Universidade Federal de Ouro Preto

Instituto de Ciências Exatas e Biológicas

Programa de Pós-Graduação em Ciência da Computação
PPGCC

Dissertação

Uma nova formulação para otimização multi-objetivo em redes de filas finitas gerais e com único servidor

Gabriel Lima de Souza

Ouro Preto
2020

UFOP

Gabriel Lima de Souza

Uma Nova Formulação para Otimização Multi-objetivo em Redes de Filas Finitas Gerais e com Único Servidor

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Computação do Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Gladston Juliano Prates Moreira
Coorientador: Anderson Ribeiro Duarte

Ouro Preto
2020

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S729n Souza, Gabriel Lima de.
Uma nova formulação para otimização multi-objetivo em redes de filas finitas gerais e com único servidor. [manuscrito] / Gabriel Lima de Souza. - 2020.
59 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Gladston Juliano Prates Moreira.
Coorientador: Prof. Dr. Anderson Ribeiro Duarte.
Dissertação (Mestrado Acadêmico). Universidade Federal de Ouro Preto. Departamento de Computação. Programa de Pós-Graduação em Ciência da Computação.
Área de Concentração: Ciência da Computação.

1. Algoritmos genéticos. 2. Otimização combinatória. 3. Inteligência artificial. I. Duarte, Anderson Ribeiro. II. Moreira, Gladston Juliano Prates. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 004.8

Bibliotecário(a) Responsável: Celina Brasil Luiz - CRB6-1589

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Gabriel Lima de Souza

Uma Nova Formulação para Otimização Multi-Objetivo em Redes de Filas Finitas Gerais e com Único Servidor

Membros da banca

Prof. Dr. Gladston Juliano Prates Moreira, UFOP
Prof. Dr. Anderson Ribeiro Duarte, UFOP
Prof. Dr. Frederico Rodrigues Borges da Cruz, UFMG
Prof. Dr. Ivair Ramos Silva, UFOP

Versão final
Aprovado em 05 de agosto de 2020.

De acordo

Prof. Dr. Gladston Juliano Prates Moreira

Documento assinado eletronicamente por **Gladston Juliano Prates Moreira**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 05/08/2020, às 15:58, 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 **0072704** e o código CRC **61E4DCA1**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: 3135591692 - www.ufop.br

Agradecimentos

Encerra-se mais um ciclo. Agradeço meus pais, José Vicente e Vânia, pelo grande apoio e paciência durante essa caminhada. Vocês são essenciais.

Agradeço ao Sindicato ASSUFOP, por representar e lutar por melhorias aos Técnicos Administrativos em Educação da UFOP. Esta luta garantiu direitos fundamentais, que me proporcionou dedicar maior tempo a este trabalho.

Aos amigos do PPGCC pela companhia na sala de pesquisa, pelas dicas na construção de um código e também por várias noites de descontração pelas ladeiras da nossa querida Ouro Preto.

Aos camaradas do PCdoB, por entenderem minha ausência nos últimos meses e conduzirem com maestria e ousadia nossa participação no cenário politico atual. Seguimos sonhando por manhãs de sol e socialismo.

Agradeço a todos os professores do PPGCC. Em especial Marco, Cristiano, Alan, Gladston, Marcone, Puca e Eduardo pelo conhecimento a mim compartilhado. Um agradecimento especial a Mariana por resolver toda burocracia do nosso programa, sempre me receber com alegria e prontamente me instruir em tudo que é necessário.

Aos vários amigos que a vida me presenteou, em especial Marcelo, Helgem, Clebis e Junio que com as conversas diárias tornaram essa etapa mais tranquila e agradável. Vocês são demais.

Ao Professor Gladston Moreira, pela paciência e tranquilidade. Topou o desafio de orientar esse trabalho e nesse processo me fez crescer e aprender.

Ao Professor Frederico Cruz que acompanhou este estudo desde o início, sempre disponível, me orientando e mostrando os caminhos a seguir. Obrigado por tudo.

Ao Professor Anderson, muito mais que professor e orientador, durante toda essa caminhada virou grande amigo. Uma referência desde minha graduação e é até hoje. Sempre apresentando novos desafios e a quem sempre pude recorrer nas horas difíceis. Muito obrigado.

Resumo

Uma nova formulação de programação matemática é proposta para um problema de otimização em redes de filas. A soma das probabilidades de bloqueio de uma rede de filas acíclicas finitas de servidor único e tempo de serviço geral é minimizada juntamente com o tamanho total da área de espera e as taxas gerais de serviço. Um algoritmo genético multiobjetivo (MOGA) e um algoritmo multiobjetivo de otimização por enxame de partículas (MOPSO) é adaptado para resolver esse difícil problema estocástico. O algoritmo resultante produz um conjunto de soluções eficientes para mais de um objetivo. A implementação dos algoritmos de otimização depende do método de expansão generalizado (GEM), uma ferramenta clássica usada para avaliar o desempenho de redes de filas finitas. Um conjunto de experimentos computacionais é apresentado para evidenciar a eficácia e eficiência da abordagem proposta. As informações obtidas a partir da análise de uma rede complexa podem ajudar no planejamento desses tipos de redes de filas.

Palavras-chave: Redes de filas, Objetivos conflitantes, Algoritmo genético, Otimização por enxame de partículas, Método de expansão generalizado.

Abstract

A new mathematical programming formulation is proposed for an optimization problem in queueing networks. The sum of the blocking probabilities of a general service time, single server, finite, acyclic queueing network is minimized, as are the total buffer sizes and the overall service rates. A multi-objective genetic algorithm (MOGA) and a particle swarm optimization (MOPSO) algorithm are combined to solve this difficult stochastic problem. The derived algorithm produces a set of efficient solutions for multiple objectives in the objective function. The implementation of the optimization algorithms is dependent on the generalized expansion method (GEM), a classical tool used to evaluate the performance of finite queueing networks. A set of computational experiments is presented to attest to the efficacy and efficiency of the proposed approach. Insights obtained from the analysis of a complex network may assist in the planning of these types of queueing networks.

Keywords: Queueing networks, Conflicting objectives, Genetic algorithm, Particle swarm optimization, Generalized expansion method.

Lista de ilustrações

Figura 1 – Uma rede de filas com topologia incluindo séries, fusão e divisão composta por filas de entradas markovianas e atendimento geral. . . 3
Figura 2 – Classificação dos problemas de otimização segundo Yang (2010) [1] . 10
Figura 3 – Pontos dominados (■) e não-dominados (●) retirados de Cruz et al. (2012) [2]. 13
Figura 4 – Esquema ilustrativo de utilização do Método de Expansão Generalizado. 21
Figura 5 – Algoritmo NSGA-II 26
Figura 6 – Algoritmo PSO multi-objetivo 28
Figura 7 – Rede de filas com topologia mista (adaptada de MacGregor Smith & Cruz [3]). 31
Figura 8 – Soluções para $s^2 = 0{,}5$. 33
Figura 9 – Soluções para $s^2 = 1{,}0$. 34
Figura 10 – Soluções para $s^2 = 1{,}5$. 36

Sumário

1 INTRODUÇÃO . . . 1
1.1 Motivação . . . 1
1.2 Objetivos . . . 2
1.2.1 Objetivos Gerais . . . 2
1.2.2 Objetivos Específicos . . . 3
1.3 Contribuições . . . 3

2 FUNDAMENTAÇÃO TEÓRICA . . . 5
2.1 Teoria das Filas . . . 5
2.1.1 Notação de Kendall . . . 6
2.2 Problemas de Otimização . . . 9
2.2.1 Otimização Multi-objetivo . . . 11
2.3 Heurísticas de Otimização . . . 13
2.3.1 Algoritmo Genético NSGA-II . . . 14
2.3.2 Algoritmo por Enxame de Partículas PSO . . . 14

3 ABORDAGEM DO PROBLEMA E ASPECTOS METODOLÓGICOS 17
3.1 O Problema de Otimização . . . 17
3.1.1 Formulação Mono-objetivo . . . 17
3.1.2 Uma Possível Formulação Matemática Multi-objetivo . . . 18
3.1.3 Uma Nova Formulação Matemática . . . 19
3.2 Método de Expansão Generalizado . . . 21
3.2.1 Reconfiguração de rede . . . 21
3.2.2 Estimação de parâmetros . . . 22
3.2.3 Eliminação da retroalimentação . . . 22
3.3 Abordagem algorítmica utilizada . . . 23
3.3.1 Detalhamento do algoritmo genético NSGA-II . . . 25
3.3.2 Detalhamento do algoritmo por enxame de partículas PSO . . . 27

4 RESULTADOS EXPERIMENTAIS . . . 31
4.1 Sistema Hipoexponencial . . . 32
4.2 Sistemas Exponenciais . . . 33
4.3 Sistemas Hiperexponenciais . . . 35
4.4 Observações . . . 37

5 CONCLUSÃO . . . 39

REFERÊNCIAS **41**

1 Introdução

Problemas de otimização multi-objetivo surgem naturalmente em várias áreas do conhecimento, como engenharia, economia, logística, entre outras. A principal característica destes problemas é o conflito entre suas funções objetivo. Normalmente encontrar o ponto ótimo de qualquer problema não é uma tarefa trivial. Geralmente são problemas complexos e de solução não óbvia.

No estudo das filas, não é diferente. É comum a presença de objetivos conflitantes nos quais se busca encontrar um ponto de equilíbrio. Seja na distribuição do número de atendentes, no aumento ou diminuição do espaço de espera, ou até mesmo diminuir o tempo médio de espera na fila. A busca por esses pontos de equilíbrio (ótimo) é sempre um processo muito complexo.

1.1 Motivação

É comum verificar aborrecimentos e desconforto decorrentes de situações de atendimento em sistemas de filas. Muitas pessoas já passaram pelo incômodo de aguardar mais do que o desejado, para serem atendidas por algum estabelecimento ou servidor. Por que isso ocorre?

Sempre que se busca por determinado serviço e este seviço não é oferecido de imediato, a formação de filas para este atendimento ocorre. Em suma, toda vez que a demanda por determinado serviço é maior que a oferta de recursos para executá-lo, verifica-se a formação de filas de espera.

Uma fila é formada quando há estocasticidade no atendimento e no fluxo de chegada dos novos clientes. Esse fenômeno ocorre de forma recorrente no dia a dia, por exemplo, em filas de banco, supermercados, postos de combustível, no tráfego em uma avenida congestionada, ou até mesmo no fluxo de dados em uma rede de computadores.

A discussão de sistemas de filas remete usualmente a associações com as clássicas filas do cotidiano. Porém, existem processos que podem ser modelados como sistemas de filas que não são claramente perceptíveis em nossa rotina diária. O tráfego de dados em uma rede de computadores, mencionado anteriormente, é um dos casos. Como ocorre com as tarefas aguardando para serem processados em uma CPU [4–7], matéria prima aguardando um processo produtivo [8, 9], entre diversos outros exemplos.

Uma solução ingênua para a gestão destes sistemas seria aumentar a capacidade de atendimento. Entretanto, essa escolha, muitas vezes, não se mostra viável por razões

básicas: indisponibilidade econômica, limitação do espaço físico, entre outras. Diante deste cenário, é imprescindível discutir os sistemas de filas, investigar estes sistemas a fim de propor as melhores configurações para distribuição de recursos econômicos e físicos. O interesse dessa investigação está na busca por atender as necessidades dos gestores e também daqueles que buscam a melhor oferta possível para os seus serviços de interesse.

No estudo das redes de filas, o espaço para clientes em espera, usualmente denominado áreas de circulação (do inglês *buffer*) entre as filas do sistema e o número de servidores, são informações preponderantes para a obtenção de um balanço ótimo de custo e qualidade na prestação dos serviços na rede. Outro fator que não pode passar despercebido, é a oferta do espaço de espera. Toda fila em que a área de circulação é finita, existe a chance de, em algum momento, estar bloqueada para a chegada de novos clientes. Essa chance é denominada probabilidade de bloqueio. Dessa forma, uma configuração natural para o problema de otimização em redes de filas inclui objetivos conflitantes: por exemplo, o aumento da capacidade de atendimento implica em atender mais clientes de forma mais ágil, porém, também implica em aumento de custo para manter o serviço em funcionamento.

Problemas de otimização com este tipo de conflito remetem a uma abordagem multi-objetivo, ou seja, que observa diversos objetivos simultaneamente. Uma solução ótima do ponto de vista de abordagens multi-objetivo não é necessariamente composta pela solução ótima de cada um dos objetivos individuais. Na verdade, trata-se de uma composição conjunta de soluções eficientes em diversos objetivos simultaneamente, usualmente denominada solução Pareto-ótimo.

A avaliação de escolhas de soluções específicas em um conjunto de soluções Pareto-ótimo fica a cargo do gestor tomador de decisão. Em geral, a prioridade para a solução mais adequada, dentre todas as possíveis, é feita de acordo com sua necessidade específica do problema sob investigação. Além disso, devido aos conflitos existentes entres as diversas funções objetivo que compõe o problema, dificilmente as soluções ótimas individuais de cada função objetivo irão coincidir para formar uma solução ótima do ponto de vista multi-objetivo.

1.2 Objetivos

1.2.1 Objetivos Gerais

Este estudo tem foco em discussões sobre redes de filas com servidor único, fluxo de chegada markoviano e dinâmica de atendimento geral. Será abordada no estudo uma rede de filas complexa quanto a sua topologia, que envolva filas dispostas em série,

divisões de fluxo para duas ou mais filas, e fusões de fluxo em uma única fila. A entrada no sistema será única através da primeira fila do sistema e percorrerá o sistema através de um vetor de probabilidades de roteamento preestabelecido para situações de divisão. O interesse está em obter mecanismos para obtenção de soluções que determinem a alocação de recursos em áreas de circulação e servidores que forneçam um balanço adequado entre os custos do sistema e a eficácia dos serviços prestados.

1.2.2 Objetivos Específicos

De forma mais específica, o interesse está em formular matematicamente o problema de otimização multi-objetivo na investigação sobre redes de filas. Além disso, apresentar um método heurístico eficaz que seja capaz de otimizar simultaneamente os objetivos envolvidos na formulação proposta. Uma configuração de rede de filas com a estrutura de interesse dessa investigação pode ser vista na Figura 1. Maior clareza sobre a notação utilizada, será elucidado na subseção 2.1.1.

