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

Otimização Multiobjetivo para Filas Markovianas em Rede por Estratégia de Enxame de Partículas

Gean Gabriel de Amorim Ribeiro

Ouro Preto-MG
2025

Gean Gabriel de Amorim Ribeiro

Otimização Multiobjetivo para Filas Markovianas em Rede por Estratégia de Enxame de Partículas

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

2025

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

R484o Ribeiro, Gean Gabriel de Amorim.
Otimização multiobjetivo para filas markovianas em rede por estratégia de enxame de partículas. [manuscrito] / Gean Gabriel de Amorim Ribeiro. - 2025.
59 f.: il.: color., gráf., tab.. + Diagramas.

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

1. Teoria das Filas - Rede de filas. 2. Otimização por Enxame de Partículas. 3. Alocação de recursos. I. Duarte, Anderson Ribeiro. II. Barbosa, Josino José. III. Universidade Federal de Ouro Preto. IV. 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

Gean Gabriel de Amorim Ribeiro

Otimização multiobjetivo para filas markovianas em rede por estratégia de enxame de partículas

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 04 de setembro de 2025

Membros da banca

Dr. Anderson Ribeiro Duarte - Orientador - Universidade Federal de Ouro Preto
Dr. Fernando Luiz Pereira de Oliveira - Universidade Federal de Ouro Preto
Dr. Helgem de Souza Ribeiro Martins - 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 04/09/2025

Documento assinado eletronicamente por **Anderson Ribeiro Duarte**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 09/09/2025, às 16:22, 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 **0974615** e o código CRC **16AC1664**.

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

Agradecimentos

A conclusão deste trabalho representa o ápice de uma jornada de muito aprendizado, persistência e dedicação, que não teria sido possível sem Deus.

Agradeço profundamente também à minha família, pelo apoio incondicional, incentivo constante e paciência em todos os momentos desta caminhada.

Minha gratidão especial ao meu orientador, Prof. Dr. Anderson Ribeiro Duarte, cuja confiança, paciência e orientação foram a bússola deste projeto. Seu rigor metodológico e dedicação não apenas guiaram o desenvolvimento deste trabalho, mas também contribuíram para meu crescimento como estatístico e pesquisador.

À Universidade Federal de Ouro Preto (UFOP) e ao Departamento de Estatística (DEEST), pelo ambiente acadêmico de excelência e pelos recursos que possibilitaram a realização deste estudo.

Estendo também meu agradecimento aos professores do Instituto de Ciências Exatas e Biológicas (ICEB), em especial aos de Estatística, são professores dedicados às suas profissões, que me proporcionaram uma bagagem valiosa de conhecimento, metodologias e maturidade. Cada desafio proposto foi essencial para o meu desenvolvimento como estudante, e suas trajetórias permanecem como inspiração para a minha formação profissional. Aos colegas de curso e de pesquisa, pelo companheirismo, pelas discussões produtivas e pelo apoio mútuo, que tornaram esta caminhada mais leve e enriquecedora.

Aos amigos — tanto os de longa data quanto os que conquistei ao longo desta jornada — sou grato pela amizade, pelo incentivo e pela presença em cada etapa.

A todos, deixo aqui o meu mais sincero e profundo, muito obrigado!

Resumo

Este estudo aborda a modelagem e otimização multiobjetivo da alocação de recursos em redes de filas finitas, um desafiador problema de investigação científica e fundamental em sistemas operacionais sujeitos a restrições físicas e orçamentárias. Em ambientes como linhas de produção, centros de atendimento e redes de telecomunicações, decisões que envolvem a alocação de áreas de espera (do inglês, *buffers*) e servidores exigem o equilíbrio entre múltiplos critérios de desempenho conflitantes — como taxa de atendimento (do inglês, *throughput*), taxa de bloqueio e custo computacional. Tais decisões configuram um problema de otimização multiobjetivo, intrinsecamente não linear, com variáveis mistas e sem solução analítica de forma fechada, o que demanda o uso de métodos heurísticos e estocásticos. O estudo propôs o uso do MOPSO como abordagem de solução robusta, detalha sua estrutura algorítmica baseada em dominância de Pareto, arquivamento elitista de soluções não-dominadas, estratégias de preservação de diversidade e mecanismos adaptativos de atualização de partículas. A modelagem estocástica das redes foi conduzida com base em sistemas do tipo $M/G/1/k$, concedem a quantificação de métricas de desempenho fundamentais. A formulação multiobjetivo resultante foi implementada computacionalmente no *framework* do MOPSO, com operadores de reparo e ajustes dinâmicos que asseguram a factibilidade e eficiência das soluções geradas. A avaliação empírica consistiu na replicação e análise crítica de experimentos previamente consolidados na literatura, isto inclui a comparação entre MOPSO e uma versão do clássico algoritmo genético NSGA-II em redes mistas submetidas a diferentes regimes de variabilidade dos tempos de serviço ($cv^2 = 0{,}5; 1{,}0; 1{,}5$). Os resultados interpretados evidenciaram a superioridade do MOPSO em todos os cenários: produziu fronteiras de Pareto mais densas, convergentes e estáveis, com menor desvio-padrão e menor demanda de recursos para atingir desempenho equivalente ao NSGA-II. Tais achados reforçam a adequação do MOPSO como ferramenta preferencial para otimização multiobjetivo de redes de filas sob incertezas operacionais, com ampla aplicabilidade prática. Espera-se, assim, que este estudo forneça uma base sólida para a otimização multiobjetivo em sistemas de filas complexos e ofereça subsídios práticos para o planejamento e operação de redes de filas em diversas áreas, como manufatura, telecomunicações e transporte; além de fomentar novas investigações voltadas à otimização sob múltiplos critérios em ambientes estocásticos.

Palavras-chave: Otimização Multiobjetivo; Enxame de Partículas; Rede de Filas; Teoria de Filas; Otimização Estocástica; Alocação de Recursos.

Abstract

This study investigates the modeling and multiobjective optimization of resource allocation in finite queueing networks—an inherently complex and fundamental research problem in the domain of operating systems, particularly under physical and budgetary constraints. In environments such as production lines, call centers, and telecommunications networks, resource allocation decisions involving buffers and servers require trade-offs among multiple conflicting performance objectives, including throughput, blocking probability, and computational cost. These decisions form a multi-objective optimization problem characterized by intrinsic nonlinearity, mixed-variable types, and the absence of a closed-form analytical solution, thus necessitating the application of heuristic and stochastic solution methods. To address this challenge, the study proposes the use of MultiObjective Particle Swarm Optimization (MOPSO) as a robust optimization approach. The algorithmic framework of MOPSO is detailed, highlighting its reliance on Pareto dominance, elitist archiving of non-dominated solutions, diversity-preserving mechanisms, and adaptive particle update strategies. The queueing network is modeled stochastically using $M/G/1/k$ systems, which provide the means to quantify key performance indicators. The formulated multiobjective problem is computationally implemented within the MOPSO framework, incorporating repair operators and dynamic adjustment techniques to ensure both the feasibility and efficiency of the generated solutions. Empirical validation involved replicating and critically analyzing benchmark experiments established in the literature. This included comparative evaluations between MOPSO and a variant of the classical NSGA-II genetic algorithm across heterogeneous queueing networks subjected to varying service time variability levels ($cv^2 = 0.5; 1.0; 1.5$). The results consistently demonstrated the superior performance of MOPSO across all tested scenarios, yielding denser, more convergent, and more stable Pareto fronts, with lower standard deviations and reduced resource consumption—while achieving performance on par with, or exceeding, that of NSGA-II. These outcomes underscore the effectiveness of MOPSO as a preferred method for multiobjective optimization of queuing networks under operational uncertainty and its wide-ranging applicability in practice. Consequently, this study aims to provide a rigorous foundation for multi-objective optimization in complex queuing systems and to offer actionable insights for the design and operation of resource-constrained networks in domains such as manufacturing, telecommunications, and transportation. It also paves the way for future research on multi-criteria optimization in stochastic and uncertain environments.

Keywords: Multiobjective Optimization; Particle Swarm; Queueing Network; Queueing Theory; Stochastic Optimization; Resource Allocation.

Lista de ilustrações

Sumário

1 Introdução

A crescente complexidade dos sistemas de filas em redes de serviços — como centros de atendimento, sistemas de tráfego de dados, ambientes produtivos com servidores limitados, entre outros — exige abordagens de modelagem e otimização cada vez mais sofisticadas. Diversas métricas de desempenho, como o tempo médio de espera, taxa de bloqueio, utilização dos servidores e capacidade de *buffers*, frequentemente entram em conflito. Esse fato torna a busca por soluções eficientes um desafio de natureza multiobjetivo (Duarte *et al*. (2024) [1]). Diante da limitação dos métodos analíticos tradicionais — que, embora matematicamente rigorosos, nem sempre são aplicáveis a sistemas grandes, estocásticos ou não lineares — surge a necessidade de técnicas flexíveis e heurísticas para apoiar o processo de tomada de decisão.

Entre os métodos que têm ganhado destaque está o algoritmo *Particle Swarm Optimization* (PSO), proposto por Kennedy & Eberhart (1995) [2]. Inspirado no comportamento coletivo de enxames — como bandos de pássaros e cardumes de peixes — o PSO representa uma poderosa meta-heurística baseada na movimentação e interação de partículas em um espaço de busca. Cada partícula representa uma solução candidata que ajusta sua trajetória com base em experiência individual e na experiência coletiva do grupo. A simplicidade estrutural e a eficácia exploratória do PSO têm motivado uma extensa variedade de aplicações, e seu modelo evolutivo tem sido continuamente aperfeiçoado com elementos como fatores de inércia (Shi e Eberhart (1998) [3]), estratégias topológicas, técnicas de controle adaptativo e extensões multiobjetivo — conhecidas como MOPSO (Poli *et al*. (2007) [4]). Este relatório tem por objetivo consolidar o conhecimento teórico e metodológico necessário para a compreensão aprofundada do PSO, com foco na sua estrutura algorítmica, nos fundamentos que o motivam e nas principais variantes já propostas, além disso, propor uma estrutura adequada para a adaptação do algoritmo PSO ao clássico problema de otimização em redes de filas.

A investigação se desdobra em três frentes complementares: (i) apresentar didaticamente o funcionamento do PSO clássico, com base em seus componentes fundamentais e lógica de atualização; (ii) explicar as adaptações necessárias para a formulação multiobjetivo, com a inclusão de mecanismos como arquivamento de soluções não dominadas, seleção de líderes e estratégias de preservação da diversidade em frentes de Pareto; e (iii) discutir, de forma teórica e geral, a aplicação do PSO — especialmente de sua versão multiobjetivo — em estudos de Filas em redes, sem a realização de experimentações, mas com foco na modelagem, formulação e abordagens já existentes na literatura. A proposta visa, portanto, oferecer uma base sólida para o domínio conceitual do algoritmo PSO e sua adaptabilidade a sistemas de filas, o que contribui

para a formação crítica do pesquisador e para o entendimento das potencialidades desta técnica como ferramenta de otimização em contextos em que múltiplos objetivos conflitantes e restrições operacionais complexas se fazem presentes.

1.1 Objetivos

Os objetivos do presente estudo são:

- apresentação de uma revisão da bibliográfica na área de modelagem e otimização de redes de filas finitas e aplicações do algoritmo *Particle Swarm Optimization*;
- utilização da linguagem TEX, que é padrão na confecção de textos estatísticos em vários níveis de pesquisa;
- apresentação e detalhamento do algoritmo *Particle Swarm Optimization* em versão multiobjetivo;
- investigação e adaptação do algoritmo *Particle Swarm Optimization*, como ferramenta de resolução do BSAP.

O conteúdo está organizado da seguinte maneira: o primeiro capítulo tem caráter introdutório, em sequência, o capítulo 2 aborda os fundamentos e a estrutura do algoritmo Otimização por Enxame de Partículas (PSO) clássico; o capítulo 3 discute suas extensões para otimização multiobjetivo e apresenta uma aplicação específica em redes de filas; por fim, o capítulo 4 apresenta as conclusões alcançadas através dessa investigação e também propostas de continuidade desse estudo.

2 Abordagem do Problema e Aspectos Metodológicos

Os métodos de otimização desempenham um papel fundamental nas atividades do cotidiano. Seja na tentativa de reduzir gastos durante as compras ou ao escolher o trajeto mais eficiente para evitar o tráfego, busca-se constantemente soluções melhores. Mesmo que de forma inconsciente, há um esforço em representar esses desafios por meio de modelos matemáticos com o objetivo de maximizar ganhos ou minimizar perdas.

Entre suas múltiplas aplicações, destaca-se a otimização em sistemas de filas, em que é possível aumentar a eficiência pelo ajuste de variáveis como a quantidade de atendentes, o número de vagas para espera, entre outros fatores que compõem esse tipo de estrutura.

A relevância dos problemas de otimização se evidencia na necessidade de aprimorar processos em diferentes domínios da vida moderna. Por essa razão, a pesquisa científica nessa área é amplamente explorada. Além de ser bastante investigada, trata-se de uma linha de estudo envolvente, graças à sua ampla gama de possibilidades práticas.

No contexto do dia a dia, os desafios de otimização estão constantemente presentes. De forma geral, otimizar significa encontrar a alternativa mais eficaz para executar uma tarefa, visando sempre um desempenho superior. Por exemplo, ao comprar produtos, o objetivo costuma ser gastar menos e obter maior qualidade ou quantidade. Na organização de um trajeto, busca-se o percurso mais curto ou menos congestionado.

Sob a ótica acadêmica, muitos desses problemas são representados por modelos matemáticos. Nesse campo, a otimização está diretamente relacionada à maximização ou minimização de uma função que represente algum aspecto relevante do problema. Segundo Yang (2010) [5], os estudos na área de otimização abrangem uma extensa gama de situações e contextos. Toda vez que se procura atingir um certo grau de excelência ou eficiência, está-se, de fato, diante de um problema de otimização.

Diversos desses problemas são tratados por meio de métodos heurísticos, que buscam soluções satisfatórias, embora não garantam a obtenção do ótimo global. No cotidiano das pessoas, é comum o uso de estratégias heurísticas, nas quais as decisões são tomadas com base em experiências anteriores. Muitas vezes, essas escolhas produzem bons resultados, ainda que não sejam as melhores possíveis, e podem variar em eficácia conforme a situação.

Dessa forma, os métodos heurísticos geram soluções que, apesar de subótimas,

costumam se aproximar dos ótimos conforme a eficiência da técnica utilizada e sua capacidade de adaptação ao problema em questão. Tais métodos, em geral, apresentam maior rapidez e simplicidade em comparação com abordagens exatas, o que os torna mais viáveis para diversas aplicações. Entre as técnicas mais comuns destacam-se as heurísticas construtivas e as de busca local, que partem de uma solução inicial e a aprimoram gradativamente.

Este estudo concentra-se na utilização de métodos heurísticos de otimização, com foco específico em uma técnica que será detalhada ao longo do texto. Ao tratar de Redes de Filas, o trabalho refere-se diretamente a situações do cotidiano em que ocorrem filas de espera, como em agências bancárias, unidades de saúde, congestionamentos no trânsito ou até mesmo em ambientes digitais, como sistemas de atendimento por telefone (*call centers*). O objetivo é investigar a aplicação de estratégias de otimização para melhorar a eficiência operacional desses sistemas.

O processo de otimização envolve encontrar a melhor forma de alocar recursos para maximizar ou minimizar determinadas métricas de desempenho. Tais métricas podem ser representadas por variáveis discretas ou contínuas, como o número de servidores ou a capacidade de *buffers*.

Para tratar problemas de otimização em sistemas de filas de forma eficaz, é necessário seguir uma metodologia estruturada, que inclua a definição clara da função objetivo, a identificação dos parâmetros relevantes, o estabelecimento do espaço de busca e das restrições envolvidas. A escolha de algoritmos adequados também é essencial na obtenção de soluções próximas ao ótimo. Neste estudo, o foco recai sobre o estudo e aplicação de uma técnica heurística específica conhecida como Otimização por Enxame de Partículas, ou (PSO, do inglês *Particle Swarm Optimization*), que será explorada com maior profundidade nas seções seguintes.

2.1 Fundamentos do PSO

O clássico algoritmo PSO foi proposto por Kennedy & Eberhart (1995) [2]. Ao passar dos anos sua aplicabilidade foi difundida e suas versões foram aprimoradas. Inicialmente serão apresentados detalhes de sua concepção, evolução até convergir para a aplicação particular em redes de filas.

