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

Otimização Multiobjetivo de *Buffers* e Servidores em Redes de Filas Markovianas via Algoritmo *Simulated Annealing*

Adriano Pires Cunha

Ouro Preto-MG
2025

Adriano Pires Cunha

Otimização Multiobjetivo de *Buffers* e Servidores em Redes de Filas Markovianas via Algoritmo *Simulated Annealing*

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.

Orientadora: Anderson Ribeiro Duarte

Ouro Preto
2025

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C972o Cunha, Adriano Pires.
Otimização multiobjetivo de buffers e servidores em redes de filas markovianas via algoritmo Simulated Annealing. [manuscrito] / Adriano Pires Cunha. - 2025.
56 f.: il.: color., gráf., tab..

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

1. Teoria das filas - Redes de filas. 2. Markov, Processos de. 3. Otimização. I. Duarte, Anderson Ribeiro. II. Universidade Federal de Ouro Preto. III. Título.

CDU 519.872

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526

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

FOLHA DE APROVAÇÃO

Adriano Pires Cunha

Otimização multiobjetivo de buffers e servidores em redes de filas markovianas via algoritmo *Simulated Annealing*

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 01 de setembro de 2025

Membros da banca

Dr. Anderson Ribeiro Duarte - Orientador - Universidade Federal de Ouro Preto
Dr. Helgem de Souza Ribeiro Martins - Universidade Federal de Ouro Preto
Dr. 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 01/09/2025

Documento assinado eletronicamente por **Anderson Ribeiro Duarte**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 09/09/2025, às 16:20, 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 **0974599** e o código CRC **E36A60A3**.

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

Agradecimentos

Agradeço, em primeiro lugar, aos meus pais, por serem o alicerce de toda a minha jornada. Seu apoio incondicional e sacrifício foram fundamentais para que eu pudesse não apenas ingressar, mas também concluir este curso, especialmente ao me incentivarem a seguir minha vocação na Estatística.

Minha gratidão se estende à cidade de Ouro Preto, que foi muito mais que um cenário para os meus estudos; foi o lugar onde construí laços e fiz amizades que, tenho certeza, levarei para toda a vida.

Ao corpo docente do Departamento de Estatística da Universidade Federal de Ouro Preto, agradeço por todo o conhecimento compartilhado e pelo suporte constante, que foram essenciais para o meu desenvolvimento acadêmico e intelectual. Em especial, agradeço ao meu orientador, pela paciência, pela confiança depositada em meu potencial e pelas valiosas oportunidades que me proporcionou.

Por fim, agradeço à UFOP, por ter sido o espaço de inúmeras oportunidades e aprendizados ao longo desses anos.

Resumo

Este estudo aborda a resolução de problemas de otimização em sistemas de redes de filas, com o objetivo explícito de aumentar sua eficiência operacional. Especificamente, visa otimizar a alocação de recursos e melhorar o desempenho do sistema por meio da aplicação de algoritmos de otimização apropriados. Para tanto, o algoritmo *Simulated Annealing* (SA) é empregado para resolver problemas de otimização em redes de filas caracterizadas por processos de chegada e serviço Markovianos, múltiplos servidores e capacidades de *buffer* finitas. Dentre os desafios relevantes nesse contexto, o Problema de Alocação de Buffers e Servidores (BSAP) se destaca como uma questão significativa e amplamente estudada. Este trabalho propõe uma estratégia multiobjetivo para abordar o BSAP, destaca sua aplicabilidade a uma variedade de sistemas do mundo real que podem ser modelados como redes de filas. A relevância prática desta pesquisa reside em seu potencial para apoiar uma tomada de decisão mais eficaz no projeto e gerenciamento de tais sistemas. O BSAP constitui um problema de programação não linear para o qual não existe uma solução analítica de forma fechada e para o qual os métodos computacionais tradicionais se mostram inadequados. Consequentemente, técnicas heurísticas e livres de derivadas tornam-se essenciais. Nesse contexto, o algoritmo *Multiobjective Optimization Simulated Annealing* (MOSA) surge como um método promissor para a obtenção de soluções de alta qualidade para gerenciamento de recursos. O estudo avalia a eficácia do algoritmo na geração de soluções custo eficientes que respondam às variações nos custos relativos de servidores e *buffers*, além de acomodar as características estruturais específicas de cada topologia de rede de filas.

Palavras-chave: Redes de Filas; Servidores; *Buffers*; Otimização; *Simulated Annealing*.

Abstract

This study addresses the resolution of optimization problems in queueing network systems, with the explicit objective of enhancing their operational efficiency. Specifically, it aims to optimize resource allocation and improve system performance through the application of appropriate optimization algorithms. To this end, the Simulated Annealing (SA) algorithm is employed to solve optimization problems in queueing networks characterized by Markovian arrival and service processes, multiple servers, and finite buffer capacities. Among the relevant challenges in this context, the Buffer and Server Allocation Problem (BSAP) stands out as a significant and widely studied issue. This work proposes a multi-objective approach to address BSAP, highlighting its applicability to a variety of real-world systems that can be modeled as queueing networks. The practical relevance of this research lies in its potential to support more effective decision-making in the design and management of such systems. BSAP constitutes a nonlinear programming problem for which no closed-form analytical solution exists, and for which traditional computational methods prove inadequate. Consequently, heuristic and derivative-free techniques become essential. In this context, the Multiobjective Optimization Simulated Annealing (MOSA) algorithm emerges as a promising method for obtaining high-quality solutions for resource management. The study evaluates the algorithm's effectiveness in generating cost-efficient solutions that are responsive to variations in the relative costs of servers and buffers, while also accommodating the structural characteristics specific to each network topology.

Keywords: Queueing Networks; Servers; Buffers; Optimization; Simulated Annealing

Lista de ilustrações

Figura 1 – Rede de filas com topologia mista adaptada de Smith e Cruz (2005) [1]. 3
Figura 2 – Mudança de Estado com Variação de Temperatura Não Planejada. . 11
Figura 3 – Mudança de Estado com Variação de Temperatura Planejada. 12
Figura 4 – Cone de Dominância em Problema de Otimização Bidimensional. . . 20
Figura 5 – Fronteira de Pareto Hipotética. 21
Figura 6 – Topologias Experimentais com Redes Variadas. 25
Figura 7 – Análise Comparativa das Probabilidades de Bloqueio para Topologia em Série: Soluções Iniciais versus Soluções Otimizadas. 26
Figura 8 – Análise Comparativa das Probabilidades de Bloqueio para Topologia de Divisão: Soluções Iniciais versus Soluções Otimizadas. 27
Figura 9 – Análise Comparativa das Probabilidades de Bloqueio para Topologia de Fusão: Soluções Iniciais versus Soluções Otimizadas. 27
Figura 10 – Análise Comparativa das Probabilidades de Bloqueio para Topologia Mista: Soluções Iniciais versus Soluções Otimizadas. 28
Figura 11 – Redução na Alocação de Servidores para Topologia em Série: Comparação entre Soluções Iniciais e Otimizadas. 29
Figura 12 – Redução na Alocação de Servidores para Topologia de Divisão: Comparação entre Soluções Iniciais e Otimizadas. 29
Figura 13 – Redução na Alocação de Servidores para Topologia de Fusão: Comparação entre Soluções Iniciais e Otimizadas. 30
Figura 14 – Redução na Alocação de Servidores para Topologia Mista: Comparação entre Soluções Iniciais e Otimizadas. 30
Figura 15 – Redução na Alocação de *Buffers* para Topologia em Série: Comparação entre Soluções Iniciais e Otimizadas. 31
Figura 16 – Redução na Alocação de *Buffers* para Topologia de Divisão: Comparação entre Soluções Iniciais e Otimizadas. 31
Figura 17 – Redução na Alocação de *Buffers* para Topologia de Fusão: Comparação entre Soluções Iniciais e Otimizadas. 32
Figura 18 – Redução na Alocação de *Buffers* para Topologia Mista: Comparação entre Soluções Iniciais e Otimizadas. 32
Figura 19 – Fronteiras de Pareto para Topologia em Série, Organizadas por Valor do Parâmetro α. 33
Figura 20 – Fronteiras de Pareto para Topologia em Divisão, Organizadas por Valor do Parâmetro α. 34
Figura 21 – Fronteiras de Pareto para Topologia em Fusão, Organizadas por Valor do Parâmetro α. 34

Figura 22 – Fronteiras de Pareto para Topologia em Mista, Organizadas por Valor do Parâmetro α. 35

Sumário

1 INTRODUÇÃO 1
1.1 Motivação 2
1.2 Objetivos 4
1.2.1 Objetivos Gerais 4
1.2.2 Objetivos Específicos 4

2 FUNDAMENTAÇÃO TEÓRICA 5
2.1 Métodos de Otimização 5
2.2 Teoria de Filas 8
2.2.1 Terminologia Básica 8
2.2.2 Notação de Kendall 9
2.2.3 Métricas de Desempenho 10
2.2.4 Otimização em Redes de Filas: 10

3 ASPECTOS METODOLÓGICOS 13
3.1 Formulações para o Problema de Alocação de *Buffers* 13
3.1.1 BAP Primal 13
3.1.2 BAP Dual 14
3.2 Formulações para o Problema de Alocação de Servidores (SAP) . . 14
3.2.1 SAP Primal 15
3.2.2 SAP Dual 15
3.3 Formulação para o Problema de Alocação Conjunta de *buffers* e Servidores 16
3.3.1 BSAP Mono-objetivo 16
3.3.2 BSAP Multiobjetivo 17
3.3.3 Foco deste Estudo 19
3.4 Otimização Multi-objetivo 19
3.5 Algoritmo *Simulated Annealing* 21
3.5.1 Princípios do Algoritmo 21
3.5.2 Adaptação para Otimização Multiobjetivo 22
3.6 *Fine-tuning* do Algoritmo *Simulated Annealing* para Otimização Multiobjetivo em Rede de Filas 22
3.6.1 Estrutura de Vizinhança 23
3.6.2 Esquema de Resfriamento 23
3.6.3 Pseudo-código do Algoritmo 24

4 RESULTADOS ALCANÇADOS 25
4.1 Análise Comparativa da Eficácia do Algoritmo 26
4.1.1 Análise das Probabilidades de Bloqueio 26
4.1.2 Redução na Alocação de Recursos 28
4.2 Análise das Fronteiras de Pareto 33
4.2.1 Características das Fronteiras por Topologia 35
4.2.2 Influência do Parâmetro de Ponderação 36
4.3 Avaliação Quantitativa da Performance 36
4.3.1 Métricas de Melhoria 36
4.4 Implicações dos Resultados 37

5 CONSIDERAÇÕES FINAIS 39

REFERÊNCIAS 41

1 Introdução

Problemas práticos do cotidiano muitas vezes envolvem contextos nos quais a otimização está presente. Este cenário desperta grande interesse para investigações em estratégias de otimização e suas aplicabilidades. Em problemas de otimização é possível identificar duas abordagens principais clássicas para encontrar soluções: os métodos exatos e os heurísticos. Os métodos exatos se concentram em alcançar a solução ótima com total precisão, por meio da aplicação de técnicas matemáticas rigorosas adequadas para problemas de escala gerenciável. Por outro lado, os métodos heurísticos são adaptados para contextos em que a obtenção da solução ótima é impraticável devido à complexidade ou tamanho do problema. Eles visam encontrar soluções boas o suficiente de uma maneira eficiente, muitas vezes com o sacrifício da precisão absoluta por tempos de execução viáveis. A seleção entre essas abordagens depende do equilíbrio necessário entre precisão e praticidade. Diante disso, os métodos heurísticos constituem uma abordagem valiosa para dilemas de otimização complexos e de grande escala.

As heurísticas de otimização representam um conjunto de métodos destinados a explorar soluções aproximadas para problemas complexos, cujas soluções ótimas são desafiadoras ou inviáveis de serem obtidas por métodos exatos. A palavra heurística, derivada do grego "*heuriskein*", significa encontrar ou descobrir, reflete a essência desses métodos de buscar soluções práticas através de aproximações. Diferentemente dos métodos exatos, as heurísticas não garantem a obtenção da solução ótima, mas procuram soluções próximas que sejam suficientemente boas dentro de um tempo computacional razoável, são particularmente úteis em problemas de otimização em que a solução ótima é desconhecida ou o processo de busca por essa solução se mostra exaustivo, custoso ou até mesmo impraticável.

Para seguir na exploração de métodos heurísticos, é imprescindível abordar a aplicação dessas técnicas em contextos específicos, como nos sistemas de filas em rede. Esses sistemas representam um desafio intrigante para a otimização, em que a eficácia das heurísticas é posta à prova em um ambiente marcado por complexidade e a necessidade de decisões rápidas. A otimização de sistemas de filas em rede impacta diretamente aspectos vitais de diversas áreas — desde a eficiência de processos de manufatura até a gestão de fluxos em redes de comunicação e a organização de serviços de saúde — isso demonstra a versatilidade e a capacidade das heurísticas de oferecer soluções robustas em situações nas quais as variáveis são muitas e os modelos exatos se tornam impraticáveis.

Neste estudo, o propósito específico de explorar a aplicabilidade e eficácia da

heurística *Simulated Annealing* (SA) no contexto da otimização em sistemas de filas em rede, especificamente para o problema de filas finitas, multiservidores, com chegadas e atendimentos markovianos foi delimitado. A escolha deste método se justifica por sua conhecida capacidade de escapar de mínimos locais, inspirada no processo físico de recozimento (ou resfriamento) de metais. Essa característica permite uma busca eficiente por soluções (próximas da) ótimas em espaços de solução complexos. O SA alterna entre aceitar soluções que melhoram o objetivo e, com uma probabilidade decrescente, soluções que o deterioram, dessa forma, permite portanto escapar de mínimos locais. Em analogia ao procedimento de resfriamento de metais, este processo é regido pela "temperatura" do sistema, que gradualmente diminui de acordo com um esquema de resfriamento predefinido.

Dentro da modelagem de sistemas de filas, um problema recorrente e de grande aplicabilidade prática consiste na determinação do número ideal de servidores e do tamanho adequado dos *buffers* (áreas de espera). Esta questão de dimensionamento afeta diretamente tanto o desempenho operacional quanto os custos associados aos sistemas. Um número excessivo de servidores ou *buffers* muito grandes podem resultar em recursos ociosos e investimentos desnecessários, enquanto configurações subdimensionadas tendem a gerar congestionamentos, perda de clientes e degradação da qualidade de serviço. Por meio da aplicação do *Simulated Annealing*, o objetivo é encontrar um balanceamento que otimize métricas relevantes, como o tempo médio de espera dos clientes, a taxa de utilização dos servidores e o custo total do sistema. A busca por um equilíbrio ótimo entre essas métricas, muitas vezes conflitantes entre si, caracteriza a natureza multiobjetivo do problema abordado, cuja complexidade combinatória o torna particularmente adequado para abordagens heurísticas como a proposta neste trabalho.

1.1 Motivação

Um exemplo prático que ilustra a importância da otimização do número de servidores e *buffers* em sistemas de filas em rede é o funcionamento de um grande hospital. Nesse ambiente, diversos serviços críticos, como o atendimento de emergência, diagnósticos por imagem (como tomografias e ressonâncias), e laboratórios de análises clínicas, dependem da eficiência e rapidez no atendimento. A demanda por esses serviços pode variar significativamente, podem ocorrer picos de solicitações que exigem uma gestão eficaz das filas para evitar atrasos que podem comprometer a saúde dos pacientes.

A otimização do número de servidores (por exemplo, médicos, técnicos de diagnóstico e enfermeiros) e dos *buffers* (como leitos de espera para exames ou espaços para coleta de amostras) é crucial. O desafio reside em garantir que os pacientes sejam

atendidos prontamente, com a qualidade do serviço mantida e com os tempos de espera o menor possível, ao mesmo tempo em que se busca a utilização eficiente dos recursos e o controle dos custos operacionais. Em situações de alta demanda, um número insuficiente de servidores ou uma alocação inadequada de *buffers* pode resultar em congestionamentos significativos, afetando negativamente a experiência do paciente, seu prognóstico e a sustentabilidade financeira da instituição.