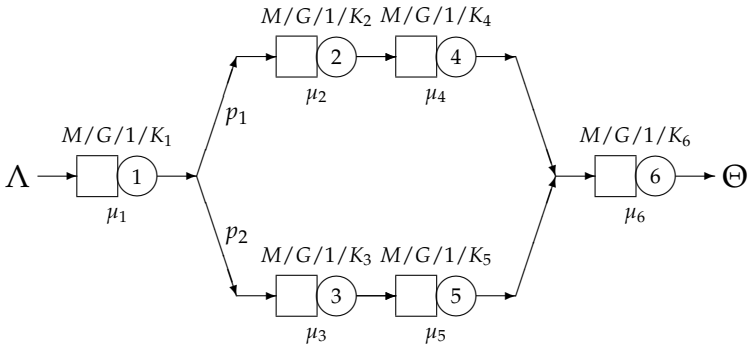


Figura 1 – Uma rede de filas com topologia incluindo séries, fusão e divisão composta por filas de entradas markovianas e atendimento geral.

1.3 Contribuições

É pertinente e de grande aplicabilidade o estudo em rede de filas. Este trabalho pretende avançar nesse estudo, deixando como principais contribuições:

- Uma revisão bibliográfica atualizada acerca das investigações em problemas de redes de filas;
- Uma nova formulação matemática multi-objetivo para o problema sob investigação;
- Um método heurístico eficiente para produzir soluções para o problema de rede de filas.

2 Fundamentação Teórica

Este capítulo objetiva apresentar conceitos associados à teoria de filas, problemas de otimização e métodos heurísticos para solucionar problemas de otimização.

2.1 Teoria das Filas

As filas fazem parte do cotidiano das pessoas. A maioria das pessoas tem na memória alguma situação marcante referente a alguma fila. A maneira pela qual a sociedade atual se organiza faz das filas algo fundamental no dia a dia. As filas são observadas nas avenidas engarrafadas com o tráfego pesado das grandes cidades, espera por atendimentos em bancos, supermercados, postos de combustível e restaurantes. Ocorre espera em filas numa infinidade de serviços e atividades.

Um fator que tende a passar despercebido para a maioria dos usuários é que, assim como os clientes, os gestores dos serviços também não se interessam pelo crescimento das filas, isto porque elas aumentam o custo de qualquer atividade e também acarreta em insatisfação para os clientes. Se as filas não interessam aos gestores nem aos clientes, então elas existem por quê?

De maneira simplista, é porque existe em certos momentos uma maior demanda para um determinado serviço que os recursos disponibilizados para ofertá-lo. Logo, toda situação em que o recurso para oferecer tal serviço é limitado e a procura pelo referido serviço é estocástica, tem-se a formação de filas de espera. Isto ocorre por uma grande gama de razões: a dificuldade de implantar novos servidores; a necessidade de maior tempo para execução do serviço; a espera da conclusão de outras etapas do processo. Em algumas situações, pode até mesmo ser economicamente inviável disponibilizar o serviço a todo tempo, isto de acordo com sua demanda. Pode também ocorrer alguma limitação do espaço físico disponibilizado para a instalação da estação de oferta do serviço em questão.

Todos os problemas mencionados, em geral, podem ser solucionados pelo aumento do valor investido em gestão e implantação do sistema de filas. Também podem ser minorados através de conhecimento determinísitico sobre a quantidade do referido serviço que poderá ser disponibilizada. Isso leva a observar o problema por outro prisma: "qual o custo de espera do usuário?"; "quantos clientes podem esperar em cada fila?"; "qual o ganho em reduzir a oferta deixando clientes em fila?". Estudos associados à teoria das filas tentam, e em muitos casos com sucesso, através da utilização de recursos e ferramentas computacionais, matemáticas e estatísticas, responder a essas e várias

outras perguntas.

Um sistema de filas pode ser descrito como: clientes que chegam e esperam pelo serviço quando não são atendidos de imediato, e ainda, saem do sistema após serem completamente atendidos quanto à sua demanda. O termo cliente é bastante usual e não implica necessariamente em um cliente humano, mas aquele que tem demanda pelo serviço. Visto de outra forma, pode ser uma máquina, um indivíduo, um veículo, uma tarefa, entre outras possibilidades.

Sistemas de filas se encontram presentes em situações de incerteza sobre o fluxo. Seja este um fluxo de produtos, usuários, tarefas, entre outros. As configurações de filas em rede, em que para cada fila tem-se uma taxa de chegada λ e uma taxa de atendimento μ, determinam uma generalização natural para diversos sistemas a serem modelados matematicamente.

São comuns as situações de limitação da área de circulação, ou seja, área de espera nas filas (do inglês *buffers*) por um determinado serviço. Uma fila com k espaços de espera é dita uma fila finita, e denota-se por P_k a probabilidade de se ter exatamente k clientes, entre os que aguardam o serviço e aqueles que já estão sendo servidos. Usualmente, afirma-se que P_k é a probabilidade de bloqueio da fila. Uma vez que um cliente chega na busca por um serviço e todos os lugares de espera estão ocupados, o cliente é dito bloqueado no sistema. Por razões óbvias, probabilidades de bloqueio altas implicam em ineficiência do sistema de filas.

2.1.1 Notação de Kendall

Uma notação descritiva para modelos de filas bastante usual e adequada é a conhecida notação de Kendall [10]. Utiliza-se para descrever uma fila, a forma resumida $A/B/m/k$ em que: A descreve a distribuição do tempo entre chegadas; B representa distribuição do tempo de serviço; m ilustra o número de canais de serviço em paralelo; e k denota o espaço total disponibilizado para os clientes em espera mais o número de clientes que estão sendo atendidos. Este esquema fica mais claro ao observar as representações na Tabela 1.

Um exemplo simples seria considerar uma fila $M/D/2/\infty$ que indica um processo de fila com tempo entre chegadas com distribuição exponencial, tempo de serviço determinístico, dois servidores em paralelo para prestar os atendimento, e sem restrição no tamanho máximo da capacidade do sistema.

Quando é omitido o termo k, na notação de Kendall, entende-se que a fila tem capacidade infinita. Por exemplo, uma fila $M/G/1$ tem chegadas com distribuição exponencial, serviço com distribuição geral, ou seja, alguma distribuição de probabilidades genérica, um único servidor, e não há limite na capacidade do sistema e o atendimento

Tabela 1 – Parametrização para filas de acordo com a notação de Kendall [10].

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

é por ordem de chegada.

Pode parecer estranho utilizar o símbolo M para a distribuição exponencial, em vez de alguma simbologia como, por exemplo, E. Essa definição evita alguma confusão com E_p, símbolo utilizado para a distribuição Erlang com p estágios. Entretanto, o fato mais marcante está na propriedade da falta de memória (do inglês *memoryless*) da distribuição exponencial conhecida como propriedade markoviana em estudos de processos estocásticos. Do termo markoviano surge o símbolo M para mencionar a distribuição exponencial.

As características de um sistema de filas são basicamente:

i) Padrão de chegadas dos clientes;

ii) Padrão de serviço dos canais de atendimento;

iii) Capacidade do sistema;

iv) Número de canais de serviço;

v) Número de estágios de serviço.

O padrão de chegadas, em situações rotineiras de qualquer fila é um processo estocástico. Dessa forma, o conhecimento da distribuição de probabilidade que rege os tempos entre as sucessivas chegadas é preponderante. É necessário também, saber se as chegadas ocorrem de forma unitária ou se podem ocorrer chegadas em blocos, como é o caso da chegada de passageiros de um vôo para utilização de esteiras de bagagens em esquemas de desembarque em aeroportos. No caso específico das chegadas em bloco, é necessário modelar a distribuição de probabilidades que descreve o tamanho dos blocos. Outro fator que merece destaque quanto ao processo de chegadas é a maneira como ele evolui com o tempo. Um processo de chegada que não se altera com o tempo, isto é, aquele cuja distribuição da probabilidade é independente do tempo, é chamado de processo de chegada estacionário.

Já, com respeito ao padrão de serviço, é também necessário descrever um modelo de distribuição de probabilidade para avaliar o tempo necessário para a execução do serviço para os clientes. O serviço também pode ocorrer de maneira unitária ou em blocos de serviços. Isto claro depende bastante da natureza do serviço que está para ser prestado. É comum pensar em clientes sendo atendidos a cada instante por um determinado servidor, porém existem casos em que vários clientes podem ser atendidos simultaneamente pelo mesmo servidor, por exemplo um computador com sistema de processamento em paralelo, ou mesmo várias pessoas sendo transportadas simultaneamente por uma embarcação ou um serviço metroviário.

Em alguns casos, o processo de serviço pode depender do número de usuários em espera. Essa situação representa processos estocásticos dependentes do estado. O padrão de serviço, assim como o padrão de chegada, pode ser estacionário ou não estacionário, em relação ao tempo.

É importante salientar a diferença entre dependência no tempo e dependência no estado. A dependência no tempo não é influenciada pelo número de clientes em espera no sistema, mas sim do instante temporal no qual o atendimento ocorrerá, ou seja, avaliar quanto tempo o sistema se encontra em operação é bastante relevante. Já a dependência no estado, não é influenciada pelo momento, mas somente pelo estado do sistema em um dado momento. Aqui entende-se por estado como a quantidade de clientes que estão no sistema naquele momento. Note, por exemplo, que um sistema de filas pode ser não estacionário e dependente do estado.

Mesmo em sistemas com taxas de serviço elevadas, é comum que clientes sejam atrasados pela espera em fila. Isso na verdade tem associação com a relação entre a taxa de serviço e a taxa de chegada dos clientes. Em geral, clientes chegam e saem em intervalos irregulares de tempo, consequentemente o comprimento da fila não tende a assumir um padrão determinístico, mas sim aleatório. A distribuição de probabilidade para o comprimento da fila será fruto de dois processos separados (chegada e saída dos clientes) que geralmente são assumidos mutuamente independentes.

Agora, quanto à capacidade do sistema, em alguns sistemas de filas existem limitações físicas quanto ao tamanho das áreas de espera pelo serviço. Caso o comprimento da fila chegue a determinado tamanho, não existe condição, espaço físico, para que novos clientes aguardem em espera, o que configura a situação de bloqueio. Neste caso, somente é permitido adicionar um novo cliente à área de espera quando se tem um novo espaço disponível fruto da conclusão do serviço de algum cliente anterior, que é uma situação comum nas conhecidas filas finitas, isto é, o espaço das áreas de circulação é restrito, colocando um limite máximo ao tamanho do sistema.

O número de canais de serviço refere-se ao número de estações de atendimento aos clientes, ou seja, o número de servidores, em paralelo, que podem atender usuários

simultaneamente. Os sistemas podem ser de um único canal ou multicanal. Estes dois sistemas diferem porque o primeiro possui uma fila única e o segundo admite uma fila para cada canal. Por exemplo, uma fila de casa lotérica normalmente é um sistema de um único canal. Os clientes aguardam em uma fila única para serem atendidos por qualquer um dos servidores. Já em uma fila de caixa de pagamentos de supermercado existem várias filas para cada um dos servidores, ou seja, um sistema de filas multicanal. Em geral, admite-se que os serviços em canais parelelos trabalham independentemente um do outro.

Por fim, para o número de estágios de serviço, têm-se sistemas de filas com apenas um estágio de serviço, como as tradicionais filas de caixas de pagamentos de supermercado, ou também sistemas mais complexos com vários estágios. Um exemplo de sistemas de vários estágios pode ser encontrado em estudos de tráfego, uma avenida com vários semáforos, ou mesmo uma planta produtiva de uma determinada indústria em que um determinado produto resultante precisa percorrer diversas etapas para ter seu processo de manufatura concluído.

Em alguns sistemas multiestágio pode ocorrer a reciclagem, comum em processos industriais de fabricação. A inspeção do controle de qualidade é feita após determinado estágio e os produtos que não atingem determinadas especificações técnicas, voltam para alguma etapa anterior do sistema para serem reprocessados. De maneira análoga, em uma rede de computadores é comum haver perda de pacotes no envio de dados, requerendo que o servidor reenvie o mesmo pacote de dados pela sequência de nós do sistema.

A otimização de sistemas de rede de filas finitas interessa a vários aspectos da vida real, com a possibilidade de ajudar a compreender e aperfeiçoar vários sistemas presentes na vida das pessoas. Potenciais usuários destes modelos de otimização baseados em redes de filas finitas incluem cientistas da computação, engenheiros de produção entre outros pesquisadores. De fato, tais modelos podem auxiliar na compreensão e melhoria de vários sistemas reais, incluindo sistemas de manufatura [8], de produção [9] e de saúde [11], sistemas de tráfego de veículos e de pedestres [12–14], sistemas de computação e de comunicação [4–6], aplicações baseadas na *web*, configuradas em camadas [7] e com requisitos de qualidade de serviço (QoS) definidos em termos de tempo de resposta, taxa de atendimento (ou *throughput*), disponibilidade e segurança [15], entre outros [16,17].

2.2 Problemas de Otimização

Estudos de otimização podem incluir uma variedade grande de problemas cujo objetivo é buscar por alguma otimalidade, algum ponto ótimo [1]. A busca por

soluções ótimas está presente em todo lugar, da engenharia ao mercado financeiro, do planejamento de férias à rotina diária. Todo negócio bem administrado busca maximizar os lucros, minimizar os gastos e até mesmo maximizar a satisfação dos seus clientes.

Por exemplo, quando se planeja um conjunto de atividades de férias, o interesse é maximizar o bem estar e/ou minimizar o custo com a viagem dos sonhos. De fato, sempre ocorre uma busca constante por soluções ótimas para os problemas. Sejam estes problemas do trabalho, do lazer ou de qualquer atividade corriqueira. Essa busca, por outro lado, nem sempre garante a obtenção de soluções para tais problemas. Não é exagero dizer que a busca por tais soluções é tão antiga quanto a existência da humanidade. O princípio do menor esforço permeia a existência humana desde os primeiros momentos, assim como um animal que quer garantir seu alimento fruto da sua caça, enquanto ao mesmo tempo pretende preservar a maior quantidade possível de energia nesse processo. As pessoas querem plantar o maior número de árvores frutíferas em um pomar utilizando ao máximo sua área de plantio. A idéia de otimização pode explicar muitos dos comportamentos humanos. Existem formas de classificação para os mais diversos problemas de otimização. Uma classificação resumida pode ser observada na Figura 2.

