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

Alocação Ideal de Recursos em Sistemas de Filas M/G/1/K

Denis Costa da Silva

Ouro Preto-MG

Fevereiro de 2021

Denis Costa da Silva

Alocação Ideal de Recursos em Sistemas de Filas M/G/1/K

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

Orientador

Dr. Anderson Ribeiro Duarte

UNIVERSIDADE FEDERAL DE OURO PRETO – UFOP
DEPARTAMENTO DE ESTATÍSTICA – DEEST

Ouro Preto-MG

Fevereiro de 2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S586a Silva, Denis Costa da .
Alocação ideal de recursos em sistemas de filasM/G/1/K. [manuscrito] / Denis Costa da Silva. - 2021.
93 f.: il.: color., gráf..

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

1. Otimização matemática. 2. Programação não-linear. 3. Teoria das filas. I. Duarte, Anderson Ribeiro. II. Universidade Federal de Ouro Preto. III. Título.

CDU 311

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 ESTATISTICA

FOLHA DE APROVAÇÃO

Denis Costa da Silva

Alocação Ideal de Recursos em Sistemas de Filas M/G/1/K

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

Aprovada em 10 de Fevereiro de 2021

Membros da banca

Doutor em Estatística Anderson Ribeiro Duarte - Orientador(a) (UFOP)
Doutor em Estatística Helgem de Souza Martins - (UFOP)
Mestre e doutorando em Ciência da Computação Gabriel Lima de Souza - (UFOP)

Anderson Ribeiro Duarte, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 10/02/2021

Documento assinado eletronicamente por **Anderson Ribeiro Duarte**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 10/02/2021, às 16:49, 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 **0134741** e o código CRC **8FF232D3**.

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

Dedico este trabalho aos meus pais, incentivadores desta jornada. À Luíza, minha amada esposa, que me apoiou, motivou e perseverou durante essa longa caminhada.

E aos meus filhos de quatro patas que tornaram tudo mais divertido.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela oportunidade de conquistar os meus objetivos durante este longo período de estudos.

À UFOP e ao Departamento de Estatística pela paciência e por proporcionar a conclusão deste sonho. A todos os professores do DEEST, em especial meu estimado orientador Dr. Anderson Ribeiro Duarte, que no decorrer da minha trajetória me permitiu aprender, sempre compartilhando os seus conhecimentos com excelência e dedicação, durante todos os anos de nossa vivência.

Aos meus queridos amigos e companheiros na batalha durante o curso de Estatística, em especial o Philipe, Wellington, Marcos, Pedro, Willian, Gabriela e Leilayne; e todos que contribuíram de alguma forma nesta caminhada. As repúblicas que convivi e levei muitos “pontos” felizes para minha vida. A Gerencianet por acreditar e confiar no meu potencial! .

LISTA DE FIGURAS

RESUMO

A discussão do problema de otimização em redes de filas finitas é desafiadora como um problema de investigação científica. O interesse particular deste estudo foi comparar soluções produzidas em diferentes contextos de pesquisa. O objetivo é obter soluções que ajustem a capacidade de atendimento dos servidores e a quantidade de áreas de espera (do inglês, buffer). No entanto, soluções capazes de atender aos requisitos de desempenho do sistema, a taxa de saída (do inglês, throughput) ou probabilidades de bloqueios nas filas da rede de filas. É comum investigar sistemas com disponibilidade limitada de recursos. Isso diz respeito à capacidade de trabalho dos servidores e também ao espaço total do buffer. O custo total envolvido no processo é bastante afetado por esses impactos financeiros. O objetivo principal foi estudar conceitos de otimização e enfatizar, principalmente, a otimização em sistemas de filas. Os sistemas de filas estão presentes em diversos cenários: o fluxo de tráfego em grandes centros urbanos, serviços telefônicos de atendimento (call-center), serviços de emergência médica, atendimentos de serviços públicos, entre outros. A metodologia empregada nesse trabalho foi utilizada para generalizar o estudo de diversos sistemas de filas em situações reais. As conclusões apresentadas, foram obtidas através da análise de várias redes e podem auxiliar aos profissionais da área no planejamento de redes de filas gerais.

Palavras-chave: Otimização; Programação não-linear; Teoria de Filas; Áreas de circulação; Capacidade de servidores.

ABSTRACT

The discussion of the optimization problem in finite queueing networks is challenging as a scientific investigation problem. The particular interest of this study was to compare solutions produced in different search contexts. The objective is to obtain solutions that adjust the service capacity and the number of buffers. However, solutions capable of meeting system performance requirements, such as throughput, or blocking queuing probabilities in the queuing network. It is common to investigate systems with limited availability of resources. This concerning the work capacity of the servers and also for the total buffer space. The total cost involved in the process is greatly affected by these financial impacts. The main objective was to study optimization concepts and to emphasize, mainly, the optimization in queueing systems. Queueing systems are present in several scenarios: the traffic flow in large urban centers, call-centers, emergency medical services, public service calls, among others. The methodology used in this work was used to generalize the study of several queueing systems in real situations. The conclusions presented were obtained through the analysis of several networks and can assist professionals in the area in planning general queueing networks.

Keywords: Optimization; Non-linear programming; Queueing theory; Buffers; Server capacity.

SUMÁRIO

1 INTRODUÇÃO

Os problemas de otimização estão presentes em nosso cotidiano em inúmeras situações. Entretanto, segundo Yang (2010), este fato não torna a proposição de solução de tais problemas algo trivial. Diversos problemas de interpretação bastante simplista podem ser de solução bastante difícil. Um exemplo clássico é o conhecidíssimo problema do caixeiro viajante, em que um vendedor precisa visitar, por exemplo, 30 cidades, uma única vez, em uma ordenação de visitas tal que a distância total percorrida seja a menor possível. Aparentemente trata-se de um problema de definição simples e fácil, assim como sua compreensão e simplicidade do objetivo a ser minimizado. Por outro lado, é de certa forma surpreendente que não se conheça ainda um algoritmo eficiente para ele.

Estudos mais recentes, criados ao longo das últimas décadas para o problema do caixeiro viajante tendem a usar estratégias metaheurísticas. As mais atualizadas e modernas técnicas de otimizadoras são usualmente heurísticas ou metaheurísticas, como por exemplo o *Simulated Annealing* (SA), a otimização por enxame de partículas (PSO), a busca harmônica e os algoritmos genéticos (GA). São algoritmos muito poderosos para a resolução de problemas de otimização sofisticados, em todas as principais áreas da ciência e da engenharia, bem como nas aplicações industriais (CRUZ, 2009; CRUZ et al., 2012; VAN WOENSEL; CRUZ, 2014; CRUZ; DUARTE; SOUZA, 2018; AZIMI; ASADOLLAHI, 2019; MARTINS et al., 2019; SOUZA et al., 2020).

A incerteza sobre o fluxo de produtos, usuários, mensagens, entre outros, com uma taxa de chegada, e seu processamento, com uma taxa de serviço, tem-se como resultado um sistema de filas. Sua concepção em uma configuração em rede, é uma generalização bastante natural e relevante, pelos diversos sistemas reais que pode modelar.