Este cenário evidencia a necessidade de estudos que explorem metodologias de otimização robustas, como a heurística *Simulated Annealing*, para resolver problemas complexos em redes de filas. A capacidade dessa heurística de encontrar boas soluções em cenários onde métodos exatos são inviáveis torna-a particularmente valiosa. No contexto hospitalar, por exemplo, o uso do *Simulated Annealing* para encontrar um balanço entre o número de servidores, a alocação dos *buffers* e outros fatores, considerando a variação da demanda, pode melhorar significativamente a eficiência operacional e a qualidade do atendimento.

A rede complexa de filas, apresentada na Figura 1, foi adaptada da literatura (Smith e Cruz, 2005 [1]). A topologia da rede ilustrada é particularmente apropriada para estudos experimentais, pois engloba configurações topológicas fundamentais como fusão (*merge*) de fluxos, divisão (*split*) e múltiplos estágios em série.

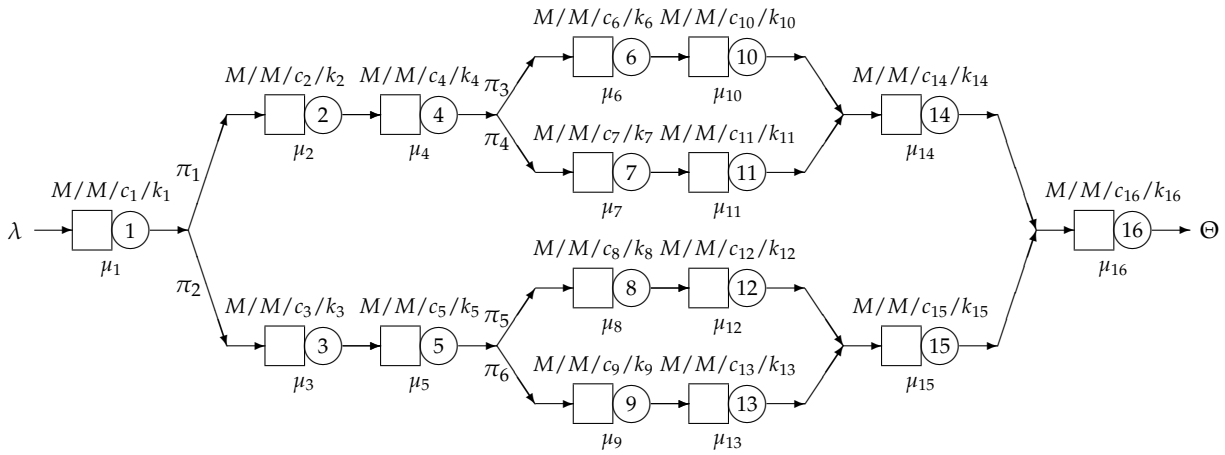


Figura 1 – Rede de filas com topologia mista adaptada de Smith e Cruz (2005) [1].

No âmbito da otimização da alocação de recursos em redes de filas, a etapa de desenvolver uma formulação matemática adequada para o problema sob investigação é de considerável relevância. Um dos interesses deste trabalho é, portanto, discutir as diversas possibilidades de formulação para otimizar a utilização de recursos em redes de filas acíclicas. Nesse contexto, e como base para análises subsequentes ou comparativas, será apresentada uma formulação com objetivo único para o Problema de Alocação de Servidores e *buffers* (BSAP). É crucial salientar que tais formulações não são estanques ou únicas; elas são, em essência, moldadas pelos objetivos específicos do pesquisador e pelas necessidades do projeto em questão. Contudo, a literatura científica

já apresenta um conjunto de formulações mais recorrentes, assim como modelos mais generalizados aplicáveis a certas classes de problemas de otimização.

1.2 Objetivos

Este estudo tem objetivos de caráter científico e aplicado. As investigações em redes de filas tem interesse científico explícito em estudos voltados para os Processos Estocásticos. Por outro lado, a modelagem de redes de filas tem impactos econômicos bem definidos para a gestão em variados setores, isso ilustra o caráter aplicado dessa investigação.

1.2.1 Objetivos Gerais

- apresentar uma revisão da bibliografia na área de modelagem e otimização de redes de filas finitas;
- utilizar a linguagem LaTeX, que é padrão na confecção de textos estatísticos em vários níveis de pesquisa;
- apresentar formulações matemáticas para o problema de otimização em redes de filas.

Além desses objetivos gerais, este trabalho busca fornecer os resultados específicos que seguem.

1.2.2 Objetivos Específicos

- detalhar uma versão de formulação matemática capaz de otimizar simultaneamente o consumo de recursos em áreas de espera e servidores alocados;
- apresentar uma versão específica do clássico algoritmo de otimização *Simulated Annealing*;
- discutir os resultados obtidos pelo algoritmo de otimização para redes de filas complexas.

2 Fundamentação Teórica

A investigação desse estudo envolve estudos de otimização em uma aplicação específica de Teoria de filas. Para tanto, uma introdução teórica em problemas de otimização combinatória e Teoria de filas se faz importante.

Um problema de otimização combinatória é um problema de maximização ou de minimização com um conjunto associado de instâncias de estudo. Sem perda de generalidade todas as definições apresentadas se restringirão aos problemas de minimização, uma vez que todo problema de maximização pode facilmente ser convertido em um problema de minimização.

Definição 1. *Uma instância de um problema de otimização combinatória é um par* $(\mathcal{S}, f)$*, em que* $\mathcal{S}$ *é o conjunto finito de soluções candidatas e* $f : \mathcal{S} \to \mathbb{R}$ *é uma função que atribui a cada elemento* $s \in \mathcal{S}$ *um valor* $f(s)$*. O valor de* $f(s)$ *também é usualmente denominado por valor da função objetivo. O objetivo de um problema de otimização combinatória é a obtenção de uma solução* $s \in \mathcal{S}$ *com valor mínimo da função objetivo, ou seja,* $f(s_{opt}) \leqslant f(s') \; \forall \; s' \in \mathcal{S}$*. Dessa forma,* s_{opt} *é chamada de solução globalmente ótima de* $(\mathcal{S}, f)$*.*

Em um problema dessa natureza, o termo problema refere-se à questão geral a ser respondida, geralmente tendo vários parâmetros ou variáveis com valores não especificados. Já o termo instância refere-se a um problema com valores especificados para todos os parâmetros.

Dado um problema específico de otimização, alguma metodologia de solucionar o referido problema se faz necessária. Dessa forma alguma discussão acerca dessas metodologias deve ser abordada.

2.1 Métodos de Otimização

Existem diversos níveis de classificação e subdivisão para as metodologias de otimização mais difundidas. Dentre essas divisões, os métodos podem ser classificados como métodos exatos e métodos heurísticos.

Os métodos exatos de otimização são frequentemente chamados de métodos determinísticos. Eles são caracterizados pela previsibilidade de seus passos, dado um ponto de partida conhecido. De outra forma, um método determinístico sempre levará à mesma resposta se iniciar do mesmo ponto. Estes métodos opõem-se aos métodos estocásticos ou aleatórios, que incorporam elementos de aleatoriedade em seus processos (Arora, 2017 [2]).

Os métodos heurísticos são regras práticas derivadas da experiência, sem uma prova conclusiva de validade. Eles são projetados para encontrar soluções boas, mas não necessariamente ótimas, para problemas de otimização. Estes métodos podem ser tanto determinísticos quanto estocásticos. Eles ganharam destaque nos anos 80 como ferramentas adicionais para superar as limitações das heurísticas convencionais (Mataifa, Krishnamurthy e Kriger, 2022 [3]).

Os métodos exatos seguem uma abordagem algorítmica rigorosa baseada em princípios matemáticos estabelecidos, enquanto os métodos heurísticos dependem de regras práticas e muitas vezes de experimentação, também são relativamente mais rápidos. Dentre os métodos exatos, é possível enumerar alguns exemplos: algoritmos de programação linear, métodos de pontos interiores, métodos de ramificação e corte, entre outros [4]. Já dentre os métodos heurísticos: algoritmos genéticos, otimização por enxame de partículas, Busca Tabu, e *Simulated Annealing*, entre outros.

Os métodos exatos frequentemente são a base para o estudo teórico em otimização e têm suas raízes nos primeiros desenvolvimentos da Matemática e da Computação. Dentre os métodos heurísticos, muitos são inspirados em fenômenos naturais, como a evolução (algoritmos genéticos) ou o comportamento de animais (otimização por colônia de formigas), estes são usualmente denominados algoritmos bioinspirados.

Cada método tem seu próprio conjunto de valências, limitações e contextos ideais de aplicação, isso torna a escolha entre eles dependente das necessidades específicas e características do problema a ser resolvido.

Adicionalmente, cumpre ressaltar que uma parcela significativa dos problemas de otimização encontrados em contextos práticos, especialmente em sistemas de maior complexidade, não se restringe à otimização de um único critério. Frequentemente, é necessário considerar múltiplos objetivos de forma simultânea, os quais podem, inclusive, ser conflitantes entre si. Tais problemas configuram o campo da otimização multiobjetivo. Nesses cenários, a noção de uma única solução ótima global é usualmente substituída pela busca por um conjunto de soluções de compromisso, conhecidas como soluções Pareto-ótimas. Estas representam os diferentes balanços possíveis entre os objetivos concorrentes, e a seleção de uma solução final frequentemente depende de preferências adicionais do tomador de decisão. Métodos heurísticos, dada a sua flexibilidade, são frequentemente adaptados para explorar o espaço de soluções em problemas multiobjetivo, seja por meio da agregação de objetivos em uma função escalar única ou pela incorporação de conceitos de dominância no processo de busca.

Os métodos exatos baseiam-se em uma definição matemática rigorosa do problema e em um algoritmo preciso para buscar a solução ótima. Já os métodos heurísticos dependem de regras práticas, que podem não garantir uma solução ótima, mas são eficientes para encontrar soluções promissoras em problemas complexos.

Os métodos exatos seguem um processo sistemático e previsível, com passos bem definidos. São mais adequados para problemas bem definidos e de menor escala, em que uma solução precisa é necessária. Por outro lado, os métodos heurísticos empregam uma abordagem mais experimental e adaptativa, muitas vezes inspirada em processos naturais ou fenômenos biológicos. São úteis em problemas de grande escala ou com complexidade elevada, em que uma solução exata pode ser difícil, ou até mesmo impraticável de ser obtida.

Existem diversos exemplos de problemas usualmente solucionados através das duas metodologias. Por meio de métodos exatos podem ser elencados:

- problemas de programação linear, maximização de lucro ou minimização de custos em operações de negócios com restrições lineares. O método exato *simplex* [5] ou a programação linear inteira são eficazes [6];
- problemas de fluxo de rede, como encontrar o caminho mais curto ou o fluxo máximo em uma rede, algoritmos como o de Dijkstra (para o caminho mais curto) [7] ou o algoritmo Ford-Fulkerson (para fluxo máximo) são métodos exatos aplicáveis [8];
- otimização de portfólio de investimentos, como determinar a alocação ótima de recursos em diferentes ativos para maximizar o retorno esperado, sujeito a um certo nível de risco, este é um problema bem definido em que os métodos exatos como a programação quadrática são adequados [9].

Já por meio de métodos heurísticos podem ser elencados:

- problema do caixeiro viajante, em que dada uma lista de cidades e as distâncias entre elas, busca-se encontrar o caminho mais curto que visita todas as cidades uma vez e retorna à cidade de origem, devido à complexidade computacional exponencial para problemas maiores, métodos heurísticos como algoritmos genéticos [10] ou otimização por colônia de formigas [11] são frequentemente usados;
- otimização de escalas de trabalho, em que o interesse é criar horários de trabalho para uma grande equipe, sob várias restrições e preferências, a natureza variada e complexa dessas restrições torna os métodos heurísticos, como algoritmos genéticos [12] ou busca tabu [13], mais viáveis;
- otimização de redes de distribuição, com interesse em projetar a rede de distribuição ótima para produtos de um armazém para vários destinos, considerando custos, tempo e capacidade [14], métodos heurísticos são úteis aqui devido à complexidade e ao tamanho do problema.

Ainda para alinhar as dicussões de revisão necessárias para o presente estudo, algumas considerações sobre Teoria de Filas são apresentadas em sequência.

2.2 Teoria de Filas

A Teoria das Filas é um campo fascinante e instigante da Matemática Aplicada e da Pesquisa Operacional. O interesse central dos estudos em filas é focado no comportamento de sistemas cuja demanda tende a superar a oferta de serviço, o que gera espera. Ao entrar em uma agência bancária e pegar um número para aguardar atendimento, o indivíduo está inserido em um sistema de filas. Da mesma forma, quando acessa um *site* e seus dados são processados em um servidor, existem diversas situações de filas envolvidas. A aplicação dessa teoria abrange desde o atendimento ao cliente em lojas e *call centers* até o gerenciamento de tráfego de redes e processos de produção industrial. Dentre diversos conceitos e definições, uma nomenclatura específica de termos é usual nos estudos de teoria de filas. Alguns desses termos precisam estar bem definidos:

2.2.1 Terminologia Básica

- **Clientes:** em teoria das filas, os "clientes" podem ser pessoas, dados ou qualquer entidade que espera por algum determinado serviço.
- **Servidores:** em sistemas de filas, os "servidores" são os prestadores do serviço em questão, são encarados como recursos do sistema, processadores de computador, entre muitos outros possíveis exemplos.
- **Taxa de Chegada (λ):** é a taxa que determina a frequência com que os clientes chegam à fila, geralmente expressa em clientes por unidade de tempo (por exemplo, clientes por hora).
- **Taxa de Serviço (μ):** é a taxa que determina a fluidez dos atendimentos, ou seja, com que razão temporal os clientes são atendidos, também expressa em clientes por unidade de tempo.
- **Disciplina da Fila:** refere-se à regra pela qual os clientes são escolhidos para serem atendidos, por exemplo, primeiro a chegar, primeiro a ser atendido (do inglês FIFO, *first in first out*), ou até mesmo esquemas específicos com regras de prioridade.
- **Capacidade da Fila:** é o número máximo de clientes que podem esperar por atendimento na fila, em alguns modelos, esta capacidade pode ser até mesmo ilimitada.

2.2.2 Notação de Kendall

Uma notação bastante difundida para modelos de filas é a bastante conhecida notação de Kendall [15]. As filas são descritas em uma forma resumida $A/B/m/k$ em que: A representa o modelo probabilístico que descreve o tempo entre chegadas; B representa o modelo probabilístico que descreve o tempo de serviço; m determina o número de servidores disponíveis; e k determina a área de circulação, ou seja o espaço total disponibilizado para os clientes em espera acrescidos do número de clientes que já estão em atendimento. Os espaços de espera sem o acréscimo dos clientes em serviço são usualmente denominados *buffers*. A representação proposta por Kendall (1953) [15] fica melhor descrita nas representações da Tabela 1. É pertinente destacar que a utilização da distribuição Exponencial (representada pela letra 'M') para os tempos entre chegadas e/ou de serviço é fundamental na caracterização de processos de filas Markovianos. Tais processos, devido à propriedade de 'ausência de memória' da distribuição exponencial, possuem uma estrutura matemática que facilita a análise e modelagem, sendo, por isso, amplamente empregados como base para estudos em Teoria de Filas.

Tabela 1 – Parametrização para filas de acordo com a notação de Kendall [15].

parâmetro	símbolo	definição
A e B	M	Exponencial
	D	Determinístico
	E_p	Erlang$_p$ ($p = 1, 2, \ldots$)
	G	Geral
m	$1, 2, \ldots$	número de servidores
k	$1, 2, \ldots$	espaço total (servidores + área de espera)

Um possível exemplo seria considerar uma fila $M/G/4/\infty$, ou seja, um processo de fila com tempo entre chegadas com distribuição exponencial, tempo de serviço é geral, quatro servidores em paralelo para executar os atendimentos, e sem limitação no tamanho máximo da capacidade de *buffers*.

As características de um sistema de filas são basicamente:

i) padrão de chegadas dos clientes;

ii) padrão de serviço dos canais de atendimento;

iii) capacidade do sistema;

iv) número de canais de serviço;

v) número de estágios de serviço.

2.2.3 Métricas de Desempenho

Em investigações associadas com modelos para sistemas de filas, o foco é centrado em melhorias de performance e desempenho dos sistemas. Nesse contexto o desempenho é relevante tanto do ponto de vista do usuário do sistema de filas quanto do gestor, ambos procuram performance elevada, desempenho ótimo aliados com máximo aproveitamento dos recursos disponíveis. Para tanto, algumas medidas de desempenho em sistemas de filas tem relevância destacada.