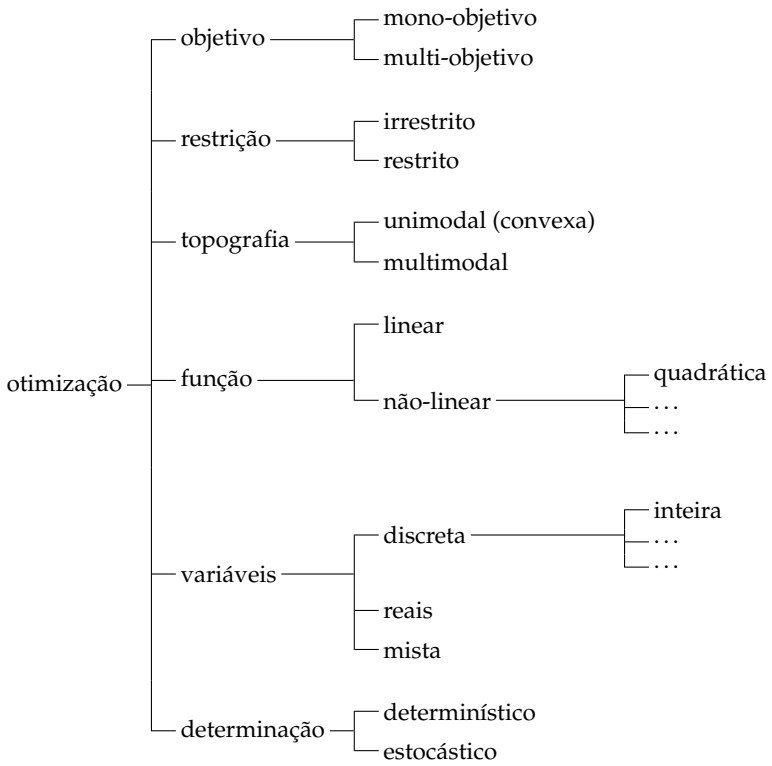


Figura 2 – Classificação dos problemas de otimização segundo Yang (2010) [1]

A discussão neste texto recairá em um problema *multi-objetivo*, *irrestrito*, *unimodal*, *não-linear geral*, *misto* e *estocástico*. Existem diversas técnicas para resolução de problemas

de otimização. Essas técnicas variam significativamente de acordo com a natureza do problema. Ocorre portanto, uma exigência cada vez maior de conhecimento do problema e de um vasto número de técnicas disponíveis. A complexidade de um problema de otimização depende, e muito, da sua função objetivo e do seu conjunto de restrições.

Otimização, por si só, pode significar muitas coisas diferentes. Um atleta de 100 metros rasos, por exemplo, deseja otimizar sua corrida desenvolvendo sua técnica e suas habilidades a fim de terminar a mesma prova em um melhor tempo. No foco deste estudo, busca-se descrever matematicamente os problema de otimização com a seguinte forma:

$$\text{minimizar } f_i(\mathbf{x}), (i = 1, 2, \ldots, I),$$

sujeito a:

$$\begin{aligned} \phi_j(\mathbf{x}) &= 0, (j = 1, 2, \ldots, J), \\ \psi_k(\mathbf{x}) &\leqslant 0, (k = 1, 2, \ldots, K), \end{aligned}$$

em que $f_i(\mathbf{x}), \phi_j(\mathbf{x})$ e $\psi_k(\mathbf{x})$ são funções do vetor de decisão $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. Os componentes x_i de $\mathbf{x}$ são varáveis de decisão que podem ser contínuas, discretas ou uma mistura das duas. As funções $f_i(\mathbf{x})$, em que $i = 1, 2, \ldots, I$, são denominadas funções objetivo. O espaço das variáveis de decisão é denominado espaço de busca, enquanto o espaço formado pelos valores das funções objetivo é denominado espaço das soluções.

As funções objetivo podem ser lineares ou não lineares. As igualdades para ϕ_j e desigualdades para ψ_k são denominadas restrições. Os problemas de otimização podem ter naturezas mono ou multi-objetivo. Na direção do presente estudo, o interesse se concentra em abordagens multi-objetivo.

2.2.1 Otimização Multi-objetivo

Muitos problemas de otimização tem a natureza do problema na direção de otimizar funções com mais de um objetivo, ou seja, o espaço de soluções tem dimensão não unitária. Os problemas chamados multi-objetivos, em geral, são mais complexos e tem como solução não um valor, mas sim um conjuto de soluções que será abordado posteriormente.

Apesar dos diversos avanços matemáticos e computacionais, resolver um problema de otimização multi-objetivo não é uma tarefa simplista. Determinar, de forma completa, o conjunto de soluções para problemas de natureza requer um grande consumo de energia no procedimento de execução dessa tarefa. De uma forma mais ge-

neralista, um problema de otimização multi-objetivo pode ser descrito da seguinte forma:

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

sujeito a:

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

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

O conceito de solução ótima para um problema multi-objetivo foi introduzido por Vilfredo Pareto em 1896 [18]. Seu interesse estava na distribuição de recursos e no bem-estar social de uma população. Pareto afirmou que a população está em situação de equilíbrio (situação ótima) quanto à distribuição de recursos, quando uma possível redistribuição destes recursos implicar simultaneamente em alguma melhora para um dos participantes e piora para algum outro membro da população.

Atualmente este conceito é empregado nas mais diversas áreas de estudo em que é necessário encontrar pontos de equilíbrio em sistemas com diversos objetivos conflitantes. A finalidade da otimização multi-objetivo é encontrar as soluções que "minimizam" $f_i(x)$ no contexto de otimização de Pareto [19].

Para melhor compreender os conceitos de soluções em avaliações multi-objetivo, o conceito de dominância é bastante relevante. Dados os elementos $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, a relação de dominância $\prec$ é definida como:

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

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

Uma solução viável (factível) $\mathbf{x}^* \in \mathcal{X}$ é chamada solução Pareto-ótimo do problema se não houver $\mathbf{x} \in \mathcal{X}$ tal que $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}^*)$. Em uma situação de avaliar um conjunto de soluções, sub-conjunto $\mathcal{X}^*$ de $\mathcal{X}$, dentre as soluções desse sub-conjunto, é possível determinar aquelas que são soluções Pareto-ótimo que diz respeito apenas ao subconjunto $\mathcal{X}^*$ de soluções avaliadas. Estas soluções usualmente são denominadas soluções não-dominadas com respeito ao conjunto $\mathcal{X}^*$. O conjunto de todas as soluções não-dominadas, com respeito a um espaço investigado, é chamado usualmente de *fronteira*.

A Figura 3 exemplifica a associação de dominância entre duas funções objetivos para minimização.

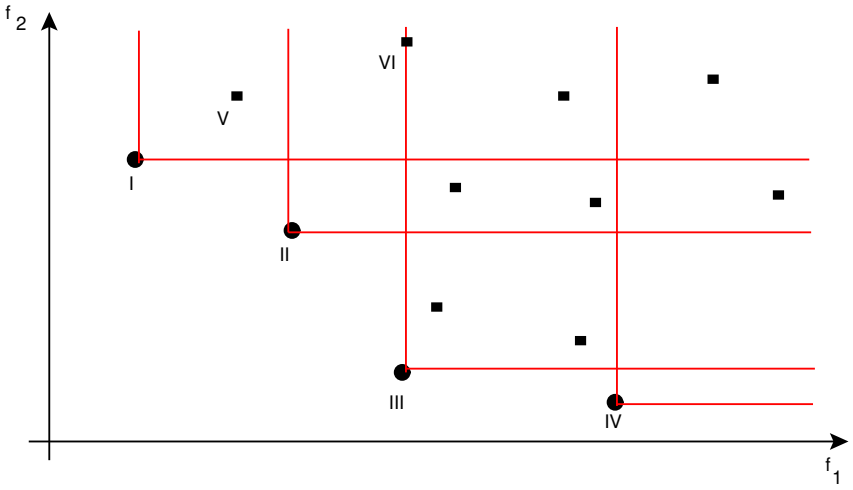


Figura 3 – Pontos dominados (▪) e não-dominados (•) retirados de Cruz et al. (2012) [2].

O ponto I, por exemplo, domina os pontos V e VI, o ponto II domina o ponto VI. Observe que o valor da função objetivo f_1 aplicada no ponto I é menor que quando aplicada em V e VI. Também ao analizar a função f_2 o valor retornado para I é menor que para os pontos V e VI. Por outro lado, para os pontos I, II, III e IV, não é possível definir uma relação hierárquica, pois para a função f_1 aplicada ao ponto I, retorna-se um valor menor que a mesma função aplicada ao ponto II, porém para a função f_2 aplicada ao ponto I o resultado é maior que em sua aplicação para o ponto II. O mesmo ocorre na comparação dos demais pontos III e IV. Portanto, os pontos I, II, III e IV são pontos não-dominados.

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

2.3 Heurísticas de Otimização

Em geral, o termo heurística está associado a arte de inventar, descobrir, ter ciência de um fato novo. Decorre do famoso termo grego *heureca* (encontrar, descobrir), termo que ficou famoso ao ser exclamado por Arquimedes quando se deu conta do fenômeno físico que hoje chamamos de empuxo.

Porém, usualmente denomina-se por heurística a habilidade de procurar a solução de algum problema por aproximação. Busca-se obter uma solução que se aproxime ao máximo da efetiva solução ótima de algum problema em estudo, solução ótima que,

em geral, é desconhecida. Em algumas situações, o processo de busca por tais soluções tende a ser exaustivo e custoso, portanto requer alguma estratégia adequada para tal busca. Procedimentos que explicitam a mecânica desse processo de busca e também critérios de parada para tal busca são usualmente denominados métodos heurísticos de otimização. Em particular, neste estudo, duas estratégias heurísticas de otimização serão abordadas: os clássicos algoritmos *Non-dominated Sorting Genetic Algorithm* (NSGA-II) [20] e *Particle Swarm Optimization Algorithm* (PSO) [21].

2.3.1 Algoritmo Genético NSGA-II

O algoritmo NSGA-II (algoritmo genético de ordenação não-dominante elitista, do inglês *Elitist Non-dominated Sorting Genetic Algorithm*) de Deb et al. [20] é composto por operadores de *seleção* e de *elitismo* que precisam estar especificamente estruturados para auxiliarem no processo de identificação de condições de otimalidade. O elitismo é baseado no conceito de dominância. Uma descrição bastante detalhada da implementação do algoritmo NSGA-II que será abordado neste estudo pode ser obtida no trabalho de Cruz et al. [2]. Porsteriormente, na sub-seção 3.3.1 será apresentado em detalhe a implementação para o NSGA-II utilizada para este estudo específico.

2.3.2 Algoritmo por Enxame de Partículas PSO

É comum na natureza várias espécies optarem por viver em conjunto e tirar proveito da sociabilidade. A vida em grupos sociais aumenta a possibilidade de acasalamento, facilita estratégias de caça e coleta de alimentos, reduz a chance de êxito em ataques de predadores, dentre vários outros benefícios. Não é a toa que os humanos vivem em sociedade e tiram proveitos disso. O termo "enxame", aqui empregado, é utilizado de forma genérica para se referir a qualquer coletivo. Um exemplo clássico é um enxame de abelhas. Porém, a metáfora pode ser estendida a outros coletivos, como uma colônia de formigas, uma revoada de pássaros.

O algoritmo PSO, sigla em inglês para *Paticle Swarm Optimization* foi proposto inicialmente por Kennedy e Eberhart em 1995 [21]. É um algoritmo de otimização inspirado no movimento de várias espécies de animais. O algoritmo busca reproduzir esse movimento de bandos, manadas ou enxames de vários animais na busca pelo seu alimento e em jornadas para regiões mais quentes ou frias. Essas atividades ocorrem sempre na busca por melhores condições para a espécie. Esse movimento é guiado pelo líder, mas também conta com a colaboração de cada um dos membros do enxame, a fim de guiar todo o enxame para uma melhor localização de bem estar da espécie.

Baseado nessas observações, surgiu o clássico algoritmo PSO que utiliza equações matemáticas para reproduzir os movimentos de enxame e fazer com que um

conjunto de pontos se movimente em busca de um posicionamento ótimo. O PSO baseia-se no conceito de cooperação. Além disso, possui as características de individualidade e sociabilidade e possui habilidades para trocar informações entre vizinhos, memorizar uma posição anterior e utilizar essas informações para melhorar a busca pelo ponto ótimo. De maneira geral, as partículas são posicionadas inicialmente de maneira aleatória no espaço de soluções factíveis. A cada iteração, sucessivos movimentos são realizados para esses pontos de acordo com as informações contidas nas próprias partículas, isso de acordo com as funções objetivo. Para movimentar cada ponto, basicamente calcula-se sua direção e o tamanho de cada movimento. Essa operação é tratada como a velocidade da uma partícula desse enxame. Porsteriormente, na sub-seção 3.3.2 será apresentado em detalhe a implementação para o PSO utilizada para este estudo específico.

3 Abordagem do Problema e Aspectos Metodológicos

O interesse deste capítulo é discutir as possibilidade de formulação matemática para o problema em investigação. Aqui é discutido o problema de otimização da utilização de recursos em redes acíclicas de filas do tipo $M/G/1/k$. Além disso, será abordada a proposição de um algoritmo de otimização específico que seja adequado para solucionar o problema de otimização em discussão.

3.1 O Problema de Otimização

Uma formulação multi-objetivo usual para esse problema de teoria de filas busca maximizar o *throughput* (Θ), ou seja, a taxa de clientes atendidos pelo sistema. E, simultaneamente, minimiza o consumo de recurso, como o espaço total de alocação em áreas de circulação ($\sum k_i$), e a soma das taxas de atendimentos ($\sum \mu_i$) dos servidores em cada fila do sistema entre outros possíveis alvos.