Uma clara compreensão de sistemas de filas é extremamente relevante na melhoria das investigações acerca de suas aplicações. Processos de Markov estão

entre os métodos estocásticos que são tradicionalmente usados para avaliação de desempenho de sistemas multiestados (YOUSSEF; ELMARAGHY, 2008). De forma mais específica, o objetivo está em discutir a formulação matemática para o problema de otimização multiobjetivo na investigação sobre redes de filas. Além disso, comparar soluções de métodos heurísticos eficazes já discutidos na literatura. Trata-se de abordar métodos que sejam capazes 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.

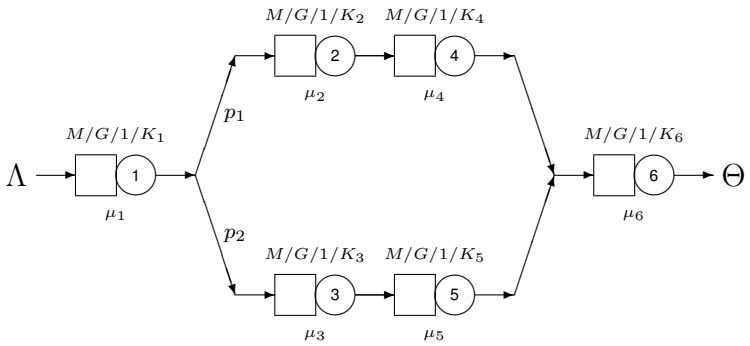


Figura 1: Uma rede de filas com topologia incluindo séries, fusão e divisão composta por filas de entradas markovianas e atendimento geral, adaptada de MacGregor Smith e Cruz (2005).

Seguindo a notação de Kendall (1953) a investigação que será discutida aborda redes de filas do tipo $M/G/1/K$. Para essas filas, há apenas um tipo de tarefa a ser executada entre as chegadas ao sistema. Tais chegadas ocorrem de acordo com um processo de Poisson a uma taxa λ. Um único servidor desempenha as tarefas e a quantidade de tempo dedicada a cada trabalho em cada servidor segue distribuição geral.

A disciplina de funcionamento das filas sob investigação é (FIFO - do idioma inglês *first in first out*) o primeiro a chegar é primeiro a ser servido. Em outras palavras, as tarefas que chegarem primeiro ao sistema serão atendidos primeiro. No que diz respeito à alocação do servidor, uma possibilidade é que o primeiro trabalho na fila seja alocado assim que qualquer servidor ficar ocioso.

A utilização de resultados de Teoria das Filas auxilia na proposição de modelos analíticos de processos que resultam em espera em fila e posteriormente na obtenção de resultados ou medidas de desempenho para explicar sobre a operacionalidade ou a produtividade dos processos.

Os modelos matemáticos são de extrema valia nos processos decisórios. Eles possibilitam o dimensionamento adequado de infraestrutura, recursos humanos e/ou financeiros. Tudo isso, com interesse em atender enfoques muitas vezes conflitantes, como são os dos usuários dos sistemas, que buscam tempos de permanência reduzidos, em contrapartida dos gerenciadores, que procuram maximizar o volume de serviços prestados.

A prestação de serviço, seja ela executada por indivíduos e/ou por máquinas, requer investimentos significativos. Existem custos operacional e de manutenção envolvidos nas tarefas. Dessa forma, a oferta de serviços além da necessidade provoca custos que devem ser evitados. Por outro lado, uma oferta de serviços abaixo dos necessários, leva a perda de clientes, o que pode acarretar a falência do sistema no que tange a sua funcionalidade.

Um sistema adequadamente dimensionado leva à manutenção do equilíbrio entre o capital disponível para o sistema em contrapartida dos retornos sociais e financeiros do mesmo, ou seja, um balanço econômico entre o custo do serviço e o custo associado à espera pelo serviço.

A solução de problemas dessa natureza está diretamente vinculada ao desempenho da rede de filas para a qual deseja-se obter a solução ótima e a configuração do número de servidores e tamanho das áreas de espera em cada nó de uma rede de filas vista como um grafo $\mathcal{G}(V, A, P)$, em que V é o conjunto de todos os vértices que compõe a rede, A é o conjunto de arcos que interconectam pares de nós e P são as respectivas probabilidades de roteamento entre os arcos.

O correto balanceamento dos recursos do sistema pode ser obtido pelo trata-

mento estocástico de informações referentes a tempos entre chegadas de clientes ao sistema e tempos de execução de serviços para estes clientes.

As observações físicas do sistema sob investigação complementam as informações necessárias para aplicações da Teoria das Filas, número de servidores em paralelo (atendimento simultâneo), existência ou não de limitações físicas para a formação da fila e disciplina de atendimento adotada.

A utilização da Teoria de filas requer algumas simplificações acerca da realidade. Algumas suposições básicas devem ser estabelecidas, como por exemplo admitir que o sistema esteja vazio no instante inicial de sua operação, a necessidade de os clientes chegarem separadamente, ou seja, não ocorrem chegadas em blocos, mesmo que apenas por pequenos intervalos de tempo, chegadas e saídas serem estatisticamente independentes e taxa de chegadas ao sistema constante. Estas suposições são preponderantes para garantir a aplicabilidade da teoria.

A otimização de sistemas de rede de filas finitos interessa a vários aspectos da vida real. Com a possibilidade de ajudar a compreender a aperfeiçoar vários sistemas presentes no cotidiano das pessoas, entre eles é possível destacar: processos industriais, sistemas de saúde, tráfego urbano, sistemas de comunicação, entre outros (MENASCÉ, 2002; CHAUDHURI et al., 2007; OSORIO; BIERLAIRE, 2009; DIMITRIOU; LANGARIS, 2010; MACGREGOR SMITH; CRUZ; VAN WOENSEL, 2010; ALVES et al., 2011).

Um foco dessa investigação é uma discussão centrada em uma rede de filas com limitação de tamanho para área de circulação, com chegadas markovianas e atendimentos gerais, ou seja, filas $M/G/1/K$ na notação de Kendall (1953). Na prática, o interesse direto está em buscar uma alocação adequada dos recursos disponíveis que seja adequada para aumentar a eficiência tanto para os usuários do serviço quanto para os gestores que ofertam o serviço.

Exemplo 1.1 *Para ilustrar melhor esse tipo de estudo, considere o exemplo de fabri-*

cação que representa o projeto conceitual de um sistema de montagem automotiva analisado por Spieckermann et al. (2000), em que buffers finitos são necessários para evitar avarias em uma área da planta e desacoplar o processo de montagem representado na figura 2. Este exemplo foi adaptado para estudos dessa natureza por Cruz, Duarte e Souza (2018).