2.1.1 Motivação e Inspiração Biológica

O PSO tem sua gênese nas primeiras simulações computacionais do comportamento coletivo em bandos de aves e cardumes de peixes, desenvolvidas por Reynolds [6] e por Heppner & Grenander na década de 1990 [7], em que simples regras locais de interação, baseadas na manutenção de distâncias ótimas entre indivíduos, geravam

dinâmicas de grupo sincronizadas e surpreendentemente complexas. Ao desfrutarem dessas ideias, Kennedy & Eberhart [2] adotaram a metáfora do "enxame de partículas" para descrever uma população de agentes que, dotados de memória limitada e comunicação restrita aos vizinhos mais próximos, atualizam suas posições em um espaço de busca multidimensional por meio de atração vetorial às melhores soluções já encontradas tanto por si mesmos (*pbest*) quanto pelo grupo como um todo (*gbest*). A escolha do nome reflete fielmente essa analogia: cada "partícula" individual explora o domínio de forma autônoma, mas permanece imersa em um "enxame" social, cujo comportamento global emerge da conjugação de memórias individuais e de normas coletivas, sempre sem recorrer a operadores de recombinação ou mutação típicos de algoritmos genéticos.

Essa dualidade cognitivo-social confere ao PSO sua notável capacidade de equilibrar exploração e explotação com mínima complexidade estrutural, o que permite convergência eficiente, implementação simples e escalabilidade, o que também explica sua ampla consolidação como método de otimização em domínios contínuos e discretos e sua aplicação em campos tão diversos quanto Engenharia e Aprendizado de Máquina (Wang *et al*. (2018) [8]). Além disso, a robustez desse paradigma pode ser compreendida à luz dos princípios fundamentais da "inteligência de enxame" (*A-life*) delineados por Millonas, (1996) [9] , segundo os quais um sistema inspirado em coletivos naturais deve satisfazer um conjunto de requisitos inter-relacionados: *proximidade* — cada agente interage apenas com vizinhos próximos no espaço e no tempo; *qualidade* — a comunidade deve reagir a indicadores de qualidade no ambiente; *resposta diversa* — deve evitar convergência prematura, manter múltiplos canais de busca; *estabilidade* — não mudar de comportamento a cada pequena perturbação ambiental; *adaptabilidade* — alterar seu modo de atuação quando justificado pelo ganho em exploração.

O PSO cumpre esses princípios ao permitir que cada partícula realize cálculos de posição e velocidade em etapas discretas de tempo (proximidade), reaja aos valores de *pbest* e *gbest* (qualidade), preserve diversidade via seleção de líderes distintos (resposta diversa), somente ajuste seu rumo quando *gbest* se modifica significativamente (estabilidade) e adapte-se à medida que novas informações surgem (adaptabilidade). A motivação biológica que fundamenta o algoritmo estende-se ainda aos conceitos de "nostalgia" e "conformismo" social, uma vez que a memória autobiográfica representada por *pbest* imprime às partículas uma tendência de retorno ao ponto de maior satisfação histórica, ao passo que a atração pelo melhor global, *gbest*, espelha a influência de normas sociais sobre o comportamento coletivo e, resulta em uma dinâmica que equilibra de modo singular a iniciativa individual e a conformidade ao sucesso do grupo (Wang *et al*. (2018) [8]).

2.2 Componentes da Partícula

O algoritmo PSO, conforme formulado originalmente por Kennedy & Eberhart (1995) [2], baseia-se na analogia com o comportamento coletivo de enxames, atribui a cada partícula a função de um agente computacional que realiza uma busca autônoma, mas socialmente influenciada, no espaço de soluções. Essa busca é conduzida por mecanismos de memória individual e influência grupal, os quais definem a trajetória das partículas em direção a regiões promissoras do espaço de busca. A essência desse modelo está na representação vetorial da posição e da velocidade de cada agente, elementos que não apenas caracterizam seu estado atual, mas também determinam sua capacidade de aprendizado e adaptação ao longo do tempo.

Seja $D \in \mathbb{N}$ o número de variáveis de decisão do problema. A posição de uma partícula i na iteração t é descrita por um vetor $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \in \mathbb{R}^D$, cujos componentes são geralmente inicializados aleatoriamente dentro dos limites inferiores e superiores de cada dimensão, $x_d \in [x_d^{\min}, x_d^{\max}]$. Dessa forma, fica garantida uma distribuição inicial ampla e não tendenciosa do enxame no espaço de busca. A velocidade correspondente, expressa por $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, define a magnitude e a direção do deslocamento da partícula e funciona como vetor de transição que promove o movimento entre diferentes pontos do espaço de soluções. Além desses atributos cinemáticos, o PSO incorpora elementos de aprendizagem inspirados em princípios de cognição social e adaptabilidade. Cada partícula mantém um vetor de memória $\mathbf{pbest}_i(t)$, que armazena a melhor posição visitada por ela até a iteração t, de acordo com o valor da função-objetivo f. Este comportamento, muitas vezes interpretado como "nostalgia computacional", é crucial para garantir a intensificação local em torno de soluções previamente satisfatórias [8,9]. A atualização da memória individual ocorre de forma condicional, conforme a regra:

$$\mathbf{pbest}_i(t) = \begin{cases} x_i(t), & \text{se } f(x_i(t)) \text{ supera } f(pbest_i(t-1)) \\ pbest_i(t-1), & \text{caso contrário} \end{cases} \tag{2.1}$$

A afirmação de que $f(x_i(t))$ supera $f(pbest_i(t-1))$ remete aos contextos tanto de minimização quanto maximização. Para complementar a aprendizagem individual, o PSO emprega um mecanismo de aprendizagem coletiva representado pelo vetor $\mathbf{gbest}(t)$, o qual sintetiza a melhor solução global identificada por qualquer partícula do enxame até o instante t. Essa posição globalmente ótima é definida como:

$$\mathbf{gbest}(t) = \arg\operatorname{opt}_{1 \leqslant j \leqslant N} f\big(pbest_j(t)\big), \tag{2.2}$$

em que N é o número total de partículas, opt remete a ideia de otimizar tanto no senso

de minimização quanto de maximização. O vetor $\mathbf{gbest}(t)$ representa, portanto, uma forma de consenso emergente no enxame e exerce um papel de liderança coletiva: ele influencia simultaneamente todas as partículas e, serve como referencial de excelência alcançada pelo grupo. Em termos comportamentais, essa componente social evoca o conceito de conformidade à norma coletiva, atua como vetor de convergência que impulsiona as partículas a se aproximarem de regiões do espaço onde o sucesso já foi constatado. O contraste entre $\mathbf{pbest}_i(t)$ e $\mathbf{gbest}(t)$ expressa, assim, a dualidade fundamental do PSO: a tensão entre a autonomia da experiência individual e a pressão social de grupo, cuja interação equilibrada resulta em um comportamento coletivo eficiente na exploração do espaço de soluções [2, 8].

A movimentação das partículas é modulada por um conjunto de parâmetros que determina a intensidade relativa das forças envolvidas nesse processo de aprendizado. O peso de inércia ω regula a influência da velocidade anterior, atua como fator de persistência que favorece a exploração do espaço quando elevado e a estabilização da trajetória quando reduzido, como proposto por Shi & Eberhart (1998) [3]. Os coeficientes de aceleração c_1 e c_2, conhecidos respectivamente como parâmetros cognitivo e social, controlam a atratividade das melhores posições individuais e globais. Em conjunto, esses parâmetros permitem calibrar o comportamento do enxame entre os extremos da exploração irrestrita e da rápida convergência. Para evitar que o processo de atualização se torne determinístico — o que comprometeria a robustez da busca —, cada termo direcional é multiplicado por fatores aleatórios r_1 e r_2, sorteados a cada iteração conforme distribuições uniformes no intervalo $[0, 1]$, ou seja: $r_1, r_2 \sim \mathcal{U}(0, 1)$. Essa variabilidade estocástica, embora simples, desempenha papel fundamental na introdução de diversidade populacional e na prevenção da convergência prematura a mínimos locais.

Com todos esses componentes definidos — posição, velocidade, memórias **pbest** e **gbest**, além dos parâmetros de controle ω, c_1, c_2, r_1 e r_2 — tem-se o aparato conceitual completo sobre o qual se fundamenta o comportamento de cada partícula em um sistema PSO. Na sequência, a Seção 2.3 formalizará a equação de atualização cinemática que define a movimentação iterativa das partículas, além de discutir os mecanismos auxiliares de estabilização que asseguram a viabilidade numérica do algoritmo em problemas complexos e de alta dimensionalidade.

2.3 Atualização de Velocidade e Posição

A dinâmica evolutiva do PSO fundamenta-se em um processo iterativo de aprendizado vetorial, no qual cada partícula atualiza sua trajetória combinando memórias individuais e informações sociais. Originalmente proposto por Kennedy & Eberhart

(1995) [2], esse esquema não se reduz a um mero deslocamento no espaço de busca, mas sim a um mecanismo de ajuste coordenado: a partícula i altera sua velocidade na iteração $t+1$ de acordo com

$$\mathbf{v}_i(t+1) = w\,\mathbf{v}_i(t) + c_1\,r_1\,(\mathbf{pbest}_i(t) - \mathbf{x}_i(t)) + c_2\,r_2\,(\mathbf{gbest}(t) - \mathbf{x}_i(t)), \tag{2.3}$$

em que w representa o fator de inércia que regula a influência do impulso anterior e, assim, modula a transição gradativa entre exploração de novas regiões e intensificação local (Shi e Eberhart (1998) [3]). Os coeficientes c_1 e c_2 equilibram, respectivamente, a nostalgia pela melhor posição já visitada e a atração em direção ao melhor ponto global do enxame, enquanto os termos r_1 e r_2 são variáveis aleatórias independentes com distribuição uniforme no intervalo $[0, 1]$. A nova posição da partícula é então determinada pela adição vetorial da velocidade recém-atualizada à sua posição:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \tag{2.4}$$

Essa dinâmica iterativa —— baseada na interação entre a experiência individual (via $\mathbf{pbest}_i$) e inteligência coletiva (via $\mathbf{gbest}$) — confere ao PSO uma capacidade adaptativa notável. A presença dos fatores aleatórios r_1 e r_2 garante que mesmo partículas posicionadas em regiões similares do espaço possam explorar de forma diferenciada, o que reforça a robustez do processo de busca. O parâmetro de inércia w, introduzido por Shi & Eberhart (1998) [3], regula o equilíbrio entre exploração e intensificação. Valores altos de w favorecem deslocamentos amplos e, promovem a exploração do espaço de busca; enquanto valores menores tendem a restringir a movimentação das partículas e, intensificar a busca local. O esquema de aplicação iterativa de movimentos nas particulas pode ser ilustrado pela analogia apresentada na Figura 1.

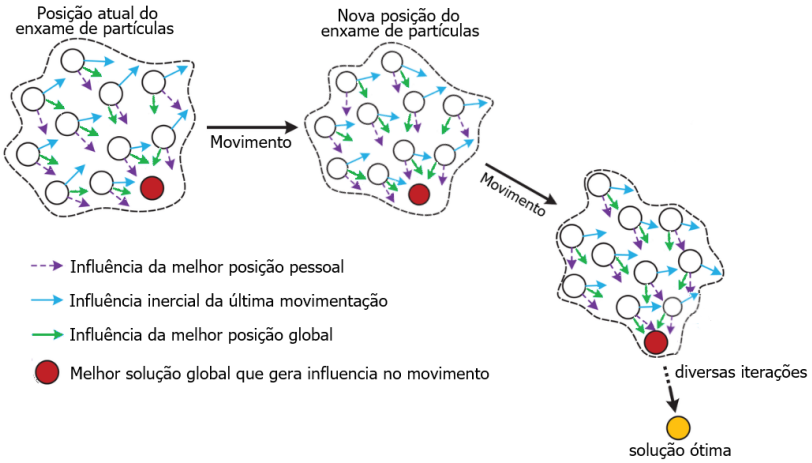

Figura 1 – Analogia para esquema de movimentação de partículas no algoritmo PSO.

Estratégias adaptativas, como o decaimento linear de w ao longo das iterações, são comumente utilizadas para melhorar a eficiência do processo de otimização. Para mitigar oscilações excessivas ou evitar que as partículas escapem dos limites definidos do espaço de busca, empregam-se técnicas de controle. Uma das abordagens mais utilizadas é a limitação de velocidade (*velocity clamping*), proposta por Eberhart & Shi (2000) [10], a qual impõe um valor máximo ($v_{\max}$) a cada componente da velocidade:

$$|v_{id}| \leqslant v_{\max}, \quad \forall d = 1, \ldots, D. \tag{2.5}$$

esse limitador evita saltos abruptos que comprometam a estabilidade numérica e coerência das trajetórias. Outra abordagem consolidada é a introdução do fator de constrição (χ), proposto por Clerc & Kennedy (2002) [11], que reformula a equação de velocidade para assegurar convergência estável. A nova equação é dada por:

$$\mathbf{v}_i(t+1) = \chi \left[\mathbf{v}_i(t) + \phi_1 \, r_1 \, (\mathbf{pbest}_i - \mathbf{x}_i) + \phi_2 \, r_2 \, (\mathbf{gbest} - \mathbf{x}_i) \right], \tag{2.6}$$

em que $\phi = \phi_1 + \phi_2 > 4$ é uma condição necessária para estabilidade teórica. O fator χ é calculado por:

$$\chi = \frac{2}{\left| 2 - \phi - \sqrt{\phi^2 - 4\phi} \right|}. \tag{2.7}$$

Essa formulação aprimora o controle sobre as oscilações das partículas e melhora o comportamento assintótico do algoritmo, o que a torna particularmente eficaz em problemas de alta complexidade e superfícies irregulares de busca.

Portanto, a atualização de velocidade e posição no PSO vai além de uma simples regra de movimento: trata-se de um mecanismo sofisticado de aprendizado distribuído, no qual partículas interagem entre si e com o ambiente por meio de componentes determinísticos e estocásticos. Assim, a combinação desses mecanismos — ajuste adaptativo pela equação original, contenção pela limitação de velocidade e amortecimento pelo fator de constrição — confere ao PSO uma capacidade robusta para navegar de forma eficiente em problemas complexos, ao manter o equilíbrio delicado entre a exploração global e a intensificação local.

2.4 Principais Variantes

Desde sua formulação original como um modelo inspirado em comportamentos sociais auto-organizados (Souza *et al*. (2023) [12]), o algoritmo PSO tem sido amplamente estendido para contemplar contextos computacionalmente diversos e aplicações de

elevada complexidade. Tais variantes buscam reforçar a capacidade do PSO em equilibrar eficientemente dois mecanismos essenciais à busca metaheurística: a exploração global de regiões promissoras no espaço de soluções e a intensificação local sobre áreas já identificadas como potencialmente ótimas. A literatura científica tem promovido sucessivas modificações que aumentam a robustez do algoritmo em domínios multimodais, discretos, de alta dimensionalidade ou multiobjetivo —– contexto este que será explorado em profundidade nos capítulos subsequentes.

Entre as primeiras inovações relevantes, destaca-se a introdução do peso de inércia variável. Nesta versão, proposta por Shi & Eberhart (1998) [3], o fator de inércia é reduzido progressivamente ao longo das iterações, este precesso promove uma transição suave do comportamento exploratório — desejável no início da busca — para a intensificação — crítica nas fases finais. Essa estratégia mostrou-se eficiente em *benchmarks* multimodais e aplicações de engenharia de controle, como a sintonia de parâmetros de controladores PID.

Complementarmente, Clerc & Kennedy (2002) [11] como referido, introduziram o fator de constrição (χ) (2.7), que oferece uma estrutura teórica mais rigorosa para garantir a estabilidade do sistema dinâmico das partículas. A equação modificada (2.6) regula o crescimento da velocidade das partículas de modo a assegurar convergência assintótica controlada, tende a evitar comportamentos caóticos mesmo em espaços de busca de alta complexidade. Essa abordagem é particularmente eficaz em problemas de engenharia robusta, nos quais a viabilidade numérica é condição *sine qua non* — para esses problemas é absolutamente necessário que o algoritmo funcione bem do ponto de vista numérico, ou seja, que ele gere resultados estáveis, corretos e computacionalmente viáveis.