Um dos alvos desse estudo é propor uma nova formulação matemática para o problema que leve em conta como objetivo de otimização a soma das probabilidades de bloqueio em cada uma das filas da rede ($\sum P_{k_i}$), uma vez que este é um problema no qual a escolha do algoritmo para sua solução é bastante dependente da formulação da programação matemática. Para contextualizar essa discussão, algumas formulações mais enxutas serão inicialmente relatadas.

3.1.1 Formulação Mono-objetivo

Inicialmente será descrita uma formulação para minimizar o espaço de alocação de área de circulação (*buffers*), problema conhecido na literatura como *Buffer alocation Problem* (BAP).

O problema é definido através de um digrafo $\mathcal{D}(V, A)$ em que V é um conjunto finito de vértices (filas) e A é um conjunto finito de arestas (conexões entre as filas). O BAP, em sua formulação primal [3], é descrito da seguinte forma:

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

sujeito a:

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

minimizar a área total de circulação em uma rede com m filas, sujeito a um limiar de atendimento $\Theta_{\min}$.

Embora bastante semelhante a um problema de otimização linear inteira, $\Theta(\mathbf{K})$ é difícil de definir porque é uma função que envolve as taxas de chegada e de serviço e a topologia da rede de filas.

Existe outra formulação intimamente relacionada, uma espécie de BAP dual, que busca maximizar o *throughput*, $\Theta(\mathbf{K})$, restrito a um limite máximo para a alocação total de área de circulação ao longo da rede de filas, $K_{\max}$, descrito a seguir. Esta formulação pode ser vista claramente como uma analogia ao clássico problema da mochila estocástico [3,22], que pode ser definido da seguinte forma:

$$\text{maximizar } \Theta(\mathbf{K}), \tag{3.3}$$

sujeito a:

$$\begin{aligned} &\sum_{i=1}^{m} c_i K_i \leqslant K_{\max}, \\ &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{3.4}$$

maximizar o *throughput* , $\Theta(\mathbf{K})$, sujeito a um limite máximo, K_{max}, para distribuição da área total de circulação alocada ao longo da rede de filas.

Embora as duas formulações do BAP apresentadas possam ser usadas para auxiliar o desenvolvimento de algoritmos eficientes para solucionar problemas de rede de filas, neste trabalho consideramos um algoritmo baseado na formulação multi-objetivo.

3.1.2 Uma Possível Formulação Matemática Multi-objetivo

O problema de otimização de redes filas $M/G/1/K$, descrito, pode ser reformulado para uma versão multi-objetivo que compreende a minimização do espaço total alocado em área de circulação e a soma total das taxas de serviço, isso simultaneamente com a maximização da taxa de atendimento geral da rede, o *throughput*. Tal reformulação pode ser descrita da seguinte maneira (ver [2,23]):

$$\text{minimizar } F(\mathbf{K}, \boldsymbol{\mu}) = \Big[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu}) \Big], \tag{3.5}$$

sujeito a:

$$\begin{aligned} &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\mu_i \geqslant 0, \forall i \in \{1, 2, \ldots, m\}. \end{aligned} \tag{3.6}$$

em que $f_1(\mathbf{K})$ representa o espaço total de alocação, $f_2(\boldsymbol{\mu})$ representa a taxa de serviço total e $f_3(\mathbf{K}, \boldsymbol{\mu})$ representa o *throughput*:

$$\Big[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu})\Big] \triangleq \left[\sum_{i=1}^{m} K_i, \sum_{i=1}^{m} \mu_i, -\Theta(\mathbf{K}, \boldsymbol{\mu})\right].$$

Observe que o *throughput* (Θ) em $f_3(\mathbf{K})$ aparece com um sinal negativo, pois na prática, seria um objetivo a ser maximizado. Muitas abordagens modelam o *throughput* como uma restrição, uma desvantagem dessa abordagem é que a restrição do *throughput* deve ser relaxada. No entanto, encontrar um limite adequado não é uma tarefa trivial.

3.1.3 Uma Nova Formulação Matemática

A literatura apresenta várias formulações possíveis para o problema de otimização com base na busca por uma alocação ótima de recursos que fornece o máximo *throughput* do sistema, (Θ) [2, 13, 23–30]. Tal alocação ótima, na prática, tende a ser uma alocação que minimize a chance de ocorrências de bloqueios entre os clientes que estão sendo servidos através das filas da rede. A probabilidade de bloqueio está diretamente vinculada à taxa de atendimento do sistema através da seguinte expressão:

$$\theta = \lambda(1 - P_k) \tag{3.7}$$

indicando que minimizar a probabilidade de bloqueio está intimamente ligado a maximizar a taxa de atendimento.

A nova formulação matemática que será proposta para otimização, concentra-se nas probabilidades de bloqueio entre as filas do sistema. Essa investigação prioriza a minimização da soma das probabilidades de bloqueio no sistema, minimização do espaço total alocado em áreas de circulação e da soma das taxas de serviço entre as filas da rede. As variáveis de decisão K_i e μ_i indicam, respectivamente, o espaço de alocação e a taxa de serviço para a i-ésima fila $M/G/1/k$ do sistema. O problema de otimização em estudo pode ser formulado por:

$$\text{minimizar } F(\mathbf{K}, \boldsymbol{\mu}) = \Big[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu})\Big], \tag{3.8}$$

sujeito a:

$$\begin{aligned} &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\mu_i \geqslant 0, \forall i \in \{1, 2, \ldots, m\}. \end{aligned} \tag{3.9}$$

em que $f_1(\mathbf{K})$ representa o espaço total de alocação, $f_2(\boldsymbol{\mu})$ representa a taxa de serviço total e $f_3(\mathbf{K}, \boldsymbol{\mu})$ representa a soma das probabilidades de bloqueio:

$$[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu})] \triangleq \left(\sum_{i=1}^{m} k_i, \sum_{i=1}^{m} \mu_i, \sum_{i=1}^{m} P_{k_i} \right).$$

Em uma única fila $M/G/1/K$, a estimativa da probabilidade de bloqueio P_k, para qualquer fila da rede, pode ser obtida através de uma formula fechada computacionalmente eficiente e precisa. O método proposto por MacGregor Smith [31] é baseado em uma aproximação de dois momentos apresentada por Kimura [32]:

$$P_k = \frac{\rho^{\left(\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+2(k-1)}{2+\sqrt{\rho}s^2-\sqrt{\rho}} \right)}(\rho-1)}{\rho^{\left(2\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+(k-1)}{2+\sqrt{\rho}s^2-\sqrt{\rho}} \right)}-1}, \tag{3.10}$$

em que $\rho < 1$ representa a taxa de utilização do sistema, sendo $\rho = \lambda/\mu$, a razão entre a taxa total de chegada e a taxa de serviço e $s^2 = \mathrm{Var}(T_s)/\mathbb{E}^2(T_s)$ é o quadrado do coeficiente de variação do tempo de serviço (T_s). Vários estudos anteriores confirmam que esta aproximação de P_k é bastatnte precisa [3,13,33].

Uma fração P_k das chegadas não pode ingressar no sistema. Assim, P_k representa a probabilidade de um cliente chegar quando não há mais espaço de espera. Portanto, somente a fração $(1 - P_k)$ das chegadas pode ser atendida pela fila [34]. Investigações sobre problemas em redes de filas são abordadas por várias perspectivas [29,35–37]. Abordagens através de métodos de otimização são bastante comuns, por exemplo: o algoritmo Powell [30], algoritmos genéticos [2] e o *simulated annealing* [23] foram utilizados com sucesso. Essas abordagens usam o *througput* (Θ), que geralmente é obtido usando um método de avaliação de desempenho aproximado bem conhecido, o Método de Expansão Generalizado (GEM) [38].

O Método de Expansão Generalizado visa atualizar o cálculo das medidas de desempenho do sistema. O método leva em consideração o efeito de atraso gerado por vários possíveis bloqueios que ocorrem no fluxo de clientes ao longo das filas da rede. O Método de Expansão Generalizado resolve um conjunto de equações não lineares simultâneas através de procedimentos iterativos. Esse fato leva a algum aumento no esforço computacional para a realização dos cálculos necessários. Por outro lado, o aumento na precisão do cálculo de medidas como as verdadeiras probabilidades ao longo da rede de filas é notório.

3.2 Método de Expansão Generalizado

O Método de Expansão Generalizado é um algoritmo utilizado em muitas situações e com bastante sucesso para a estimação de medidas em desempenho de redes de filas acíclicas finitas com configurações de rede arbitrária [24]. O GEM é uma combinação da decomposição nó-a-nó e tentativas repetidas, em que cada fila é avaliada em separado para execução de correções com o intuito de contabilizar os efeitos de inter-relacionamentos entre as filas finitas da rede. O GEM considera que os bloqueios fila a fila, ocorrem se, após o serviço estar concluso em alguma fila, a fila subsequente já tem sua área de circulação completamente ocupada, ou seja, existe um cliente em serviço no servidor único da fila e além disso, todos os espaços de espera da fila se encontram preenchidos.

A Figura 4 descreve bem o GEM. É importante observar que a distribuição exponencial é uma aproximação de boa qualidade para os tempos entre saídas de clientes de uma fila $M/G/1/K$ na rede.

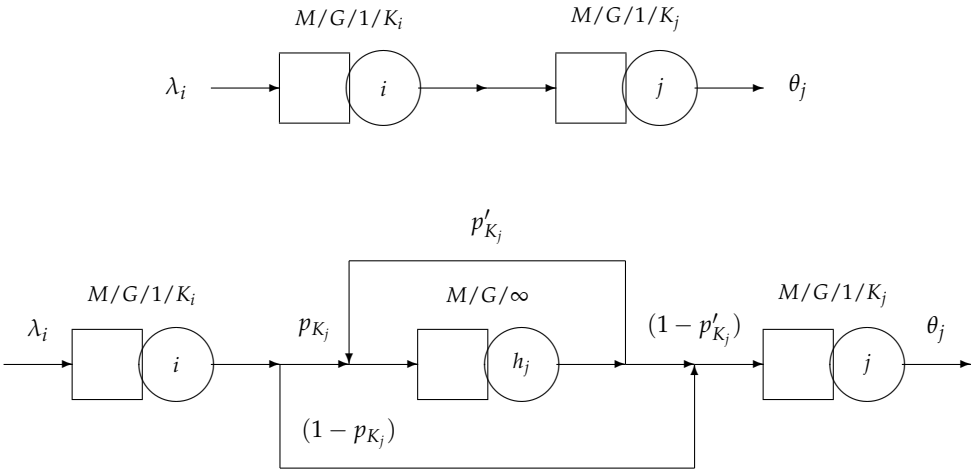


Figura 4 – Esquema ilustrativo de utilização do Método de Expansão Generalizado.

Um resultado clássico, que é a "quase-reversibilidade"pode ser obtido para uma ampla gama de filas finitas. Em particular, para as filas finitas gerais dependentes do estado, $M/G/c/c$ [39]. A considerar que clientes perdidos pelo bloqueio são incluídos, o processo de saída segue distribuição Poisson. Diversos estudos apresentam resultados empíricos nesta direção [3,9,13,27,28,40]. O GEM pode ser dividido em três estágios mais relevantes: *reconfiguração de rede*, *estimação dos parâmetros*, e *eliminação da retroalimentação*.

3.2.1 Reconfiguração de rede

Na reconfiguração de rede, um nó auxiliar h_j é criado e é modelado como uma fila $M/G/\infty$ com taxa de serviço μ_h posicionando antes de cada fila finita j. Ao deixar o

nó i, em direção ao nó seguinte, j, o cliente pode ser bloqueado com probabilidade P_{K_j}, ou desbloqueado com probabilidade $(1 - P_{K_j})$. Em condição de bloqueio, os clientes são redirecionados para o nó h_j, e passam por um período de espera, isso enquanto o nó j estiver ocupado. Posteriormente a este período de espera, o cliente pode ser novamente bloqueado, com uma probabilidade P'_{K_j}, para um segundo período de espera. O nó h_j contabiliza o tempo que um cliente deverá aguardar, até que possa, de fato, ser aceito no nó j. Além disso, contabiliza a taxa de chegada efetiva (isto é, descontado os efeitos de bloqueios) ao nó j.

3.2.2 Estimação de parâmetros

O procedimento de estimação de parâmetros tem principal interesse em estimar os valores P_K, P'_K, e a taxa μ_h (para simplificar foi omitido o subscrito referente ao nó j).

A probabilidade de bloqueio, P_K, é obtida pela aproximação descrita na equação (3.10). Já a probabilidade de ocorrer um segundo bloqueio, P'_K, é obtida por uma aproximação via técnicas de difusão, desenvolvida por Labetoulle & Pujolle [41]:

$$P'_K = \left(\frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda\Big((r_2^K - r_1^K) - (r_2^{K-1} - r_1^{K-1})\Big)}{\mu_h\Big((r_2^{K+1} - r_1^{K+1}) - (r_2^K - r_1^K)\Big)} \right)^{-1}, \tag{3.11}$$

em que r_1 e r_2 são as raízes do polinômio $\lambda - (\lambda + \mu_h + \mu_j)x + \mu_h x^2 = 0$, isso com $\lambda = \lambda_j - \lambda_h(1 - p'_K)$, em que λ_h é a taxa de chegada real ao nó artificial criado e λ_j a taxa de chegada real para o nó finito j, obtidas através da expressão $\lambda_j = \tilde{\lambda}_i(1 - p_K) = \tilde{\lambda}_i - \lambda_h$, em que $\tilde{\lambda}_i$ é a taxa de atendimento na fila antecessora.

Por fim, a taxa μ_h é obtida por meio de teoria da renovação:

$$\mu_h = \frac{2\mu_j}{1 + \sigma_j^2 \mu_j^2}, \tag{3.12}$$

em que σ_j^2 é a variância do tempo de serviço.