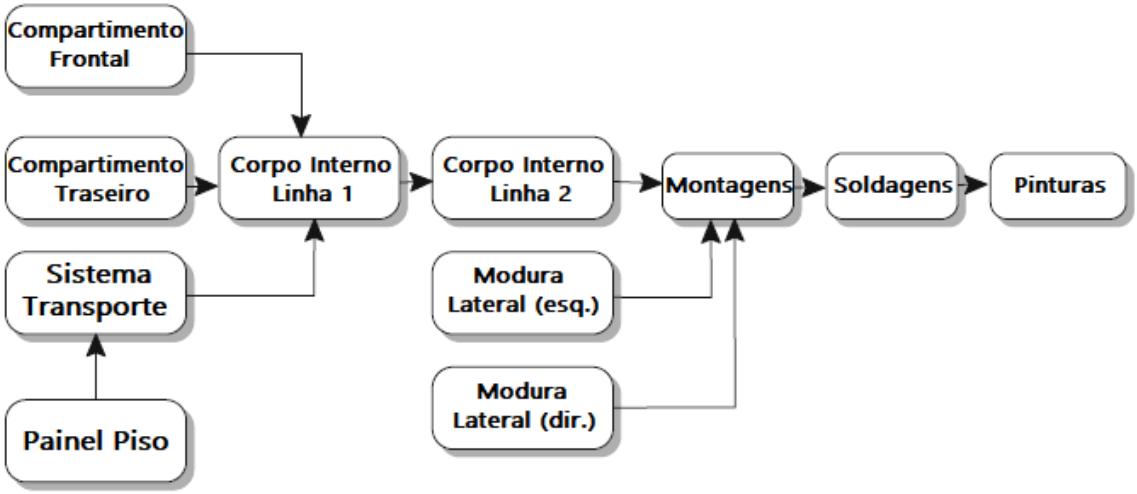


Figura 2: Um sistema de montagem automotiva adaptado de Spieckermann et al. (2000).

Uma representação simplificada em termos de uma rede de filas é dada na figura 3.

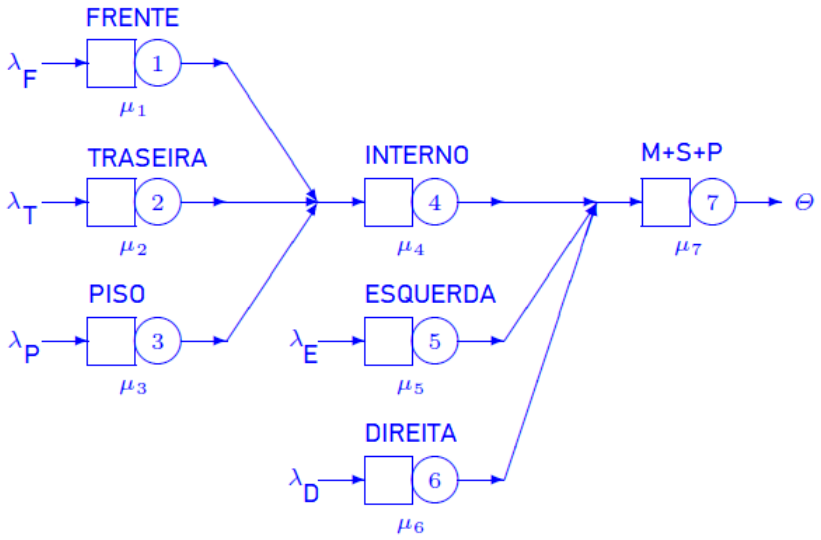


Figura 3: Uma representação em rede de filas finitas do sistema de montagem proposto por Spieckermann et al. (2000).

Observe que algumas partes dos processos que correspondiam às filas foram

mescladas em uma única fila para simplificar a construção do diagrama representativo, porém isso ocorre sem nenhuma perda de generalidade, obviamente levando em consideração as taxas de serviço combinadas das respectivas filas. Um mau dimensionamento das áreas de espera nas filas ou do número de servidores poderiam levar à perda significativa da eficiência e rentabilidade do processo, dados os altos custos envolvidos. Em casos como estes, a utilização de métodos que otimizem simultaneamente o número de servidores e a capacidade das áreas de espera se faz necessária.

1.1 OBJETIVOS

Este estudo apresenta duas formulações distintas para o problema de otimização dos recursos em redes de filas M/G/1/K. São duas abordagens já discutidas em trabalhos anteriores (CRUZ et al., 2012; CRUZ; DUARTE; SOUZA, 2018; SOUZA et al., 2020). Além disso, os trabalhos mencionados discutem distintas estratégias de otimização que são utilizadas aqui. O interesse central está em comparar as soluções fornecidas pelas diversas abordagens mencionadas. A grande inovação está em comparar tais soluções no espaço das variáveis de decisão. Com base nesse propósito de pesquisa, os passos para alcançar os objetivos do presente estudo são:

- apresentação de uma revisão bibliográfica na área de teoria de filas e problemas de otimização em filas;
- discutir a formulação matemática do problema de otimização em redes de filas;
- discutir a proposição das heurísticas de otimização: *Elitist Non-dominated Sorting Genetic Algorithm* - NSGA-II de Deb et al. (2002), *Simulated Annealing Algorithm* - SAA de Kirkpatrick et al. (1983) e Černỳ (1985) e *Paticle Swarm Optimization* - PSO de Kennedy e Eberhart (1995);
- estabelecer estudos comparativos entre experimentos computacionais realizados com os algoritmos em estudo;

- elaboração de discussões e conclusões acerca do tema de pesquisa e proposições de propostas de continuidade.

O presente relato é iniciado com um capítulo introdutório que aborda aspectos de pesquisa abordados durante a concepção desse estudo, bem como o delineamento prévio de objetivos a serem trabalhados. Posteriormente, o segundo capítulo apresenta uma relevante revisão da bibliografia existente sobre esse tema de pesquisa. A seguir, o capítulo de Aspectos Metodológicos detalha a formulação do problema de otimização sob investigação e também os algoritmos de otimização utilizados. O quarto capítulo apresenta de forma mais detalhada todo o conjunto de resultados alcançados, detalhando a análise comparativa do conjunto de soluções sob investigação. Por fim, o último capítulo apresenta as conclusões alcançadas através dessa investigação e também propostas de continuidade desse estudo.

2 REVISÃO DE LITERATURA

As filas são fenômenos comuns que ocorrem o tempo todo. Elas podem ser encontrados em qualquer lugar. Todo mundo já entrou em uma fila pelo menos uma vez, por exemplo, quando volta para casa no tráfego, quando paga contas em uma agência bancária entre outros. As filas diárias são muito frequentes e, em alguns casos, nem são percebidas. As filas acontecem, por exemplo, durante os processos de fabricação (LI; ENGINARLAR; MEERKOV, 2006; HU; MEERKOV, 2006; DIMITRIOU; LANGARIS, 2010), em aeroportos, portos e sistemas de distribuição de produtos (VAN WOENSEL et al., 2008), ou em computadores e sistemas de comunicação (AHMED; OUYANG, 2007; CHEN; HU; JI, 2010; TANG et al., 2010). As filas podem aumentar ou diminuir a qualidade dos serviços ou os preços dos produtos, dependendo da eficiência da distribuição e logística (VAN WOENSEL; CRUZ, 2009). Assim, a organização de sistemas de filas para diminuir o seu tamanho médio pode ser uma maneira de reduzir custos e maximizar a eficiência de um sistema.