Outro eixo fundamental de diversificação do PSO diz respeito às estruturas topológicas de vizinhança, que definem como as partículas interagem para atualizar suas soluções. No modelo clássico (*gbest*), todas as partículas se orientam pela melhor solução global encontrada até o momento, o que favorece convergência rápida, mas aumenta o risco de aprisionamento em ótimos locais. Para contornar essa limitação, van den Bergh & Engelbrecht (2004) [13] propuseram a topologia local (*lbest*), em que cada partícula se comunica apenas com um subconjunto do enxame, geralmente estruturado em anéis ou grades, o qual promove diversidade e retarda a convergência prematura (Souza *et al*. (2023) [12]). Posteriormente, Poli *et al*. (2007) [4] desenvolveram esquemas de vizinhança dinâmicos, nos quais a estrutura de interação é adaptada durante a execução do algoritmo. Tais estratégias têm se mostrado vantajosas em aplicações que exigem adaptação contínua, como redes inteligentes, sistemas bioinspirados e projetos de antenas com múltiplos objetivos.

Além disso, adaptações foram feitas para estender o PSO a problemas com va-

riáveis discretas ou binárias. O modelo *Binary* PSO, proposto por Kennedy & Eberhart (1997) [14], converte o vetor de velocidade em uma função de probabilidade via *sigmoide*, de modo que cada dimensão da partícula representa uma decisão binária (ligado/desligado). Essa variante abriu caminho para a aplicação do PSO em seleção de atributos, roteamento em redes, alocação de recursos em telecomunicações e problemas combinatórios em geral.

Outro ramo promissor de desenvolvimento envolve a paralelização e hibridização do PSO. Versões paralelas dividem o enxame em subpopulações que operam concorrentemente, como descrito em Blackwell & Bentley (2002) [15], de maneira que permitem aceleração significativa da execução em problemas de grande escala como otimização fluidodinâmica. Em paralelo, algoritmos híbridos combinam o PSO com outros paradigmas, como algoritmos genéticos (Angeline (1998) [16]), busca local determinstica (Mezura-Montes e Coello Coello (2005) [17]), e algoritmos baseados em aprendizado. Tais hibridizações fortalecem o algoritmo ao reunir diferentes mecanismos de exploração e refinamento, especialmente úteis em aplicações estruturais, alocação de recursos industriais e engenharia de *software*.

Todas essas variantes convergem naturalmente para a generalização multiobjetivo do PSO, o*MultiObjective PSO* (MOPSO). Desenvolvido por Coello Coello *et al*. (2002) [18] e posteriormente refinado por Mostaghim & Teich (2003) [19], o MOPSO introduz mecanismos de arquivamento baseados em dominância de Pareto e estratégias de seleção diversificadas para líderes, como *crowding-distance* e ε*-dominance*. Diferentemente do PSO convencional, no qual um único líder orienta o movimento do enxame, no MOPSO cada partícula pode ser guiada por diferentes soluções não dominadas, ele favorece a manutenção de uma fronteira de soluções equilibradas entre objetivos conflitantes.

2.5 Fundamentos da Otimização Multiobjetivo

Em inúmeros problemas de Engenharia e Ciência Aplicada, como o dimensionamento de redes de filas finitas em centros de atendimento, a configuração de *buffers* em redes de telecomunicações e o balanceamento de linhas de produção industriais, torna-se imprescindível considerar simultaneamente múltiplos critérios de desempenho que, em geral, são conflitantes (Souza *et al*. (2023) [12]). A título ilustrativo, considere um sistema de filas com limitação de capacidade de área (*buffer*), a ampliação da capacidade reduz a probabilidade de bloqueio e aumenta a taxa de atendimento, mas eleva significativamente os custos operacionais e o consumo de recursos computacionais.

De modo semelhante, a diminuição das taxas de serviço pode gerar ganhos de eficiência energética ou de disponibilidade de servidores, ao custo de aumento do

tempo de espera dos clientes ou pacotes. Nessas situações, a busca por uma solução única é conceitualmente inadequada, pois não há um ponto no espaço de soluções que satisfaça todos os objetivos de forma ótima e simultânea. Insistir em uma única "solução ótima" é inadequado, pois todo ganho em um objetivo implica necessariamente em sacrifício em outro (Souza *et al*. (2023) [12]).

Para lidar com esses *trade-offs* inerentes, adota-se o paradigma da Otimização Multiobjetivo (MOO), cuja essência consiste em buscar não um ponto único, mas um conjunto de soluções eficientes, também chamadas não-dominadas, que representam diferentes compromissos (*trade-offs*) entre objetivos em conflito (Coello Coello *et al*. (2002) [18]). Essas soluções formam a chamada fronteira de Pareto, definida como o conjunto de alternativas para as quais não é possível melhorar um dos critérios sem deteriorar ao menos um dos demais. A fronteira oferece ao tomador de decisão uma visão abrangente e estruturada das possíveis escolhas, isto permite selecionar aquela que melhor atende às suas restrições contextuais ou preferências de desempenho (Deb *et al*. (2002) [20]). A formulação matemática usual de um problema de otimização multiobjetivo é dada por:

$$\min_{\mathbf{x}\in\Omega} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})]\,, \tag{2.8}$$

em que $\mathbf{x} \in \Omega \subseteq \mathbb{R}^D$ representa o vetor de variáveis de decisão em um espaço viável Ω, com D dimensões. Cada função $f_i : \Omega \to \mathbb{R}$, para $i = 1, 2, \ldots, m$, corresponde a um objetivo a ser minimizado (ou, de forma equivalente, maximizado por inversão de sinal), em que m é definido como o número total de critérios em conflito. Portanto, nessa formulação vetorial, cada vetor $\mathbf{x}$ é avaliado por M funções objetivo, não existe hierarquia natural entre elas. Como Deb *et al*. (2002) [20] enfatiza, quando os objetivos são conflitantes, a otimização leva naturalmente a um conjunto de soluções de compromisso, pois melhorar um objetivo geralmente exige sacrificar outro.

No contexto da dominância de Pareto, uma solução $\mathbf{x}^{(1)} \in \Omega$ é dita dominante sobre outra $\mathbf{x}^{(2)} \in \Omega$, denotada por $\mathbf{x}^{(1)} \prec \mathbf{x}^{(2)}$, se satisfaz as seguintes condições:

$$\begin{aligned} &\forall\, i \in \{1, \ldots, m\}, f_i(\mathbf{x}^{(1)}) \leqslant f_i(\mathbf{x}^{(2)}), \text{ e existe pelo} \\ &\text{menos um valor } j \in \{1, \ldots, m\}; f_j(\mathbf{x}^{(1)}) < f_j(\mathbf{x}^{(2)}). \end{aligned} \tag{2.9}$$

Dessa forma, uma solução $\mathbf{x}^* \in \Omega$ é não-dominada se não existe outro vetor $\mathbf{x} \in \Omega$ tal que $\mathbf{x} \prec \mathbf{x}^*$. O conjunto de todas essas soluções não-dominadas define, então, a fronteira de Pareto, que representa as alternativas ótimas em sentido fraco; ou seja, soluções que não são superadas simultaneamente em todos os objetivos por nenhuma outra solução factível (Deb *et al*. (2002) [20]).

Dada uma solução $\mathbf{x}^* \in \Omega$ 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 2 ilustra esse efeito em um problema hipotético de minimização de dois objetivos conflitantes. A linha azul representa o cone delimitado pela solução $\mathbf{x}^*$ e a região sombreada indica o conjunto de soluções menos vantajosas sob a ótica da dominância de Pareto, ou seja, as soluções dominadas pela solução $\mathbf{x}^*$.

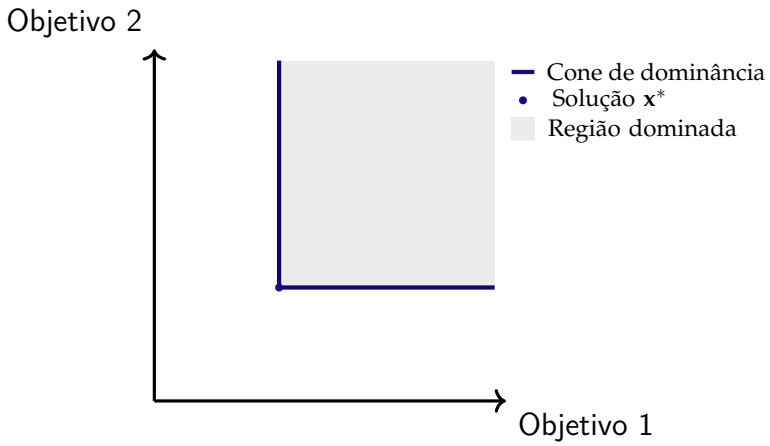

Figura 2 – Ilustração para cone de dominância em um problema de otimização multiobjetivo bidimensional.

Cada solução candidata 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 soluções sob investigação. Em outras palavras, restrito ao espaço de soluções investigado, o conjunto das soluções não-domindas ou fronteira de pareto é o conjunto de soluções ótimas restrito ao espaço de soluções investigado. Quanto maior a abrangência do conjunto de soluções investigado maior a capacidade de que sua fronteira de pareto seja uma aproximação eficiente para a fronteira de pareto considerando todas as possíveis soluções factíveis.

Uma ilustração bastante didática para a Fronteira de Pareto é apresentada na Figura 3, é possível visualizar um espaço objetivo com dois critérios conflitantes. Agora, a linha azul representa a fronteira de Pareto, formada por soluções eficientes (não-dominadas), nas quais a melhoria de um objetivo implica na piora de outro. Pontos cinzas representam alternativas dominadas, e a região sombreada indica o conjunto de soluções menos vantajosas sob a ótica da dominância de Pareto. A construção de uma aproximação com grau de eficiencia elevado dessa fronteira exige, simultaneamente, a convergência das soluções em direção ao verdadeiro Pareto *front* e a diversidade ao longo de sua extensão, de modo a cobrir regiões distintas de interesse. Estratégias clássicas de agregação de objetivos em um único índice — por ponderação ou penalização — frequentemente apresentam fragilidades diante de frentes não-convexas ou

descontínuas, pois podem deixar de explorar regiões ótimas dispersas no espaço de soluções (Coello Coello *et al*. (2002) [18]).

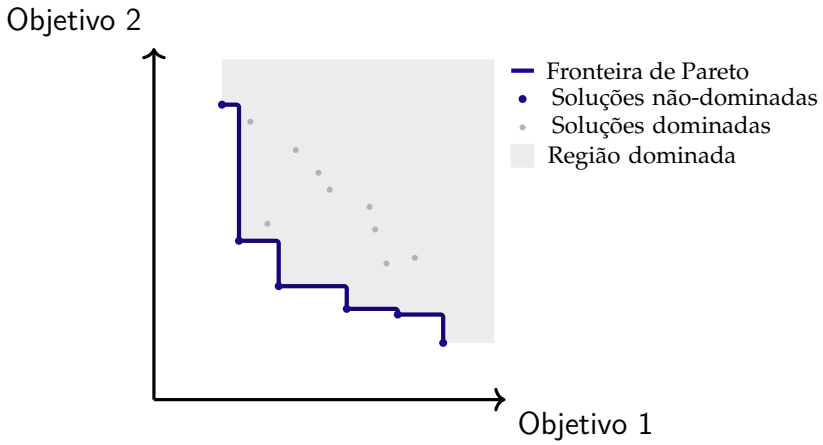

Figura 3 – Fronteira de Pareto em problema de otimização multiobjetivo bidimensional.

Ao aproximar a fronteira de Pareto, buscam-se dois objetivos principais: convergência e diversidade. De um lado, é preciso aproximar as soluções o mais próximo possível da fronteira real (convergência); de outro, espera-se que as soluções estejam bem distribuídas ao longo de toda a fronteira (diversidade). Como Deb (2011) [21] aponta, o algoritmo ideal deve encontrar um conjunto de pontos na fronteira de Pareto e garantir que eles cubram toda a gama de *trade-offs*. Em termos práticos, a ausência de diversidade leva a concentração de soluções em apenas uma região da fronteira, enquanto a falta de convergência produz soluções de baixa qualidade. Nesse sentido, algoritmos como o NSGA-II incorporam mecanismos explícitos de diversidade para assegurar bom espalhamento das soluções (Deb (2002) [22]).

Métodos clássicos de agregação linear (como a soma ponderada dos objetivos) têm sido muito usados para converter o problema multiobjetivo em vários problemas monoobjetivos. Contudo, esses métodos apresentam sérias limitações. É necessário escolher a priori os pesos dos objetivos, e cada execução normalmente produz apenas um ponto de Pareto por vez. Além disso, a abordagem por agregação linear não consegue gerar soluções em regiões não convexas da fronteira de Pareto, de modo que algumas soluções eficientes podem não ser encontradas. Mesmo nas regiões convexas, variações nos pesos podem levar a distribuições desiguais das soluções, este fato dificulta uma cobertura uniforme da fronteira (Fonseca e Fleming (1995) [24]).

É nesse contexto que emergem os algoritmos evolutivos e, em especial, Coello Coello *et al*. (2002) [18] propuseram o MOPSO. Inspirado no PSO original de Kennedy & Eberhart (1995) [2], o MOPSO incorpora mecanismos de arquivamento de soluções não-dominadas em um repositório externo, seleciona líderes ao longo desse arquivo de modo a preservar diversidade e guia cada partícula não mais por um único melhor

global, mas por líderes distintos pertencentes à fronteira aproximada [19, 25].

A arquitetura específica, e também a estrutura multiobjetivo se tornam particularmente poderosas em contextos como o de redes de filas finitas, em que métricas como *throughput*, taxa de bloqueio e custo de infraestrutura competem entre si. Essa aplicação específica será aprofundada por meio de uma conexão entre os fundamentos do MOPSO à sua implementação em ambientes estocásticos e com restrições operacionais.

2.6 Extensões do PSO para Otimização Multiobjetivo (MOPSO)

O PSO, em sua formulação original, foi concebido para resolver problemas de otimização com um único objetivo. No entanto, em muitos domínios aplicados, como o projeto e operação de redes de filas com capacidade finita, os sistemas envolvem objetivos múltiplos e mutuamente conflitantes. Essa característica exige abordagens capazes de explorar soluções que reflitam compromissos ótimos entre diferentes métricas de desempenho, o que caracteriza a natureza multiobjetivo do problema.

A extensão do PSO para problemas multiobjetivo resultou no chamado *MultiObjective Particle Swarm Optimization* (MOPSO), proposto inicialmente por Coello Coello *et al.* (2002) [18] como uma adaptação estrutural do algoritmo clássico. A ideia central é substituir a noção de ótimo global (*gbest*) por um conjunto de soluções não-dominadas, organizadas conforme a dominância de Pareto. Assim, o papel de liderança e aprendizado coletivo é redistribuído entre múltiplas regiões da fronteira de Pareto para preservar a diversidade populacional e viabilizar a geração de múltiplas soluções de compromisso em uma única execução.

No contexto de redes de filas finitas, esse arcabouço é particularmente vantajoso. Ele permite a formulação de soluções que conciliam métricas conflitantes, como *throughput*, probabilidade de bloqueio, custo operacional, entre outras; além de respeitar as restrições físicas e orçamentárias do sistema. O resultado é um conjunto de configurações eficientes que ilustram os *trade-offs* possíveis entre eficiência operacional, confiabilidade do serviço e investimento de recursos, conforme ilustrado em estudos como Cruz *et al.* (2008) [26] e Souza (2020) [27]. A subsequente Seção 2.7 detalha a arquitetura geral do MOPSO, explica como esses conceitos se traduzem em estruturas de dados e procedimentos algorítmicos que viabilizam o processamento paralelo e a busca adaptativa ao longo da fronteira de Pareto.

2.7 Arquitetura Geral do MOPSO

O MOPSO implementa três mecanismos centrais para lidar com a complexidade de problemas com múltiplos critérios conflitantes. Primeiro, um arquivo elitista ($A(t)$)

funciona como memória externa, a cada iteração, esse arquivo é atualizado: novas soluções não dominadas ingressam nele, enquanto aquelas que passam a ser dominadas são removidas (Fieldsend *et al*. (2003) [28]). Essa estrutura, formalmente descrita por

$$A(t) = \left\{ \mathbf{x}_i \in \Omega \;\middle|\; \nexists\, \mathbf{x}_j \in \Omega \text{ tal que } \mathbf{x}_j \prec \mathbf{x}_i \right\}, \tag{2.10}$$

em que $\prec$ representa a relação de dominância de Pareto, assegura que o repositório sempre represente o melhor conhecimento coletivo disponível até o instante t. Esse conjunto pode ser interpretado como uma aproximação amostral de uma medida de probabilidade concentrada na fronteira de Pareto. No entanto, à medida que o número de objetivos cresce, o tamanho de $A(t)$ tende a aumentar exponencialmente. Para preservar a diversidade e controlar a cardinalidade do arquivo, são utilizadas políticas como a *crowding distance* que retira soluções em regiões densamente povoadas (Deb (2002) [20]), o ε-dominância que agrupa soluções próximas (Benítez *et al*. (2005) [29]), e a discretização por grade regulares do espaço objetivo (Mostaghim e Teich (2003) [19]).