3.2.3 Eliminação da retroalimentação

Dadas as visitas repetidas ao nó artificial h_j, em decorrência da retroalimentação, um forte lastro de dependência no processo de chegada ao nó j é produzido. A eliminação desse efeito é obtida por um acréscimo adequado ao tempo de serviço no nó i, durante sua primeira passagem através do nó de retenção. A taxa de serviço ajustada,

para o nó h_j, μ'_h, é então:

$$\mu'_h = (1 - p'_K)\mu_h. \tag{3.13}$$

O objetivo do GEM é propiciar uma estratégia de aproximação para as taxas de serviço dos nós i, isso claro, que levem em conta o bloqueio após serviço, causados por possíveis bloqueios no nós subsequentes ao nó j:

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_K(\mu'_h)^{-1}. \tag{3.14}$$

Para cada fila finita j posterior a uma fila finita i, um conjunto de equações não lineares simultâneas para as variáveis P_K, P'_K, e μ_h, associadas com outras variáveis auxiliares, como λ e $\tilde{\lambda}_i$. A solução dessas equações de forma simultanea executadas recursivamente possibilita o cálculo das medidas de desempenho da rede de filas:

$$\begin{aligned}
\lambda &= \lambda_j - \lambda_h(1 - P'_K), && (3.15)\\
\lambda_j &= \tilde{\lambda}_i(1 - P_K), && (3.16)\\
\lambda_j &= \tilde{\lambda}_i - \lambda_h, && (3.17)\\
P'_K &= \left(\frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda\Big((r_2^K - r_1^K) - (r_2^{K-1} - r_1^{K-1})\Big)}{\mu_h\Big((r_2^{K+1} - r_1^{K+1}) - (r_2^K - r_1^K)\Big)}\right)^{-1}, && (3.18)\\
z &= (\lambda + 2\mu_h)^2 - 4\lambda\mu_h, && (3.19)\\
r_1 &= \frac{[(\lambda + 2\mu_h) - \sqrt{z}]}{2\mu_h}, && (3.20)\\
r_2 &= \frac{[(\lambda + 2\mu_h) + \sqrt{z}]}{2\mu_h}, && (3.21)\\
& && (3.22)
\end{aligned}$$

As equações (3.15) a (3.18) se referem às chegadas e também à retroalimentação do nó artificial h_j. As equações (3.19) a (3.21) são utilizadas para a resolução da Equação (3.18), em que z uma quantidade auxiliar, utilizada para simplificar o procedimento. Por fim, a aquação (3.10) fornece a probabilidade de bloqueio para a fila $M/G/1/K$. Assim, na prática são cinco equações para serem resolvidas, ou seja, as equações (3.15) a (3.18) e a também a equação (3.10).

3.3 Abordagem algorítmica utilizada

A otimização do desempenho em redes acíclicas de filas finitas com servidor único e atendimento geral é objeto de diversos estudos. A versão do problema de

otimização que considera a minimização do espaço em áreas de circulação e a soma das taxas de serviço simultaneamente com a maximização do *throughput* foi discutida anteriormente em um formato bastante interessante [23]. Uma abordagem que acoplava duas heurísticas em formatos multi-objetivo, que é uma metologia mista capaz de melhorar as soluções sem um aumento significativo nos custos coputacionais. Inicialmente o problema foi tratado através do algoritmo genético multi-objetivo em sua versão NSGA-II [2], e, posteriormente, as soluções obtidas passaram por um segundo processo de otimização, em uma estratégia de pós-processamento.

O algoritmo genético utilizado, para algumas situações, produzia um conjunto solução de Pareto abaixo do adequado. Não é possível garantir que toda combinação de alocação de capacidade em área de esperar $\mathbf{K} = (K_1, \ldots, K_m)$ foram investigados. Em outras palavras, os operadores genéticos não garantem pesquisar todas as soluções possíveis que preservam a mesma capacidade total, $\sum K_i$, e fornecer o melhor valor possível para o *throughput* (Θ). É possível existir alguma recombinação de áreas de circulação entre os valores K_i's que poderia gerar alguma melhoria para o *throughput* do sistema. Esse fato foi avaliado através de uma estratégia de pós-processamento.

A partir do conjunto inicial de soluções fornecidas pelo algoritmo genético multi-objetivo mencionado anteriormente, uma heurística de otimização que pode se adaptar bem à natureza do problema em questão é o algoritmo clássico *simulated annealing*. Uma busca exaustiva não seria viável do ponto de vista computacional, mesmo para um pequeno valor de m. O *simulated annealing* é um algoritmo inspirado no processo de recozimento de sistemas físicos usados com sucesso em muitos problemas de otimização combinatória [42,43].

Os princípios básicos do algoritmo têm origem na termodinâmica, uma analogia ao recozimento de sólidos. O *Simulated Annealing* é capaz de fornecer uma estrutura para o desenvolvimento de um algoritmo de otimização genérico capaz de escapar dos ótimos locais na busca pelo ótimo global. Desde sua introdução como um método de otimização combinatória, o *Simulated Annealing* tem sido amplamente utilizado em vários campos, como circuitos integrados para projetos auxiliados por computador, processamento de imagens, redes neurais e assim por diante. O método não utiliza uma estratégia (isto é, por exemplo, uma lei para convergência total), assumindo na maioria dos casos um mínimo ou máximo que não é global, mas que se configura como uma boa opção para resolver o problema em questão, o que pode ser visto no artigo de Spinellis et al. [44] que descreve um método para problemas dessa natureza.

De uma maneira simplista, o *Simulated Annealing* depende da função objetivo e de um critério de vizinhança entre as soluções candidatas. O objetivo é reproduzir uma cadeia de Markov cujo espaço de estado consiste em um conjunto de soluções possíveis para o problema de otimização em estudo. O *Simulated Annealing* opera da

seguinte maneira. O $i-$ésimo estado de uma cadeia de Markov é uma solução possível, então alguma solução vizinha possível para essa solução é selecionada aleatoriamente. Dado um estado vizinho escolhido, ele será o estado seguinte da cadeia de Markov se efetivamente for superior à solução anterior. Por outro lado, se ainda assim, a solução vizinha for inferior à solução corrente, a mesma poderá ser selecionada para ser o próximo estado da cadeia através de um sorteio aleatorio com probabilidade p ou a cadeia permanecerá no estado corrente com probabilidade $1 - p$. A escolha de p em geral depende do número de etapas já realizadas pela cadeia de Markov e também do aumento ou diminuição em alguma métrica associada à função objetivo.

A estratégia de pós-processamento proposta através da utilização do NSGA-II e do *Simulated Annealing* pareceu bastante adequada ao problema de otimização estocástica [23]. As soluções aprimoradas através do *Simulated Annealing* utilizam exatamente a mesma capacidade total de áreas de espera que as soluções produzidas pelo algoritmo genético NSGA-II. Como o algoritmo de pós-processamento apenas redistribui essa área total ao longo das filas do sistema, não são esperados custos adicionais ao sistema, mas a possibiliade de melhoria no desempenho do sistema.

O sucesso obtido nessa investigação de pós-processamento motivou a proposição de uma estratégia mais flexível para o algoritmo de pós-processamento [45]. Uma estratégia que não investiga somente soluções com apenas uma redistribuição de áreas de circulação ao longo das filas do sistema. Dessa forma, um algoritmo de otimização robusto que busque melhorias em soluções já conhecidas possa atuar de forma eficiente. Uma proposta bastante interessante para este propósito seria acoplar as soluções fornecidas pelo algoritmo genético NSGA-II ao clássico algoritmo *Paticle Swarm Optimization* (PSO) proposto inicialmente por Kennedy e Eberhart [21].

A seguir, serão descritos os algoritmos de otimização multi-objetivo que serão utilizados. Como já mencionado anteriormente, serão utilizados: um algoritmo genético (MOGA, do inglês, *Multi-objective Genetic Algorithm*); e um algoritmo por enxame de partículas (MOPSO, do inglês *Multi-objective Particle Swarm Optimization*), para solucionar o problema de otimização da rede de filas $M/G/1/K$. O MOGA em questão parte da clássica formulação NSGA-II (*Elitist Non-dominated Sorting Genetic Algorithm*) proposta por Deb et al. [20]. Já o MOPSO proposto deriva do algoritmo PSO multi-objetivo proposto por Coello-Coello & Lechuga [46].

3.3.1 Detalhamento do algoritmo genético NSGA-II

Na aplicação dos algoritmos genéticos multi-objetivos para problemas de otimização multi-objetivo, operadores de *seleção* e *elitismo* precisam ser definidos de acordo com a estrutura do problema, visando identificar adequadamente as condições de otimalidade. O elitismo é baseado no conceito de dominância. Uma versão resumida do

algoritmo utilizado pode ser verificada na Figura 5.

```
algoritmo
  P_1 ← GeraPopulaçãoInicial(popSize)
  para i = 1 até numGen faça
    /* gera filhos por cruzamento e mutação */
    Q_i ← FaçaPopulaçãoNova(P_i)
    /* combina pais e filhos */
    R_i ← P_i ∪ Q_i
    /* encontre fronteiras não-dominadas F = (F_1, F_2, ...) */
    F ← OrdenaçãoNãoDominante(R_i)
    /* encontre nova população pela distância de aglomeração*/
    P_{i+1} ← GeraPopulaçãoNova(R_i)
  fim para
  P_{numGen+1} ← ExtraiPareto(P_numGen)
  escreva P_{numGen+1}
fim algoritmo
```

Figura 5 – Algoritmo NSGA-II

Para a seleção de pontos não dominados, isso em cada fronteira não-dominada $(\mathcal{F}_1, \mathcal{F}_2, \dots)$, a seleção é executada até que o número de indivíduos necessários para a próxima iteração é obtida. Se, após a adição de um grupo de indivíduos da fronteira $\mathcal{F}_i$, o número máximo de indivíduos for excedido, algumas ações deverão ser tomadas, incluindo o cálculo de uma medida de diversidade, (*crowding distance* [20]), para garantir uma maior diversidade na população. Para iterações futuras apenas os pontos com a maior *crowding distance* são preservados.

Os operadores de cruzamento e mutação dependem fortemente da aplicação em estudo. Para o problema de *design* de rede de filas, o mecanismo *cruzamento uniforme* [47] foi selecionado devido à sua eficiência na identificação, herança e proteção de genes comuns, além de recombinar genes não comuns [48, 49]. O cruzamento uniforme é realizado para cada variável com uma certa probabilidade (`rateCro`). O operador de cruzamento uniforme usado no algoritmo é o *operador de cruzamento binário simulado* (SBX) [50, 51], que é muito conveniente para MOGA's com código real. A partir de pais $(x_{i,(\bullet,t)})$, filhos $(x_{i,(\bullet,t+1)})$ são obtidos de acordo com as equações 3.23 e 3.24. Observe que o operador foi projetado para criar soluções que possuem um poder de pesquisa semelhante a um cruzamento de ponto único de GA's com código binário [50]:

$$x_{i,(1,t+1)} = 0.5\Big[(1+\beta)x_{i,(1,t)} + (1-\beta)x_{i,(2,t)}\Big], \tag{3.23}$$

$$x_{i,(2,t+1)} = 0.5\Big[(1-\beta)x_{i,(1,t)} + (1+\beta)x_{i,(2,t)}\Big], \tag{3.24}$$

em que a função densidade de probabilidade descrita pela Eq. 3.25 modela a variável aleatória β,

$$f(\beta) = \begin{cases} 0.5(\eta+1)\beta^{\eta}, & \text{se } \beta \leq 1, \\ 0.5(\eta+1)\frac{1}{\beta^{\eta+2}}, & \text{caso contrário,} \end{cases} \tag{3.25}$$

Diversos valores (β) podem ser gerados por um ajuste η para produzir filhos que são mais (η pequeno) ou menos (η elevado) semelhantes aos pais.

Já o esquema de mutação ocorre com uma probabilidade específica (`rateMut`) para cada gene individual (ou seja, para cada variável de decisão K_i ou μ_i), em que perturbações gaussianas são adicionadas às variáveis de decisão [50], ou seja, $k_i + \varepsilon_i$ e $\mu_i + \varepsilon_{N+i}$, para todos os $i \in N$, com $\varepsilon_i \sim \mathcal{N}(0,1)$, $i \in \{1, 2, \ldots, 2N\}$.

Finalmente, para garantir a viabilidade das restrições da formulação matemática, após o cruzamento e a mutação, os valores das variáveis inteiras devem ser arredondados e todas as variáveis reajustadas aplicando operadores de reflexão da seguinte maneira:

$$k_{\text{rfl}_i} = k_{\text{lowlim}} + |k_i - k_{\text{lowlim}}|, \tag{3.26}$$

e

$$\mu_{\text{rfl}_i} = \mu_{\text{lowlim}_i} + |\mu_i - \mu_{\text{lowlim}_i}|, \tag{3.27}$$

em que k_{lowlim} é o limite inferior da alocação total, ou seja, $k_{\text{lowlim}} = 1$ e μ_{lowlim_i} é o limite inferior de alocação de serviço (para garantir que $\rho < 1$ seja atendido). Observe que k_i e μ_i são os valores resultantes após cruzamento e mutação, e k_{rfl_i} e μ_{rfl_i} são os resultados após a reflexão. O esquema proposto sempre gera soluções viáveis sem evitar ou favorecer qualquer solução específica.

3.3.2 Detalhamento do algoritmo por enxame de partículas PSO

O funcionamento adequado de uma proposta de implementação do algoritmo PSO passa inicialmente pela boa definição do que representará cada partícula na formulação matemática do problema sob investigação. Aqui, cada partícula deve representar uma solução possível para a alocação dos recursos (espaço de área de circulação

e taxas de serviço) que otimizam a rede de filas em estudo. Portanto, nesta formulação específica, cada partícula pode ser representada pela ℓ-upla $(x_1, x_2, \ldots, x_\ell) = (K_1, K_2, \ldots, K_m, \mu_1, \mu_2, \ldots, \mu_m)$, com $\ell = 2m$.

É importante destacar aqui que o problema de otimização multi-objetivo que está sendo tratado é um problema misto, que envolve números reais e inteiros. Assim, uma estratégia de ajuste de partículas deve ser definida. De fato, mudanças nas capacidades são realizadas e, em seguida, valores inteiros são usados, pois $K_i \geqslant 1$ é sempre respeitado. Da mesma forma, as restrições associadas às taxas de serviço também são respeitadas, pois é necessário garantir que $\rho < 1$. De outra forma, a taxa de chegada da fila deve ser estritamente menor que a taxa de serviço μ. Essas considerações garantem a viabilidade das soluções investigadas.