Para ilustrar algumas situações, considere a montagem manual, na qual os seres humanos podem ser vistos como servidores. Como observado por Wang et al. (2009), a montagem manual é, por definição, realizada pelos trabalhadores e, portanto, o sistema é centrado no ser humano e seu desempenho depende em grande parte dos seres humanos. Na prática, é possível que, mesmo que todas as tarefas sejam similares, o tempo médio de conclusão da tarefa possa diferir de pessoa para pessoa (pois é impossível para cada trabalhador ter exatamente a mesma eficiência). Assim, os indivíduos podem ser considerados servidores heterogêneos. Este caso seria um problema mais sofisticado, no qual as taxas de serviço são heterogêneas, dificultando sobremaneira a perfeita alocação da quantidade de servidores em cada fila do sistema.

Outros exemplos incluem problemas de alocação de servidores em tarefas de uma rede de filas, mas com servidores homogêneos, ou seja, com a mesma capacidade de serviço. Um exemplo de uma situação real está ligado ao sequenciamento

de tarefas em informática entre diversos processadores. Para a situação em questão, suponha a disponibilidade de diversos processadores idênticos, ou seja, com a mesma taxa de serviço para serem alocados da forma mais conveniente. De outra forma, cada uma das filas de uma rede entre tarefas poderia receber uma quantidade específica de processadores que tornasse a execução completa da tarefa mais eficiênte. Como exemplo final, suponha o transporte de mercadorias por caminhões idênticos em potência e capacidade, dessa forma os servidores (caminhões) seriam plenamente homogêneos.

Abensur et al. (2003) utilizam estudos de teoria das filas e também de simulação para obtenção de medidas de desempenho de atendimentos realizados por caixas eletrônicos do serviço bancário e, assim, obter informações para à formulação e avaliação de estratégias viáveis para a gestão do autoatendimento bancário.

Changfu e Zhenyu (2009) estabelecem modelos matemáticos e de simulação para analisar e avaliar o impacto do ajuste estrutural no sistema de filas do ambiente e-Business. Kamali et al. (2009) propõe investigar problemas de tráfego, o monitoramento é realizado através de modelos de simulação, o interesse está em medir a eficiência do serviço heterogêneo e das diferentes tecnologias que chegam ao servidor de rede.

Camelo et al. (2010) utilizaram Teoria de filas para analisar características da dinâmica de atendimento de navios para transporte de minério de ferro e manganês e obter as medidas de desempenho em dois píeres de embarque. Uma simulação foi realizada com interesse em verificar como seria a operação de um outro píer adicional que seria construído. Doy et al. (2006) abordaram o serviço de e-mails de uma rede de computadores através da simulação de um modelo de filas, para avaliar seu desempenho.

Diversas investigações associam a Teoria das filas com mecanismos de otimização. Potenciais usuários de modelos de otimização baseados em redes de filas

incluem cientistas da computação, engenheiros de produção estatísticos entre outros. De fato, estes modelos auxiliam na compreensão e melhoria de vários sistemas reais, incluindo sistemas de manufatura (ALVES et al., 2011), de produção (MACGREGOR SMITH; CRUZ; VAN WOENSEL, 2010) e de saúde (BRUIN et al., 2007), sistemas de tráfego de veículos e de pedestres (CRUZ; MACGREGOR SMITH; QUEIROZ, 2005; CRUZ; DUARTE; VAN WOENSEL, 2008; CRUZ et al., 2010), sistemas de computação e de comunicação (CHEN; HU; JI, 2010), aplicações baseadas na *web*, configuradas em camadas (CHAUDHURI et al., 2007) e com requisitos de qualidade de serviço (QoS) definidos em termos de tempo de resposta, *throughput*, disponibilidade e segurança (MENASCÉ, 2002).

Uma abordagem já bem conhecida para otimização em rede de filas pode ser observada por Cruz et al. (2012) e Cruz, Duarte e Souza (2018). A formulação proposta busca maximizar o *throughput* (θ) do sistema, simultaneamente com a minimização do espaço total de espera alocado nas filas do sistema ($\sum K_i$), e também a minimização da soma das taxas de serviço ($\sum \mu_i$), para uma topologia pré-especificada no sistema de filas do tipo M/G/1/K.

Posteriormente, uma nova abordagem para otimização em rede de filas é apresentada por Souza et al. (2020). A formulação proposta busca minimizar a soma das probabilidades de bloqueio em todas as filas do sistema simultaneamente com a minimização do espaço total de espera alocado nas filas do sistema ($\sum K_i$), e também a minimização da soma das taxas de serviço ($\sum \mu_i$), para uma topologia pré-especificada no sistema de filas do tipo M/G/1/K.

Dado que será utilizada uma proposta de heurística otimizadora para obter soluções para o problema de otimização que será investigado, algo deve ser abordado sobre essa proposta. Martins et al. (2019) utilizaram o conhecido método de Powell de forma bem sucedida em problemas de otimização em rede de filas. Cruz et al. (2012) fizeram uso do algoritmo genético alcançando bons resultados nesse tipo de pesquisa. Cruz, Duarte e Souza (2018) abordaram um mecanismo de otimização

que acopla o algoritmo genético com o algoritmo *Simulated Annealing* com resultados promissores. Souza et al. (2020) abordaram uma estratégia de otimização que utiliza o algoritmo genético conjuntamente com o algoritmo *Particle Swarm Optimization* com resultados bastante interessantes. A escolha por uma heurística se deve ao fato que para valores elevados de N, uma busca exaustiva se tornaria inviável do ponto de vista computacional.

Esta investigação pretende avaliar as propostas discutidas por Cruz et al. (2012), Cruz, Duarte e Souza (2018) e Souza et al. (2020) e estabelecer parâmetros de comparação entre elas. Essa comparação terá foco centrado no espaço das variáveis decisórias do problema, ou seja, a alocação das áreas de circulação (*buffers*) e as taxas de serviços associadas aos servidores ao longo da rede de filas. É importante salientar que investigações no espaço das variáveis para este problema específico ainda não são recorrentes na literatura da área. Uma discussão mais profunda sobre a formulação matemática do problema e também das técnicas de otimização será apresentada no capítulo 3.

3 ASPECTOS METODOLÓGICOS

Existem algumas possibilidades de formulação para o problema de otimização já tratadas na literatura. Pode-se optar por uma abordagem mono-objetivo, em que a função objetivo seria expressa em termos do *throughput* ou alguma medida de desempenho específica alcançada pelo sistema de filas e um conjunto de restrições associadas aos *buffers* alocados nas filas do sistema e às taxas de serviço em cada servidor na rede de filas. Uma outra possível opção remete a alguma abordagem multiobjetivo na qual podem ser incluídos objetivos associados aos *buffers*, às taxas de serviço ou outras variáveis de decisão associadas ao problema.