O segundo pilar do MOPSO é a redefinição da atração social. Cada partícula i deixa de usar um **gbest** único e passa a ser atraída por um *líder individual* $\ell_i(t) \in A(t)$, escolhido de forma a incentivar a exploração de regiões distintas da fronteira de Pareto. Essa seleção pode seguir os seguintes critérios:

$$\ell_i(t) = \arg\min_{\mathbf{x} \in A(t)} \text{crowding}(\mathbf{x}) \quad \text{ou} \quad \ell_i(t) \sim \text{grid-sampling}\big(A(t)\big). \tag{2.11}$$

Essa descentralização evita o colapso do enxame em um único nicho e favorece a diversidade de soluções ao longo do *front* de Pareto. A equação de atualização da velocidade mantém a forma vetorial do PSO original, substitui **gbest** por $\ell_i(t)$ e interpreta os termos estocásticos $r_1, r_2 \sim \mathcal{U}(0,1)$ como ruído branco aditivo, que induz uma dinâmica de difusão discreta — análoga a um processo de Langevin [30], o que previne o aprisionamento em bacias locais.

Por fim, para evitar estagnação do enxame, o MOPSO emprega operadores de diversidade que introduzem perturbações nas posições das partículas. São utilizadas mutações uniformes, gaussianas ou adaptativas, e reinicializações parciais de partículas estagnadas, conforme sugerido por Mezura-Montes & Coello (2005) [17]. Esses mecanismos asseguram cobertura ampla da fronteira de Pareto, especialmente em regiões não convexas ou pouco exploradas.

A Figura 4 esquematiza os métodos de seleção de líderes, a Figura 5 apresenta o fluxograma completo do ciclo de atualização de velocidade e posição, destaca as fases de consulta ao arquivo elitista, escolha de $\ell_i(t)$ e aplicação de operadores de diversidade.

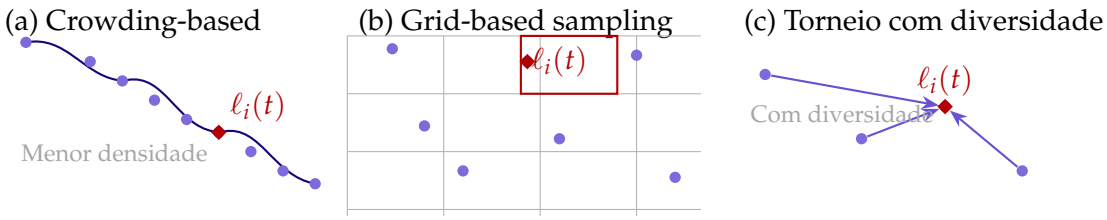

Figura 4 – Estratégias de seleção do líder $\ell_i(t)$ no MOPSO.

Na Figura 4(a) o método é baseado em *crowding distance*, o líder é escolhido em regiões de menor densidade da fronteira de Pareto para promover a diversidade; na Figura 4(b) é utilizada a amostragem em grade regular, uma célula na grade do espaço objetivo é selecionada aleatoriamente, e um líder é escolhido dentro dela; por fim na Figura 4(c) é ilustrado o torneio com diversidade, um líder é selecionado de um grupo de candidatos com base em aptidão e critérios de diversidade.

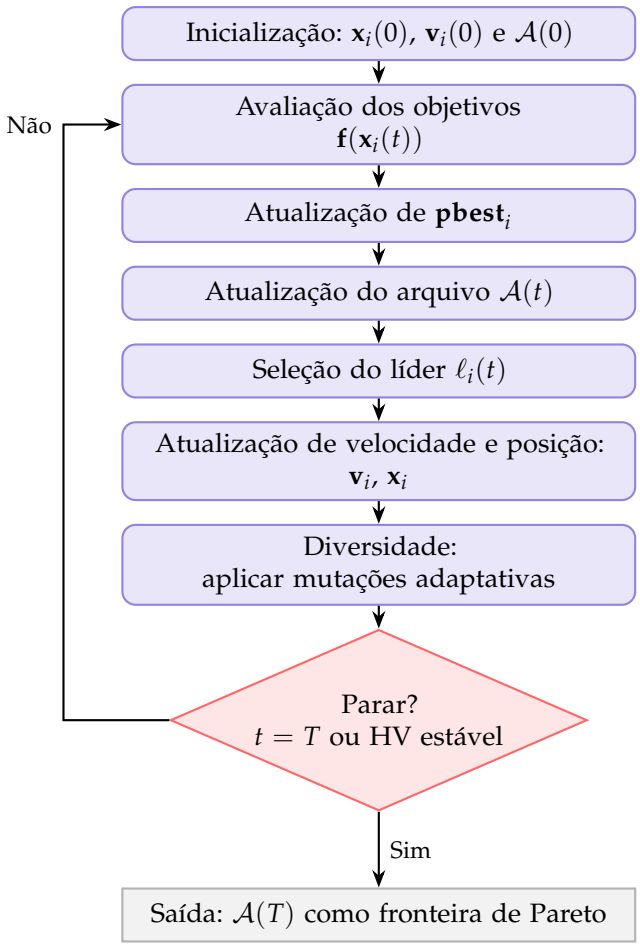

Figura 5 – Fluxograma simplificado com a representaçãodo do ciclo essencial do algoritmo MOPSO.

Vale observar que, embora o MOPSO amplie o poder de representação do PSO clássico, ele também impõe maior custo computacional, especialmente na manutenção do arquivo elitista e no cálculo da dominância de Pareto. Estratégias como arquivamento

e seleção por amostragem são, portanto, indispensáveis para preservar a escalabilidade do algoritmo em contextos de alta dimensionalidade ou com fronteiras densamente povoadas.

A presença de ruído estocástico nas equações de atualização permite associar o comportamento das partículas a uma dinâmica difusiva de tipo Langevin discreto. Sob essa perspectiva, o MOPSO pode ser interpretado como uma heurística baseada em simulações estocásticas de trajetórias em um espaço objetivo com múltiplos atratores, o que lhe confere maior capacidade de escapar de regiões subótimas.

Em síntese, a arquitetura do MOPSO integra aprendizado individual (via *pbest*), aprendizado social adaptativo (via líderes ℓ_i) e memória histórica ($A(t)$), compondo um sistema estocástico que balanceia exploração e intensificação ao longo da fronteira de Pareto.

2.7.1 Fluxo Algorítmico do MOPSO

A estrutura operacional do MOPSO mantém o cerne adaptativo do PSO clássico — isto é, a atualização iterativa de velocidades e posições das partículas com base em componentes cognitivos e sociais —, mas adiciona mecanismos imprescindíveis para lidar com múltiplos objetivos conflitantes. Nesse contexto, cada iteração do MOPSO consiste em um ciclo bem definido de avaliação, arquivamento, seleção de líderes, atualização vetorial e aplicação de operadores de diversidade, de modo que a população de partículas percorra de forma adaptativa e distribuída o espaço dos objetivos em busca de uma aproximação satisfatória da fronteira de Pareto (Deb (2002) [20]).

O ciclo de vida do MOPSO inicia-se com a fase de "Inicialização Populacional", na qual cada partícula i é posicionada aleatoriamente em uma solução $\mathbf{x}_i(0)$ dentro do espaço viável $\Omega \subset \mathbb{R}^D$, e recebe uma velocidade $\mathbf{v}_i(0)$ sorteada uniformemente no mesmo domínio. Imediatamente, avalia-se o vetor de funções-objetivo $\mathbf{f}(\mathbf{x}_i(0)) = (f_1(\mathbf{x}_i(0)), \ldots, f_m(\mathbf{x}_i(0)))$, definindo-se a memória individual — melhor posição pessoal até o momento —, denotada por $\mathbf{pbest}_i \leftarrow \mathbf{x}_i(0)$. Concomitantemente, constrói-se o primeiro arquivo elitista $A(0)$, que armazena todas as soluções não dominadas encontradas na população inicial, conforme a equação vista (2.10) [18, 20]. Esse arquivo elitista representa a memória coletiva das melhores soluções não dominadas até o instante t e, funciona como repositório dinâmico ao longo da execução.

A cada iteração $t \in \{1, 2, \ldots, T\}$, desenvolve-se um ciclo composto por cinco etapas interligadas. Inicialmente, para cada partícula i avalia $\mathbf{f}(\mathbf{x}_i(t))$. Quando $\mathbf{x}_i(t)$ domina a posição armazenada em $\mathbf{pbest}_i(t-1)$, realiza-se a atualização $\mathbf{pbest}_i(t) \leftarrow \mathbf{x}_i(t)$; caso contrário, $\mathbf{pbest}_i(t)$ permanece inalterada. Em seguida, procede-se à atualização do arquivo elitista $A(t)$: se $\mathbf{x}_i(t)$ não for dominada por nenhuma solução já existente em

$A(t-1)$, insere-se $\mathbf{x}_i(t)$ em $A(t)$, remove-se simultaneamente quaisquer $\mathbf{z} \in A(t)$ que sejam agora dominadas por $\mathbf{x}_i(t)$. Uma vez que o arquivo pode crescer rapidamente em problemas com múltiplos objetivos, aplica-se, sempre que $|A(t)|$ ultrapassa $N_{\max}$, políticas de contenção de cardinalidade, tais como *crowding distance* (Deb *et al.*, 2002 [20]), *ε-dominance* (Benítez *et al.*, 2005 [29]) ou *grid sampling* (Mostaghim & Teich, 2003 [19]), para garantir $|A(t)| \leqslant N_{\max}$. Essas políticas visam preservar a diversidade no conjunto de soluções, para remover prioridades de regiões densamente agrupadas do espaço objetivo ou agrupar soluções próximas em células discretizadas.

Após atualização do arquivo $A(t)$, procede-se à seleção de um *líder individual* $\boldsymbol{\ell}_i(t) \in A(t)$ para cada partícula i. Diferentemente do PSO clássico, em que todas as partículas são atraídas pelo mesmo **gbest**, o MOPSO atribui, a cada partícula, um líder distinto, escolhido com base em critérios de diversidade. Um método comumente adotado consiste em selecionar $\boldsymbol{\ell}_i(t)$ como a solução em $A(t)$ de menor *crowding distance*, garante que as partículas sejam direcionadas a regiões de fronteira menos saturadas. Alternativamente, pode-se empregar amostragem uniforme em uma grade regular do espaço objetivo, de modo a preservar cobertura ampla e uniforme da fronteira de Pareto [18, 19]. Formalmente, a seleção de $\boldsymbol{\ell}_i(t)$ pode ser descrita pela Equação (2.11), essa estratégia assegura que partículas distintas sejam guiadas por diferentes regiões da fronteira de pareto e, estimula a exploração paralela de múltiplos *trade-offs* entre os objetivos. Com o líder $\boldsymbol{\ell}_i(t)$ selecionado, procede-se à atualização vetorial de cada partícula i. A equação de velocidade, adaptada para o contexto multiobjetivo, é dada por:

$$\mathbf{v}_i(t+1) = \omega\,\mathbf{v}_i(t) + c_1 r_1\big(\mathbf{pbest}_i(t) - \mathbf{x}_i(t)\big) + c_2 r_2\big(\boldsymbol{\ell}_i(t) - \mathbf{x}_i(t)\big), \tag{2.12}$$

seguida pela atualização da posição,

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \tag{2.13}$$

Nesse esquema, ω é o fator de inércia, c_1 e c_2 são os coeficientes de aceleração cognitiva e social, respectivamente, e $r_1, r_2 \sim \mathcal{U}(0,1)$ são variáveis aleatórias uniformes que introduzem variação estocástica e, evitam que partículas idênticas sigam trajetórias coincidentes. Na eventualidade de buscar maior estabilidade numérica e convergência amortecida, pode-se utilizar o fator de constrição χ, conforme sugerido por Clerc & Kennedy (2002) [11]. Nesse caso, a equação de velocidade (2.12) assume a forma:

$$\mathbf{v}_i(t+1) = \chi\Big[\mathbf{v}_i(t) + \varphi_1 r_1\big(\mathbf{pbest}_i(t) - \mathbf{x}_i(t)\big) + \varphi_2 r_2\big(\boldsymbol{\ell}_i(t) - \mathbf{x}_i(t)\big)\Big], \tag{2.14}$$

em que $\varphi = \varphi_1 + \varphi_2 > 4$ e $\chi = \dfrac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}$, garante estabilização assintótica do processo dinâmico (Clerc e Kennedy (2002) [11]).

Por fim, para mitigar a estagnação do enxame em áreas limitadas do espaço objetivo e preservar a capacidade exploratória, o MOPSO incorpora operadores de diversidade adicionais. Quando uma partícula não apresenta melhora em sua memória $\mathbf{pbest}_i$ ao longo de um número pré-definido de gerações G, aplica-se um operador de mutação à sua posição $\mathbf{x}_i(t)$. As mutações podem ser uniformes — deslocamentos aleatórios dentro dos limites admissíveis de cada dimensão; gaussianas — perturbações normalmente distribuídas centradas em $\mathbf{x}_i(t)$; ou adaptativas — cujas amplitudes diminuem à medida que o algoritmo converge (Mezura-Montes e Coello Coello (2005) [17]). Além disso, pode-se reinicializar parcialmente partículas fortemente estagnadas, isso as induz na direção de regiões pouco exploradas por heurísticas de dispersão ou sorteio randômico. Esses mecanismos asseguram injeção de variabilidade no enxame e evitam que o algoritmo aprisione-se em regiões subótimas ou que a fronteira de Pareto resulte excessivamente concentrada.

Nesses procedimentos, o critério de parada mais comum é o número máximo de iterações T; contudo, métricas de convergência multiobjetivo, como *hypervolume* (HV) ou *Inverted Generational Distance* (IGD), também podem ser empregadas para avaliar dinamicamente a necessidade de encerramento, especialmente quando a fronteira de Pareto já se encontra suficientemente estável (Zitzler (2003) [31]). Ao final da execução, o arquivo $A(T)$ contém o conjunto de soluções não dominadas que define a aproximação final da fronteira de Pareto. Esses pontos representam as configurações de compromisso entre os objetivos conflitantes e fornecem ao tomador de decisão opções variadas, cada qual com características distintas em termos de *trade-offs*.

O ciclo completo de cada iteração do MOPSO pode ser resumido pelo fluxograma esquemático da Figura 5, que ilustra as etapas de avaliação, atualização de memória individual $\mathbf{pbest}_i$, manutenção do arquivo elitista $A(t)$, seleção de líder $\boldsymbol{\ell}_i(t)$, atualização vetorial e aplicação de operadores de diversidade (mutação adaptativa e reinicialização). A cada passo, o arquivo elitista pode crescer ou ser reduzido via políticas de contenção, e os líderes de cada partícula são escolhidos de maneira a maximizar a cobertura da fronteira. Ao final da última iteração $t = T$, o conjunto $A(T)$ representa a melhor aproximação da fronteira de Pareto obtida pelo enxame, isto fornece ao tomador de decisão um amplo leque de soluções diferenciadas em termos de *trade-offs* conflitantes.

Em paralelo ao fluxograma, o pseudocódigo apresentado a seguir (**Algoritmo** 1) descreve de forma sintética as operações centrais do MOPSO com arquivamento e operadores adaptativos de diversidade, conforme as propostas de Coello Coello *et*

al.,(2002) [18], Mostaghim & Teich,(2003) [19] e Mezura-Montes & Coello,(2005) [17].