- **Número Médio de Clientes na Fila (L_q):** retrata o valor esperado de clientes na fila aguardando pelo serviço.

- **Tempo Médio de Espera na Fila (W_q):** trata-se do tempo esperado que um cliente qualquer aguarda em fila antes da execução do seu serviço.

A análise dessas métricas permite aos gestores otimizar o funcionamento de sistema de filas, e ajustar o número adequado de servidores ou *buffers* ou modificar o processo de funcionamento em algum ponto com intuito de melhorar a eficiência e a satisfação dos usuários.

2.2.4 Otimização em Redes de Filas:

Em muitos sistemas reais, os processos de espera e serviço não ocorrem em unidades isoladas, mas sim em configurações onde múltiplas filas estão interconectadas, com clientes ou entidades transitando entre diferentes estágios de serviço. Tais arranjos são denominados redes de filas. A complexidade inerente a essas redes, advinda das interdependências entre seus componentes e da variedade de configurações possíveis (como filas em série, em paralelo, ou com retroalimentação), impõe desafios significativos tanto para a modelagem analítica quanto para a otimização de seu desempenho.

O objetivo da aplicação de estratégias de otimização em sistemas de filas é estabelecer o equilíbrio ideal entre custo envolvido e benefício dos usuários. Por exemplo, que seja minimizando o tempo de espera para os clientes, seja maximizando a utilização dos servidores. Em cenários reais, em que os modelos exatos podem ser impraticáveis devido à complexidade ou variabilidade das chegadas e serviços, os métodos heurísticos como o *Simulated Annealing* podem oferecer uma alternativa valiosa para explorar soluções aproximadas, proporcionar soluções práticas para problemas complexos de filas em rede.

Este estudo está focado na resolução de problemas de otimização em sistemas de filas. Neste sentido, a otimização consiste no processo de buscar a configuração de utilização de recursos de sistema que seja eficiente para o funcionamento do sistema. Para solucionar problemas de otimização em sistemas de filas, uma sequência sistemática de etapas deve ser seguida, identificar claramente a função objetivo, os parâmetros relevantes e suas restrições. Além disso, a escolha por estratégias adaptáveis de otimização para o problema em estudo tem um papel crucial na busca por soluções ótimas. O interesse desse estudo está diretamente ligado com a aplicação do algoritmo *Simulated Annealing*. Trata-se de um método metaheurístico de concepção simples e bastante difundido para resolver problemas de otimização difíceis e que usualmente demandam alguma simulação computacional onerosa em termos de processamento.

Kirkpatrick, Gelatt e Vecchi (1983) [16] introduziram o algoritmo *Simulated Annealing* para problemas de otimização combinatória. Os conceitos principais são baseados em uma analogia com o recozimento físico de materiais. Nesse procedimento de recozimento, um sólido é conduzido para um estado de baixa energia após o aumento controlado de sua temperatura. Coincidentemente, uma versão similar foi desenvolvida paralelamente na mesma época por Černý (1985) [17], as duas versões foram naturalmente unificadas no meio acadêmico.

O procedimento de recozimento físico pode ser resumido em duas etapas, o sólido é levado para uma temperatura extremamente elevada até "desmanchar" a sua estrutura. Em seguida o sólido é resfriado de acordo com um esquema de diminuição de temperatura especificamente estabelecido e controlado para atingir um estado sólido de energia mínima. Em fase líquida, as partículas do material são distribuídas de forma completamente aleatória. As Figuras 2 e 3 apresentadas por Leijoto (2024) [18] resumem o funcionamento esquemático desse procedimento físico.

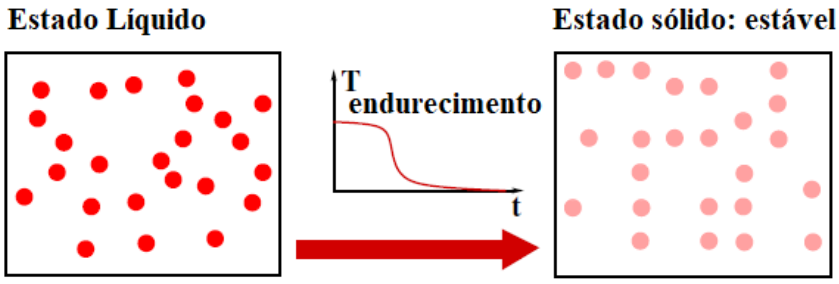


Figura 2 – Mudança de Estado com Variação de Temperatura Não Planejada.

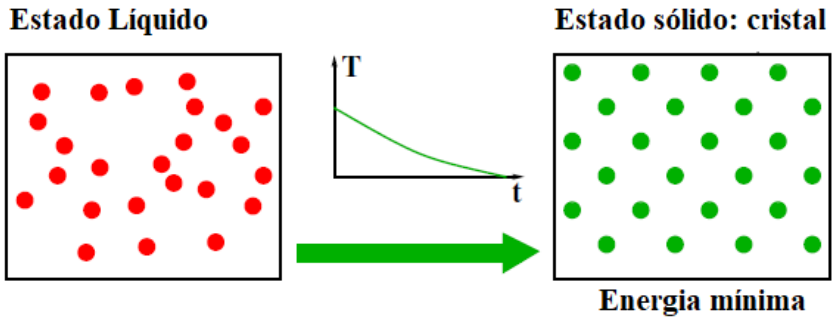


Figura 3 – Mudança de Estado com Variação de Temperatura Planejada.

3 Aspectos Metodológicos

A otimização de redes de filas requer formulações matemáticas adequadas que possam representar o problema de maneira precisa e tratável. Este capítulo apresenta diversas formulações matemáticas para a otimização em redes de filas, enfocando particularmente os problemas de alocação de *buffers* e servidores. Embora existam múltiplas possibilidades de modelagem matemática, serão apresentadas as formulações mais relevantes e recorrentes na literatura, com ênfase nas abordagens mono-objetivo e multiobjetivo. Posteriormente, será detalhado o algoritmo metaheurístico Simulated Annealing, técnica de otimização adotada neste estudo para resolver o problema de alocação ótima de recursos em redes de filas Markovianas.

3.1 Formulações para o Problema de Alocação de *Buffers*

O Problema de Alocação de *Buffers* (*Buffer Allocation Problem* - BAP) consiste em determinar o tamanho ideal dos espaços de espera em cada fila de uma rede, visando otimizar alguma medida de desempenho do sistema ou minimizar custos. A seguir são apresentadas duas formulações complementares para este problema: a formulação primal e a formulação dual.

3.1.1 BAP Primal

A formulação primal do BAP, inicialmente apresentada por MacGregor Smith e Cruz (2005) [1], busca minimizar o custo total associado à alocação de *buffers*, garantindo que o sistema mantenha uma taxa mínima de atendimento. Esta formulação é expressa matematicamente como:

$$\text{minimizar} \sum_{i=1}^{m} w_i b_i \tag{3.1}$$

sujeito a:

$$\Theta(B) \geqslant \Theta_{min} \tag{3.2}$$

$$b_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\} \tag{3.3}$$

em que:

- b_i representa o tamanho do buffer na i-ésima fila da rede;
- w_i representa o peso (custo) associado a cada unidade de buffer na i-ésima fila;

- $B = (b_1, b_2, \ldots, b_m)$ é o vetor de alocações de *buffers*;
- $\Theta(B)$ é a taxa de atendimento da rede de filas com a configuração B;
- Θ_{min} é a taxa mínima de atendimento aceitável;
- m é o número de filas na rede.

Esta formulação é particularmente útil em contextos onde o objetivo principal é a redução de custos, como em instalações com espaço físico limitado ou em sistemas em que o custo de implementação de áreas de espera é significativo.

3.1.2 BAP Dual

A formulação dual do BAP inverte a relação entre função objetivo e restrição. Nesta abordagem, busca-se maximizar a taxa de atendimento da rede, sujeito a um limite orçamentário para a alocação total de *buffers*. Matematicamente, tem-se:

$$\text{maximizar } \Theta(B) \tag{3.4}$$

sujeito a:

$$\sum_{i=1}^{m} w_i b_i \leqslant B_{max} \tag{3.5}$$

$$b_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\} \tag{3.6}$$

em que B_{max} representa o limite máximo de recursos disponíveis para alocação em *buffers* ao longo da rede de filas.

A formulação dual é adequada para cenários em que existe um orçamento predefinido para implementação de *buffers*, e o objetivo é maximizar o desempenho do sistema dentro deste limite financeiro. Esta formulação também pode ser vista como uma abordagem de otimização de "melhor valor pelo dinheiro".

3.2 Formulações para o Problema de Alocação de Servidores (SAP)

De forma análoga ao BAP, o Problema de Alocação de Servidores (*Server Allocation Problem* - SAP) visa determinar o número ideal de servidores em cada nó da rede de filas. A seguir, são apresentadas as formulações primal e dual para este problema.

3.2.1 SAP Primal

A formulação primal do SAP busca minimizar o custo total com servidores, mantendo uma taxa mínima de atendimento. Esta formulação é expressa como:

$$\text{minimizar} \sum_{i=1}^{m} w_i c_i \tag{3.7}$$

sujeito a:

$$\Theta(C) \geqslant \Theta_{min} \tag{3.8}$$

$$c_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\} \tag{3.9}$$

onde:

- c_i representa o número de servidores na i-ésima fila da rede;
- w_i representa o peso (custo) associado a cada servidor na i-ésima fila;
- $C = (c_1, c_2, \ldots, c_m)$ é o vetor de alocações de servidores;
- $\Theta(C)$ é a taxa de atendimento da rede com a configuração C de servidores.

Esta formulação é particularmente relevante em ambientes onde servidores representam um investimento significativo, como em centrais de atendimento, hospitais ou sistemas computacionais.

3.2.2 SAP Dual

A formulação dual do SAP, por sua vez, busca maximizar a taxa de atendimento, respeitando um limite orçamentário para contratação ou alocação de servidores. Matematicamente:

$$\text{maximizar } \Theta(C) \tag{3.10}$$

sujeito a:

$$\sum_{i=1}^{m} w_i c_i \leqslant C_{max} \tag{3.11}$$

$$c_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\} \tag{3.12}$$

em que C_{max} representa o limite máximo de recursos disponíveis para alocação de servidores ao longo da rede.

A formulação dual é adequada quando há um orçamento fixo para contratação ou implementação de servidores, e deseja-se maximizar o desempenho do sistema dentro desta restrição orçamentária.

3.3 Formulação para o Problema de Alocação Conjunta de *buffers* e Servidores

O Problema de Alocação de *Buffers* e Servidores (*Buffer and Server Allocation Problem* - BSAP) representa uma abordagem mais abrangente, que considera simultaneamente a alocação de ambos os recursos. Esta integração permite uma otimização mais completa do sistema, capturando as interações entre as decisões de alocação de *buffers* e servidores.

3.3.1 BSAP Mono-objetivo

A formulação mono-objetivo do BSAP, proposta por van Woensel *et al*. (2010) [19], utiliza um parâmetro α para ponderar a importância relativa entre o custo de *buffers* e o custo de servidores. Esta formulação é representada como:

$$\text{minimizar} \sum_{i=1}^{m} \alpha b_i + (1-\alpha) c_i \tag{3.13}$$

sujeito a:

$$\Theta(\mathbf{b}, \mathbf{c}) \geqslant \Theta_{\min} \tag{3.14}$$

$$0 \leqslant \alpha \leqslant 1 \tag{3.15}$$

$$b_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\} \tag{3.16}$$

$$c_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\} \tag{3.17}$$

onde:

- α é o parâmetro que pondera a importância relativa entre *buffers* e servidores;
- $\Theta(\mathbf{b}, \mathbf{c})$ é a taxa de atendimento da rede com as configurações $\mathbf{b} = (b_1, \ldots, b_m)$ de *buffers* e $\mathbf{c} = (c_1, \ldots, c_m)$ de servidores.

O parâmetro α permite ajustar a formulação de acordo com o contexto específico do problema. Quando α está próximo de 1, o custo dos *buffers* torna-se dominante na função objetivo; quando α está próximo de 0, o custo dos servidores ganha maior relevância.

De forma análoga às formulações anteriores, também pode ser definida uma versão dual do BSAP mono-objetivo, que busca maximizar a taxa de atendimento sujeito a uma restrição orçamentária combinada para *buffers* e servidores.

3.3.2 BSAP Multiobjetivo

A formulação multiobjetivo do *Buffer and Server Allocation Problem* (BSAP) equilibra dois critérios antagônicos: (i) o custo total de recursos e (ii) a soma das probabilidades de bloqueio. Os *trade-offs* foram modelados assim:

$$\min_{\mathbf{b},\mathbf{c}} \mathbf{F}(\mathbf{b},\mathbf{c}) = \left[f_1(\mathbf{b},\mathbf{c}), f_2(\mathbf{b},\mathbf{c})\right] \tag{3.18}$$

em que

$$f_1(\mathbf{b},\mathbf{c}) = \sum_{i=1}^{m} \left[\alpha\, b_i + (1-\alpha)\, c_i\right] \tag{3.19}$$

$$f_2(\mathbf{b},\mathbf{c}) = \sum_{i=1}^{m} P_{k_i}\,. \tag{3.20}$$

Sujeito a:

$$\Theta(\mathbf{b},\mathbf{c}) \geqslant \Theta_{\min} \tag{3.21}$$

$$0 \leqslant \alpha \leqslant 1 \tag{3.22}$$

$$k_i = b_i + c_i, \quad b_i \geqslant 0, \quad c_i \geqslant 1, \quad \forall i \in \{1,2,\ldots,m\} \tag{3.23}$$

$$b_i, c_i \in \mathbb{N}, \tag{3.24}$$

em que:

- b_i é o número de posições de *buffer* de espera na fila i;
- c_i é o número de servidores na fila i;
- $k_i = b_i + c_i$ é a capacidade total da fila i (incluindo posições de serviço);
- P_{k_i} é a probabilidade de bloqueio na fila i.

Considere uma fila $M/M/c/k$ com taxa de chegada λ e taxa de serviço μ. Seja p_{j_i} a probabilidade de que a $i-$ésima fila do tipo $M/M/c/k$ na rede tenha j entidades (incluindo entidades em espera e em serviço). Nesse contexto, p_{j_i} é expresso da seguinte forma:

$$p_{j_i} = \begin{cases} \frac{1}{j_i!}\left(\frac{\lambda_i}{\mu_i}\right)^j p_{0_i} & \text{se } j_i < c_i \\ \frac{1}{c_i!}\left(\frac{\lambda_i}{\mu_i}\right)^{c_i}\left(\frac{\lambda_i}{c_i\mu_i}\right)^{j_i-c_i} p_{0_i} & \text{se } c_i \leqslant j_i \leqslant k_i \\ 0 & \text{caso contrário.} \end{cases} \tag{3.25}$$

Agora, defina $r_i = \frac{\lambda_i}{\mu_i}$ e $\rho_i = \frac{r_i}{c_i} = \frac{\lambda_i}{c_i\mu_i}$, para reescrever p_{j_i} como

$$p_{j_i} = \begin{cases} \frac{r_i}{j_i!} p_{0_i} & \text{se } j_i < c_i \\ \frac{r_i^{c_i}\rho_i^{j_i-c_i}}{c_i!} p_{0_i} & \text{se } c_i \leqslant j_i \leqslant k_i \\ 0 & \text{caso contrário.} \end{cases} \tag{3.26}$$

Além disso, é sabido que $\sum_{j_i=0}^{k_i} p_{j_i} = p_{0_i} + \sum_{j_i=1}^{k_i} p_{j_i} = p_{0_i}\left(1 + \sum_{j_i=1}^{c_i}\frac{r_i}{j_i!} + \frac{r_i^{c_i}}{c_i!}\sum_{j_i=c_i+1}^{k_i}\rho_i^{j_i-c_i}\right)$, o que leva à expressão para p_{0_i}:

$$p_{0_i} = \frac{1}{\left(1 + \sum_{j_i=1}^{c}\frac{r_i^{j_i}}{j_i!} + \frac{r_i^{c_i}}{c_i!}\sum_{j_i=c_i+1}^{k_i}\rho_i^{j_i-c_i}\right)}. \tag{3.27}$$

Observe que é possível dividir em dois casos, se $\rho_i \neq 1$ ou se $\rho_i = 1$. Assim, tem-se:

$$\sum_{j_i=c_i+1}^{k_i}\rho_i^{j_i-c_i} = \begin{cases} \frac{\rho_i(1-\rho_i^{k_i-c_i})}{1-\rho_i} & \text{se } \rho_i \neq 1 \\ k_i - c_i + 1 & \text{caso contrário.} \end{cases}$$

Consequently, the expression for p_{0_i} becomes:

$$p_{0_i} = \begin{cases} \left(1 + \sum_{j=1}^{c_i}\frac{r^j}{j!} + \frac{r_i^{c_i}\rho_i(1-\rho^{k_i-c_i})}{c_i!(1-\rho_i)}\right)^{-1} & \text{se } \rho_i \neq 1 \\ \left(1 + \sum_{j=1}^{c_i}\frac{r_i^j}{j!} + \frac{r_i^{c_i}(k_i-c_i+1)}{c_i!}\right)^{-1} & \text{caso contrário,} \end{cases} \tag{3.28}$$

A probabilidade de bloqueio p_{k_i}, que representa a probabilidade de uma entidade não conseguir entrar na fila para atendimento, é dada por:

$$p_{k_i} = \left(\frac{r_i^{c_i}\rho_i^{k_i-c_i}}{c_i!}\right) p_{0_i},$$

que leva à seguinte expressão de forma fechada para p_{k_i}

$$p_{k_i} = \begin{cases} \dfrac{r_i^{c_i}\rho_i^{k_i-c_i}}{\left(c_i! + c_i! \sum_{j=1}^{c_i} \dfrac{r^j}{j!} + \dfrac{r_i^{c_i}\rho_i(1-\rho^{k_i-c_i})}{(1-\rho_i)}\right)} & \text{se } \rho_i \neq 1 \\ \dfrac{r_i^{c_i}}{\left(c_i! + c_i! \sum_{j=1}^{c_i} \dfrac{r_i^j}{j!} + r_i^{c_i}(k_i - c_i + 1)\right)} & \text{caso contrário,} \end{cases} \tag{3.29}$$

em que $r_i = \dfrac{\lambda_i}{\mu_i}$, $\rho_i = \dfrac{r_i}{c_i}$, λ_i é a taxa de chegada na $i-$ésima fila na rede, e μ_i é a taxa de serviço dos servidores na $i-$ésima fila na rede.

