Swaminathan, R. Tarjan, and Y. Zhou. Server allocation algorithms for tiered systems. *Algorithmica*, 48(2):129–146, 2007. Citado na página 2.

[4] J. Chen, C. Hu, and Z. Ji. An improved ARED algorithm for congestion control of network transmission. *Mathematical Problems in Engineering*, 2010, 2010. Citado na página 1.

[5] S. Chowdhury and S. P. Mukherjee. Estimation of traffic intensity based on queue length in a single m/m/1 queue. *Communications in Statistics - Theory and Methods*, 42(13):2376–2390, 2013. Citado na página 10.

[6] C. A. Coello Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, pages 1051–1056, 2002. Citado na página 13.

[7] F. R. B. Cruz, A. R. Duarte, and G. L. Souza. Multi-objective performance improvements of general finite single-server queueing networks. *Journal of Heuristics*, 24(5):757–781, 2018. Citado na página 10.

[8] F. R. B. Cruz, A. R. Duarte, and T. van Woensel. Buffer allocation in general single-server queueing networks. *Computers & Operations Research*, 35(11):3581 –3598, 2008. Citado 2 vezes nas páginas 1 e 10.

[9] F. R. B. Cruz, G. Kendall, L. While, A. R. Duarte, and N. C. L. Brito. Throughput maximization of queueing networks with simultaneous minimization of service rates and buffers. *Mathematical Problems in Engineering*, 2012(Article ID 692593):19 pages, 2012. Citado 3 vezes nas páginas 13, 7 e 10.

[10] F. R. B. Cruz, R. C. Quinino, and L. L. Ho. Control charts for traffic intensity monitoring of Markovian multiserver queues. *Quality and Reliability Engineering International*, 36(1):354–364, 2020. Citado na página 10.

[11] I. Dimitriou and C. Langaris. A repairable queueing model with two-phase service, start-up times and retrial customers. *Computers and Operations Research*, 37(7):1181–1190, 2010. Citado na página 2.

[12] V. Inzillo, F. De Rango, and A. A. Quintana. A self clocked fair queuing MAC approach limiting deafness and round robin issues in directional MANET. In *2019 Wireless Days (WD)*, pages 1–6. IEEE, 2019. Citado na página 1.

[13] D. G. Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of embedded Markov chains. *Annals Mathematical Statistics*, 24:338–354, 1953. Citado na página 2.

[14] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995. Citado 3 vezes nas páginas 3, 8 e 11.

[15] L. Kerbache and J. MacGregor Smith. The generalized expansion method for open finite queueing networks. *European Journal of Operational Research*, 32:448–461, 1987. Citado 2 vezes nas páginas 10 e 11.

[16] J. MacGregor Smith and F. R. B. Cruz. The buffer allocation problem for general finite buffer queueing networks. *IIE Transactions*, 37(4):343–365, 2005. Citado 4 vezes nas páginas 13, 2, 9 e 15.

[17] J. MacGregor Smith, F. R. B. Cruz, and T. van Woensel. Topological network design of general, finite, multi-server queueing networks. *European Journal of Operational Research*, 201(2):427–441, 2010. Citado na página 2.

[18] H. S. R. Martins, F. R. B. Cruz, A. R. Duarte, and F. L. P. Oliveira. Modeling and optimization of buffers and servers in finite queueing networks. *OPSEARCH*, 56(1):123–150, 2019. Citado na página 10.

[19] D. A. Menascé. QoS issues in web services. *IEEE Internet Computing*, 6(6):72–75, 2002. Citado na página 2.

[20] C. Osorio and M. Bierlaire. An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, 196(3):996–1007, 2009. Citado na página 2.

[21] D. Qi, Z. Li, X. Zi, and Z. Wang. Weighted likelihood ratio chart for statistical monitoring of queueing systems. *Quality Technology & Quantitative Management*, 14(1):19–30, 2017. Citado na página 10.

[22] G. L. Souza. Uma nova formulação para otimização multi-objetivo em redes de filas finitas gerais e com único servidor. Master's thesis, Universidade Federal de Ouro Preto, Ouro Preto, 2020. Citado 2 vezes nas páginas 5 e 6.

[23] G. L. Souza, A. R. Duarte, G. J. P. Moreira, and F. R. B. Cruz. A novel formulation for multi-objective optimization of general finite single-server queueing networks. In *Proceedings of the 2020 Congress on Evolutionary Computation. CEC'20*, pages 1–8. IEEE, 2020. Citado na página 3.

[24] G. L. Souza, A. R. Duarte, G. J. P. Moreira, and F. R. B. Cruz. Post-processing improvements in multi-objective optimization of general single-server finite queueing networks. *IEEE Latin America Transactions*, 21(3):381–388, 2023. Citado na página 3.

[25] T. van Woensel and F. R. B. Cruz. Optimal routing in general finite multi-server queueing networks. *PLoS ONE*, 9(7):e102075, 07 2014. Citado na página 10.

[26] X. S. Yang. *Engineering optimization: An introduction with metaheuristic applications*. Wiley Publishing, 2010. Citado 2 vezes nas páginas 13 e 5.