Algoritmo 1 MOPSO — *Multi-Objective Particle Swarm Optimization*

▷ /* ***Input:*** *População inicial* $\mathcal{P}_0 = \{\mathbf{x}_i(0)\}_{i=1}^{N}$, *velocidades* $\{\mathbf{v}_i(0)\}$, n^o *de iterações* T */
▷ /* ***Input:*** *parâmetros* ω, c_1, c_2, *tamanho máximo do arquivo* $N_{\max}$ */
▷ /* ***Output:*** *Arquivo elitista final* $\mathcal{A}(T)$ *com soluções não dominadas* */

para cada partícula $\mathbf{x}_i(0) \in \mathcal{P}_0$ **faça**
    Avaliar $f(\mathbf{x}_i(0))$
    $\mathbf{pbest}_i \leftarrow \mathbf{x}_i(0)$
fim para
$\mathcal{A}(0) \leftarrow$ soluções não dominadas de $\mathcal{P}_0$
/* *Loop Principal* */
para $t = 1$ até T **faça**
    para cada partícula $i = 1, \ldots, N$ **faça**
        Avaliar $f(\mathbf{x}_i(t-1))$
        Selecionar líder $\boldsymbol{\ell}_i(t) \in \mathcal{A}(t-1)$ com base em diversidade
        /* *Atualização de velocidade* */
        $\mathbf{v}_i(t) \leftarrow \omega \mathbf{v}_i(t-1) + c_1 r_1(\mathbf{pbest}_i - \mathbf{x}_i(t-1)) + c_2 r_2(\boldsymbol{\ell}_i(t) - \mathbf{x}_i(t-1))$
        /* *Atualização de posição* */
        $\mathbf{x}_i(t) \leftarrow \mathbf{x}_i(t-1) + \mathbf{v}_i(t)$
        /* *Atualização da memória pessoal* */
        se $\mathbf{x}_i(t) \prec \mathbf{pbest}_i$ **então**
            $\mathbf{pbest}_i \leftarrow \mathbf{x}_i(t)$
        fim se
    fim para
    /* *Atualização do arquivo elitista* */
    $\mathcal{A}(t) \leftarrow$ não-dominadas de $\mathcal{A}(t-1) \cup \{\mathbf{pbest}_i\}_{i=1}^{N}$
    se $|\mathcal{A}(t)| > N_{\max}$ **então**
        Aplicar política de contenção (e.g., *crowding distance*, *grid sampling*)
    fim se
    /* *Operadores de diversidade* */
    para partícula i estagnada por G gerações **faça**
        Aplicar operador de diversidade (mutação adaptativa ou reinicialização)
    fim para
fim para

retorna $\mathcal{A}(T)$

Em suma, o fluxo algorítmico do MOPSO articula a avaliação estocástica de partículas, o refinamento de memórias individuais e elitistas, a seleção distribuída de líderes e a injeção controlada de variabilidade, este resulta em um processo robusto, paralelo e flexível para aproximar a fronteira de Pareto em problemas com múltiplos critérios. Sequencialmente, esse arcabouço será aplicado à modelagem de redes de filas finitas, para demonstrar como cada componente do algoritmo pode ser mapeado para decisões operacionais reais — como alocação de capacidade de *buffer*, controle de taxa de chegada ou dimensionamento de servidores.

2.7.2 Parâmetros e Estratégias Recomendadas

A eficácia do algoritmo MOPSO na aproximação da fronteira de Pareto está diretamente relacionada à escolha apropriada de seus parâmetros de controle e estratégias dinâmicas. Tais parâmetros regulam o comportamento exploratório e de intensificação das partículas ao longo do processo evolutivo e impactam a qualidade das soluções obtidas, a taxa de convergência e a capacidade do algoritmo em manter a diversidade da população sem comprometer a eficiência computacional. Este tópico detalha as práticas recomendadas, justificadas tanto empiricamente quanto teoricamente, com base na literatura especializada em otimização multiobjetivo e métodos estocásticos populacionais.

Um dos primeiros elementos a ser definido na configuração do MOPSO é o tamanho do enxame, usualmente representado por N. Este parâmetro define o número total de partículas para explorar o espaço de busca e influenciar diretamente a capacidade do algoritmo em cobrir regiões diversas da fronteira de Pareto. Estudos clássicos, como o de Coello Coello *et al*. (2002) [18], sugerem que valores entre 50 e 200 partículas são apropriados, é comum a adoção de $N = 100$ como um valor de referência. A escolha de N deve equilibrar o ganho de diversidade advindo de um maior número de agentes com o aumento linear do custo computacional por geração. A adoção de valores maiores para N amplia a diversidade exploratória, mas demanda mais avaliações de função e, portanto, maior custo computacional.

Outro parâmetro de grande relevância é o fator de inércia, denotado por (ω), introduzido por Shi & Eberhart (1998) [3] como uma forma de controlar a persistência do movimento das partículas em direção às suas trajetórias anteriores. A recomendação amplamente adotada é que ω decresça linearmente ao longo das iterações, com variação típica de 0,9 nas fases iniciais — em que se privilegia a exploração global do espaço — até 0,4 nas fases finais, em que o foco passa a ser o refinamento local das soluções. Tal estratégia de decaimento permite transitar de uma busca ampla, propensa à descoberta de regiões promissoras, para uma intensificação progressiva em torno de soluções de alta qualidade.

Em complemento ao papel de (ω), os coeficientes de aceleração cognitiva e social, c_1 e c_2, controlam a intensidade com que cada partícula é atraída por sua própria melhor posição histórica (*pbest*) e por seu líder social ($\ell_i(t)$), respectivamente. Kennedy & Eberhart (1995) [2] propuseram os valores estáticos $c_1 = c_2 = 2{,}0$, o que assegura um equilíbrio entre aprendizado individual e colaboração entre partículas. No entanto, abordagens dinâmicas continuam sob contínua investigação, em que se reduz gradualmente c_1 ao mesmo tempo em que c_2 é aumentado, ou vice-versa. Essa variação adaptativa tem demonstrado ser particularmente eficaz em problemas cujas fronteiras de Pareto apresentam regiões convexas e não convexas alternadas (Mezura-Montes e

Coello Coello (2005) [17]).

No tocante ao gerenciamento do arquivo externo $A(t)$, que armazena as soluções não-dominadas obtidas até a iteração t, deve ser mantido com tamanho limitado a fim de evitar a degradação do desempenho computacional e a redundância de soluções. Benítez *et al*. (2005) [29] sugerem que o tamanho do repositório seja mantido em torno de 50 a 100 soluções. Caso o número de elementos exceda esse limite, é necessário adotar políticas de contenção baseadas em critérios de diversidade. Para mitigar o risco de convergência prematura e ampliar a capacidade de exploração em regiões inexploradas da fronteira, é recomendada a utilização de operadores de mutação adaptativa.

Segundo Mezura-Montes & Coello Coello (2005) [17], a aplicação de mutações a cerca de 5% das partículas a cada 10 gerações é suficiente para reintroduzir diversidade sem comprometer a estabilidade geral do processo. Tais mutações podem assumir diferentes formas: a mutação uniforme consiste na substituição aleatória do valor de uma variável por outro dentro do seu domínio viável; a mutação gaussiana aplica ruído aditivo de distribuição normal à posição da partícula; e a mutação adaptativa ajusta a intensidade do distúrbio de acordo com o progresso do algoritmo, com maior intensidade nas fases iniciais e mais sutil à medida que a população converge. Em contraste com o PSO tradicional, o MOPSO prescinde da definição de topologias fixas de vizinhança, como anel ou estrela. A função de cooperação entre partículas é mediada unicamente pelo arquivo elitista e pelo mecanismo de seleção de líderes, os quais operam de maneira descentralizada com base em critérios de diversidade. Dessa forma, a estrutura social do enxame torna-se implicitamente adaptativa, ajustando-se dinamicamente à distribuição das soluções ao longo da fronteira.

O número de iterações T, que define o horizonte temporal do algoritmo, é geralmente fixado entre 100 e 1000 ciclos. Entretanto, para problemas de grande escala ou com topologias de fronteira altamente fragmentadas, torna-se vantajoso o uso de métricas dinâmicas de convergência, como o *hypervolume* e o *Inverted Generational Distance* (IGD), que possibilitam o monitoramento contínuo da qualidade das soluções ao longo do tempo evolutivo. Tais métricas, discutidas em profundidade na Seção 2.7.3, permitem a detecção de estagnação precoce e a aplicação de medidas corretivas.

Tabela 1 – **Parâmetros recomendados para a implementação do MOPSO**

Parâmetro	Intervalo	Valor típico	Referência
N (enxame)	50–200	100	Coello Coello *et al*. (2002)
ω (inércia)	$[0{,}4,\ 0{,}9]$	Decaimento: $0{,}9 \to 0{,}4$	Shi & Eberhart (1998)
$c_1,\ c_2$ (aceleração)	$[1{,}5,\ 2{,}5]$	$c_1 = c_2 = 2{,}0$	Kennedy & Eberhart (1995)
$\lvert\mathcal{A}(t)\rvert$ (arquivo)	50–100	100	Benítez *et al*. (2005)
Taxa de mutação	—	5% a cada 10 gerações	Mezura-Montes & Coello (2005)
T (iterações)	100–1000	500	—

Conclui-se, portanto, que o sucesso do MOPSO em aplicações práticas — como no dimensionamento ótimo de redes de filas finitas — exige uma sintonia fina e contextualizada de seus parâmetros principais. A seleção adequada dos valores de $(N, \omega, c_1, c_2, |A(t)|, T)$, bem como a implementação criteriosa de operadores de diversidade, devem levar em conta as características topológicas da fronteira de Pareto, a escala do espaço de busca e os *trade-offs* entre objetivos conflitantes. Tais decisões de projeto são fundamentais para assegurar a robustez e a eficiência do algoritmo no tratamento de problemas multiobjetivo de alta complexidade. Tão importante quanto a configuração parametrizada é a avaliação objetiva da qualidade das soluções produzidas ao longo da execução. Em sequência serão abordadas as principais métricas de convergência e diversidade empregadas para esse fim.

2.7.3 Avaliação da Fronteira de Pareto

A avaliação rigorosa de algoritmos de otimização multiobjetivo, como o MOPSO, requer métricas capazes de mensurar simultaneamente quão próximo um conjunto de soluções está do Pareto ideal e qual a extensão de sua cobertura ao longo da fronteira. Em aplicações voltadas para Engenharia de desempenho de redes de filas finitas, tais métricas permitem comparar diretamente configurações que envolvem compromissos entre taxa de bloqueio, *throughput* e custo de infraestrutura, para assegurar que diferentes regiões do espaço de *trade-offs* sejam exploradas de modo satisfatório.

Uma das métricas mais utilizadas na literatura é o hipervolume (do inglês, *Hypervolume*, HV), que avalia o volume total do espaço objetivo dominado por um conjunto de soluções não-dominadas, em relação a um ponto de referência previamente definido. Seja $A \subset \mathbb{R}^m$ um conjunto de soluções não dominadas e $r = (r_1, r_2, \ldots, r_m) \in \mathbb{R}^m$ um vetor de referência, a métrica é dada por:

$$HV(A) = \text{volume}\left(\bigcup_{\mathbf{x}\in A} [f_1(\mathbf{x}), r_1] \times [f_2(\mathbf{x}), r_2] \times \cdots \times [f_m(\mathbf{x}), r_m]\right). \tag{2.15}$$

Essa métrica é considerada estritamente monotônica, ou seja, o valor do hipervolume aumenta somente quando uma solução melhora efetivamente em algum dos objetivos sem piorar nos demais. Por isso, o hipervolume tem a vantagem de capturar simultaneamente aspectos de convergência e de diversidade, sendo particularmente sensível à presença de soluções em regiões periféricas da fronteira. Em redes de filas finitas, o HV permite comparar diretamente grupos de soluções que representem diferentes escolhas de capacidade de *buffer* ou alocação de servidores, pois conjuntos que atinjam menor probabilidade de bloqueio e maior taxa de atendimento dominam naturalmente os de desempenho inferior.

Complementarmente, a métrica conhecida como distância geracional invertida (*Inverted Generational Distance*, IGD) tem sido amplamente adotada para avaliar a proximidade de uma fronteira aproximada A em relação a uma fronteira de referência ideal P^*. Essa métrica calcula a média das distâncias entre cada ponto $\mathbf{x}^* \in P^*$ e seu vizinho mais próximo $\mathbf{x} \in A$, conforme a seguinte expressão:

$$IGD(A, P^*) = \frac{1}{|P^*|} \sum_{\mathbf{x}^* \in P^*} \min_{\mathbf{x} \in A} \|\mathbf{x} - \mathbf{x}^*\|, \tag{2.16}$$

no qual $\|\mathbf{x} - \mathbf{x}^*\|$ é a distância euclidiana entre o ponto $\mathbf{x} \in A$ e $\mathbf{x}^* \in P^*$. A IGD avalia a cobertura global e a proximidade ao ótimo teórico, mas depende do conhecimento prévio de P^*, condição nem sempre factível em problemas complexos como redes de filas com topologias amplas e parametrizações variáveis [21, 31].

Adicionalmente, a métrica de dispersão, ou *Spread* (Δ), quantifica a uniformidade da distribuição dos pontos ao longo da fronteira. Se d_i representa a distância euclidiana entre soluções consecutivas ordenadas e $\bar{d}$ a distância média, então:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|A|-1} |d_i - \bar{d}|}{d_f + d_l + (|A| - 1)\bar{d}}, \tag{2.17}$$

em que d_f e d_l são as distâncias entre as soluções extremas da fronteira de referência e as extremidades do conjunto aproximado. Valores menores de Δ indicam soluções distribuídas de modo mais uniforme, o que significa, no contexto de filas finitas, uma representação ampla dos *trade-offs* possíveis entre taxa de bloqueio e atendimento. No entanto, essa métrica não fornece indicação sobre a proximidade dos pontos em relação ao Pareto ideal [17, 31].

Nenhuma dessas métricas, isoladamente, é suficiente para caracterizar a qualidade de uma fronteira de Pareto. O HV evidencia convergência e cobertura, mas pode não penalizar adequadamente buracos em regiões esparsas. A IGD mede a proximidade absoluta ao Pareto ideal, mas depende do conhecimento de P^*. A dispersão avalia a uniformidade, mas não reflete quão próximas as soluções estão do ótimo. Assim, recomenda-se o uso conjunto de HV, IGD e Δ para uma avaliação abrangente: HV fornece uma medida de dominância do espaço objetivo; IGD avalia a proximidade à fronteira ideal; Δ assegura cobertura uniforme. Em redes de filas finitas — em que taxas de bloqueio, *throughput* e custo coexistem em conflito —, essa abordagem multifacetada é essencial para uma análise detalhada dos compromissos operacionais possíveis [17, 19, 21].

No que tange ao critério de parada, o uso de um número máximo de gerações T limita o tempo de execução independentemente da convergência observada [18]. Contudo, esse critério pode encerrar prematuramente ou prolongar inutilmente a

execução. Por isso, recomenda-se complementá-lo com a verificação da estabilização do hipervolume. Define-se o encerramento da execução quando:

$$\max_{t-G\leqslant\tau\leqslant t} |HV(\tau) - HV(t)| < \varepsilon, \tag{2.18}$$

ou seja, quando a variação do HV nas últimas G gerações for inferior a um limiar ε. Essa condição evita iterações desnecessárias e assegura que a fronteira aproximada esteja estável [17, 31].

Em resumo, o uso rigoroso de métricas de avaliação constitui uma etapa fundamental na validação da eficácia de algoritmos como o MOPSO. Ao quantificar de maneira objetiva aspectos de convergência, diversidade e representatividade, tais métricas fornecem subsídios sólidos para a análise comparativa de diferentes estratégias de otimização multiobjetivo e para o desenho de soluções robustas em sistemas complexos. A próxima seção apresenta a aplicação prática do MOPSO à modelagem matemática de redes de filas com capacidade finita, contextualiza os conceitos abordados até aqui em cenários reais de decisão operacional.

3 Resultados Alcançados

3.1 Aplicação de MOPSO para Redes de Filas Finitas

A crescente complexidade dos sistemas de filas em rede, como centros de atendimento, sistemas de tráfego de dados e ambientes produtivos com servidores limitados, exige abordagens de modelagem e otimização cada vez mais sofisticadas. A natureza inerentemente multiobjetivo desses problemas, em que métricas de desempenho como tempo médio de espera, taxa de bloqueio, utilização de servidores e capacidade de *buffers* frequentemente entram em conflito, torna a busca por soluções eficientes um desafio substancial.

Diante das limitações dos métodos analíticos tradicionais, que nem sempre são aplicáveis a sistemas de grande porte, estocásticos ou não lineares, surge a necessidade de técnicas flexíveis e heurísticas para apoiar o processo de tomada de decisão. O algoritmo MOPSO emerge como uma meta-heurística promissora, inspirada no comportamento coletivo de enxames, capaz de explorar espaços de busca multidimensionais e identificar conjuntos de soluções de compromisso.