O *throughput* global da rede é definido pela soma das entradas não bloqueadas nas filas de saída da rede, na suposição de t filas com saída final da rede a representação seria dada como:

$$\Theta(\mathbf{b}, \mathbf{c}) = \sum_{i=1}^{t} \lambda_i \left(1 - P_{k_i}\right)$$

3.3.3 Foco deste Estudo

Dentre as formulações apresentadas, este trabalho concentra-se na exploração da formulação BSAP multiobjetivo, conforme definida na Seção 3.3.2. Esta escolha se justifica pela natureza intrinsecamente conflitante dos objetivos envolvidos na otimização de redes de filas: a minimização de custos (tanto em *buffers* quanto em servidores) frequentemente se contrapõe à minimização das probabilidades de bloqueio.

Ao adotar uma abordagem multiobjetivo, este estudo visa fornecer uma compreensão mais abrangente dos *trade-offs* envolvidos na alocação de recursos em redes de filas Markovianas e oferecer aos tomadores de decisão uma estimativa do conjunto de soluções Pareto-ótimas que representam diferentes compromissos entre custo e desempenho.

Para resolver este problema multiobjetivo complexo, o clássico algoritmo metaheurístico *Simulated Annealing* será empregado, adaptado para lidar com múltiplos objetivos simultaneamente.

3.4 Otimização Multi-objetivo

Para abordar os problemas de otimização multiobjetivo, é importante salientar que o conceito de ordem fica prejudicado. O conceito de ordenação é natural em espaços unidimensionais, mas não é trivial em espaços multidimensionais. Para definir o conceito de ordenação em espaços multidimensionais, os conceitos de dominância

e fronteira de Pareto precisam ser elucidados [20]. Suponha uma generalização de formulação de otimização multiobjetivo dada por:

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})]\,, \tag{3.30}$$

em que $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^m$ representa o vetor de variáveis de decisão em um espaço viável $\mathcal{X}$, com m dimensões. Portanto, nessa formulação vetorial, cada vetor $\mathbf{x}$ é avaliado por k funções objetivo, não existe hierarquia natural entre elas. Como Deb, (2002) [20] enfatiza, para objetivos que são conflitantes, a otimização multiobjetivo conduz para um conjunto de soluções de compromisso, melhorar um objetivo geralmente exige deteriorar outro.

No contexto da dominância de Pareto, uma solução $\mathbf{x}^* \in \mathcal{X}$ é dita dominante sobre outra $\mathbf{x}^{'} \in \mathcal{X}$, denotada por $\mathbf{x}^* \prec \mathbf{x}^{'}$, se existe ao menos um $j \in \{1, \ldots, m\}$; $f_j(\mathbf{x}^*) < f_j(\mathbf{x}^{'})$ e para os demais valores $i \in \{1, \ldots, m\}$ tem-se $f_i(\mathbf{x}^*) \leqslant f_i(\mathbf{x}^{'})$.

Dada uma solução $\mathbf{x}^* \in \mathcal{X}$ no espaço de objetivos, ela determina um cone de dominância, todas as soluções na região do cone de dominância são dominadas por $\mathbf{x}^*$. A Figura 4 ilustra esse efeito em uma problema hipotético de minimização de dois objetivos.

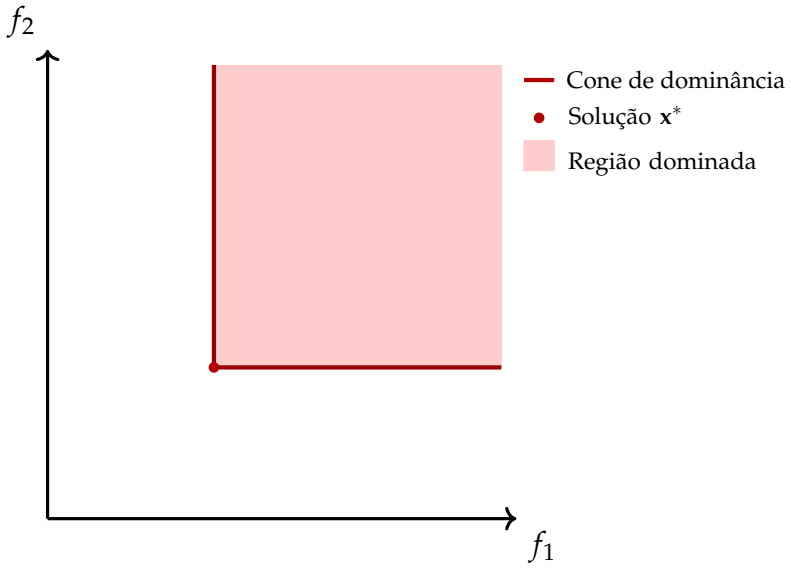


Figura 4 – Cone de Dominância em Problema de Otimização Bidimensional.

A Figura 4 ilustra um espaço bi-objetivo. A linha vermelha representa o cone delimitado pela solução $\mathbf{x}^*$ e a região sombreada indica o conjunto de soluções dominadas pela solução $\mathbf{x}^*$.

Cada solução em estudo determina um cone de dominância, ao investigar um subconjunto de soluções candidatas, a coleção das soluções não interiores à qualquer cone de dominância determina o conjunto de soluções ótimas do subconjunto de solu-

ções sob investigação, ou seja, o conjunto das soluções não-domindas ou fronteira de pareto. Uma ilustração para a fronteira de pareto é apresentada na Figura 5.

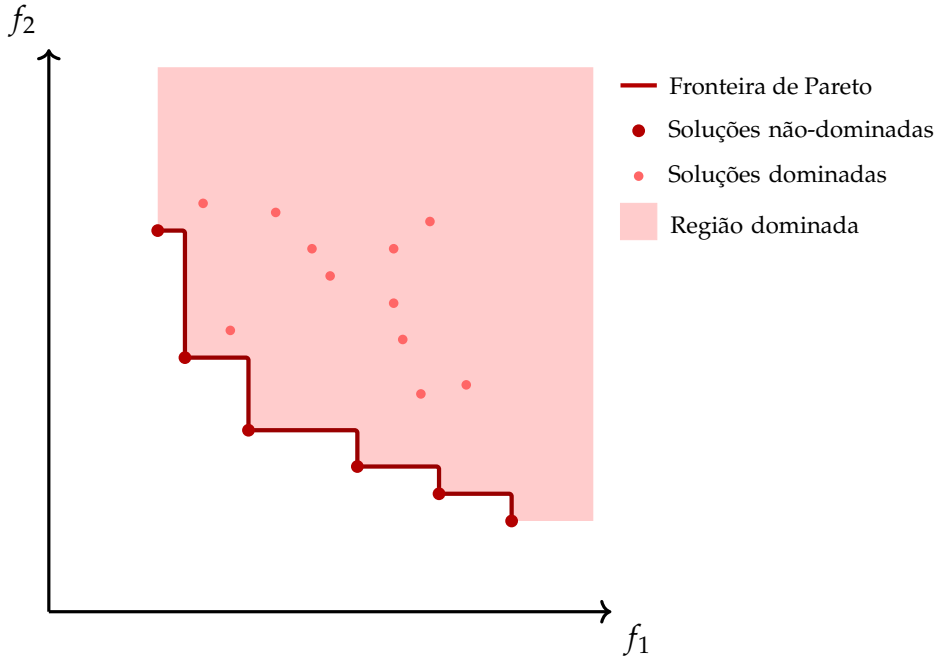


Figura 5 – Fronteira de Pareto Hipotética.

Novamente a Figura 5 apresenta o espaço objetivo com dois critérios conflitantes. Agora, a linha vermelha representa a fronteira de Pareto, formada por soluções eficientes (não-dominadas), nas quais a melhoria de um objetivo implica na piora de outro. Pontos menores representam alternativas dominadas, e a região sombreada indica o conjunto de soluções menos inferior sob a ótica da dominância de Pareto.

3.5 Algoritmo *Simulated Annealing*

O algoritmo *Simulated Annealing* (SA) é uma metaheurística inspirada no processo físico de recozimento de metais, conforme discutido na fundamentação teórica. Essa seção apresenta uma descrição detalhada do algoritmo, sua adaptação para o contexto multiobjetivo e sua aplicação específica ao problema BSAP.

3.5.1 Princípios do Algoritmo

O algoritmo SA é baseado no algoritmo Metropolis (Metropolis *et al.*, 1953 [21]), que simula a evolução de um sistema físico para o equilíbrio térmico. No contexto de otimização, o algoritmo explora o espaço de soluções de maneira controlada, aceitando ocasionalmente soluções piores para escapar de ótimos locais. A probabilidade de aceitar uma solução inferior é determinada pela seguinte expressão:

$$P(\text{aceitar}) = \begin{cases} 1, & \text{se } f(j) < f(i) \\ \exp\left(\dfrac{f(i) - f(j)}{T}\right), & \text{caso contrário} \end{cases} \tag{3.31}$$

em que:

- $f(i)$ é o valor da função objetivo da solução atual i;
- $f(j)$ é o valor da função objetivo da solução candidata j;
- T é o parâmetro de temperatura atual.

Esta probabilidade diminui à medida que a temperatura T é reduzida, seguindo um esquema de resfriamento predefinido. Em temperaturas elevadas, o algoritmo aceita mais frequentemente soluções piores, o que promove uma exploração ampla do espaço de busca. Em temperaturas baixas, o algoritmo se torna mais seletivo, o que refina a busca em regiões promissoras.

3.5.2 Adaptação para Otimização Multiobjetivo

Para aplicar o algoritmo SA ao problema BSAP multiobjetivo, algumas adaptações são necessárias. Ao contrário da otimização mono-objetivo, em que existe uma única medida de qualidade, na otimização multiobjetivo é necessário considerar a dominância de Pareto. Neste estudo, adota-se uma estratégia baseada na dominância para decidir a aceitação de novas soluções:

(i) se uma solução candidata domina a solução atual (melhor em pelo menos um objetivo e não pior nos demais objetivos), ela é aceita com probabilidade 1;

(ii) se uma solução candidata e a solução atual são não-dominadas entre si (cada uma é melhor em pelo menos um objetivo de otimização), a aceitação é baseada em uma medida de distância ao ponto ideal, com probabilidade determinada pela equação (3.31);

(iii) se a solução candidata é dominada pela solução atual, sua aceitação segue a probabilidade dada pela equação (3.31).

3.6 *Fine-tuning* do Algoritmo *Simulated Annealing* para Otimização Multiobjetivo em Rede de Filas

A estrutura de vizinhança na busca por soluções candidatas e o esquema de redução de temperatura estabelece a adapatação adequada para ajustar o algoritmo SA

ao problema específico em investigação.

3.6.1 Estrutura de Vizinhança

A definição de uma estrutura de vizinhança adequada é crucial para o desempenho do algoritmo SA, esse estrutura é usada para estabelecer a solução candidata que será avaliada a cada iteração do algoritmo. Para o problema BSAP, foram definidas seis estruturas de vizinhança, com igual probabilidade de serem selecionadas em cada iteração.

1. **Incremento de *buffer***: Aumentar em uma unidade o buffer de uma fila aleatória i:

$$\text{PerturbSolution}_1 = (b_1, ..., b_{i-1}, b_i + 1, b_{i+1}, ..., b_m) \tag{3.32}$$

2. **Incremento de servidor**: Adicionar um servidor a uma fila aleatória i:

$$\text{PerturbSolution}_2 = (c_1, ..., c_{i-1}, c_i + 1, c_{i+1}, ..., c_m) \tag{3.33}$$

3. **Troca de *buffers***: Trocar os tamanhos dos *buffers* entre duas filas aleatórias i e j:

$$\text{PerturbSolution}_3 = (b_1, ..., b_{i-1}, b_j, b_{i+1}, ..., b_{j-1}, b_i, b_{j+1}, ..., b_m) \tag{3.34}$$

4. **Troca de servidores**: Trocar o número de servidores entre duas filas aleatórias i e j:

$$\text{PerturbSolution}_4 = (c_1, ..., c_{i-1}, c_j, c_{i+1}, ..., c_{j-1}, c_i, c_{j+1}, ..., c_m) \tag{3.35}$$

5. **Decremento de *buffer***: Reduzir em uma unidade o buffer de uma fila aleatória i (desde que $b_i > 0$):

$$\text{PerturbSolution}_5 = (b_1, ..., b_{i-1}, b_i - 1, b_{i+1}, ..., b_m) \tag{3.36}$$

6. **Decremento de servidor**: Remover um servidor de uma fila aleatória i (desde que $c_i > 1$):

$$\text{PerturbSolution}_6 = (c_1, ..., c_{i-1}, c_i - 1, c_{i+1}, ..., c_m) \tag{3.37}$$

Estas estruturas de vizinhança permitem uma exploração diversificada do espaço de soluções, além disso considera tanto modificações incrementais quanto trocas entre diferentes filas.

3.6.2 Esquema de Resfriamento

O esquema de resfriamento define como a temperatura decresce ao longo das iterações do algoritmo. Neste trabalho, adota-se um esquema de resfriamento com redução logaritmica, definido por:

$$T_i = \frac{1}{\log(i+1)} \tag{3.38}$$

em que $i > 0$, T_i é a temperatura na iteração i. A temperatura inicial T_0 é um parâmetro crucial que afeta o desempenho do algoritmo. Uma temperatura inicial muito alta pode levar a uma exploração excessiva sem refinamento suficiente, enquanto uma temperatura muito baixa pode causar convergência prematura para ótimos locais.

3.6.3 Pseudo-código do Algoritmo

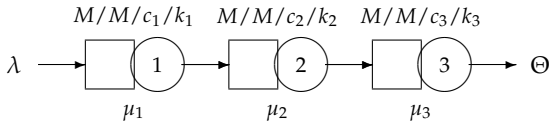
O pseudo-código 1 apresenta a versão ajustada do algoritmo *Simulated Annealing* Multiobjetivo (MOSA) adaptado para o problema BSAP.