3.1 FORMULAÇÃO MONO-OBJETIVO

Inicialmente será apresentada aqui 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 grafo direcionado $\mathcal{G}(V, A, P)$ em que V é um conjunto finito de vértices (filas) e A é um conjunto finito de arestas (conexões entre as filas) e P são as respectivas probabilidades de roteamento entre as arestas. O BAP, em sua formulação primal (MACGREGOR SMITH; CRUZ, 2005), é descrito da seguinte forma:

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

sujeito a:

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

minimizar a área total de circulação em uma rede com m filas, sujeito a um limiar de atendimento $\Theta_{\min}$. É importante notar que $\Theta(\mathbf{K})$ é difícil de definir porque é uma função não linear que envolve as taxas de chegada e de serviço e a topologia da rede de filas.

Uma outra formulação intimamente relacionada com a anterior, uma espécie de BAP dual, busca maximizar o *throughput*, $\Theta(\mathbf{K})$, restrito a um limite máximo para a alocação total de áreas de circulação ao longo da rede de filas $K_{\max}$, descrito na formulação a seguir. Esta proposta de formulação é uma visível analogia para o clássico problema da mochila estocástico (MACGREGOR SMITH; CRUZ, 2005), que pode ser definido da seguinte forma:

$$\text{maximizar } \Theta(\mathbf{K}), \tag{3}$$

sujeito a:

$$\begin{aligned} &\sum_{i=1}^{m} c_i K_i \leqslant K_{\max}, \\ &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{4}$$

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 consideram-se estudos baseados em formulações multiobjetivo.

3.2 UMA POSSÍVEL FORMULAÇÃO MATEMÁTICA MULTIOBJETIVO

O problema de otimização de redes filas $M/G/1/K$, descrito, pode ser reformulado para uma versão multiobjetivo 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 foi descrita por Cruz et al. (2012) e também por Cruz, Duarte e Souza

(2018) da seguinte maneira:

$$\text{minimizar } F(\mathbf{K}, \boldsymbol{\mu}) = \Big[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu})\Big], \tag{5}$$

sujeito a:

$$\begin{aligned} &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\mu_i \geqslant 0, \forall i \in \{1, 2, \ldots, m\}. \end{aligned} \tag{6}$$

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*:

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

Note que o *throughput* (Θ) em $f_3(\mathbf{K}, \boldsymbol{\mu})$ 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 como pode ser visto no trabalho de Cruz, Duarte e van Woensel (2008).

3.3 UMA SEGUNDA POSSIBILIDADE DE FORMULAÇÃO MATEMÁTICA MULTIOBJETIVO

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) como pode ser visto nos trabalhos de Kerbache e MacGregor Smith (2000), Cruz, Duarte e van Woensel (2008), Cruz (2009), van Woensel et al. (2010), Cruz et al. (2010), Cruz et al. (2012), van Woensel e Cruz (2014), Cruz, Duarte e Souza (2018), Martins et al. (2019).

A 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:

$$\theta = \lambda(1 - P_k) \tag{7}$$

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

Uma segunda formulação matemática para otimização, apresentada por Souza et al. (2020), concentra-se nas probabilidades de bloqueio entre as filas do sistema. Essa proposta 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:

$$\text{minimizar } F(\mathbf{K}, \boldsymbol{\mu}) = \Big[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu})\Big], \tag{8}$$

sujeito a:

$$\begin{aligned} &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\mu_i \geqslant 0, \forall i \in \{1, 2, \ldots, m\}. \end{aligned} \tag{9}$$

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:

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

Para 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 fórmula fechada computacionalmente eficiente e precisa. O método proposto por MacGregor Smith (2004) é

baseado em uma aproximação de dois momentos apresentada por Kimura (1996):

$$P_k = \frac{\rho^{\left(\frac{2+\sqrt{\rho}\,CV^2-\sqrt{\rho}+2(k-1)}{2+\sqrt{\rho}\,CV^2-\sqrt{\rho}}\right)}(\rho-1)}{\rho^{\left(2\frac{2+\sqrt{\rho}\,CV^2-\sqrt{\rho}+(k-1)}{2+\sqrt{\rho}\,CV^2-\sqrt{\rho}}\right)}-1}, \tag{10}$$

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 $CV^2 = \text{Var}(T_s)/\mathbb{E}^2(T_s)$ é o quadrado do coeficiente de variação do tempo de serviço (T_s). Vários estudos anteriores como MacGregor Smith (2003), MacGregor Smith e Cruz (2005) e Cruz, Duarte e van Woensel (2008) confirmam que esta aproximação de P_k é bastante precisa.

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 (GROSS et al., 2009).

Investigações sobre problemas em redes de filas são abordadas por várias perspectivas (CHOWDHURY; MUKHERJEE, 2013; VAN WOENSEL; CRUZ, 2014; QI et al., 2017). Abordagens através de métodos de otimização são bastante comuns, por exemplo: o algoritmo Powell (MARTINS et al., 2019), algoritmos genéticos (CRUZ et al., 2012), *Simulated Annealing* (CRUZ; DUARTE; SOUZA, 2018) e o *Particle Swarm Optimization* (SOUZA et al., 2020) foram utilizados com sucesso. Essas abordagens usam o método de avaliação de desempenho aproximado bem conhecido, o Método de Expansão Generalizado (GEM) proposto em Kerbache e MacGregor Smith (1987) para estimar medidas de desempenho nas filas da rede.

3.4 DETALHAMENTO DO ALGORITMO GENÉTICO NSGA-II

A versão para o algoritmo genético NSGA-II implementado nos estudos de Cruz et al. (2012), Cruz, Duarte e Souza (2018) e Souza et al. (2020) será apre-

sentada superficialmente aqui, detalhes mais específicos podem ser obtidos nos três trabalhos. Esta descrição foi inspirada na descrição apresentada em Souza (2020). Na aplicação dos algoritmos genéticos multiobjetivo para problemas de otimização multiobjetivo, 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 4.

algoritmo
$P_1 \leftarrow$ **GeraPopulaçãoInicial**$(\texttt{popSize})$
para $i = 1$ **até** $\texttt{numGen}$ **faça**
/ gera filhos por cruzamento e mutação */*
$Q_i \leftarrow$ **FaçaPopulaçãoNova**(P_i)
/ combina pais e filhos */*
$R_i \leftarrow P_i \cup Q_i$
/ encontre fronteiras não-dominadas* $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$ **/*
$\mathcal{F} \leftarrow$ **OrdenaçãoNãoDominante**(R_i)
/ encontre nova população pela distância de aglomeração*/*
$P_{i+1} \leftarrow$ **GeraPopulaçãoNova**(R_i)
fim para
$P_{\texttt{numGen+1}} \leftarrow$ **ExtraiPareto**$(P_{\texttt{numGen}})$
escreva $P_{\texttt{numGen+1}}$
fim algoritmo

Figura 4: Pseudocódigo para o algoritmo genético NSGA-II (CRUZ et al., 2012).