Em sequência, a aplicação prática do MOPSO para o complexo problema de alocação de recursos em redes de filas finitas, transita dos fundamentos teóricos para a implementação e análise concreta em um contexto de sistema estocástico. As subseções subsequentes detalham a abordagem de modelagem, a formulação do problema multiobjetivo, as adaptações computacionais específicas do MOPSO, o delineamento experimental e uma análise dos resultados, o que cessa em uma compreensão abrangente dos *trade-offs* inerentes a esses sistemas.

3.1.1 Modelagem Estocástica de Redes de Filas Finitas

Uma representação precisa de sistemas de filas finitas exige a construção de modelos estocásticos que capturem a dinâmica e a incerteza inerentes a esses ambientes. Redes de filas finitas são sistemas complexos no quais recursos limitados, como servidores e *buffers*, impactam diretamente o desempenho, isto inclui a taxa de atendimento (*throughput*) e a probabilidade de bloqueio. A modelagem estocástica é, portanto, indispensável para compreender o comportamento desses sistemas. Representações por grafos são bastante adequadas para representar os componentes essenciais de uma rede de filas, que incluem nós, em que cada um representa uma fila individual. Estes nós são caracterizados por processos de chegada, que podem seguir distribuições como a de Poisson (Markoviana), distribuições de tempo de serviço (Exponencial, Geral,

entre outras), uma capacidade de *buffer* finita e um número específico de servidores. As interconexões entre esses nós definem o fluxo de clientes através da rede. A notação de Kendall, como $M/G/c/k$, é fundamental para descrever filas com chegadas Markovianas, tempos de serviço gerais, c servidor e capacidade finita k. Esta notação é utilizada de forma consistente para caracterizar cada nó da rede e, permitir uma representação padronizada e rigorosa. A grande maioria dos sistemas reais opera sob a restrição de recursos limitados, o que eleva a alocação conjunta de *buffers* e servidores, conhecida como *Buffer and Servers Allocation Problem* (BSAP), a um problema instigante na teoria das filas que foi formalizado por Cruz & Van Woensel (2014) [32].

Versões de interesse prático, porém menos abrangentes que o BSAP são o *Buffer Allocation Problem* (BAP) apresentado inicialmente por MacGregor Smith & Cruz (2005) [33], neste contexto, consideram-se redes acíclicas compostas por nós do tipo $M/G/1/k$, em que os tempos entre chegadas seguem distribuições exponenciais Markovianas e os tempos de serviço seguem distribuições geral, cada nó possui um único servidor e a capacidade total do sistema, em que se inclui a fila e o cliente em serviço, está limitada a k_i vagas e o *Server Allocation Problem* (SAP) apresentado por Duarte (2024) [34], neste contexto, consideram-se redes acíclicas compostas por nós do tipo $M/M/c$, em que os tempos entre chegadas e os tempos de serviço seguem distribuições exponenciais Markovianas, cada nó possui um c_i servidores e a capacidade total do sistema é ilimitada. A Figura 6 ilustra a dinâmica de um sistema $M/G/1/k$ isolado e, demostra os processos de chegada, atendimento e o ponto de bloqueio.

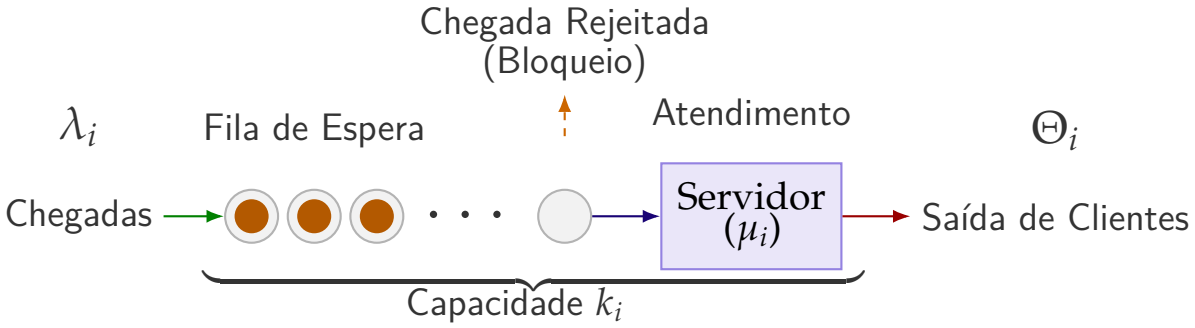

Figura 6 – Representação esquemática de um sistema de filas $M/G/1/k$

A estrutura da rede contempla m nós, indexados por $i \in \{1, \ldots, m\}$, em que cada nó opera de maneira independente, mas está interconectado por rotas de tráfego com os demais. Cada nó i é caracterizado por um conjunto de variáveis fundamentais, sumarizadas na Tabela 2: λ_i, que denota a taxa média de chegada de entidades (clientes ou pacotes); μ_i, a taxa média de atendimento do servidor; k_i, a capacidade total finita do sistema; e $\rho_i = \lambda_i/\mu_i$, o fator de utilização, que expressa a carga média imposta ao servidor em relação à sua capacidade de processamento.

Tabela 2 – Variáveis Fundamentais da Modelagem de Redes de Filas Finitas

Variável	Significado	Intervalo de Valores Típicos
λ_i	Taxa média de chegada ao nó i	$(0, \infty)$
μ_i	Taxa média de atendimento do nó i	$(0, \infty)$
k_i	Capacidade total do sistema (fila + servidor) no nó i	Inteiro positivo ($k_i \geqslant 1$)
ρ_i	Fator de utilização do nó i (λ_i/μ_i)	$(0, 1)$ (para estabilidade)
p_{k_i}	Probabilidade de bloqueio no nó i	$[0, 1]$
Θ_i	*Throughput* efetivo do nó i	$(0, \lambda_i]$
L_i	Número médio de clientes no sistema no nó i	$[0, k_i]$
W_i	Tempo médio de permanência/ espera no nó i	$[0, \infty)$
π_{ji}	Probabilidade de roteamento do nó j para o nó i	$[0, 1]$

A condição de estabilidade local, $\rho_i < 1$, é uma premissa indispensável para que o sistema não experimente um crescimento indefinido da fila e, garante sua permanência em regime estacionário, conforme destacado por Souza *et al*. (2023) [12].

As métricas de desempenho são preponderantes para avaliar a eficiência de uma rede de filas. Entre elas, destacam-se o tempo médio de espera, a taxa de bloqueio, a utilização dos servidores e a capacidade dos *buffers*, métricas que frequentemente entram em conflito. O bloqueio ocorre quando um cliente chega a uma fila que atingiu sua capacidade máxima, o cliente é então forçado a sair do sistema ou aguardar em um *buffer* externo. A minimização da probabilidade de bloqueio é um objetivo crítico, visto que bloqueios excessivos reduzem a performance geral da rede. A taxa de atendimento (*throughput*), definida como o número de trabalhos ou clientes atendidos por unidade de tempo, é outra métrica fundamental a ser maximizada, pois ela pode indicar a eficiência do sistema.

Para uma avaliação rigorosa do desempenho de cada nó individual, recorre-se usualmente a três métricas clássicas e amplamente consolidadas na teoria de filas. A primeira é a *probabilidade de bloqueio* (p_{k_i}), que quantifica a fração de clientes que, ao chegarem, encontram o sistema completamente ocupado e, consequentemente, são rejeitados. Para o modelo $M/G/1/k$, não existe uma expressão fechada, para modelo $M/M/1/k$, a expressão exata para essa probabilidade é dada por:

$$p_{k_i} = \frac{(1-\rho_i)\rho_i^{k_i}}{1-\rho_i^{k_i+1}}, \tag{3.1}$$

Esta fórmula, amplamente reconhecida na literatura, provê uma estimativa fechada da chance de bloqueio em função do fator de utilização (ρ_i) e da capacidade do sistema (k_i) (Gross e Harris (2008) [35]).

Para o modelo $M/G/1/k$, em uma única fila, uma estimativa adequadada (em uma formula fechada computacionalmente eficiente e precisa) para probabilidade de bloqueio P_{k_i} foi proposta por MacGregor Smith [36] com base em uma aproximação de dois momentos apresentada por Kimura [37]:

$$P_{k_i} = \frac{\rho_i^{\left(\dfrac{2+\sqrt{\rho_i}\,\mathrm{cv}_i^2-\sqrt{\rho_i}+2(k_i-1)}{2+\sqrt{\rho_i}\,\mathrm{cv}_i^2-\sqrt{\rho_i}}\right)}(\rho_i-1)}{\rho_i^{\left(2\,\dfrac{2+\sqrt{\rho_i}\,\mathrm{cv}_i^2-\sqrt{\rho_i}+(k_i-1)}{2+\sqrt{\rho_i}\,\mathrm{cv}_i^2-\sqrt{\rho_i}}\right)}-1}, \tag{3.2}$$

em que $\mathrm{cv}_i^2 = \mathrm{Var}(T_{s_i})/\mathbb{E}^2(T_{s_i})$ é o quadrado do coeficiente de variação do tempo de serviço (T_{s_i}). Vários estudos anteriores confirmam que a aproximação de P_{k_i} é precisa para uma ampla gama de valores [26, 33, 38].

Em seguida, o *throughput efetivo* (Θ_i) do nó, definido como a taxa média de saída de clientes, decorre diretamente da fração de chegadas não bloqueadas e é expresso por:

$$\Theta_i = \lambda_i(1 - p_{k_i}). \tag{3.3}$$

Complementarmente, a quantidade média de clientes no sistema (L_i) é utilizada para determinar o *tempo médio de permanência ou espera* (W_i), por meio da aplicação da Lei de Little [39], uma relação fundamental em sistemas de filas. Formalmente, tem-se:

$$W_i = \frac{L_i}{\Theta_i}, \tag{3.4}$$

em que L_i pode ser obtido por meio das fórmulas clássicas da teoria de filas para sistemas $M/M/1/k$, mas também não possui formato analítico fechado para filas $M/G/1/k$. Em redes interconectadas, em que os fluxos de tráfego podem ser complexos, estimativas aprimoradas para L_i e p_{k_i} podem ser obtidas via o Método de Expansão Generalizada (do inglês, *Generalized Expansion Method* GEM), que corrige iterativamente as taxas de chegada e as probabilidades de bloqueio em redes acíclicas finitas [40].

O GEM pode ser definido como uma técnica de aproximação iterativa que ajusta localmente as taxas de chegada (λ_i) e as probabilidades de bloqueio (p_{k_i}) em cada nó de uma rede de filas finitas, até que as variações entre as iterações caiam abaixo de um limiar ε [33]. Esse processo é essencial para lidar com as interdependências entre os nós e o fenômeno de propagação de bloqueio, fornece estimativas de desempenho que justificam o custo computacional associado quando o GEM é integrado em esquemas de otimização heurística.

Em cenários nos quais múltiplos nós se interconectam, a modelagem explícita do roteamento de clientes torna-se imperativa para capturar corretamente as dependências estocásticas. Denota-se por p_{ji} a probabilidade de que um cliente, após ser atendido no nó j, seja encaminhado ao nó i, a taxa efetiva de chegada ao nó i é dada pela equação de balanceamento de fluxo:

$$\lambda_i = \sum_{j=1}^{n} \lambda_j \Theta_j p_{ji}, \tag{3.5}$$

em que Θ_j representa o *throughput* efetivo do nó j, calculado conforme a Equação (3.3). A equação (3.5) está sujeita à condição de normalização:

$$\sum_{i=1}^{n} p_{ji} = 1, \quad \forall j \in \{1, \ldots, n\} \text{ com saídas}, \tag{3.6}$$

o que garante a conservação de fluxo probabilístico ao longo da rede. Essa formulação permite representar uma vasta gama de topologias de rede — desde configurações simples em série ou paralelas até estruturas complexas com múltiplas ramificações e fusões de fluxos — o que mantem o rigor na caracterização das interdependências estocásticas entre os nós.

A validade dessa aproximação em topologias não triviais é reforçada por estudos como o de van Woensel & Cruz (2014) [41], que discutem políticas de roteamento ótimo em redes de filas com múltiplos servidores sob condições de congestionamento e, confirmam a utilidade do GEM como base para estimativas robustas em algoritmos de otimização.

A complexidade inerente às redes de filas finitas demanda a análise de diferentes topologias, as quais moldam o comportamento dinâmico do sistema e suas dependências internas. Entre os arranjos mais comuns destacam-se:

- **redes em série (tandem):** a saída de um nó serve como entrada direta para o próximo, forma-se uma cadeia linear de processamento;
- **redes em paralelo (*split/merge*):** os clientes são distribuídos entre múltiplos nós simultaneamente ou são posteriormente reunificados;
- **redes mistas:** combinam elementos de arranjos em série e paralelo;
- **redes cíclicas:** combinam elementos de arranjos em série e paralelo e permite incluir ciclos de retroalimentação.

A literatura clássica apresenta diagramas esquemáticos representando tais topologias — como redes *tandem*, redes com servidor central e configurações com ciclos

fechados. Estas representações visuais são fundamentais para a compreensão estrutural do sistema, as quais permitem a identificação clara das rotas de tráfego, das conexões entre os nós e dos pontos potenciais de congestionamento.

A topologia adotada impacta diretamente na propagação de bloqueios entre os nós, na dificuldade de caracterizar o comportamento em regime estacionário e na complexidade computacional envolvida na avaliação de desempenho e na otimização. Redes mais densamente conectadas, por exemplo, exigem métodos mais sofisticados de modelagem e técnicas de otimização multiobjetivo que considerem as interdependências locais e globais do sistema [35,41]. As Figuras 7, 8, 9, 10 e 11 ilustram uma redes em diversas topologias (série, fusão, divisão, mista e cíclica respectivamente).

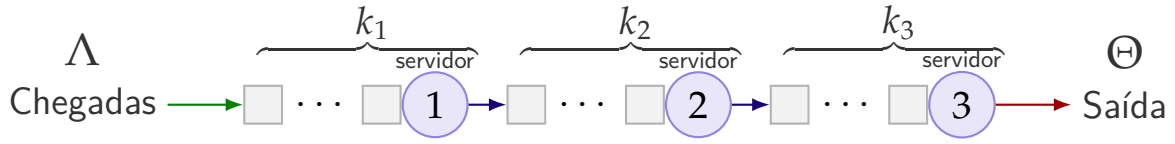

Figura 7 – Topologia de Rede em Série (*tandem*) com clientes que percorrem nós $M/G/1/k_i$ de forma sequencial.

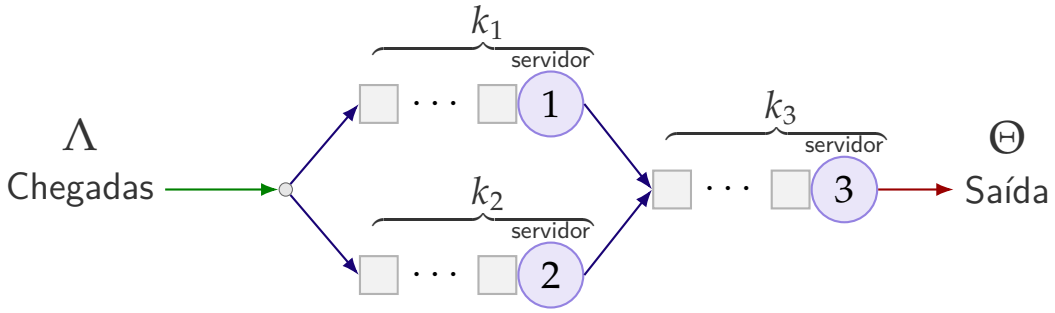

Figura 8 – Topologia de Rede em Paralelo (*merge*) com clientes que percorrem nós $M/G/1/k$ e ocorre a reunificação de dois ou mais nós com o efeito de agregação de tráfego.

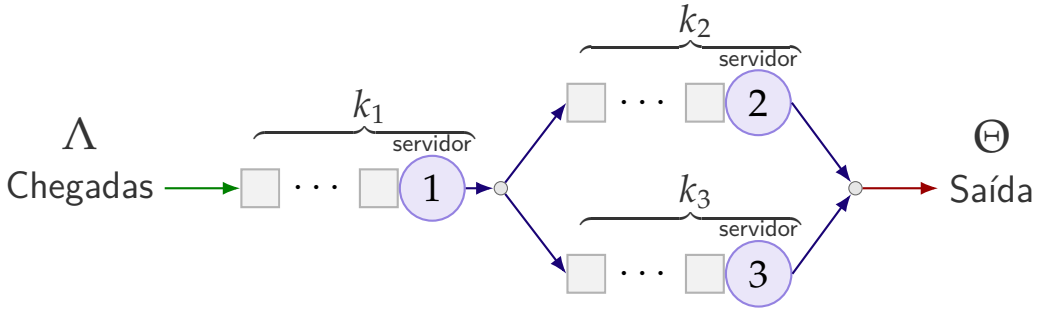

Figura 9 – Topologia de Rede em Paralelo (*split*) com clientes que percorrem nós $M/G/1/k$ e ocorre a separação de dois ou mais nós com o efeito de divisão de tráfego.