Algoritmo 1 Pseudo-código para o Algoritmo MOSA

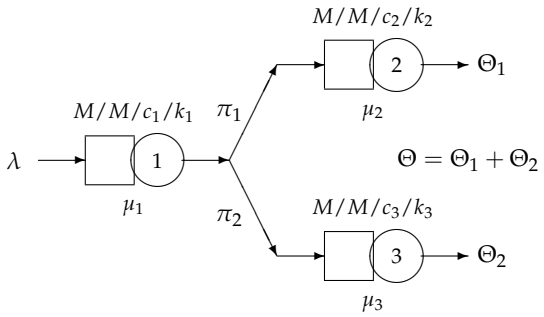
▷ /* Inicialize a população de soluções $P = \{S_1, S_2, ..., S_w\}$ */
▷ /* Inicialize o arquivo externo A para armazenar soluções não-dominadas */
▷ /* Defina a temperatura inicial T_0 e o limiar T_{min} */
para cada solução $S_j \in P$ **faça**
  $T = T_0$
  $S_{atual} = S_j$
  $i = 0$
  while $T_i > T_{min}$ **faça**
    ▷ /* Selecione aleatoriamente uma estrutura de vizinhança */
    ▷ /* Gere uma solução vizinha $S_{vizinha}$ com a estrutura selecionada */
    se $S_{vizinha}$ domina S_{atual} **então** $S_{atual} = S_{vizinha}$
      Atualize $\mathcal{A}$ com $S_{vizinha}$
    else
      se $S_{vizinha}$ e S_{atual} são não-dominadas **então** $\Delta = d(S_{atual}) - d(S_{vizinha})$
        se $\Delta > 0$ ou $\mathcal{U}(0,1) < e^{\Delta/T_i}$ **então** $S_{atual} = S_{vizinha}$
          Atualize $\mathcal{A}$ com $S_{vizinha}$
        fim se
      else $\Delta = d(S_{atual}) - d(S_{vizinha})$
        se $\mathcal{U}(0,1) < e^{\Delta/T}$ **então** $S_{atual} = S_{vizinha}$
        fim se
      fim se
      $i = i + 1$
      $T_i = {}^1/_{\log(i+1)}$
    fim se
  fim while
  $P_j = S_{atual}$
fim para
retorna $\mathcal{A}$
  ▷ /* Retorna o arquivo $\mathcal{A}$ contendo a aproximação da fronteira de Pareto */

4 Resultados Alcançados

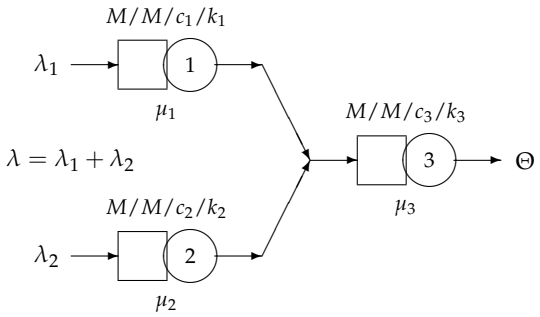
Os experimentos computacionais foram conduzidos para quatro topologias de redes de filas distintas. A Figura 6 apresenta graficamente estas estruturas topológicas, cada uma representa configurações usualmente encontradas em sistemas reais: série (Fig. 6(a)), divisão (Fig. 6(b)), fusão (Fig. 6(c)) e mista (Fig. 6(d)). A escolha dessas topologias permite uma avaliação abrangente do comportamento do algoritmo *Simulated Annealing* em diferentes cenários operacionais.



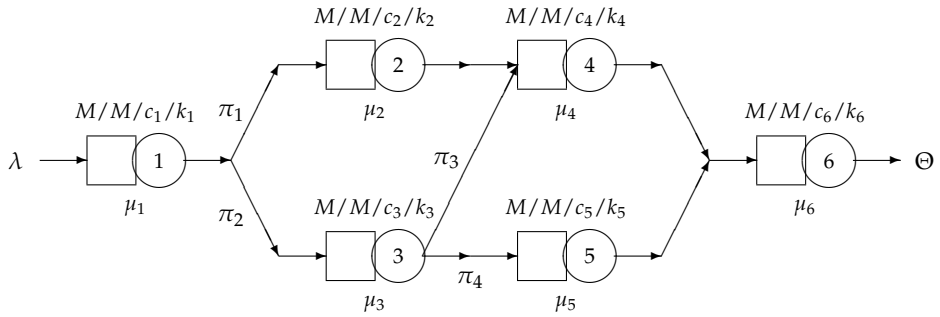
(a) Rede de Filas em Série com 3 Nós.



(b) Rede de Filas em Divisão com 3 Nós.



(c) Rede de Filas em Fusão com 3 Nós.



(d) Rede de Filas Mista com 6 Nós.

Figura 6 – Topologias Experimentais com Redes Variadas.

Para cada topologia, foram realizadas 1000 iterações independentes do algoritmo MOSA (*Multi-Objective Simulated Annealing*), com variação no parâmetro de ponderação

α entre os valores: 0,0625; 0,125; 0,25; 0,5; 0,75; 0,875 e 0,9375. Esta variação permite explorar diferentes compromissos entre a minimização de custos de *buffers* e a minimização de custos de servidores, α próximo de zero implica em *buffers* menos onerosos que os servidores, ao passo que α próximo de um implica em *buffers* mais onerosos que os servidores. O algoritmo foi implementado em linguagem `R`, no ambiente `RStudio` versão 4.3.2. Os experimentos foram conduzidos em um equipamento com processador Intel® Core™ i7-10750H 2.60GHz, Windows 11 Pro 64 bits, com 16,00 GB de memória RAM. Cada execução completa do algoritmo para uma configuração específica teve duração média de 15 minutos.

4.1 Análise Comparativa da Eficácia do Algoritmo

A avaliação da eficácia do algoritmo *Simulated Annealing* multiobjetivo foi conduzida através da análise comparativa entre as soluções iniciais e finais para cada uma das quatro topologias investigadas. Esta análise permitiu identificar não apenas a capacidade geral de melhoria do algoritmo, mas também as características de otimização e os *trade-offs* específicos inerentes a cada configuração topológica.

4.1.1 Análise das Probabilidades de Bloqueio

A redução das probabilidades de bloqueio constitui em um objetivo relevante da formulação BSAP multiobjetivo proposta. Os resultados experimentais, ilustrados nas Figuras 7 a 10, evidenciam um comportamento heterogêneo entre as topologias.

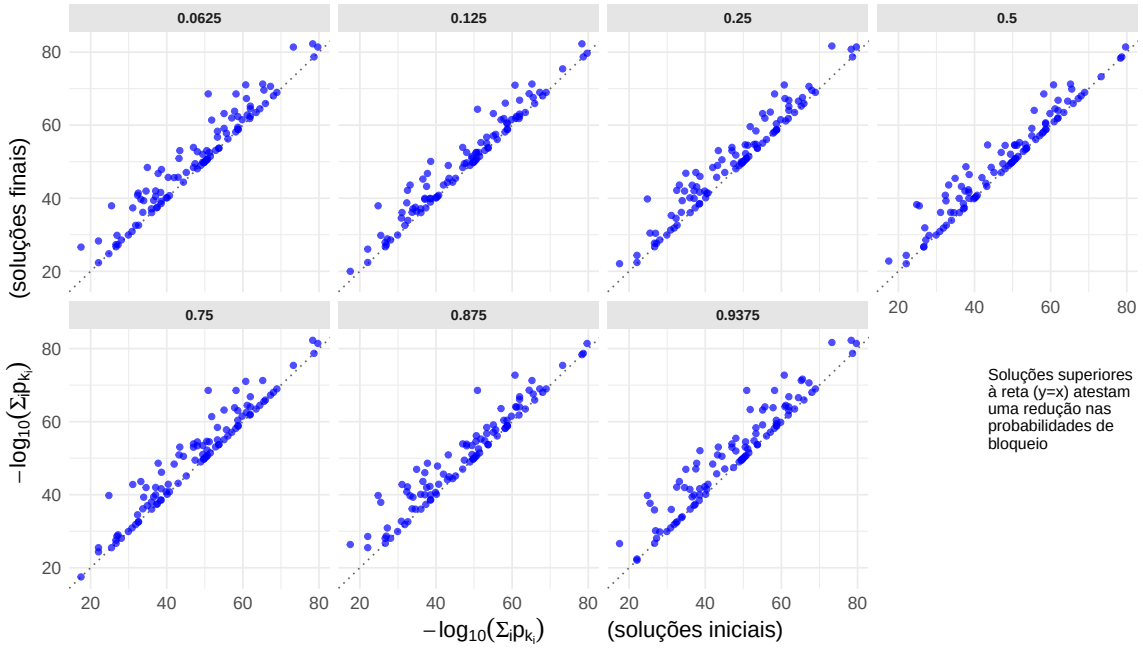


Figura 7 – Análise Comparativa das Probabilidades de Bloqueio para Topologia em Série: Soluções Iniciais versus Soluções Otimizadas.

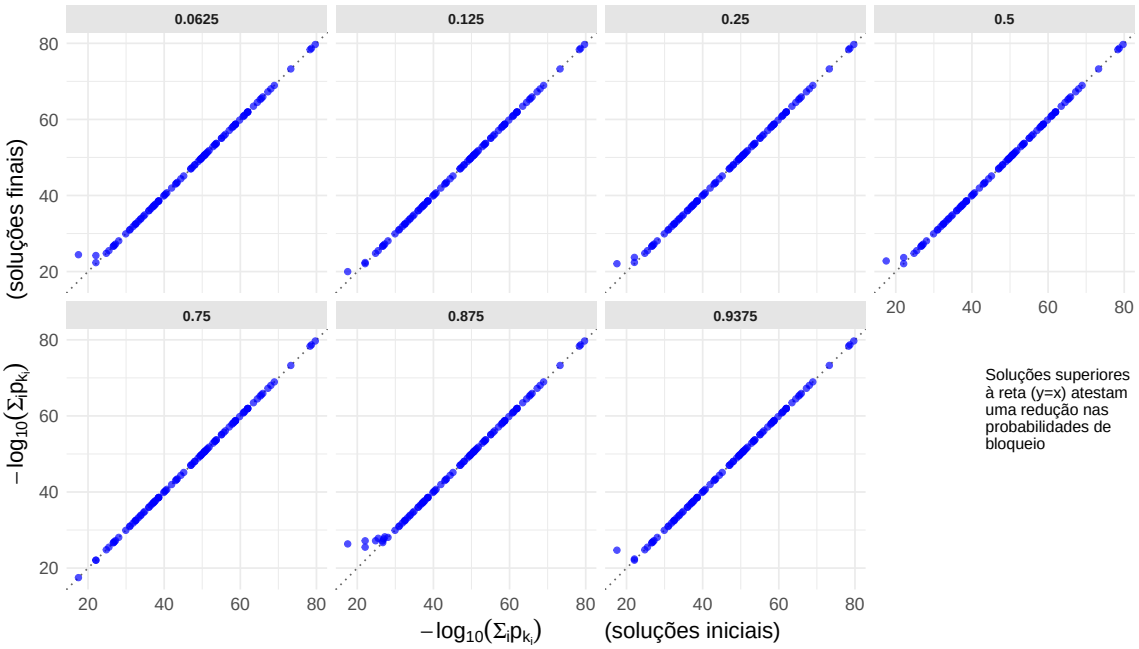


Figura 8 – Análise Comparativa das Probabilidades de Bloqueio para Topologia de Divisão: Soluções Iniciais versus Soluções Otimizadas.

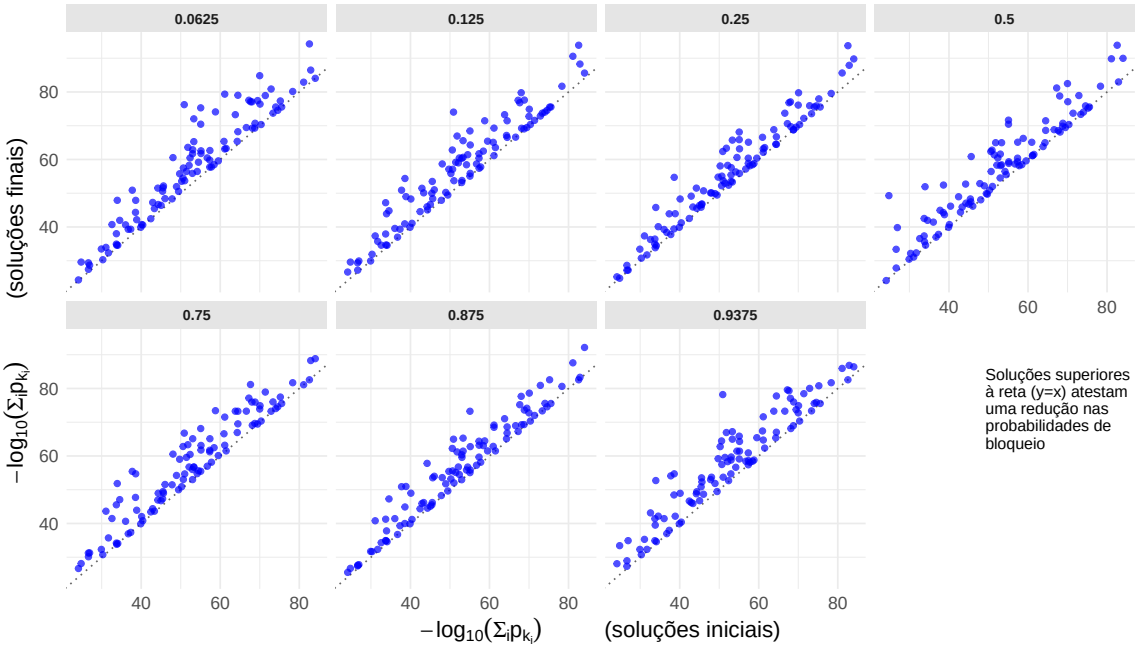


Figura 9 – Análise Comparativa das Probabilidades de Bloqueio para Topologia de Fusão: Soluções Iniciais versus Soluções Otimizadas.

A análise dos gráficos de dispersão revela que as topologias de Fusão (*merge*) e Série (veja Figuras 9 e 7) apresentaram melhorias substanciais e consistentes, com uma clara concentração de pontos otimizados acima da linha de referência $y = x$. Este

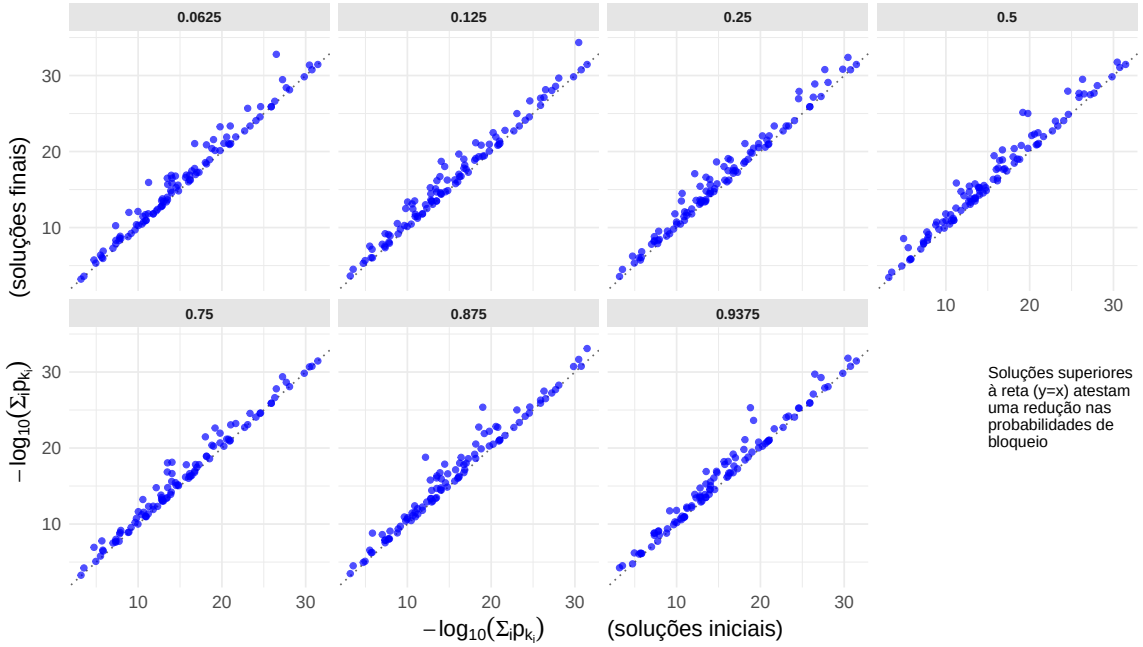


Figura 10 – Análise Comparativa das Probabilidades de Bloqueio para Topologia Mista: Soluções Iniciais versus Soluções Otimizadas.

padrão sugere que o algoritmo explora eficientemente o espaço de soluções para estas configurações.