A abordagem MOPSO proposta para o problema de otimização da rede de filas segue basicamente a execução descrita na Figura 6.

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

Figura 6 – Algoritmo PSO multi-objetivo

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

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

Os parâmetros do algoritmo MOPSO foram definidos da seguinte forma: r_1 e r_2 são números aleatórios positivos com distribuição uniforme pertencente ao intervalo $[0, 1]$, $w(t)$ é o peso da inércia. O peso da inércia foi definido $w(t) = 0{,}4$. O MOPSO aqui descrito, é uma adaptação da implementação clássica apresentada por Coello-Coello & Lechunga [46]. No entanto, versões simplificadas podem ser encontradas [52] e versões mais sofisticadas e aprimoradas [53,54] também, incluindo formulações de programação matemática com números reais e inteiros [55].

Na formulação multi-objetivo, a posição da i-ésima partícula no espaço d-dimensional de busca é representada por $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Já a velocidade da referida partícula é representada por $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. A melhor posição da i-ésima partícula durante as buscas é dada por $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})$. A velocidade e a posição das partículas são atualizadas da iteração t para a iteração $t + 1$ conforme as equações:

$$v_i^{t+1} = w^t + r_1(p_i - x_i^t) + r_2(g_i - x_i^t), \tag{3.28}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}; \tag{3.29}$$

A escolha da melhor posição da i-ésima partícula (p_i) é feita a cada iteração, da seguinte forma: se a nova posição é superior (em termos de dominância no conceito multi-objetivo) à posição p_i, a mesma é atualizada pela nova posição $x_i(t + 1)$. Se a posição atual é inferior (dominada) pela posição p_i, a posição p_i é mantida. Caso p_i não seja superior ou inferior (pertencer a mesma classe em termos de dominância no conceito multi-objetivo) à posição atual $x_i(t + 1)$, a escolha é feita de maneira aleatória entre p_i e $x_i(t + 1)$. A melhor posição global (g_i) é escolhida aleatoriamente a cada iteração entre as partículas não dominadas.

4 Resultados Experimentais

O algoritmo de otimização discutido anteriormente foi implementado em FORTRAN. O ambiente de execução para realização dos experimentos computacionais foi um Intel (R) Core (TM) i3-2310M 2,10GHz, executando Windows 10 Pro 64 bits, com 6,00 GB de memória RAM.

A rede complexa de filas, apresentada na Figura 7, foi adaptada da literatura [3] e analisada com o método proposto. Esta rede é bastante adequada para os experimentos em estudo, isso porque inclui as situações topológicas de fusão, divisão e série. Foram analisados três valores distintos para os quadrados do coeficiente de variação $s^2 = \{0{,}5; 1{,}0; 1{,}5\}$ para caracterizar sistemas que são hipoexponenciais, exponenciais (markovianos) e hiperexponenciais, respectivamente. A taxa de chegada no sistema de filas foi sempre fixada em $\Lambda = 5{,}0$. Para fins de comparação, os experimentos correspondem aos anteriormente realizados por Cruz et al. [23].

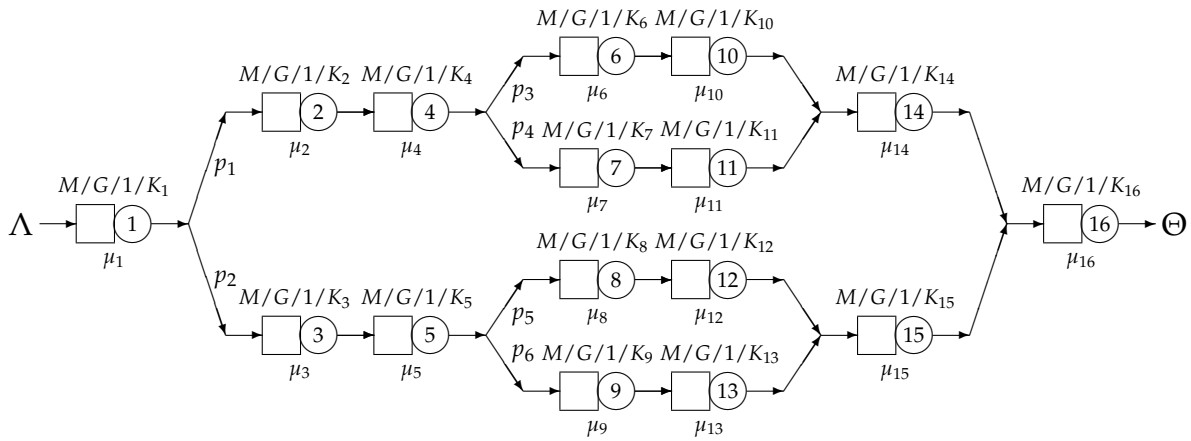


Figura 7 – Rede de filas com topologia mista (adaptada de MacGregor Smith & Cruz [3]).

Os experimentos que serão apresentados aqui contemplam a formulação matemática do problema descrita na Eq. 3.8 sujeito às restrições da Eq. 3.9. Os experimentos executados buscam configurações ótimas para a distribuição dos recursos de servidores e áreas de circulação na rede descrita na Figura 7. As probabilidades de roteamento p_i foram fixadas em 0,5 para todas as situações de divisão no sistema em todos os experimentos.

O tamanho da população no MOGA e no MOPSO foi definido como 400, e o número máximo de interações do algoritmo foi definido como 4000. Para examinar as soluções com mais detalhes, as Figuras 8, 9 e 10 apresentam os resultados obtidos para as três variações de valores para os quadrados do coeficiente de variação investigados.

Em cada uma dessas figuras, são apresentadas soluções subótimas para: (a) espaço tridimensional fornecido pelo NSGA-II, (b) espaço tridimensional fornecido pelo algoritmo MOPSO e (c) pontos no espaço bidimensional $\sum_i K_i \times \sum_i \mu_i$. No gráfico que representa a projeção, são mostrados os pontos obtidos pelo NSGA-II, assim como os pontos obtidos pelo pós-processamento do MOPSO. A formulação matemática utilizada é a descrita na Eq. 3.8 sujeito às restrições da Eq. 3.9.

4.1 Sistema Hipoexponencial

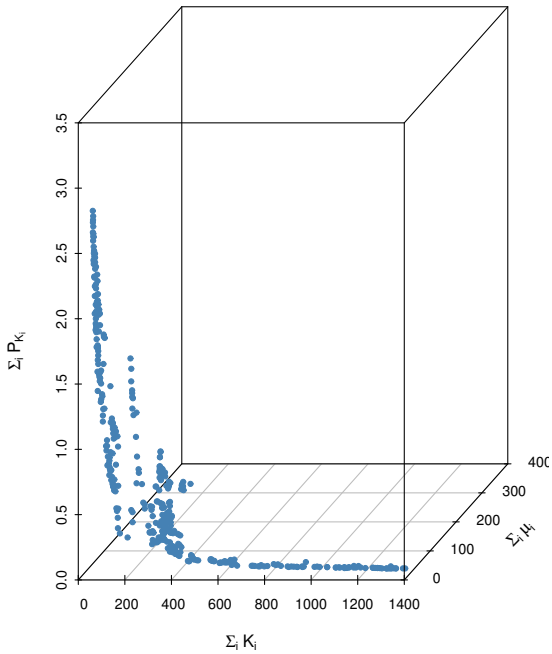
Os resultados do sistema hipoexponencial podem ser visualizados na Figura 8. Detalhes para o sistema hipoexponencial com $s^2 = 0{,}5$ podem ser visualizados através da superfície final no espaço bidimensional $\sum_i K_i \times \sum_i \mu_i$, após a convergência. Essa projeção deixa uma falsa impressão da existência de pontos dominados, mas, de fato, todos são pontos não dominados no espaço tridimensional. Fica possível verificar que o comportamento de uma determinada rede de filas não pode ser previsto sem o uso de uma abordagem algorítmica específica, como a que foi proposta nesta investigação.

Uma análise detalhada dos resultados da Figura 8 revelou que diversos pares diferentes de capacidades de alocação e taxas de serviço podem ser selecionados para uma determinada soma de probabilidades de bloqueio. É possível verificar que um grande volume de soluções candidatas foram investigadas através da estratégia de pós-processamento que foi proposta através do algoritmo MOPSO para as soluções previamente oriundas do algoritmo genético NSGA-II.

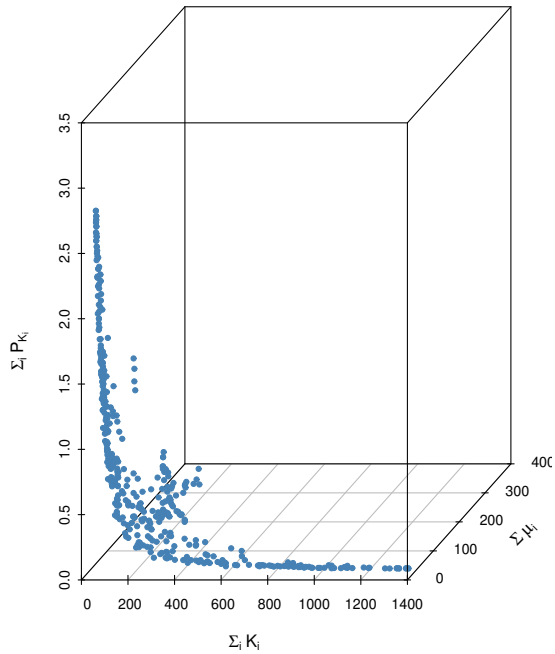
A projeção bidimensional do espaço $\sum_i K_i \times \sum_i \mu_i$, mostrada na Figura 8 - (c), resultante da abordagem NSGA-II e MOPSO, revela várias soluções com um baixo nível geral para alocação de serviços, mas que ainda são eficientes para resolver o problema sob investigação. Mesmo com alocação de baixa capacidade, o algoritmo pode produzir soluções promissoras.

É possível observar um grande número de soluções com alocação de capacidade entre aproximadamente 50 e 150, enquanto a alocação geral de serviço é inferior a 250, melhor do que algumas soluções fornecidas anteriormente pelo NSGA-II. A Figura 8 - (c) ilustra que essas soluções possuem uma soma aceitável de probabilidades de bloqueio para resolver o problema. Essa análise confirma que, para sistemas hipoexponenciais, a abordagem proposta é capaz de oferecer muitas soluções eficientes.

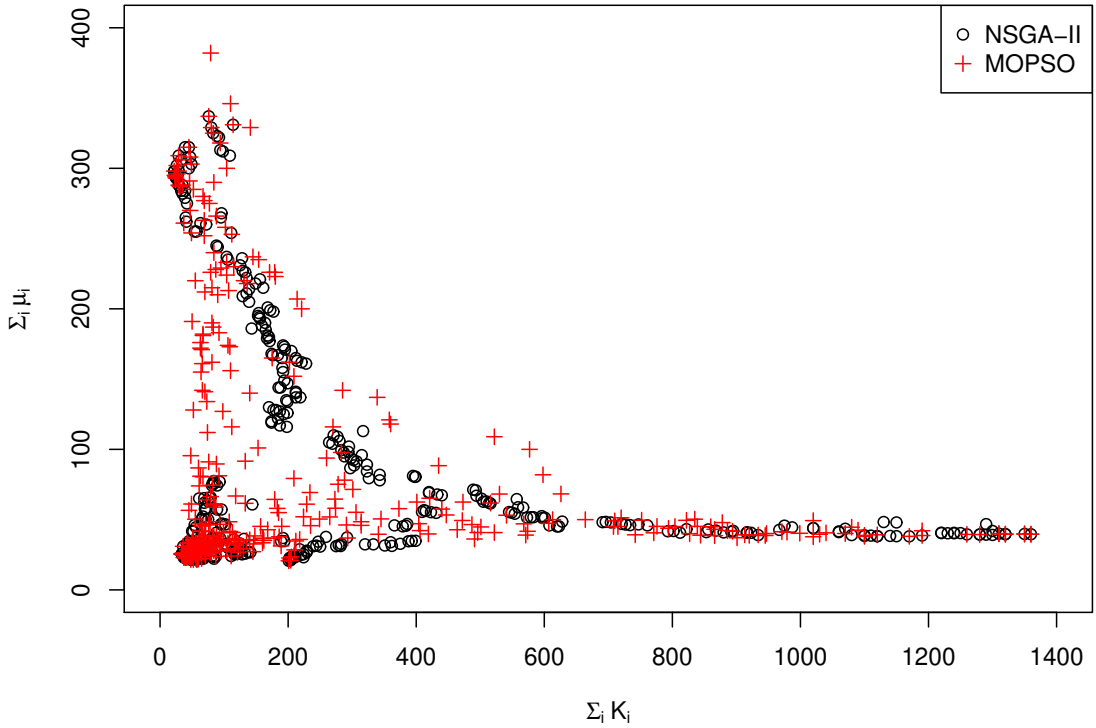
Vale ressaltar que o algoritmo MOPSO aparentemente alcançou uma distribuição muito melhor dos pontos e, ainda, melhorou sobremaneira a representação da superfície de Pareto usando seus pontos discretos, para que soluções mais diversas estejam disponíveis para escolha do agente tomador de decisão.



(a) Pareto-ótimo NSGA-II



(b) Pareto-ótimo após MOPSO



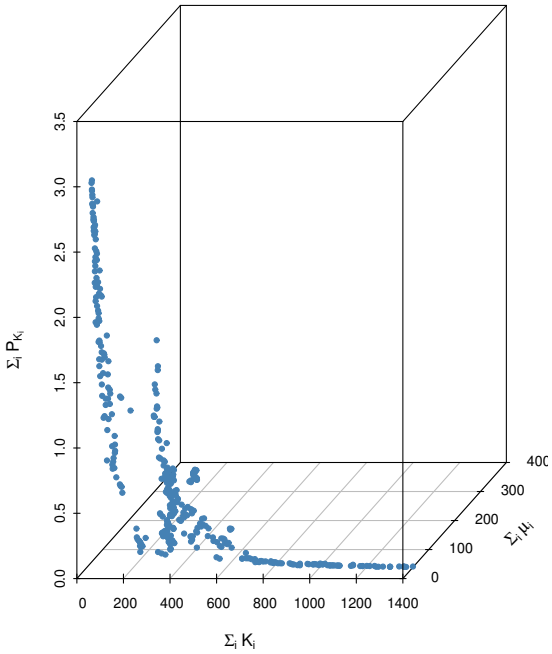
(c) Projeção bidimensional das soluções

Figura 8 – Soluções para $s^2 = 0{,}5$.