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 medida de diversidade *crowding distance* (DEB et al., 2002), 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 são projetados de acordo com a aplicação em estudo. Para o problema específico de investigação em rede de filas, o

mecanismo *cruzamento uniforme* (BÄCK; FOGEL; MICHALEWICZ, 1997) foi utilizado devido à sua eficiência na identificação, herança e proteção de genes comuns, além de recombinar genes não comuns (SYSWERDA, 1989; HU; DI PAOLO, 2009). O cruzamento uniforme é realizado para cada variável com uma determinada probabilidade (`rateCro`). O operador de cruzamento uniforme usado no algoritmo é o *operador de cruzamento binário simulado* (SBX) (DEB; AGRAWAL, 1995; DEB; BEYER, 2001), que é muito conveniente para algoritmos genéticos multiobjetivo com codificação real. A partir de pais ($x_{i,(\bullet,t)}$), filhos ($x_{i,(\bullet,t+1)}$) são obtidos de acordo com as equações 11 e 12. O operador foi concebido com interesse em criar soluções que possuem um poder de pesquisa semelhante a um cruzamento de ponto único de algoritmos genéticos com codificação binária (DEB; AGRAWAL, 1995):

$$x_{i,(1,t+1)} = 0{,}5\Big[(1+\beta)x_{i,(1,t)} + (1-\beta)x_{i,(2,t)}\Big], \tag{11}$$

$$x_{i,(2,t+1)} = 0{,}5\Big[(1-\beta)x_{i,(1,t)} + (1+\beta)x_{i,(2,t)}\Big], \tag{12}$$

em que a função densidade de probabilidade descrita pela equação 13 modela a variável aleatória β,

$$f(\beta) = \begin{cases} 0{,}5(\eta+1)\beta^{\eta}, & \text{se } \beta \leqslant 1, \\ 0{,}5(\eta+1)\frac{1}{\beta^{\eta+2}}, & \text{caso contrário}, \end{cases} \tag{13}$$

Um grande volume de 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 normais gaussianas são adicionadas às variáveis de decisão (DEB; AGRAWAL, 1995), 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:

$$k_{\mathrm{rfl}_i} = k_{\mathrm{inf\_lim}} + |k_i - k_{\mathrm{inf\_lim}}|, \tag{14}$$

e

$$\mu_{\mathrm{rfl}_i} = \mu_{\mathrm{inf\_lim}_i} + |\mu_i - \mu_{\mathrm{inf\_lim}_i}|, \tag{15}$$

em que $k_{\mathrm{inf\_lim}}$ é o limite inferior da alocação total, ou seja, $k_{\mathrm{inf\_lim}} = 1$ e $\mu_{\mathrm{inf\_lim}_i}$ 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.

3.5 DETALHAMENTO DO ALGORITMO *SIMULATED ANNEALING*

A metodologia proposta através do algoritmo genético em Cruz et al. (2012) fornece um conjunto Pareto sub-ótimo, porém não é possível garantir que todas as combinações de alocação de *buffers*, $\mathbf{K} = (K_1, \ldots, K_m)$ em uma rede de m filas tenham sido vasculhadas. Em outras palavras, os operadores genéticos do algoritmo não são capazes de assegurar a procura por algumas das possíveis soluções que preservam uma mesma capacidade total $(\sum K_i)$ e fornecem um maior desempenho, por exemplo na formulação que visa maximizar o *throughput* (θ).

A proposição de uma estratégia de pós-processamento através do algoritmo *Simulated Annealing* como descrita por Cruz, Duarte e Souza (2018). Alguma recombinação na distribuição de *buffers* entre as filas do sistema pode gerar uma substancial

melhoria no *throughput* do sistema.

Nesse cenário, o problema de otimização, para o pós-processamento foi descrito por Cruz, Duarte e Souza (2018) como:

$$\text{maximizar } \theta(\mathbf{K}, \boldsymbol{\mu}), \tag{16}$$

sujeito a:

$$\begin{aligned} &\mathbf{K} = (K_1, \ldots, K_N), K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N), \mu_i \geqslant 0, \forall i \in \{1, 2, \ldots, m\}. \end{aligned} \tag{17}$$

em que as variáveis de decisão, K_i, indicam *buffers* alocados para a i-ésima fila. Os valores μ_i para as taxas de serviço são fixos.

Utilizou-se a proposta de heurística otimizadora para fornecer soluções para o problema de otimização descrito nas equações 16 e 17, a partir do conjunto inicial de soluções candidatas formado pelas soluções fornecidas pelo algoritmo genético utilizado em Cruz et al. (2012). A escolha por uma heurística se deve ao fato que para valores elevados de m, uma busca exaustiva se tornaria inviável do ponto de vista computacional. Como já mencionado anteriormente, uma heurística de otimização que pode se adaptar bem a natureza do problema em questão é o clássico algoritmo *Simulated Annealing*.

O algoritmo *Simulated Annealing* descrito inicialmente por Kirkpatrick et al. (1983) e Černỳ (1985) é inspirado no processo de recozimento de sistemas físicos. Os princípios básicos têm origens em termodinâmica estatística, uma analogia com o recozimento de sólidos poderia fornecer uma estrutura para o desenvolvimento de um algoritmo genérico de otimização capaz de escapar de ótimos locais na busca pelo ótimo global. Desde a sua introdução, como um método de otimização combinatorial, o *Simulated Annealing* é vastamente utilizado em diversas áreas, tais como projeto de circuitos integrados auxiliado por computador, processamento de imagem, redes neurais, entre outras.

O método não utiliza uma “estratégia” (uma lei por exemplo para convergência total), o que faz assumir na maioria das vezes um mínimo ou máximo que não é o global, mais se configura como uma boa opção para solução do problema em questão, como pode ser visto em Spinellis, Papadopoulos e MacGregor Smith (2000).

Visto de forma resumida, o método depende de um funcional objetivo de otimização, neste caso $\theta(\mathbf{K})$, e de um critério de vizinhança entre as soluções candidatas. A técnica reproduz uma cadeia de Markov cujo espaço de estados é composto por um conjunto de possíveis soluções para o problema de otimização em estudo. O *Simulated Annealing* opera da seguinte forma: se o $i-$ésimo estado da cadeia de Markov é uma possível solução $\mathbf{K}^{(a)}$, então alguma solução vizinha da solução $\mathbf{K}^{(a)}$ é selecionada aleatoriamente; se o estado vizinho escolhido for $\mathbf{K}^{(b)}$, então o próximo estado da cadeia será $\mathbf{K}^{(b)}$, se este for superior a $\mathbf{K}^{(a)}$ isso quando avaliado através do funcional objetivo do problema. Caso contrário, o próximo estado da cadeia ainda poderá ser $\mathbf{K}^{(b)}$ com uma probabilidade p, ou então a cadeia se manterá em $\mathbf{K}^{(a)}$ com probabilidade $1-p$. A escolha do valor p, em geral, é dependente do número de passos já executados pela cadeia de Markov e também pelo acréscimo ou decréscimo na função objetivo, gerado pela possível troca entre as soluções $\mathbf{K}^{(a)}$ e $\mathbf{K}^{(b)}$.