Em forte contraste, a topologia de Divisão (*split*) (veja Figura 8) exibe um comportamento distinto, com a maioria dos pontos otimizados muito próximos à linha de referência, isso indica que as melhorias de desempenho alcançadas foram marginais. Este resultado pode ser atribuído à complexidade inerente ao balanceamento de múltiplos caminhos de saída, que representa um desafio significativo para o processo de otimização. A topologia Mista (veja Figura 10), por sua vez, apresentou resultados intermediários, consistentes com sua natureza estrutural híbrida.

4.1.2 Redução na Alocação de Recursos

A análise da redução na alocação de servidores e *buffers* constitui outro aspecto fundamental da avaliação da eficácia do algoritmo. As Figuras 11 a 14 ilustram os resultados obtidos para a alocação de servidores, enquanto as Figuras 15 a 18 apresentam os resultados para *buffers*. Os gráficos demonstram a capacidade do algoritmo de identificar configurações mais econômicas, com uma predominância de pontos abaixo da linha de referência $y = x$, indicando redução de recursos. Notavelmente, as topologias de Divisão (*split*) e Mista foram as que apresentaram as reduções mais expressivas de recursos, o que evidencia uma forte tendência do algoritmo em priorizar a economia de custos quando a melhoria de desempenho é estruturalmente limitada. Em

contraste, as topologias de Fusão (*merge*) e Série apresentam reduções mais modestas, concentrando-se na melhoria de performance.

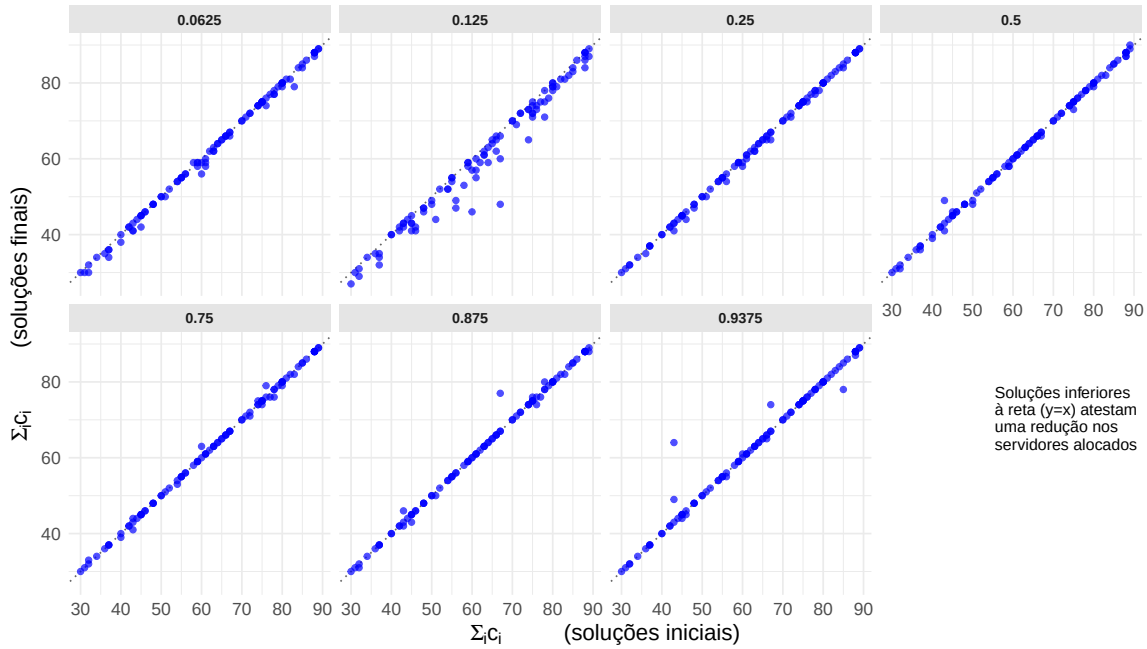


Figura 11 – Redução na Alocação de Servidores para Topologia em Série: Comparação entre Soluções Iniciais e Otimizadas.

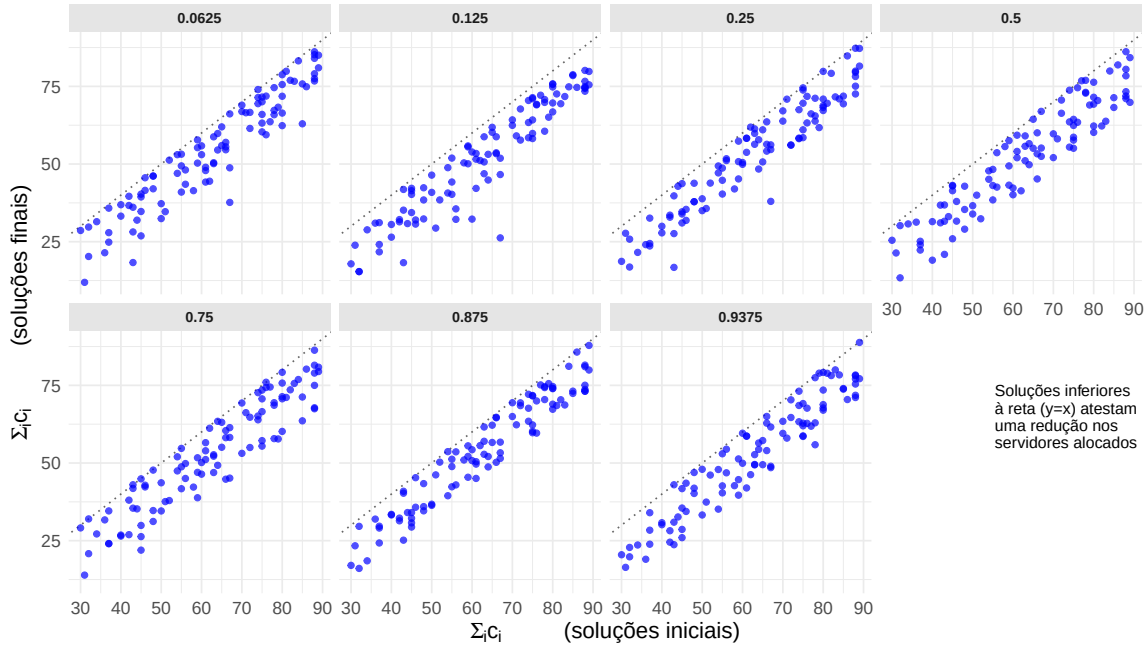


Figura 12 – Redução na Alocação de Servidores para Topologia de Divisão: Comparação entre Soluções Iniciais e Otimizadas.

Esses efeitos identificados para as topologias de Divisão (*split*) e Mista provavelmente são a justificativa de menores efeitos identificados na redução das probabilidades de bloqueio verificados anteriormente.

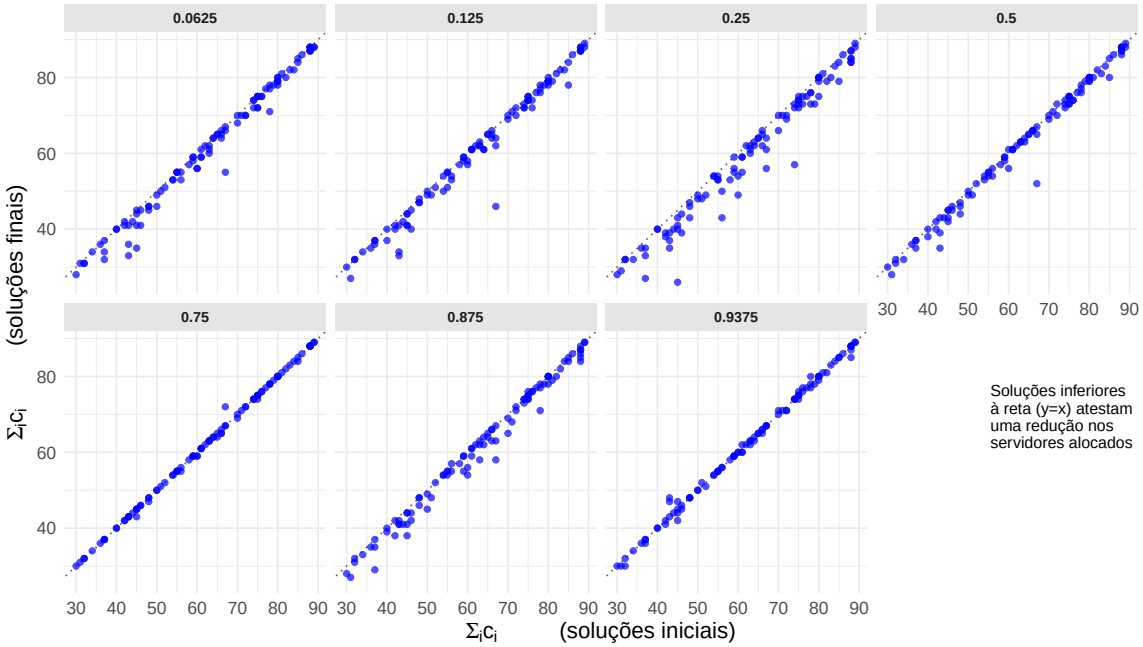


Figura 13 – Redução na Alocação de Servidores para Topologia de Fusão: Comparação entre Soluções Iniciais e Otimizadas.

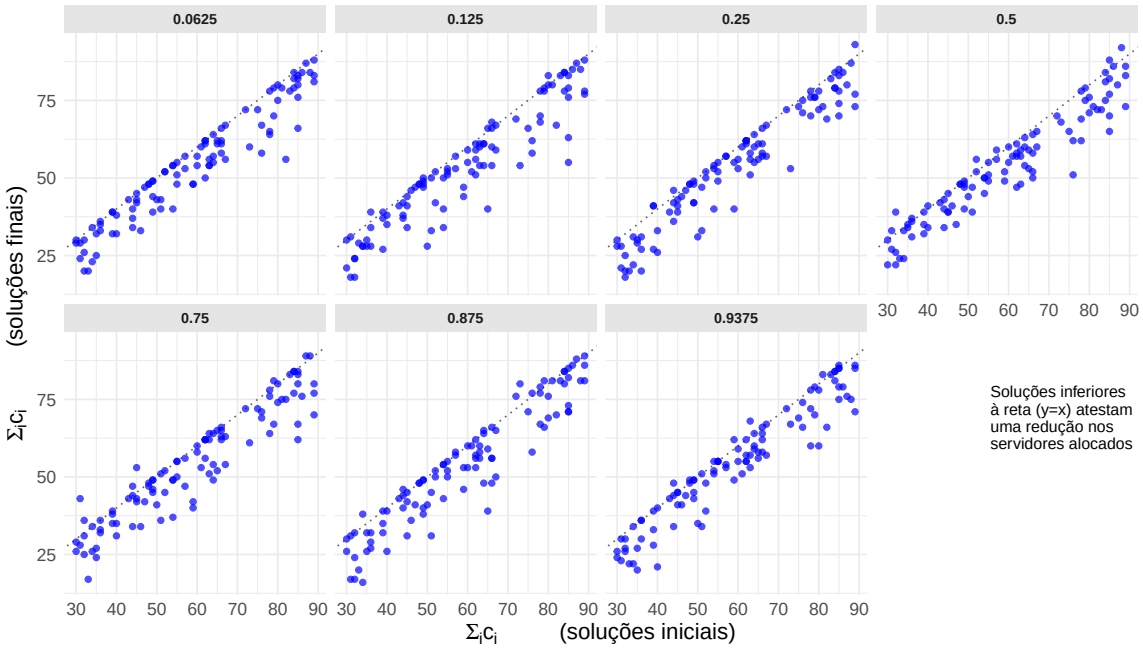


Figura 14 – Redução na Alocação de Servidores para Topologia Mista: Comparação entre Soluções Iniciais e Otimizadas.

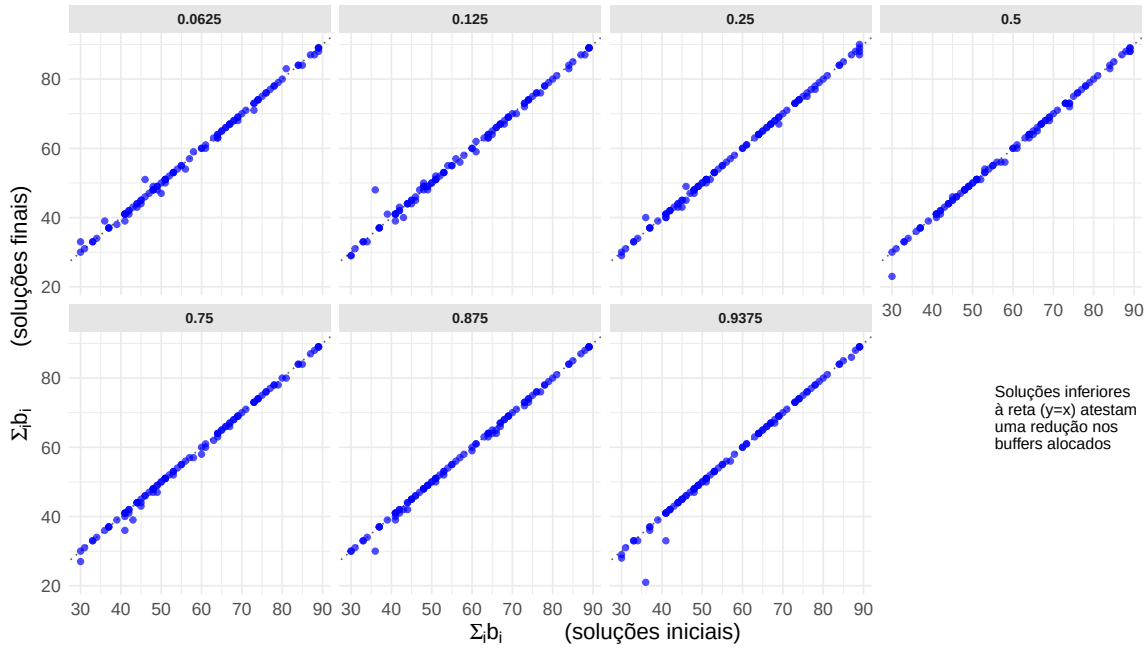


Figura 15 – Redução na Alocação de *Buffers* para Topologia em Série: Comparação entre Soluções Iniciais e Otimizadas.

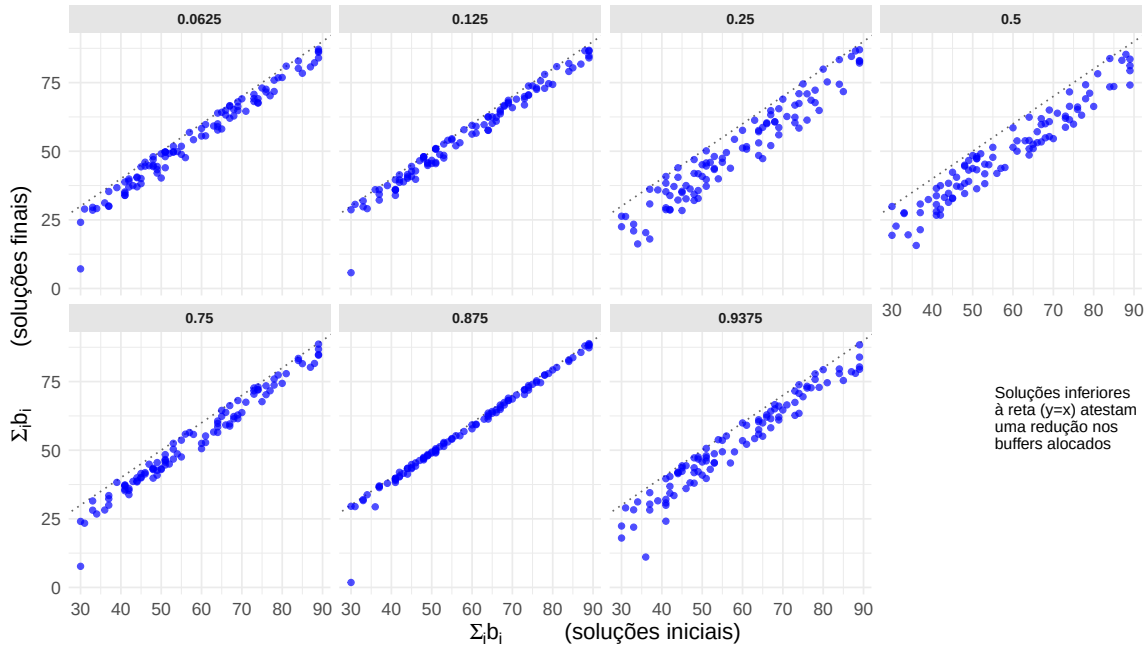


Figura 16 – Redução na Alocação de *Buffers* para Topologia de Divisão: Comparação entre Soluções Iniciais e Otimizadas.