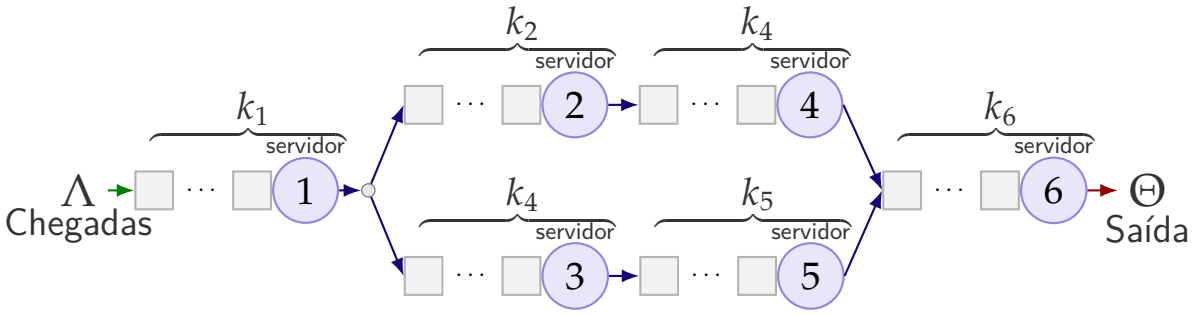

Figura 10 – Topologia de Rede Mista com clientes que percorrem nós $M/G/1/k$ com roteamentos para múltiplas filas em série e/ou em paralelo.

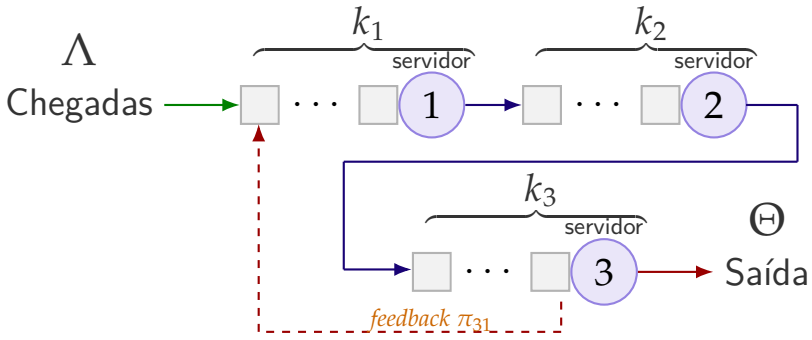

Figura 11 – Topologia de Rede Cíclica com *feedback* Inferior, clientes chegam ao nó 1 com taxa λ, percorrem sequencialmente os nós $M/G/1/k$ e podem retornar ao início após atendimento no nó 3, forma-se um ciclo de realimentação.

A modelagem de redes de filas finitas não se resume à simples agregação de filas individuais. A investigação de uma solução para o BAP, ou então o BSAP torna-se consideravelmente mais complexa em uma rede de filas interconectadas. As topologias de rede e o conceito de bloqueio indicam que um bloqueio em um nó pode repercutir em nós a montante e a jusante. Essa interdependência entre os nós, especialmente na presença de capacidades limitadas e bloqueio, gera um fenômeno de "propagação de bloqueio". O bloqueio em uma fila pode retroagir, o que impede a saída de clientes de filas anteriores, ou avançar, que impede a entrada de clientes em filas subsequentes. Isso significa que a otimização de um único nó isoladamente não garante a otimização global da rede. A otimização deve considerar a rede como um todo e, assim, buscar um equilíbrio na alocação de recursos que minimize a probabilidade de bloqueio em toda a cadeia de serviço, e não apenas em pontos isolados [33]. Essa complexidade intrínseca do problema justifica a necessidade de abordagens meta-heurísticas como o MOPSO, capazes de explorar o espaço de soluções de forma holística.

A avaliação de desempenho de redes de filas finitas, particularmente aquelas com topologias complexas e distribuições de serviço gerais ($M/G/1/k$), não possui soluções

analíticas de forma fechada na maioria dos casos. Isso implica que as funções-objetivo, como *throughput* ou probabilidade de bloqueio, não podem ser calculadas diretamente por uma fórmula simples, mas requerem métodos aproximados ou simulação. A utilização do *Generalized Expansion Method* (GEM) ou de simulação de eventos discretos para estimar as métricas de desempenho para cada solução candidata introduz um custo computacional significativo por avaliação. Essa complexidade da função-objetivo, combinada com um espaço de busca de alta dimensionalidade (devido a múltiplos *buffers* e/ou servidores), é a principal razão pela qual algoritmos evolutivos como o MOPSO são indispensáveis, pois não dependem de gradientes ou formas analíticas da função-objetivo.

A escolha do modelo $M/G/1/k$ neste estudo oferece um equilíbrio otimizado entre simplicidade analítica e fidelidade operacional. Ele é suficientemente expressivo para capturar de maneira eficaz os efeitos de saturação, a concorrência por recursos limitados e os inerentes *trade-offs* entre desempenho do sistema e o investimento em capacidade. Com essa modelagem estocástica devidamente formalizada, estabelece-se a base robusta para a formulação multiobjetivo que será detalhada na próxima seção, em que as métricas de bloqueio, *throughput* e tempo de espera convergem como funções-objetivo primárias na otimização por enxame de partículas.

3.1.2 Caso Aplicado de Utilização do MOPSO em Otimização em Redes de Filas

O estudo apresentado por Souza *et al.* (2025) [42] considera a minimização simultânea da alocação total de *buffer* e das taxas gerais de serviço na rede, maximizando seu *throughput*. O estudo apresentou uma *fine-tuning* para otimização multiobjetivo de enxames de partículas. O presente estudo replicou essa abordagem a uma rede de filas finitas acíclicas e gerais de servidor único para otimizar *throughput* aliado a redução de *buffers* alocados e capacidade de servidores por meio de suas taxas de serviço e comparou com uma abordagem clássica com o algoritmo NSGA-II [43]. A abordagem apresentada gera um conjunto de Pareto subótimo para esses objetivos conflitantes. A aplicação foi executada aqui para uma rede de filas mista apresentada na Figura 12. O experimento foi conduzido para diferentes cv^2 (0,5; 1,0 e 1,5) para caracterizar tempos de atendimento gerais, respectivamente, hipo-exponenciais, exponenciais (Markoviano) e hiperexponenciais. A versão do algoritmo MOPSO [42] foi implementada em FORTRAN para compatibilidade com implementações anteriores do NSGA-II [43] e do GEM [36, 38]. Os códigos foram gentilmente cedidos mediante solicitação para fins educacionais e de pesquisa junto aos autores. Os experimentos computacionais foram realizados em um processador Intel® Core™ i7-1165G7 de 11ª geração de 2,80 GHz rodando Windows 11 Home de 64 bits, com 8,00 GB de RAM.

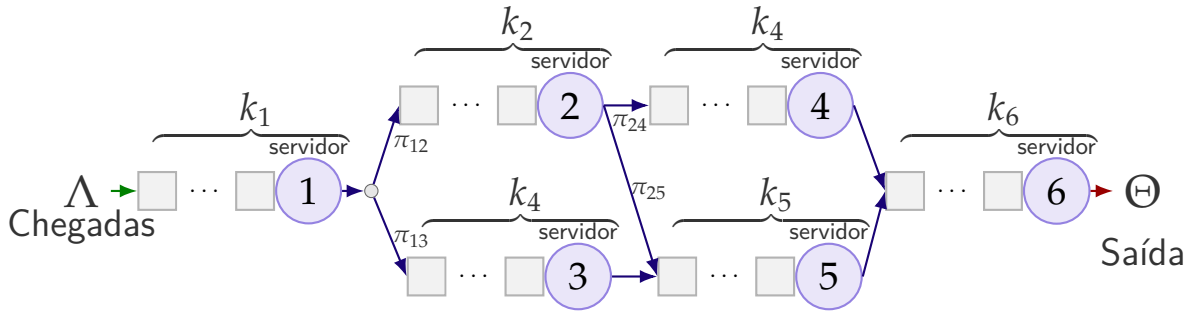

Figura 12 – Topologia mista experimental com nós $M/G/1/k$ com roteamentos para múltiplas filas em série e/ou em paralelo.

As Figuras 13, 14 e 15 ilustram uma comparação entre soluções típicas geradas pelo NSGA-II e pelo MOPSO. Cada Figura consiste de três gráficos, à esquerda, o *throughput* com a alocação total de *buffers*; no centro, o *throughput* com as taxas de serviço gerais; e à direita, o *throughput* das soluções. Com base nas evidências empíricas extraídas da Figura 13 (e subsequentes), que ilustram o desempenho do MOPSO frente ao NSGA-II sob diferentes regimes de variabilidade do tempo de serviço, observa-se um comportamento sistematicamente superior do MOPSO em cenários com baixa, média e alta variabilidade.

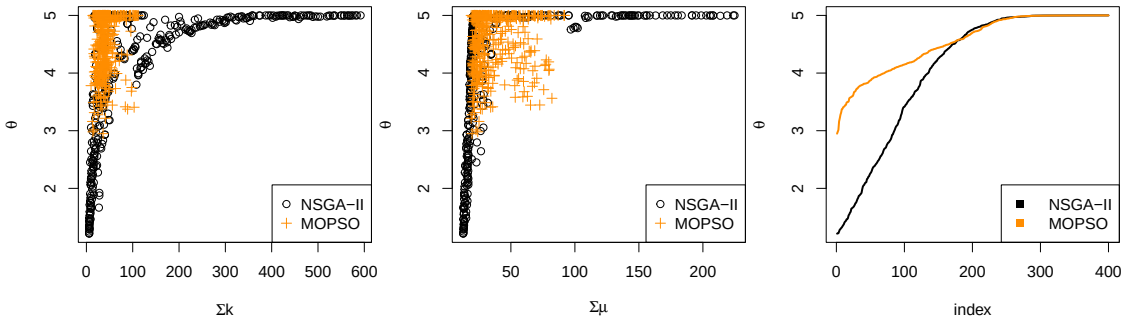

Figura 13 – Soluções para a rede de filas da Figura 12 com ($cv^2 = 0{,}5$): alocação total de buffer à esquerda, taxas de serviço no centro e *throughput* à direita.

No regime hipoexponencial, caracterizado por um coeficiente de variação quadrático ($cv^2 = 0{,}5$), a fronteira de Pareto gerada pelo MOPSO concentra-se na região superior esquerda do espaço de soluções, esse fato expõe uma dominância clara sobre o NSGA-II. Em termos operacionais, isso significa que o MOPSO é capaz de alcançar níveis elevados de *throughput* ($\Theta \approx 5$) com alocações significativamente menores de capacidade de *buffer*, tipicamente entre $\sum k \approx 30$ *e* 50, enquanto o NSGA-II requer valores superiores a $\sum k > 200$ para atingir desempenhos comparáveis. Tal comportamento indica não apenas uma utilização mais eficiente dos recursos pelo MOPSO,

mas também um mapeamento mais acurado da região de soluções de alto desempenho, aproximando-se da verdadeira fronteira de Pareto. Esse padrão de superioridade é reiterado no gráfico (b) da mesma figura, em que se analisa a relação entre *throughput* e a soma das taxas de serviço ($\sum \mu$). O MOPSO alcança níveis similares de *throughput* com taxas significativamente inferiores, o que implica menor carga computacional e energética para o sistema. A implicação prática desse achado é que a solução gerada pelo MOPSO não apenas demanda menos recursos físicos (*buffers*), mas também opera com menor exigência de capacidade dos servidores, o que pode traduzir-se em economia direta de infraestrutura ou energia. O gráfico (c) fortalece ainda mais esse argumento, as soluções obtidas por cada um dos algoritmos na fronteira final estão ordenadas por *throughput*, é fácil ver que aproximadamente metade (≈ 200) das soluções obtidas em ambos algoritmos possuem *throughput* próximo do λ de entrada na rede. Por outro lado, quando comparadas as soluções de menor *throughput*, as soluções fornecidas pelo MOPSO são notoriamente superiores. Esse efeito mostra a habilidade do MOPSO em encontrar soluções eficazes de pouco investimento, as soluções de *throughput* mais baixo são claramente as soluções com menor investimento em *buffers* e servidores. De uma forma objetivo, o MOPSO se mostra como um algoritmo hábil para fornecer boas soluções em sistemas com baixa disponibilidade de recursos.

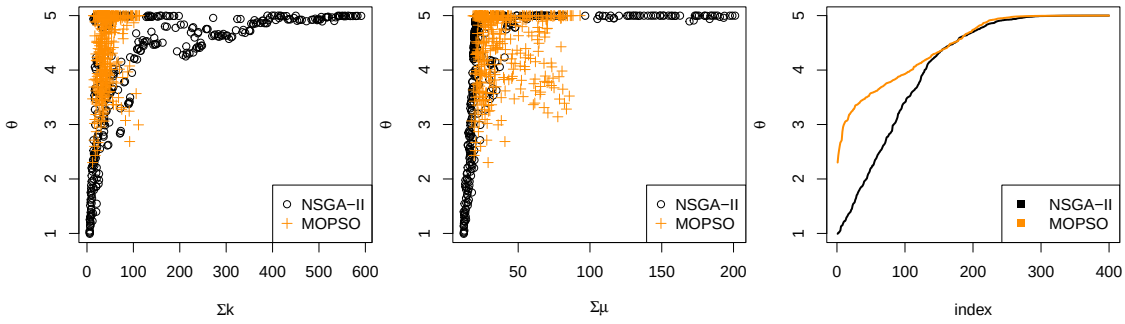

Figura 14 – Soluções para a rede de filas da Figura 12 com ($cv^2 = 1{,}0$): alocação total de buffer à esquerda, taxas de serviço no centro e *throughput* à direita.

Quando a variabilidade dos tempos de serviço aumenta e o sistema passa a operar sob regime Markoviano ($cv^2 = 1{,}0$), conforme ilustrado na Figura 14, observa-se uma leve compressão da fronteira de Pareto — um fenômeno esperado, dado o maior grau de aleatoriedade dos tempos de atendimento. Ainda assim, o MOPSO mantém sua dominância. O incremento marginal de recursos exigido para sustentar um dado *throughput* se apresenta como um deslocamento moderado da curva, mas o MOPSO continua superando o NSGA-II com folga em termos de eficiência e convergência. A curva laranja do MOPSO segue consistentemente acima da curva preta do NSGA-II,

o que evidencia, mesmo em presença de maior ruído estocástico, a eficácia do ajuste fino proposto por Souza *et al.* (2025) [42] no controle adaptativo dos parâmetros de inércia, aceleração e diversidade populacional. A capacidade do MOPSO de preservar a diversidade das soluções e explorar nichos menos saturados do espaço de busca contribui diretamente para essa performance robusta.

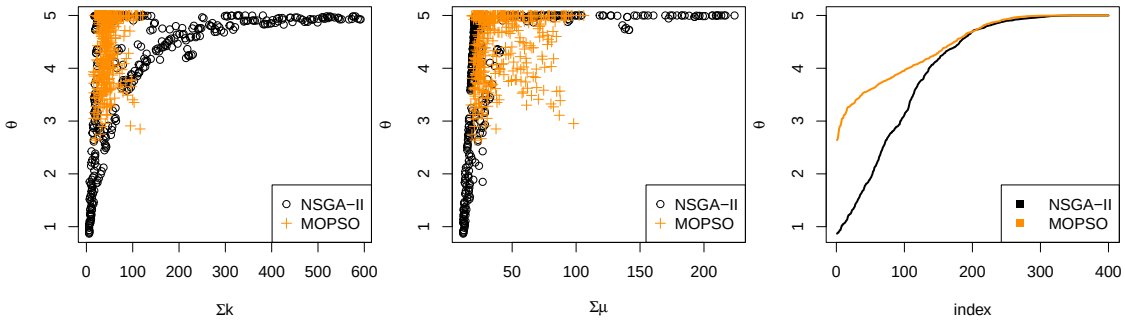

Figura 15 – Soluções para a rede de filas da Figura 12 com ($cv^2 = 1{,}5$): alocação total de buffer à esquerda, taxas de serviço no centro e *throughput* à direita.