Uma escolha computacionalmente usual de p é $e^{C\log(1+n)}$ em que a constante C é dada por $C = -\left|\theta\left(\mathbf{K}^{(a)}\right) - \theta\left(\mathbf{K}^{(b)}\right)\right|$ e $\mathtt{n}$ é o número de passos dados pela cadeia de Markov até o instante corrente. Após um conjunto de $\mathtt{nsim}$ passos sucessivos da cadeia, $\Omega_K = \{\mathbf{K}_1, \ldots, \mathbf{K}_{nsim}\}$, é possível estimar a solução ótima θ^* por $\theta(\mathbf{K}_i)$, em que $\mathbf{K}_i$ otimiza θ com respeito ao conjunto Ω_K.

É importante também, discutir a definição do conceito de vizinhança entre as possíveis alocações $\mathbf{K} = (K_1, \ldots, K_m)$. Dado dois valores aleatórios distintos i e j em $\{1, \ldots, m\}$, um possível vizinho da alocação $\mathbf{K} = (K_1, \ldots, K_m)$ pode ser definido considerando a alocação perturbada dada por $\mathbf{K}^* = (K_1, \ldots, K_{i-1}, K_i - 1, K_{i+1}, \ldots, K_{j-1}, K_j + 1, K_{j+1}, \ldots, K_m)$. Nessa proposição de vizinhança, as modificações na alocação individual de *buffers* entre as filas, mantém fixo o espaço total

alocado, porém acarretará em alterações no valor da função objetivo $\theta(.)$. Uma versão resumida do algoritmo utilizado pode ser verificada na Figura 5, trata-se da descrição apresentada em Cruz, Duarte e Souza (2018).

algoritmo
/ recebe soluções iniciais via NSGA-II*/*
$\mathbf{K}^* \leftarrow \mathbf{K} \leftarrow \mathbf{K}_1$
Inicializa temperatura T_1
Inicializa número de transições L_1
$j \leftarrow 1$
repita
para $i = 1$ **até** L_j **faça**
$\mathbf{K}_i \leftarrow$ **SoluçãoPerturbada**$(\mathbf{K})$
$\Delta \leftarrow [\theta(\mathbf{K}_i, \mu^*) - \theta(\mathbf{K}, \mu^*)]$
$\mathcal{U} \leftarrow$ **Uniforme**$(0, 1)$
se $(\Delta \geqslant 0)$ **então**
$\mathbf{K}^* \leftarrow \mathbf{K} \leftarrow \mathbf{K}_i$
senão se $(\mathcal{U} < e^{\Delta/T_j})$ **então**
$\mathbf{K} \leftarrow \mathbf{K}_i$
fim se
fim para
$j \leftarrow j + 1$
Atualiza temperatura T_j
Atualiza número de transições L_j
enquanto $(j < j_{max})$ **ou** $(T_j < T_\varepsilon)$
escreva $\mathbf{K}^*$
fim algoritmo

Figura 5: Pseudocódigo para o algoritmo *Simulated Annealing* (CRUZ; DUARTE; SOUZA, 2018).

3.6 DETALHAMENTO DO ALGORITMO POR ENXAME DE PARTÍCULAS (PSO)

Novamente a metodologia proposta através do algoritmo genético em Cruz et al. (2012) foi avaliada para alguma proposta de pós-processamento. Por outro lado, o estudo apresentado por Souza et al. (2020) apresenta a formulação descrita nas equações 8 e 9, baseada em minimizar a soma das probabilidades de bloqueio. Além disso, o estudo de Souza et al. (2020) propõe o pós processamento através de um algoritmo

multiobjetivo por enxame de partículas, o clássico *Particle Swarm Optimization*.

Uma adequada proposição de implementação de um algoritmo multiobjetivo por enxame de partículas passa inicialmente pela bem ajustada definição do que representará cada partícula na formulação matemática do problema sob investigação. Na implementação em questão, cada partícula representou uma solução possível para a alocação dos recursos (espaço de área de circulação e taxas de serviço para cada uma das filas) que otimizam a rede de filas em estudo. Portanto, nessa formulação específica de algoritmo, 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 salientar que o problema de otimização multiobjetivo que está sendo tratado através desse algoritmo 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 como já discutido anteriormente nas equações 14 e 15 sugeridas para a implementação do algoritmo genético NSGA-II.

A descrição apresentada aqui, foi inspirada na descrição apresentada em Souza (2020). Para a implementação do algoritmo multiobjetivo por enxame de partículas algumas definições paramétricas são bastante relevantes. 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 multiobjetivo por enxame de partículas 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 algoritmo multiobjetivo por enxame de partículas aqui descrito, é uma adaptação da implementação clássica apresentada por Coello Coello e Lechuga (2002). A abordagem multiobjetivo por enxame de partículas proposta para o problema de otimização 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: Pseudocódigo para o algoritmo PSO multiobjetivo (SOUZA et al., 2020).

Na formulação multiobjetivo, a posição da i-ésima partícula no espaço d-dimensional de busca é representada por $x_i = (x_{i1}, x_{i2}, \ldots, x_{i\ell})$. Já a velocidade da

referida partícula é representada por $v_i = (v_{i1}, v_{i2}, \ldots, v_{i\ell})$. A melhor posição da i-ésima partícula durante as buscas é dada por $p_i = (p_{i1}, p_{i2}, \ldots, p_{i\ell})$. A velocidade e a posição das partículas são atualizadas da iteração t para a iteração $t+1$ conforme as equações:

$$v_i^{t+1} = w^t + r_1\big(p_i - x_i^t\big) + r_2\big(g_i - x_i^t\big), \tag{18}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}; \tag{19}$$

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 multiobjetivo) 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 multiobjetivo) 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.

Com respeito à implementação utilizada para o algoritmo PSO, versões simplificadas podem ser encontradas em Trivedi, Varshney e Ramteke (2020) e versões mais sofisticadas e aprimoradas como as descritas em Fan et al. (2017) e Jia e Zhu (2017) também, incluindo formulações de programação matemática com números reais e inteiros em Zhao et al. (2013).

3.7 MÉTODO DE EXPANSÃO GENERALIZADO

A atuação de todos os algoritmos descritos anteriormente depende de estratégias adequadas para o cálculo das medidas de desempenho envolvidas no problema. Considerando filas simples, para que seja possível a maximização da taxa de atendimento $\theta(\mathbf{K}, \boldsymbol{\mu})$ é necessário algum método para estimá-la. Em uma *única fila*

$M/G/1/K$, o procedimento de estimação pode ser executado através de uma forma matemática fechada, computacionalmente eficiente, para a probabilidade p_K de ocorrência de bloqueio na fila como descrito anteriormente pela equação 10. Resultados empíricos em Cruz, Duarte e van Woensel (2008) indicam que esta aproximação para p_K é bastante acurada, para uma vasta gama de valores.

Já para uma *rede de filas*, a estimação da taxa de atendimento é consideravelmente mais complicada. 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 apresentado por Kerbache e MacGregor Smith (2000). 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. Será apresentada uma descrição do GEM de acordo com o detalhamento apresentado em Cruz, Duarte e van Woensel (2008). Esta descrição foi apresentada por Cruz, Duarte e Souza (2018) e utilizada neste texto com interesse em trazer maior clareza aos leitores.