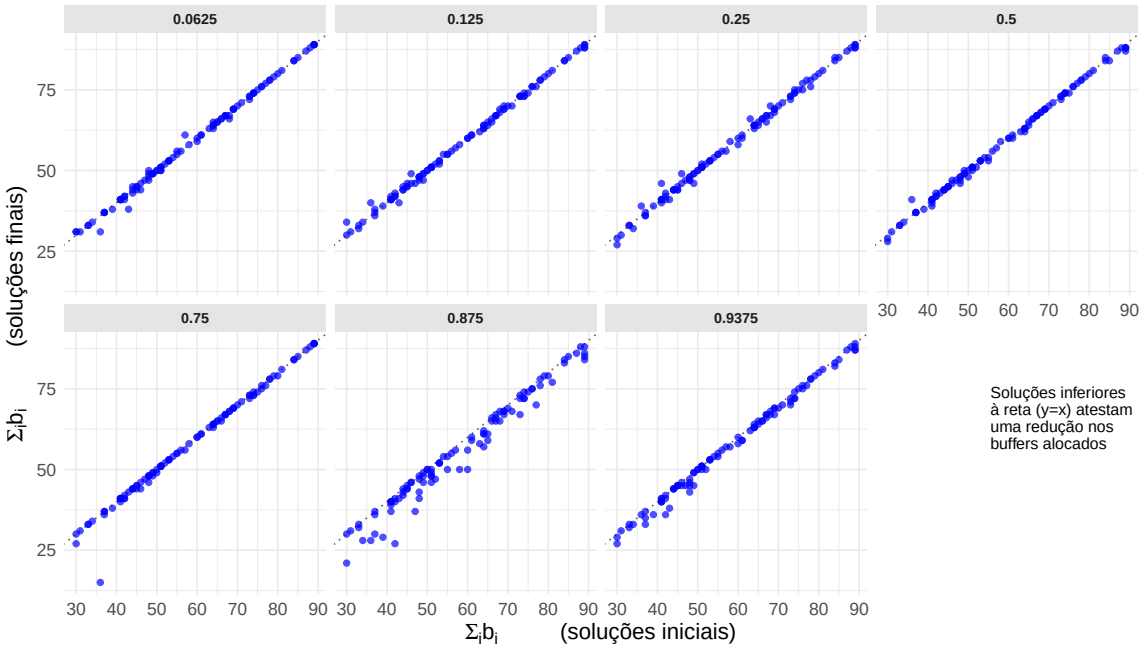


Figura 17 – Redução na Alocação de *Buffers* para Topologia de Fusão: Comparação entre Soluções Iniciais e Otimizadas.

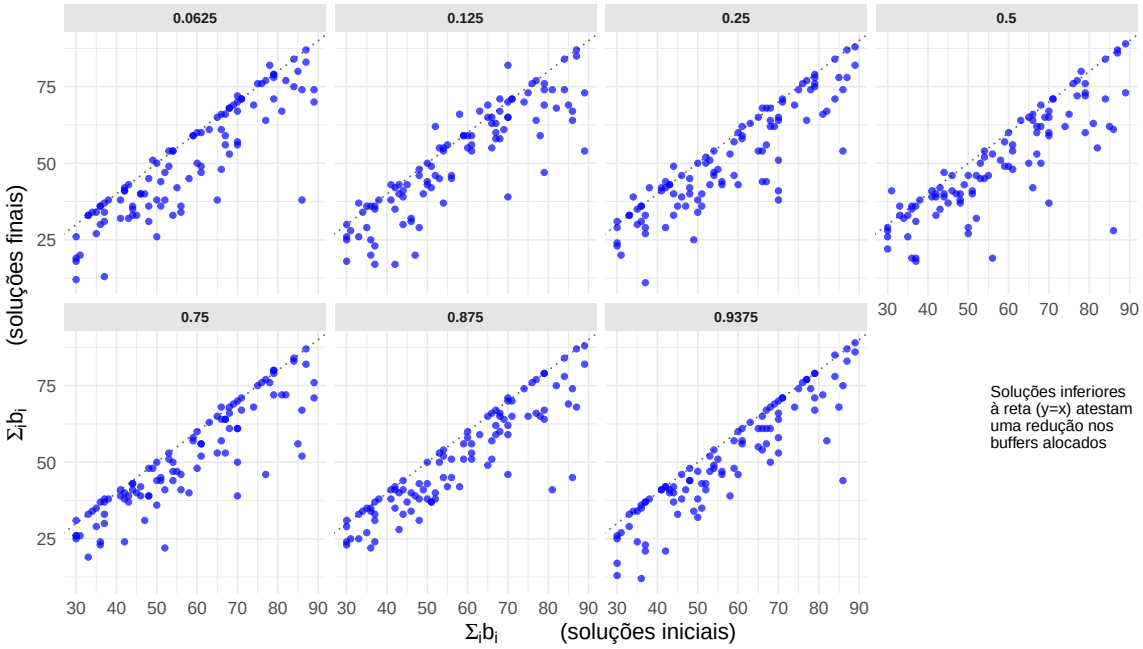


Figura 18 – Redução na Alocação de *Buffers* para Topologia Mista: Comparação entre Soluções Iniciais e Otimizadas.

É relevante observar que a abordagem multiobjetivo permite *trade-offs*. Em cenários específicos, como na topologia em Série com alta prioridade para a redução de *buffers* (valores de α elevados), o algoritmo pode optar por um leve aumento no número de servidores para alcançar uma melhoria de desempenho global, o que valida a flexibilidade da otimização.

É importante salientar que essas análises consideram os efeitos alcançados em objetivos individualizados. Apesar da análise multiobjetivo que foi executada os efeitos podem ser avaliados individualmente o permite identificar as capacidades do algoritmo em cada direção de otimização. Não obstante, uma análise conjunta dos efeitos nos objetivos de otimização é também preponderante.

4.2 Análise das Fronteiras de Pareto

A análise das fronteiras de Pareto constitui um elemento central na avaliação de algoritmos de otimização multiobjetivo. As fronteiras obtidas para cada topologia, segmentadas pelo parâmetro α, são apresentadas nas Figuras 19 a 22. Tais fronteiras evidenciam os *trade-offs* entre o custo combinado de recursos (eixo-x) e o desempenho do sistema (eixo-y), representado pela probabilidade de bloqueio. Nesse ponto os objetivos estão sob análise conjunta para aferir o desempenho do algoritmo.

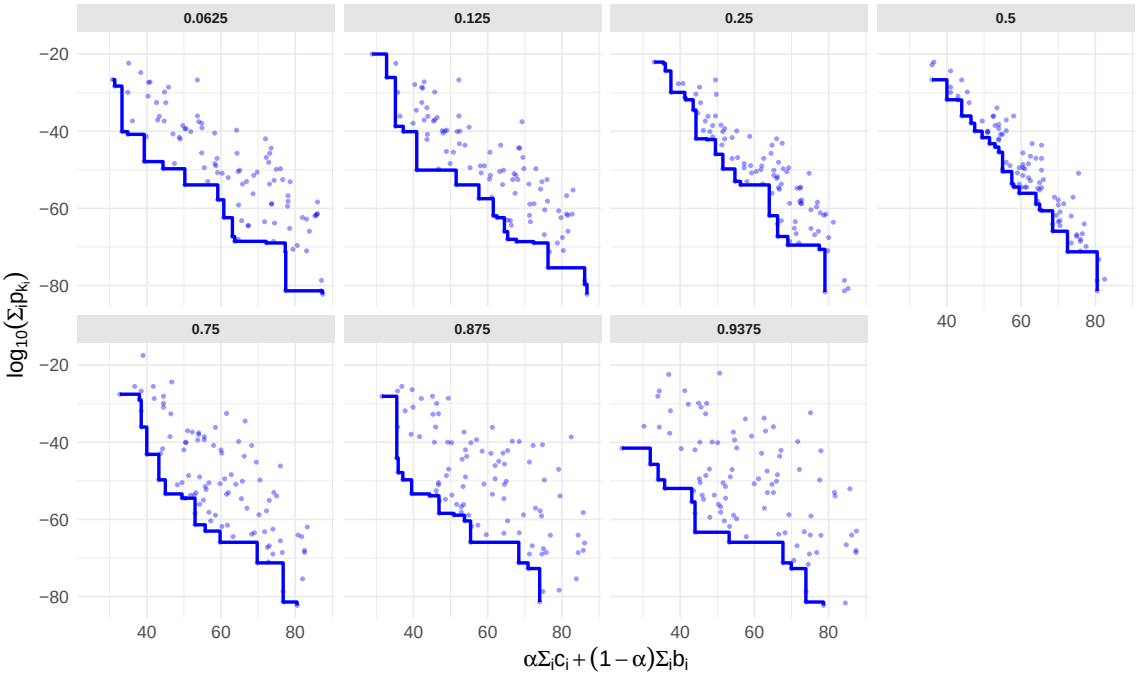


Figura 19 – Fronteiras de Pareto para Topologia em Série, Organizadas por Valor do Parâmetro α.

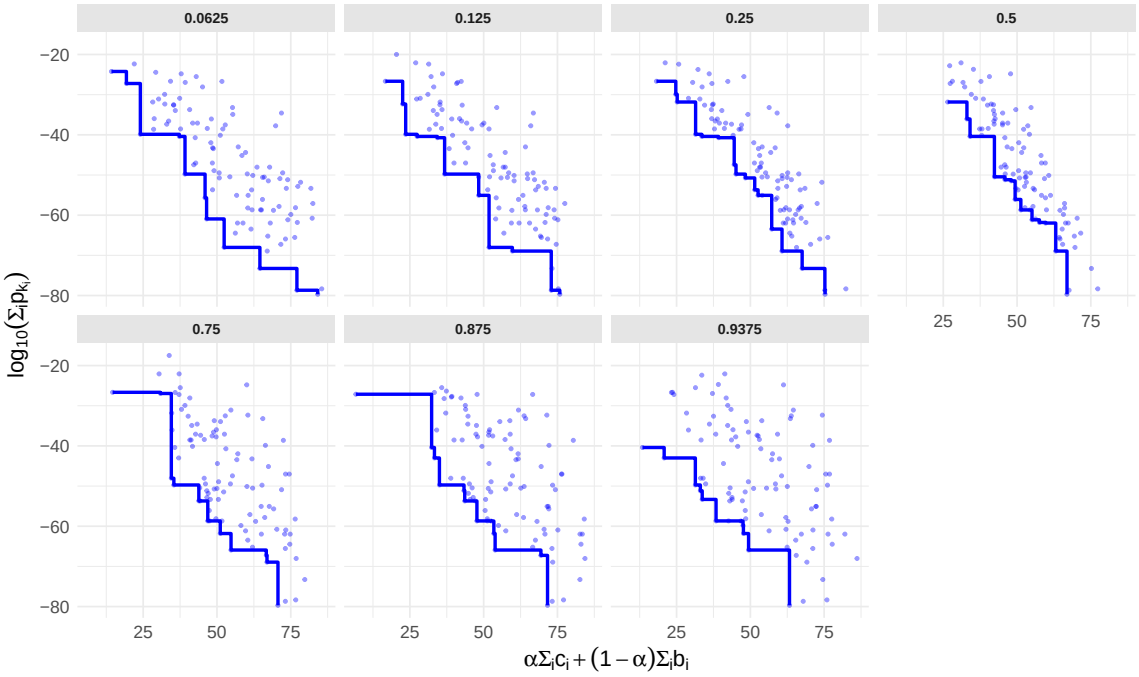


Figura 20 – Fronteiras de Pareto para Topologia em Divisão, Organizadas por Valor do Parâmetro α.

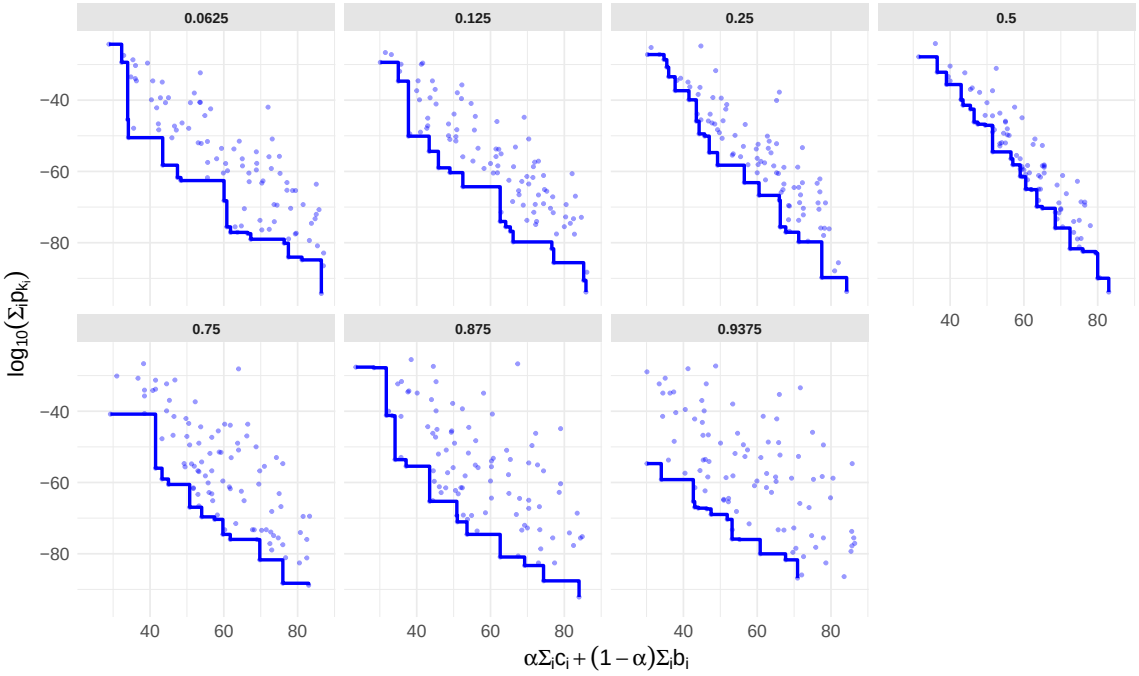


Figura 21 – Fronteiras de Pareto para Topologia em Fusão, Organizadas por Valor do Parâmetro α.