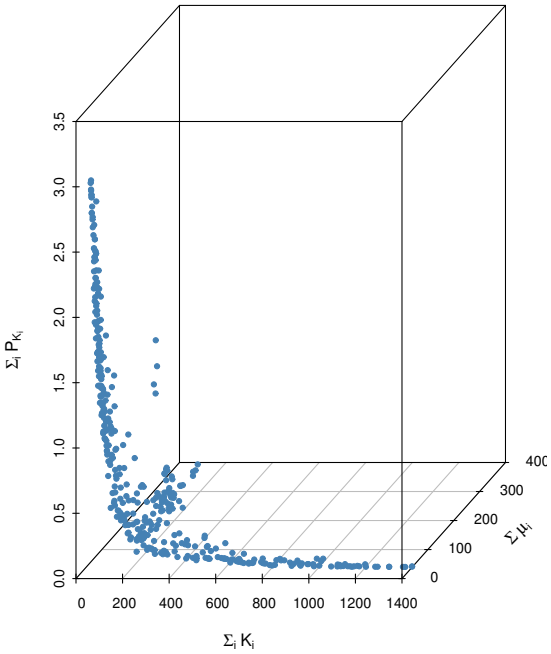
4.2 Sistemas Exponenciais

Os resultados para sistemas com quadrado do coeficiente de variação unitário ($s^2 = 1,0$), ou seja, sistemas exponenciais são mostrados através da Figura 9. No contexto em análise aqui, tem-se que o quadrado da taxa média do tempo de atendimento é

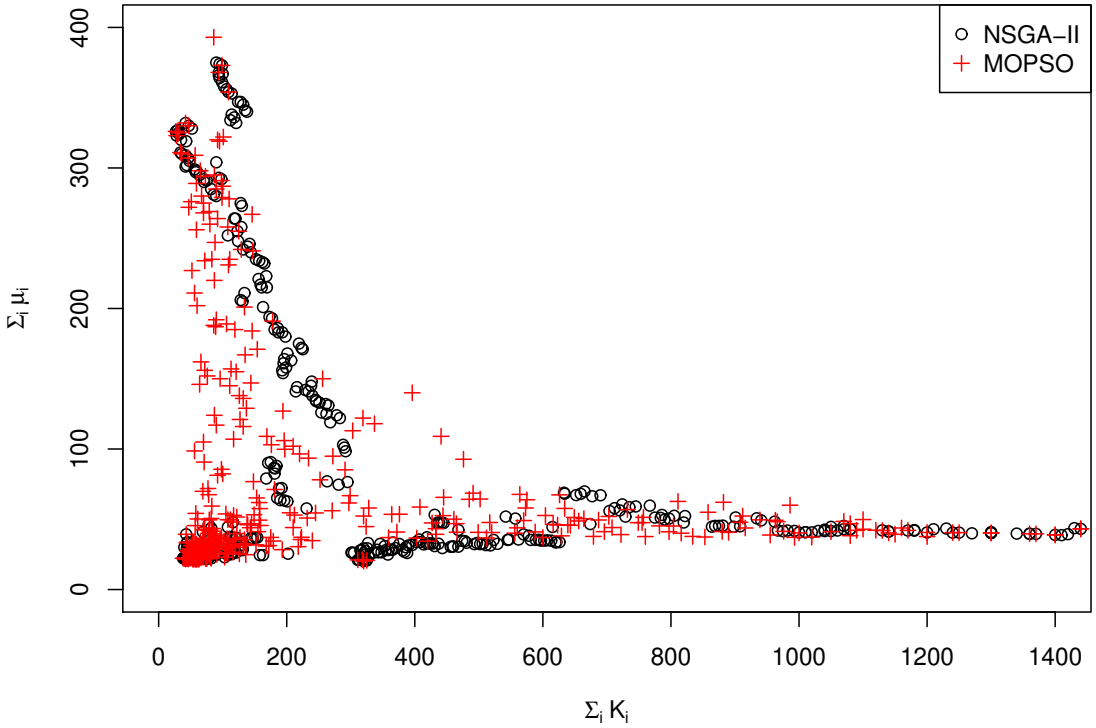
igual à variância do tempo de atendimento. Visto de outra forma, trata-se de filas do tipo $M/M/1/k$, ou seja, com atendimentos markovianos, com tempos puramente exponenciais.



(a) Pareto-ótimo NSGA-II



(b) Pareto-ótimo após MOPSO



(c) Projeção bidimensional das soluções

Figura 9 – Soluções para $s^2 = 1{,}0$.

Novamente, a análise mostra que vários pares diferentes de taxas e capacidades de serviço podem ser escolhidos para uma determinada soma de probabilidades de

bloqueio. O pós-processamento do MOPSO produziu soluções mais amplamente difundidas na região em comparação com as soluções produzidas anteriormente apenas pelo NSGA-II.

A projeção bidimensional do espaço $\sum K_i \times \sum \mu_i$, Figura 9 - (c), do NSGA-II e MOPSO, novamente mostra muitos pontos semelhantes para sistemas exponenciais. A abordagem do MOPSO conseguiu representar melhor a superfície de Pareto, utilizando pontos discretos mais representativos, distribuídos de forma mais ampla.

É notável que o MOPSO usa soluções fornecidas pelo NSGA-II com uma alocação geral de serviço acima de 200 e as substitui por outras soluções quando há uma alocação geral de serviço mais baixa, mesmo preservando a alocação de capacidade das soluções anteriores.

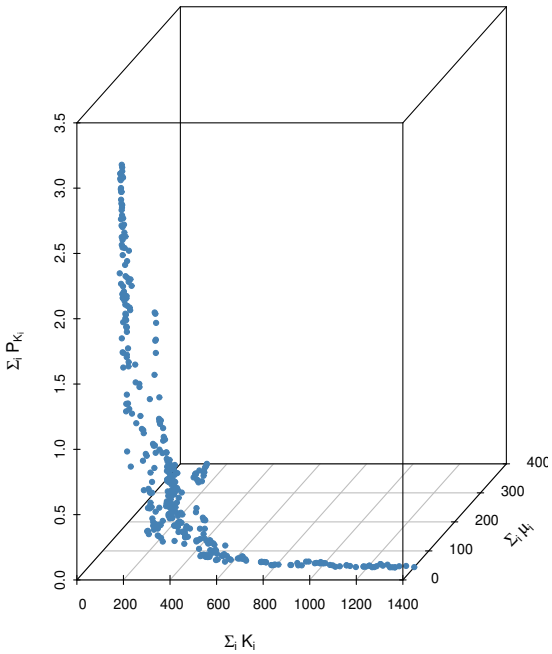
Neste caso, trata-se de soluções fornecidas pelo pós-processamento através do algoritmo MOPSO. Essas soluções são claramente menos dispendiosas em comparação com as soluções anteriores obtidas através do NSGA-II, ao mesmo tempo em que fornecem soluções eficazes em relação ao objetivo de minização das probabilidades de bloqueio.

4.3 Sistemas Hiperexponenciais

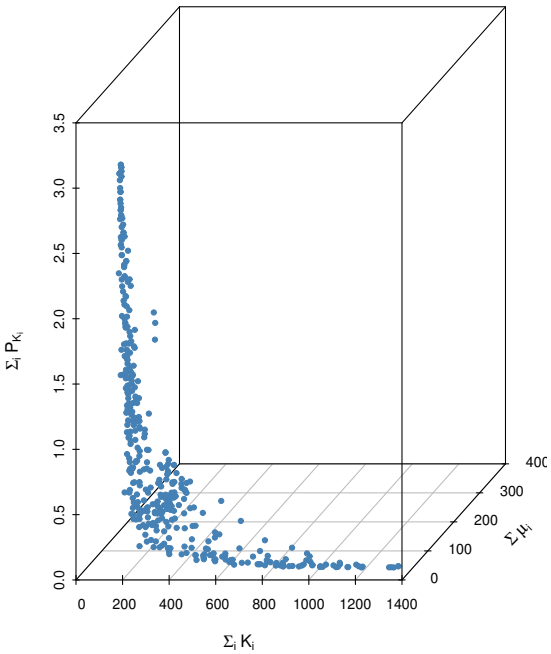
Finalmente são apresentados os resultados dos sistemas hiperexponenciais ($s^2 = 1,5$), destacados na Figura 10. Aqui, com o quadrado do coeficiente de variação superior a unidade, fica caracterizado um sistema cujos tempos de atendimento possuem uma variabilidade maior que nos casos de sistemas com tempo de atendimento markoviano.

Mais uma vez, a análise dos resultados da Figura 10 mostra uma grande diversidade de pares distintos de capacidades totais de espaço de espera e de somas totais de taxas de serviço que podem ser alcançadas para uma determinada soma de probabilidades de bloqueio. Esse achado parece muito importante em abordagens multi-objetivos.

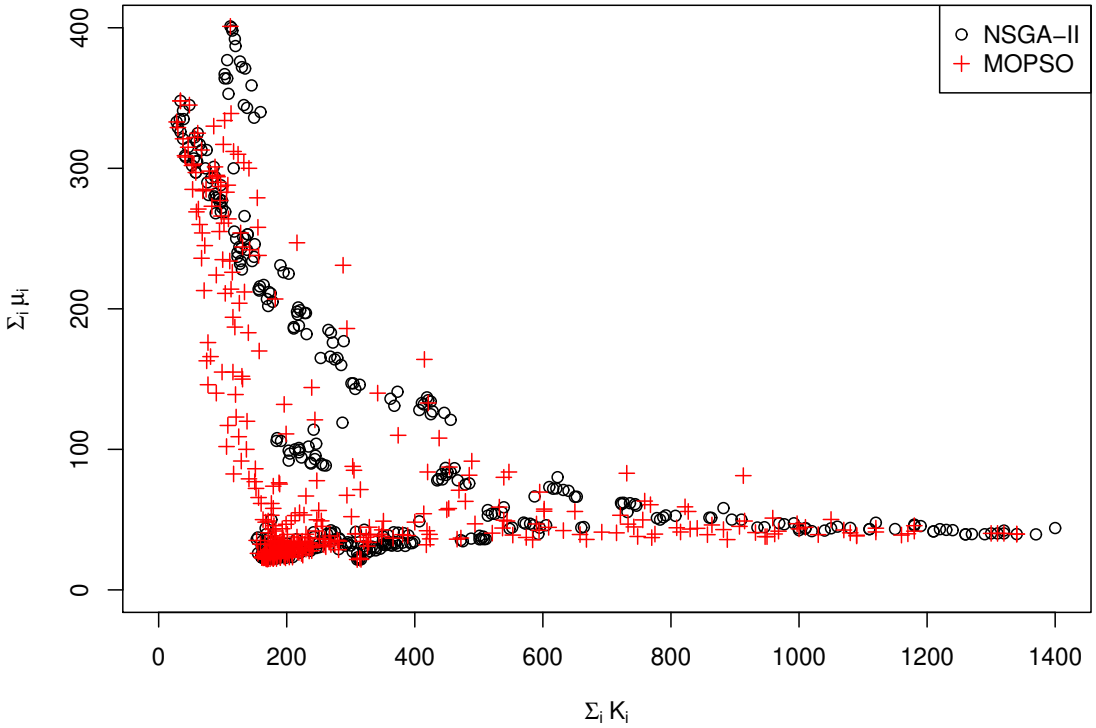
A projeção bidimensional representada para o espaço $\sum K_i \times \sum \mu_i$, Figura 10 - (c), através dos algoritmos NSGA-II e MOPSO, novamente mostra bastante similaridade entre os dois conjuntos de pontos sob análise, semelhante aos vistos nos sistemas hipoexponencial e exponencial anteriormente avaliados. A abordagem MOPSO se mostrou capaz de alcançar muitas das soluções que foram obtidas anteriormente através do algoritmo NSGA-II. O pós-processamento realizado através do algoritmo MOPSO também identifica novas soluções com alocação de capacidade entre 70 e 150, mas com alocação geral de serviço notavelmente menor do que as soluções anteriores fornecidas pelo NSGA-II.



(a) Pareto-ótimo NSGA-II



(b) Pareto-ótimo após MOPSO



(c) Projeção bidimensional das soluções

Figura 10 – Soluções para $s^2 = 1{,}5$.

Muitas das soluções fornecidas pelo pós-processamento através do algoritmo MOPSO aparentemente apresentam um menor consumo de recursos em capacidade de áreas de espera e também na soma total de taxas de serviço quando comparadas com as soluções anteriores obtidas através do NSGA-II. Isso ocorre ao mesmo tempo em que são fornecidas soluções eficazes em relação ao objetivo de minização das probabilidades

de bloqueio.

4.4 Observações

Dois fatores bastante específicos impactam na eficiência computacional do algoritmo na obtenção de soluções para o problema de otimização de rede de filas em questão. Primeiro, a escolha do algoritmo MOPSO como estratégia de pós-processamento para os resultados obtidos através do algoritmo NSGA-II e, segundo, a nova formulação de programação matemática que susbstitui um dos objetivos, aquele associado ao *througput* (Θ), pela soma soma das probabilidade de bloqueio em cada uma das filas do sistema ($\sum P_{k_i}$).

A probabilidade de bloqueio parece ser uma medida de desempenho que faz mais sentido, dado o comportamento do sistema que está sendo modelado, porque mostra a proporção de itens que não são atendidos pela respectiva fila. Para que as comparações fossem adequadas, o mesmo tamanho de população no NSGA-II também foi usado como número de partículas no enxame para o algoritmo MOPSO. Além disso, o número de iterações escolhido para o algoritmo MOPSO foi idêntico ao número de gerações usadas com o algoritmo NSGA-II.