Sob regime hiper-exponencial ($cv^2 = 1{,}5$), conforme a Figura 15, observa-se uma degradação mais acentuada da fronteira de Pareto, com deslocamento de ambas as heurísticas para regiões que exigem maior alocação de *buffers* e taxas de serviço mais elevadas, em troca de níveis inferiores de *throughput*. Apesar desse comportamento, que é uma manifestação direta do aumento da variabilidade estocástica dos tempos de serviço — como foi previsto por Kleinrock (1975) [39] e reiterado por MacGregor Smith & Cruz (2005) [33] — o MOPSO mantém uma vantagem competitiva clara. Ele novamente entrega soluções com menores valores de $\sum k$ e $\sum \mu$ para cada valor fixo de *throughput* (Θ), em comparação com as soluções geradas pelo NSGA-II. A análise quantitativa apresentada na Tabela 3 reforça de maneira estatisticamente significativa essa dominância: o MOPSO obteve um *throughput* médio, superior ao NSGA-II.

Tabela 3 – Performance Média do NSGA-II e do MOPSO.

	NSGA-II			MOPSO		
Medida	Θ	$\sum k$	$\sum \mu$	Θ	$\sum k$	$\sum \mu$
média	4,0511	143,9458	42,9435	4,4722	43,6950	39,2556
desvio padrão	1,2157	164,6712	48,6312	0,5925	20,0907	19,0219

Nota: melhores resultados em destaque.

A média de *buffers* alocados foi de apenas 43,6950 para o MOPSO, contra expressivos 143,9458 do NSGA-II, enquanto as taxas de serviço somadas ($\sum \mu$) também

se mantiveram inferiores no MOPSO, com menor variabilidade em todas as métricas avaliadas. Esse padrão de comportamento evidencia uma tendência crítica: à medida que o coeficiente de variação dos tempos de serviço aumenta, tanto o uso de recursos quanto o desempenho máximo viável do sistema se deterioram. Este resultado está em consonância com a teoria clássica das filas, segundo a qual o aumento da variabilidade em processos de chegada ou atendimento impacta negativamente o desempenho global do sistema, o que eleva o tempo médio de espera, a probabilidade de bloqueio e exige maior capacidade de amortecimento ou velocidade de serviço para preservar níveis aceitáveis de desempenho. Tais evidências são consistentes com as análises de Deb (2001) [20], Zitzler *et al.* (2003) [31] e Cruz & van Woensel (2014) [41], que apontam a variabilidade como um dos principais fatores limitantes na eficiência de sistemas de atendimento sob restrição de recursos.

Assim, a combinação das evidências gráficas e numéricas demonstradas nesta seção permite afirmar que o ajuste parametrizado do MOPSO proposto por Souza *et al.* (2025) [42] constitui uma abordagem eficaz, robusta e computacionalmente vantajosa para a otimização multiobjetivo de redes de filas com capacidade finita. Sua capacidade de alcançar frentes de Pareto mais densas e diversificadas, com maior rapidez e menor consumo de recursos, o posiciona como a heurística preferencial em contextos de Engenharia de sistemas estocásticos, especialmente sob cenários de incerteza moderada a elevada. Essa performance superior não decorre apenas de sua estrutura meta-heurística, mas também do uso estratégico de mecanismos como arquivamento elitista, seleção adaptativa de líderes e operadores de diversidade, que conferem ao algoritmo uma notável plasticidade frente a problemas de alta complexidade estrutural.

4 Considerações Finais

Este estudo apresentou uma abordagem abrangente para a otimização multiobjetivo de redes de filas finitas, com ênfase na aplicação do algoritmo *MultiObjective Particle Swarm Optimization* (MOPSO). O estudo teve como objetivo principal desenvolver, implementar e avaliar a eficácia do MOPSO na resolução dos problemas de alocação de recursos, formulado de forma a conciliar métricas conflitantes como *throughput*, alocação de *buffers* e taxas de serviço. A fundamentação metodológica contemplou desde os princípios biológicos do PSO clássico até suas adaptações para problemas multiobjetivo, destacando a importância de estruturas como o arquivo elitista, a seleção distribuída de líderes e operadores de diversidade para assegurar convergência e abrangência da fronteira de Pareto. Uma revisão abrangente da bibliografia na área de modelagem e otimização de redes de filas finitas e aplicações do algoritmo PSO foi apresentada. A execução desse estudo também contibuiu para o aprimoramento da utilização da linguagem TEX, que é padrão na confecção de textos estatísticos em vários níveis de pesquisa.

A formulação do problema de otimização foi sustentada por uma modelagem estocástica robusta das redes de filas, sobretudo do tipo $M/G/1/k$. Essa modelagem permitiu traduzir decisões operacionais em um problema computacionalmente tratável, ainda que não analiticamente resolúvel. A simulação de diferentes cenários operacionais, que diversificam os coeficientes de variação dos tempos de serviço (cv^2), possibilitou uma avaliação sistemática da robustez e estabilidade dos algoritmos testados.

Os resultados empíricos demonstraram, de forma consistente, a superioridade do MOPSO em relação ao NSGA-II, algoritmo de referência em otimização multiobjetivo. Em todos os cenários avaliados, o MOPSO apresentou soluções com maior *throughput* médio, menor consumo de recursos e menor variabilidade estatística. As fronteiras de Pareto geradas foram mais densas, convergentes e uniformemente distribuídas, mesmo em regiões não convexas do espaço objetivo — um desafio conhecido para métodos baseados em agregação linear. Essas evidências confirmam que o MOPSO é particularmente eficaz para problemas com múltiplos objetivos conflitantes e variabilidade estrutural e, oferece ao tomador de decisão um conjunto mais amplo e preciso de soluções de compromisso. Do ponto de vista prático, o MOPSO mostrou-se altamente promissor para aplicações em sistemas reais como redes logísticas, sistemas de atendimento, cadeias produtivas e telecomunicações — contextos nos quais decisões sobre capacidade, desempenho e custo devem ser tomadas de forma integrada e adaptativa. Sua habilidade em gerar múltiplas soluções eficientes simultaneamente torna-o ideal para o apoio à decisão sob diferentes cenários de orçamento, prioridade ou restrição

operacional. Do ponto de vista teórico, a integração entre modelagem estocástica rigorosa e meta-heurísticas multiobjetivo contribui significativamente para o avanço do estado da arte na área de otimização em sistemas de filas.

Entretanto, reconhecem-se limitações importantes que abrem espaço para futuros avanços. A aplicação foi limitada a redes acíclicas e ao modelo $M/G/1/k$, não abrange redes com *feedback*, múltiplos servidores ou classes de clientes prioritárias. Além disso, a implementação em FORTRAN, embora eficiente, restringe a integração com ambientes computacionais mais modernos e a exploração de paralelização em larga escala. Estudos futuros devem explorar estratégias híbridas com aprendizado de máquina para ajuste dinâmico de parâmetros, aplicação em redes cíclicas com simulação de eventos discretos e inclusão de objetivos adicionais como consumo energético ou níveis de *QoS* diferenciados. Conclui-se, portanto, que o MOPSO representa uma alternativa poderosa, flexível e confiável para a otimização multiobjetivo em redes de filas finitas, capaz de capturar relações complexas de *trade-offs* e operar de forma estável sob incertezas estocásticas. Este trabalho contribui não apenas para consolidar os fundamentos algorítmicos do MOPSO, mas também para validar empiricamente sua aplicabilidade em sistemas reais e, abre caminho para sua adoção em domínios mais amplos da Engenharia, Logística e Ciência de Dados aplicada.

Referências

[1] Duarte, A. R., F. R. B. Cruz e G. L. Souza: *A greedy post-processing strategy for multi-objective performance optimization of general single-server finite queueing networks*. Soft Computing, 28(17):9483–9494, 2024. Citado na página 1.

[2] Kennedy, J. e R. Eberhart: *Particle swarm optimization*. Em *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, páginas 1942–1948. IEEE, 1995. Citado 8 vezes nas páginas 1, 4, 5, 6, 7, 8, 14 e 22.

[3] Shi, Y. e R. Eberhart: *Parameter selection in particle swarm optimization*. Em *Evolutionary Programming VII: 7th International Conference, EP98 San Diego, California, USA, March 25–27, 1998 Proceedings 7*, páginas 591–600. Springer, 1998. Citado 5 vezes nas páginas 1, 7, 8, 10 e 22.

[4] Poli, R., J. Kennedy e T. Blackwell: *Particle swarm optimization: An overview*. Swarm intelligence, 1:33–57, 2007. Citado 2 vezes nas páginas 1 e 10.

[5] Yang, X. S.: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley Publishing, 1st edição, 2010, ISBN 0470582464, 9780470582466. Citado na página 3.

[6] Reynolds, C. W.: *Flocks, herds and schools: A distributed behavioral model*. Em *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, páginas 25–34, 1987. Citado na página 4.

[7] Heppner, F. e U. Grenander: *The ubiquity of chaos*. Sl]: Amer Assn for the Advancement, página 247, 1990. Citado na página 4.

[8] Wang, D., D. Tan e L. Liu: *Particle swarm optimization algorithm: an overview*. Soft Computing, 22(2):387–408, 2018. Citado 3 vezes nas páginas 5, 6 e 7.

[9] Millonas, M. M.: *Self-organized stigmergic communication in ant colonies*. Journal of Theoretical Biology, 179(3):275–287, 1996. Citado 2 vezes nas páginas 5 e 6.

[10] Eberhart, R. C e Y. Shi: *Comparing inertia weights and constriction factors in particle swarm optimization*. Em *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, volume 1, páginas 84–88. IEEE, 2000. Citado na página 9.

[11] Clerc, M. e J. Kennedy: *The particle swarm–explosion, stability, and convergence in a multidimensional complex space*. IEEE Transactions on Evolutionary Computation, 6(1):58–73, 2002. Citado 4 vezes nas páginas 9, 10, 19 e 20.

[12] Souza, G. L., A. R. Duarte, G. Moreira e 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 5 vezes nas páginas 9, 10, 11, 12 e 29.

[13] van den Bergh, F. e A. P. Engelbrecht: *Cooperative learning in neural networks using particle swarm optimizers*. Em *South African Computer Journal*, volume 26, páginas 84–90, 2004. Citado na página 10.

[14] Kennedy, J. e R. Eberhart: *A discrete binary version of the particle swarm algorithm*. Em *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 5, páginas 4104–4108. IEEE, 1997. Citado na página 11.

[15] Blackwell, T. e P. Bentley: *Multiobjective optimization using evolutionary algorithms*. Em *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, volume 2, páginas 1157–1163. IEEE, 2002. Citado na página 11.

[16] Angeline, P. J.: *Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences*. Lecture Notes in Computer Science, 1447:601–610, 1998. Citado na página 11.

[17] Mezura-Montes, E. e C. A. Coello Coello: *Constrained multiobjective optimization using a multiobjective evolutionary algorithm*. Journal of Heuristics, 11(3):151–175, 2005. Citado 7 vezes nas páginas 11, 16, 20, 21, 23, 25 e 26.

[18] Coello Coello, C. A., G. T. Pulido e M. S. Lechuga: *MOPSO: A proposal for multiple objective particle swarm optimization*. Em *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, volume 2, páginas 1051–1056. IEEE, 2002. Citado 9 vezes nas páginas 11, 12, 14, 15, 18, 19, 21, 22 e 25.

[19] Mostaghim, S. e J. Teich: *Covering pareto-optimal fronts using multi-objective particle swarm optimization*. Em *Proceedings of the 2003 Congress on Evolutionary Computation. CEC'03*, páginas 632–639. IEEE, 2003. Citado 6 vezes nas páginas 11, 15, 16, 19, 21 e 25.

[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 5 vezes nas páginas 12, 16, 18, 19 e 38.

[21] Deb, K.: *Multi-objective optimisation using evolutionary algorithms: an introduction*. Em *Multi-objective evolutionary optimisation for product design and manufacturing*, páginas 3–34. Springer, 2011. Citado 2 vezes nas páginas 14 e 25.

[22] Deb, K.: *A fast and elitist multi-objective genetic algorithm: NSGA-II*. IEEE Trans. on Evolutionary Computation, 6(2):182–197, 2002. Citado na página 14.

[23] Khan, S. A e S. Rehman: *Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey*. Renewable and Sustainable Energy Reviews, 19:370–384, 2013. Citado na página 14.

[24] Fonseca, C. M. e P. J. Fleming: *An overview of evolutionary algorithms in multiobjective optimization*. Evolutionary computation, 3(1):1–16, 1995. Citado na página 14.

[25] van Veldhuizen, D. A e G. B. Lamont: *Multiobjective evolutionary algorithms: Analyzing the state-of-the-art*. Evolutionary computation, 8(2):125–147, 2000. Citado na página 15.

[26] Cruz, F. R. B., A. R. Duarte e T. van Woensel: *Buffer Allocation in General Single-Server Queueing Networks*. Computers & Operations Research, 35(11):3581 –3598, 2008. Citado 2 vezes nas páginas 15 e 30.

[27] Souza, G. L., A. R. Duarte, G. Moreira e F. R. B. Cruz: *A novel formulation for multi-objective optimization of general finite single-server queueing networks*. Em *2020 IEEE congress on evolutionary computation (CEC)*, páginas 1–8. IEEE, 2020. Citado na página 15.

[28] Fieldsend, J. E., R. M. Everson e S. Singh: *Using unconstrained elite archives for multiobjective optimization*. IEEE Transactions on Evolutionary Computation, 7(3):305–323, 2003. Citado na página 16.

[29] Benítez, J. E. A., R .M. Everson e J. E. Fieldsend: *ε-dominance: An optimality theory for approximation of Pareto sets*. Evolutionary Computation, 13(4):439–459, 2005. Citado 3 vezes nas páginas 16, 19 e 23.

[30] Lax, M.: *Classical noise IV: Langevin methods*. Reviews of Modern Physics, 38(3):541, 1966. Citado na página 16.

[31] Zitzler, E., . Thiele, M. Laumanns, C. M. Fonseca e V. G. Fonseca: *Performance assessment of multiobjective optimizers: An analysis and review*. IEEE Transactions on evolutionary computation, 7(2):117–132, 2003. Citado 4 vezes nas páginas 20, 25, 26 e 38.

[32] Cruz, F. R. B. e T. van Woensel: *Buffers and servers allocation in general finite queueing networks*. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, 2(1), 2014. Citado na página 28.

[33] 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 4 vezes nas páginas 28, 30, 33 e 37.

[34] Duarte, A. R.: *The server allocation problem for Markovian queueing networks*. International Journal of Services and Operations Management, 48(2):256–271, 2024. Citado na página 28.

[35] Gross, D. e C. M. Harris: *Fundamentals of Queueing Theory*. John Wiley & Sons, 2008. Citado 2 vezes nas páginas 29 e 32.

[36] MacGregor Smith, J.: *Optimal Design and Performance Modelling of $M/G/1/k$ Queueing Systems*. Mathematical and Computer Modelling, 39(9-10):1049–1081, 2004. Citado 2 vezes nas páginas 30 e 34.

[37] Kimura, T.: *A Transform-Free Approximation for the Finite Capacity $M/G/s$ Queue*. Operations Research, 44(6):984–988, 1996. Citado na página 30.

[38] MacGregor Smith, J.: *$M/G/c/k$ blocking probability models and system performance*. Performance Evaluation, 52(4):237–267, 2003. Citado 2 vezes nas páginas 30 e 34.

[39] Kleinrock, L.: *Queueing Systems. Volume I: Theory*. Wiley-Interscience, 1975. Citado 2 vezes nas páginas 30 e 37.

[40] Kerbache, L. e J. MacGregor Smith: *The generalized expansion method for open finite queueing networks*. European Journal of Operational Research, 32(3):448–461, 1987. Citado na página 30.

[41] van Woensel, T. e F. R. B. Cruz: *Optimal routing in general finite multi-server queueing networks*. PloS one, 9(7):e102075, 2014. Citado 3 vezes nas páginas 31, 32 e 38.

[42] Souza, G. L., A. R. Duarte, F. R. B. Cruz e G. Moreira: *Optimal resource allocation in networks of general single-server finite queues*. Journal of the Brazilian Computer Society (submitted paper), páginas 1–30, 2025. Citado 3 vezes nas páginas 34, 37 e 38.

[43] Cruz, F. R. B., G. Kendall, L. While, A. R. Duarte e 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 na página 34.