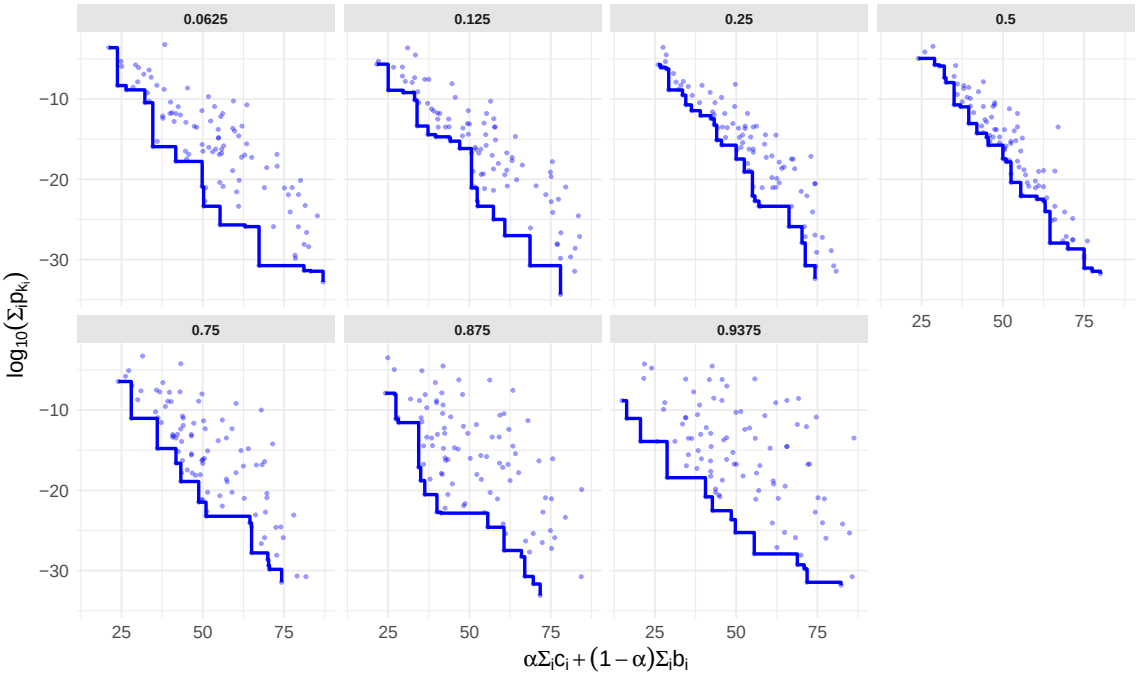


Figura 22 – Fronteiras de Pareto para Topologia em Mista, Organizadas por Valor do Parâmetro α.

4.2.1 Características das Fronteiras por Topologia

A morfologia das fronteiras de Pareto revela características intrínsecas ao espaço de soluções de cada topologia. As topologias de Fusão (*merge*) e Série exibem fronteiras bem estruturadas e com uma amplitude vertical significativa. Este comportamento indica que o algoritmo identificou eficientemente um conjunto robusto de soluções não-dominadas, em que existem múltiplos pontos de equilíbrio entre custo e desempenho.

Em forte contraste, a topologia de Divisão (*split*) apresenta fronteiras notavelmente "achatadas" no eixo vertical. Esta compressão visual confirma que, mesmo com grandes variações na alocação de recursos (eixo-x), o ganho de desempenho (eixo-y) é mínimo. O espaço de soluções parece carecer de alternativas que ofereçam melhorias de performance substanciais.Aparentemente, para esta topologia, o algoritmo priorizou a redução na alocação de recursos em detrimento de grandes melhorias na probabilidade de bloqueio.

Em contraste, a topologia de divisão apresenta fronteiras mais fragmentadas, especialmente para valores intermediários de α. Esta fragmentação pode ser atribuída à complexidade do balanceamento de fluxos em sistemas de divisão, em que pequenas mudanças na configuração podem produzir impactos desproporcionais no desempenho global.

A topologia Mista, por sua vez, exibe fronteiras com características intermediá-

rias, refletindo sua natureza estrutural híbrida e a complexidade resultante da combinação de diferentes padrões de fluxo.

4.2.2 Influência do Parâmetro de Ponderação

O parâmetro α atua como um mecanismo de ponderação para guiar a busca do algoritmo no espaço de soluções. Contudo, os resultados sugerem que sua influência interage de forma complexa com a paisagem de otimização de cada topologia, em vez de determinar o resultado de forma linear.

Observa-se que, embora valores extremos de α modulem a preferência por um tipo de recurso (servidores ou *buffers*), a configuração ótima final é uma consequência da estrutura da rede. Por exemplo, as maiores reduções de um determinado recurso não ocorreram necessariamente nos valores de α que mais o penalizavam. Este comportamento indica que o papel de α é o de guiar a exploração dentro das restrições e possibilidades impostas pela própria topologia, validando a flexibilidade da abordagem para se adaptar a diferentes cenários de custo sem desconsiderar as limitações estruturais do sistema.

4.3 Avaliação Quantitativa da Performance

Para complementar a análise visual, foi realizada uma avaliação quantitativa da performance agregada do algoritmo. Esta análise utiliza métricas estatísticas, consolidadas a partir de todas as execuções para todos os valores de α, o que permite uma comparação objetiva e robusta entre as topologias.

4.3.1 Métricas de Melhoria

A Tabela 2 resume os principais indicadores de desempenho. As métricas incluem a melhoria média na probabilidade de bloqueio (em unidades de $-\log_{10}$), as a quantidade porcentual de soluções que tiveram reduções na alocação de recursos e a taxa de sucesso (quantidade porcentual de execuções que resultaram em melhoria da probabilidade de bloqueio).

Tabela 2 – Resumo Quantitativo da Performance Agregada do Algoritmo MOSA por Topologia.

Topologia	Melhoria	Redução		
	p_k (unidades de $-\log_{10}$)	servidores (%)	*buffers* (%)	p_k (%)
Fusão (*merge*)	4,62	50,00	34,00	91,57
Série	2,94	22,14	16,29	72,00
Mista	0,93	71,00	71,14	81,57
Divisão (*split*)	0,08	99,43	99,00	2,86

Os dados quantitativos revelam alguma segmentação de comportamentos entre as topologias. A topologia de Fusão (*merge*) apresenta resultados importantes em termos de performance em p_k, ao alcançar a maior melhoria numérica de desempenho (4,62 unidades) e a mais alta taxa de sucesso (91,57%), acompanhada de reduções moderadas de recursos (50,00% servidores e 34,00% *buffers*) dentre as soluções.

A topologia em Série demonstra um comportamento menos impactante, com melhoria de desempenho em p_k (2,94 unidades) e reduções mínimas de recursos (22,14% servidores e 16,29% *buffers*) dentre as soluções, o que evidencia uma estratégia conservadora.

Em contraste, a topologia de Divisão (*split*) consolida seu perfil de "otimização agressiva de custos", apresenta as maiores reduções de recursos (99,43% servidores e 99,00% *buffers*) dentre as soluções, porém com melhoria de desempenho marginal (0,08 unidades) com redução em apenas 2,86% das soluções. A topologia Mista apresenta um perfil bastante relevante na economia de recursos (71,00% servidores e 71,14% *buffers*) dentre as soluções com melhoria modesta de desempenho (0,93 unidades),m entretanto alcançando o porcentual de 81,57% das soluções.

4.4 Implicações dos Resultados

A análise experimental fornece *insights* determinantes sobre a interação entre a estrutura da rede de filas e a eficácia do algoritmo de otimização. As diferenças de performance observadas não são aleatórias, mas sim uma consequência direta das características topológicas.

A performance consistentemente superior dos resultados da topologia de Fusão sugere que a convergência de fluxos cria um ambiente de otimização sinérgico. Melhorias realizadas em qualquer um dos ramos de entrada tendem a beneficiar o sistema como um todo, pois o fluxo consolidado permite uma otimização mais eficaz dos recursos no gargalo final. Este efeito de acoplamento positivo parece facilitar a identificação de soluções ótimas pelo algoritmo.

Os resultados para a topologia de Divisão revelam um comportamento otimizador distinto e particularmente interessante. Embora apresente a menor melhoria de desempenho em p_k (0,08 unidades), esta topologia demonstra a maior capacidade de redução de recursos, com reduções sno total de recursos alocados para praticamente todas as soluções.

Este padrão sugere que o algoritmo identifica corretamente as limitações estruturais da topologia para melhorar performance e, como resposta adaptativa, concentra-se na otimização econômica. A complexidade do balanceamento de fluxos em múltiplos

ramos de saída cria um cenário onde grandes investimentos em recursos não se traduzem em melhorias proporcionais de desempenho, levando o algoritmo a uma estratégia de "economia inteligente". Portanto, o comportamento observado não é uma limitação do otimizador, mas uma característica desejável de adaptação às restrições topológicas.

Em suma, os resultados validam a formulação BSAP multiobjetivo e a eficácia do algoritmo *Simulated Annealing* para explorá-la. A capacidade deste de encontrar diferentes pontos de equilíbrio, que variam drasticamente com a topologia, confirma a robustez da abordagem. A análise demonstra que não existe uma estratégia de otimização universal; a eficácia está intrinsecamente ligada à estrutura da rede. Esta conclusão sublinha a importância de se considerar a topologia do sistema como um fator primário em qualquer iniciativa de otimização de redes de filas.

5 Considerações Finais

O presente trabalho dedicou-se à investigação da otimização multiobjetivo em redes de filas Markovianas finitas, um problema de notável relevância tanto teórica quanto aplicada. O objetivo central foi desenvolver e avaliar uma abordagem baseada na metaheurística *Simulated Annealing* (SA) para resolver o Problema de Alocação de *Buffers* e Servidores (BSAP), buscando um equilíbrio ótimo entre dois objetivos conflitantes: a minimização do custo combinado de recursos e a minimização da probabilidade de bloqueio do sistema.

Para tal, foi apresentada uma detalhada revisão da literatura, que fundamentou os conceitos de Teoria de Filas, otimização combinatória e o funcionamento do algoritmo SA. A metodologia foi construída sobre uma rigorosa formulação matemática do problema BSAP multiobjetivo, que serviu de base para a implementação de uma versão adaptada do algoritmo, o MOSA (*Multi-Objective Simulated Annealing*). A eficácia desta abordagem foi sistematicamente testada em quatro topologias de rede distintas — série, fusão, divisão e mista — o que permitiu uma análise comparativa abrangente do comportamento do otimizador em diferentes cenários estruturais.

A principal contribuição desta investigação reside na demonstração empírica de que a eficácia e a estratégia de otimização estão intrinsecamente condicionadas pela topologia da rede. Os resultados evidenciaram comportamentos heterogêneos e reveladores. As topologias de Fusão e Série mostraram-se mais propensas a otimizações que privilegiam o desempenho, alcançou reduções substanciais na probabilidade de bloqueio com uma economia de recursos mais modesta. Em forte contraste, a topologia de Divisão, devido à sua complexidade estrutural no balanceamento de fluxos, revelou-se um cenário onde o algoritmo priorizou uma expressiva redução de custos em detrimento de melhorias marginais de desempenho. Este comportamento não denota uma falha do otimizador, mas sim uma notável capacidade de adaptação, identificando que, em certos sistemas, a busca por desempenho máximo pode ser economicamente inviável, tornando a otimização de recursos a estratégia mais racional.

Tais achados validam a robustez da formulação BSAP multiobjetivo e a flexibilidade do algoritmo *Simulated Annealing* para explorá-la. A análise das fronteiras de Pareto geradas para cada topologia fornece um mapa de *trade-offs* que é de suma importância para um gestor. Fica claro que não existe uma estratégia de alocação de recursos universal; a decisão ótima depende fundamentalmente da arquitetura do sistema e das prioridades estratégicas da organização, sejam elas focadas em custo ou em qualidade de serviço.

Apesar dos resultados promissores, reconhece-se que este estudo possui limitações que abrem caminhos para investigações futuras. A suposição de tempos de chegada e de serviço exponenciais (Markovianos), embora facilite a modelagem matemática, pode não representar com fidelidade todos os sistemas reais. Adicionalmente, os parâmetros do algoritmo SA, como o esquema de resfriamento e as estruturas de vizinhança, foram fixados; a exploração de diferentes parametrizações poderia, potencialmente, refinar ainda mais os resultados.

Nesse contexto, sugerem-se as seguintes propostas para trabalhos futuros:

- Explorar a aplicação da metodologia em redes de filas com distribuições de chegada e serviço mais gerais (e.g., G/G/c/k), o que aumentaria a aplicabilidade do modelo a uma gama mais ampla de problemas práticos.
- Realizar uma análise comparativa do desempenho do MOSA com outras metaheurísticas multiobjetivo consagradas na literatura, como os algoritmos genéticos (e.g., NSGA-II), a fim de avaliar a eficiência relativa das abordagens.
- Incorporar modelos de custo mais complexos na função objetivo, que considerem, por exemplo, o custo da perda de clientes devido ao bloqueio ou custos operacionais variáveis.
- Investigar o problema em um contexto dinâmico, onde as taxas de chegada (λ) e de serviço (μ) podem variar ao longo do tempo, exigindo estratégias de alocação de recursos adaptativas.

Em suma, este trabalho cumpriu com seus objetivos ao apresentar uma metodologia coesa e eficaz para a otimização multiobjetivo de redes de filas, validando a aplicabilidade do *Simulated Annealing* para o problema BSAP e, crucialmente, elucidando a profunda interação entre a topologia da rede e as estratégias de otimização. Espera-se que as conclusões aqui apresentadas possam contribuir para o avanço dos estudos na área da Teoria de Filas e para a tomada de decisão mais informada na gestão de sistemas de serviços complexos.

Referências

[1] MacGregor Smith, J. e F. R. B. Cruz: *The buffer allocation problem for general finite buffer queueing networks*. IIE Transactions, 37(4):343–365, 2005. Citado 2 vezes nas páginas 13 and 3.

[2] Arora, J. S.: *Introduction to optimum design*. Elsevier, 2017. Citado na página 5.

[3] Mataifa, H., S. Krishnamurthy e C. Kriger: *Volt/var optimization: A survey of classical and heuristic optimization methods*. IEEE Access, 10:13379–13399, 2022. Citado na página 6.

[4] Orlin, J. B., A. P. Punnen e A. S. Schulz: *Approximate local search in combinatorial optimization*. SIAM Journal on Computing, 33(5):1201–1214, 2004. Citado na página 6.

[5] Arsham, H.: *Classroom note: Initialization of the simplex algorithm: An artificial-free approach*. SIAM Review, 39(4):736–744, 1997. Citado na página 7.

[6] Floudas, C. A e X. Lin: *Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications*. Annals of Operations Research, 139:131–162, 2005. Citado na página 7.

[7] Shu-Xi, W.: *The improved dijkstra's shortest path algorithm and its application*. Procedia Engineering, 29:1186–1190, 2012. Citado na página 7.

[8] Bulut, M. e E. Özcan: *Optimization of electricity transmission by Ford–Fulkerson algorithm*. Sustainable Energy, Grids and Networks, 28:100544, 2021. Citado na página 7.

[9] McCarl, B. A., H. Moskowitz e H. Furtan: *Quadratic programming applications*. Omega, 5(1):43–55, 1977. Citado na página 7.

[10] Larranaga, P., C. M. H. Kuijpers, R. H. Murga, I. Inza e S. Dizdarevic: *Genetic algorithms for the travelling salesman problem: A review of representations and operators*. Artificial Intelligence Review, 13:129–170, 1999. Citado na página 7.

[11] Dorigo, M. e L. M. Gambardella: *Ant colonies for the travelling salesman problem*. Biosystems, 43(2):73–81, 1997. Citado na página 7.

[12] Cheng, R., M. Gen e Y. Tsujimura: *A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies*. Computers & Industrial Engineering, 36(2):343–364, 1999. Citado na página 7.

[13] Dell'Amico, M. e M. Trubian: *Applying tabu search to the job-shop scheduling problem*. Annals of Operations Research, 41(3):231–252, 1993. Citado na página 7.

[14] Thanh, P. N., O. Péton e N. Bostel: *A linear relaxation-based heuristic approach for logistics network design*. Computers & Industrial Engineering, 59(4):964–975, 2010. Citado na página 7.

[15] Kendall, D. G.: *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 9.

[16] Kirkpatrick, S., C. D. Gelatt e M. P. Vecchi: *Optimization by simulated annealing*. Science, 220(4598):671–680, 1983. Citado na página 11.

[17] Černỳ, V.: *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*. Journal of Optimization Theory and Applications, 45(1):41–51, 1985. Citado na página 11.

[18] Leijoto, H. D.: *Avaliação de Mecanismo Probabilístico de Cooling Schedule para o Algoritmo Simulated Annealing*. Trabalho de Conclusão de Curso de Graduação, Universidade Federal de Ouro Preto, 2024. Citado na página 11.

[19] van Woensel, T., R. Andriansyah, F. R. B. Cruz, MacGregor Smith J. e L. Kerbache: *Buffer and server allocation in general multi-server queueing networks*. International Transactions in Operational Research, 17(2):257–286, 2010. Citado na página 16.

[20] Deb, K., L. Thiele, M. Laumanns e E. Zitzler: *Scalable multi-objective optimization test problems*. Em *Proceedings of the 2002 congress on evolutionary computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, páginas 825–830. IEEE, 2002. Citado na página 20.

[21] Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller e E. Teller: *Equation of state calculations by fast computing machines*. The Journal of Chemical Physics, 21(6):1087–1092, 1953. Citado na página 21.