Dentre os três quadrados dos coeficientes de variação analisados $s^2 = \{0{,}5; 1{,}0; 1{,}5\}$, para descrever sistemas, hipoexponenciais, exponenciais e hiperexponenciais respectivamente, os tempos de CPU do NSGA-II foram de $527,0$; $529,4$ e $539,8$ segundos. Já os tempos de CPU do algoritmo MOPSO foram de $478,1$; $475,5$ e $495,95$ segundos, respectivamente. Esse resultado confirma que a nova abordagem fornece soluções eficientes em um período de tempo bastante razoável.

5 Conclusão

Foi proposta uma nova formulação matemática para o problema de otimização de redes de filas finitas gerais com servidor único. Na proposta o alvo central é minimizar a soma das probabilidades de bloqueio entre as filas de uma rede de filas acíclicas, com distribuição geral do tempo de serviço e um único servidor, que é indroduzido como um dos objetivos, em uma formulação multi-objetivo. Além disso, o problema contou com outros objetivos: a minimização da capacidade total em área de espera do sistema e a soma das taxas de serviço dentre todas as filas do sistema.

A combinação dos algoritmos genético (NSGA-II) e de otimização por enxame de partículas (MOPSO), ambos em sua formulação multi-objetivo, foi desenvolvida e implementada para o sistema de filas proposto. Foram obtidas superfícies de Pareto com soluções bastante promissoras. Nesta nova abordagem, o uso da ferramenta clássica para avaliar de maneira aproximada o desempenho de redes de filas finitas, o método da expansão generalizada (GEM), auxiliou profundamente para a obtenção das soluções eficientes com um alto grau de fidelidade.

Foram obtidas soluções de Pareto para uma configuração de redes de filas bastante complexa, com diferentes topologias (envolvendo séries, fusões e divisões). Através das soluções obtidas foi possível identificar melhor como se relacionam as grandezas otimizadas. Assim, foram identificados aqueles pontos a partir dos quais não é mais compensador aumentar as áreas de espera, ou as taxas de serviço, pois o ganho proporcionado na eficiência global do sistema pareceu insignificante. Estes resultados devem produzir impactos significativos em inúmeras situações de interesse prático.

Em relação ao tempo de processamento, a nova abordagem é comparável à abordagem anterior. Assim, experimentos futuros podem ser realizados para avaliar a flexibilidade dessa abordagem em relação ao número de interações e tamanho do enxame, seu uso em diferentes redes de filas, com diferentes topologias e variações nos números de nós, taxas de chegada e variabilidade do tempo de serviço.

As modificações na formulação matemática desse problema de otimização estocástica nas redes de filas e a mudança na heurística de otimização aplicada trouxeram melhorias para a área. A combinação do NSGA-II e do MOPSO foi bem sucedida. A nova formulação matemática, que faz uso das atualizações do GEM, possibilitou a produção de uma abordagem de otimização com bom desempenho.

Investigações futuras devem ser executadas para determinar a aplicabilidade dessa abordagem na determinação de outras condições ideais nas redes de filas. Por exemplo, esse método pode ser aplicado para otimizar redes gerais de filas finitas com

vários servidores. Além disso, pesquisas futuras devem ser conduzidas para avaliar os algoritmos em situações da vida real.

Referências

[1] Yang, X S: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley Publishing, 1st edição, 2010, ISBN 0470582464, 9780470582466. Citado 3 vezes nas páginas 11, 9, and 10.

[2] 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 7 vezes nas páginas 11, 13, 14, 18, 19, 20, and 24.

[3] 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 6 vezes nas páginas 11, 17, 18, 20, 21, and 31.

[4] Ahmed, N e X Ouyang: *Suboptimal red feedback control for buffered tcp flow dynamics in computer network*. Mathematical Problems in Engineering, 2007, 2007. Citado 2 vezes nas páginas 1 and 9.

[5] Chen, J, C Hu e Z Ji: *An improved ARED algorithm for congestion control of network transmission*. Mathematical Problems in Engineering, 2010, 2010. Citado 2 vezes nas páginas 1 and 9.

[6] Inzillo, V, F De Rango e A A Quintana: *A self clocked fair queuing mac approach limiting deafness and round robin issues in directional manet*. Em *2019 Wireless Days (WD)*, páginas 1–6. IEEE, 2019. Citado 2 vezes nas páginas 1 and 9.

[7] Chaudhuri, K, A Kothari, R Pendavingh, R Swaminathan, R Tarjan e Y Zhou: *Server allocation algorithms for tiered systems*. Algorithmica, 48(2):129–146, 2007. Citado 2 vezes nas páginas 1 and 9.

[8] Alves, F S Q, H C Yehia, L A C Pedrosa, F R B Cruz e L Kerbache: *Upper bounds on performance measures of heterogeneous $M/M/c$ queues*. Mathematical Problems in Engineering, 2011(Article ID 702834):18 pages, 2011. Citado 2 vezes nas páginas 1 and 9.

[9] MacGregor Smith, J, F R B Cruz e T van Woensel: *Topological network design of general, finite, multi-server queueing networks*. European Journal of Operational Research, 201(2):427–441, 2010. Citado 3 vezes nas páginas 1, 9, and 21.

[10] Kendall, D G: *Stochastic processes occurring in the theory of queues and their analysis by the method of embedded Markov chains*. Annals Mathematical Statistics, 24:338–354, 1953. Citado 2 vezes nas páginas 6 and 7.

[11] De Bruin, A M, A C Van Rossum, M C Visser e G M Koole: *Modeling the emergency cardiac in-patient flow: an application of queuing theory*. Health Care Management Science, 10(2):125–137, 2007. Citado na página 9.

[12] Cruz, F R B, J MacGregor Smith e R O Medeiros: *An $M/G/C/C$ state dependent network simulation model*. Computers & Operations Research, 32(4):919–941, 2005. Citado na página 9.

[13] 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 4 vezes nas páginas 9, 19, 20, and 21.

[14] Cruz, F R B, T Van Woensel, J MacGregor Smith e K Lieckens: *On the system optimum of traffic assignment in m/g/c/c state-dependent queueing networks*. European Journal of Operational Research, 201(1):183–193, 2010. Citado na página 9.

[15] Menascé, D A: *Qos issues in web services*. IEEE Internet Computing, 6(6):72–75, 2002. Citado na página 9.

[16] Osorio, C e M Bierlaire: *An analytic finite capacity queueing network model capturing the propagation of congestion and blocking*. European Journal of Operational Research, 196(3):996–1007, 2009. Citado na página 9.

[17] Dimitriou, I e C Langaris: *A repairable queueing model with two-phase service, start-up times and retrial customers*. Computers and Operations Research, 37(7):1181–1190, 2010. Citado na página 9.

[18] Pareto, V: *Cours d'économie politique*, volume 1. Librairie Droz, 1896. Citado na página 12.

[19] Moreira, G. e L. Paquete: *Guiding under uniformity measure in the decision space*. Em *Proceedings of the 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, páginas 1–6, 2019. Citado na página 12.

[20] Deb, K, A Pratap, S Agarwal e T Meyarivan: *A fast and elitist multiobjective genetic algorithm: Nsga-ii*. IEEE transactions on evolutionary computation, 6(2):182–197, 2002. Citado 3 vezes nas páginas 14, 25, and 26.

[21] Kennedy, J. e R. Eberhart: *Particle swarm optimization*. Em *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, páginas 1942–1948, 1995. Citado 2 vezes nas páginas 14 and 25.

[22] MacGregor Smith, J e N Chikhale: *Buffer allocation for a class of nonlinear stochastic knapsack problems*. Annals of operations research, 58(5):323–360, 1995. Citado na página 18.

[23] Cruz, F R B, A R Duarte e G L Souza: *Multi-objective performance improvements of general finite single-server queueing networks*. Journal of Heuristics, 24(5):757–781, 2018. Citado 6 vezes nas páginas 18, 19, 20, 24, 25, and 31.

[24] Kerbache, L e J MacGregor Smith: *Multi-objective routing within large scale facilities using open finite queueing networks*. European Journal of Operational Research, 121(1):105–123, 2000. Citado 2 vezes nas páginas 19 and 21.

[25] Cruz, F R B: *Optimizing the throughput, service rate, and buffer allocation in finite queueing networks*. Electronic Notes in Discrete Mathematics, 35:163–168, 2009. Citado na página 19.

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

[27] Andriansyah, R, T van Woensel, F R B Cruz e L Duczmal: *Performance optimization of open zero-buffer multi-server queueing networks*. Computers & Operations Research, 37(8):1472–1487, 2010. Citado 2 vezes nas páginas 19 and 21.

[28] Cruz, F R B, T van Woensel e J MacGregor Smith: *Buffer and throughput trade-offs in $M/G/1/K$ queueing networks: A bi-criteria approach*. International Journal of Production Economics, 125(2):224–234, 2010, ISSN 0925-5273. Citado 2 vezes nas páginas 19 and 21.

[29] van Woensel, T e F R B Cruz: *Optimal routing in general finite multi-server queueing networks*. PLoS ONE, 9(7):e102075, julho 2014. http://dx.doi.org/10.1371%2Fjournal.pone.0102075. Citado 2 vezes nas páginas 19 and 20.

[30] Martins, H S R, F R B Cruz, A R Duarte e F L P Oliveira: *Modeling and optimization of buffers and servers in finite queueing networks*. OPSEARCH, 56(1):123–150, 2019. Citado 2 vezes nas páginas 19 and 20.

[31] 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 na página 20.

[32] Kimura, T: *A transform-free approximation for the finite capacity $M/G/s$ queue*. Operations Research, 44(6):984–988, 1996. Citado na página 20.

[33] MacGregor Smith, J: *M/G/c/K blocking probability models and system performance*. Performance Evaluation, 52(4):237–267, 2003. Citado na página 20.

[34] Gross, D, J F Shortle, Thompson J M e Harris C M: *Fundamentals of Queueing Theory*. Wiley - Interscience, New York, NY, 4th edition edição, 2009. Citado na página 20.

[35] Chowdhury, S e S P Mukherjee: *Estimation of traffic intensity based on queue length in a single m/m/1 queue*. Communications in Statistics - Theory and Methods, 42(13):2376–2390, 2013. Citado na página 20.

[36] Qi, D, Z Li, X Zi e Z Wang: *Weighted likelihood ratio chart for statistical monitoring of queueing systems*. Quality Technology & Quantitative Management, 14(1):19–30, 2017. Citado na página 20.

[37] Cruz, F R B, R C Quinino e L L Ho: *Control charts for traffic intensity monitoring of Markovian multiserver queues*. Quality and Reliability Engineering International, 36(1):354–364, 2020. Citado na página 20.

[38] Kerbache, L e J MacGregor Smith: *The generalized expansion method for open finite queueing networks*. European Journal of Operational Research, 32:448–461, 1987. Citado na página 20.

[39] Cheah, J Y e J MacGregor Smith: *Generalized m/g/c/c state dependent queueing models and pedestrian traffic flows*. Queueing Systems, 15(1-4):365–386, 1994. Citado na página 21.

[40] Cruz, F R B, P C Oliveira e L Duczmal: *State-dependent stochastic mobility model in mobile communication networks*. Simulation Modelling Practice and Theory, 18(3):348–365, 2010. Citado na página 21.

[41] Labetoulle, J e G Pujolle: *Isolation method in a network of queues*. IEEE Transactions on Software Engineering, 4:373–381, 1980. Citado na página 22.

[42] Kirkpatrick, S, C D Gelatt e M P Vecchi: *Optimization by simmulated annealing*. Science, 220(4598):671–680, 1983. Citado na página 24.

[43] Černỳ, V: *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*. Journal of Optimization Theory and Applications, 45(1):41–51, 1985. Citado na página 24.

[44] Spinellis, D, C Papadopoulos e J M Smith: *Large production line optimization using simulated annealing*. International journal of production research, 38(3):509–541, 2000. Citado na página 24.

[45] Souza, G L, A R Duarte, G J P Moreira e F R B Cruz: *A novel formulation for multi-objective optimization of general finite single-server queueing networks*. Em *Proceedings of the 2020 Congress on Evolutionary Computation. CEC'20*, páginas 1–8. IEEE, 2020. Citado na página 25.

[46] Coello Coello, C. A. e M. S. Lechuga: *MOPSO: A proposal for multiple objective particle swarm optimization*. Em *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, páginas 1051–1056, 2002. Citado 2 vezes nas páginas 25 and 29.

[47] Bäck, T, D B Fogel e Z Michalewicz: *Handbook of evolutionary computation*. Release, 97(1):B1, 1997. Citado na página 26.

[48] Syswerda, G: *Uniform crossover in genetic algorithms*. Em *Proceedings of the 3rd international conference on genetic algorithms*, páginas 2–9, 1989. Citado na página 26.

[49] Hu, X B e E Di Paolo: *An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem*. Em *Multi-objective memetic algorithms*, páginas 71–89. Springer, 2009. Citado na página 26.

[50] Deb, K e R B Agrawal: *Simulated binary crossover for continuous search space*. Complex systems, 9(2):115–148, 1995. Citado 2 vezes nas páginas 26 and 27.

[51] Deb, K e H G Beyer: *Self-adaptive genetic algorithms with simulated binary crossover*. Evolutionary computation, 9(2):197–221, 2001. Citado na página 26.

[52] Trivedi, V, P Varshney e M Ramteke: *A simplified multi-objective particle swarm optimization algorithm*. Swarm Intelligence, 14:83–116, 2020. Citado na página 29.

[53] Fan, Z, T Wang, Z Cheng, G Li e F Gu: *An improved multiobjective particle swarm optimization algorithm using minimum distance of point to line*. Shock and Vibration, 2017:1–16, 2017. Citado na página 29.

[54] Jia, C e H Zhu: *An improved multiobjective particle swarm optimization based on culture algorithms*. Algorithms, 10(2):46, 2017. Citado na página 29.

[55] Zhao, X, Y Jin, H Ji, J Geng, X Liang e R Jin: *An improved mixed-integer multi-objective particle swarm optimization and its application in antenna array design*. Em *2013 5th IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, páginas 412–415. IEEE, 2013. Citado na página 29.