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, $M/G/c/c$, de acordo com Cheah e MacGregor Smith (1994). A considerar que clientes perdidos pelo bloqueio são incluídos, o processo de saída segue distribuição Poisson. Diversos estudos apresentam resultados empíricos nesta direção como MacGregor Smith e Cruz (2005), MacGregor Smith, Cruz e van Woensel (2010), Cruz, Duarte e van Woensel (2008), Andriansyah et al. (2010), Cruz et al. (2010) e Cruz, Oliveira e Duczmal (2010).

A Figura 7 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 na rede.

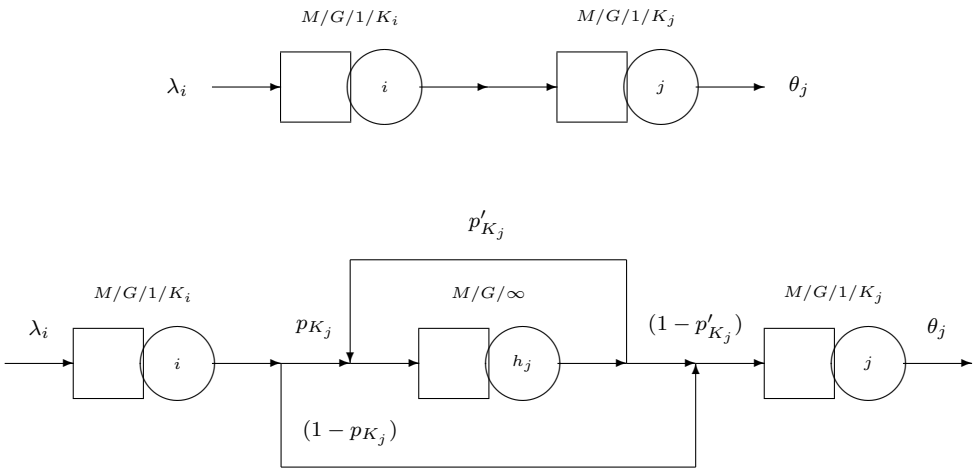


Figura 7: Esquema ilustrativo de utilização do Método de Expansão Generalizado.

O GEM pode ser dividido em três estágios mais relevantes: *reconfiguração de rede*, *estimação dos parâmetros*, e *eliminação da retroalimentação*.

3.7.1 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.

3.7.2 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 (10). Já a probabilidade de ocorrer um segundo bloqueio, P'_K, é obtida por uma aproximação via técnicas de difusão, desenvolvida por Labetoulle e Pujolle (1980):

$$P'_K = \left(\frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda\Big((r_2^K - r_1^K) - (r_2^{K-1} - r_1^{K-1})\Big)}{\mu_h\Big((r_2^{K+1} - r_1^{K+1}) - (r_2^K - r_1^K)\Big)} \right)^{-1}, \tag{20}$$

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 ainda λ_j é 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:

$$\mu_h = \frac{2\mu_j}{1 + \sigma_j^2 \mu_j^2}, \tag{21}$$

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

3.7.3 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:

$$\mu'_h = (1 - p'_K)\mu_h. \tag{22}$$

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:

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_K(\mu'_h)^{-1}. \tag{23}$$

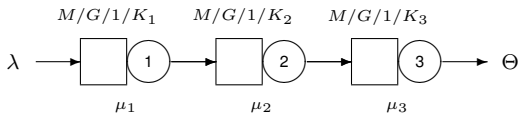
Para cada fila finita j posterior a uma fila finita i, um conjunto de equações não lineares simultâneas para as variáveis P_K, P'_K, e μ_h, associadas com outras variáveis auxiliares, como λ e $\tilde{\lambda}_i$. A solução dessas equações de forma simultânea executadas recursivamente possibilita o cálculo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

$$\begin{aligned}
\lambda &= \lambda_j - \lambda_h(1 - P'_K), && (24)\\
\lambda_j &= \tilde{\lambda}_i(1 - P_K), && (25)\\
\lambda_j &= \tilde{\lambda}_i - \lambda_h, && (26)\\
P'_K &= \left(\frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda\Big((r_2^K - r_1^K) - (r_2^{K-1} - r_1^{K-1})\Big)}{\mu_h\Big((r_2^{K+1} - r_1^{K+1}) - (r_2^K - r_1^K)\Big)} \right)^{-1}, && (27)\\
z &= (\lambda + 2\mu_h)^2 - 4\lambda\mu_h, && (28)\\
r_1 &= \frac{[(\lambda + 2\mu_h) - \sqrt{z}]}{2\mu_h}, && (29)\\
r_2 &= \frac{[(\lambda + 2\mu_h) + \sqrt{z}]}{2\mu_h}, && (30)\\
& && (31)
\end{aligned}$$

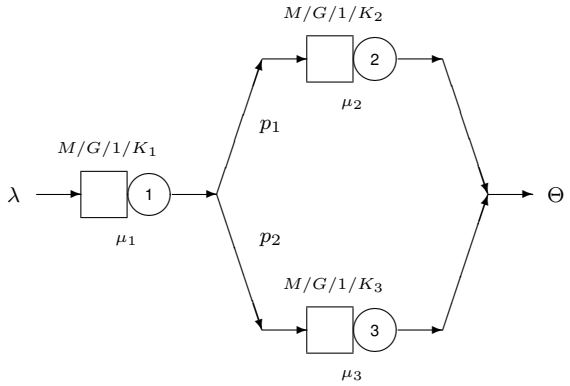
As equações (24) a (27) se referem às chegadas e também à retroalimentação do nó artificial h_j. As equações (28) a (30) são utilizadas para a resolução da equação (27), em que z uma quantidade auxiliar, utilizada para simplificar o procedimento. Por fim, a equação (10) fornece a probabilidade de bloqueio para a fila. Assim, na prática são cinco equações para serem resolvidas, ou seja, as equações (24) a (27) e a também a equação (10).

4 RESULTADOS E DISCUSSÕES

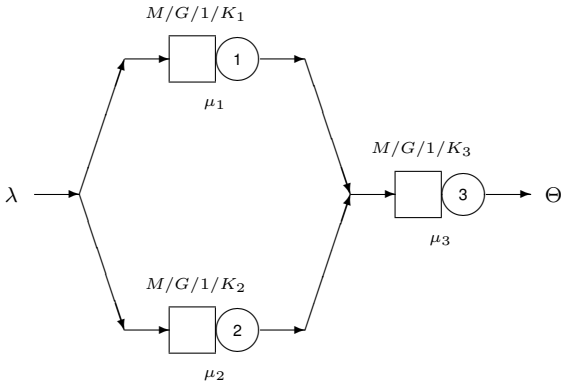
O objetivo do estudo é a comparação entre a qualidade das soluções obtidas para duas formulações matemáticas multiobjetivo e diferentes abordagens algorítmicas. Topologias específicas, incluem fusões, divisões e séries são investigadas. A figura 8 detalha as topologias sob investigação.



(a) topologia de três filas em série



(b) topologia de três filas com divisão



(c) topologia de três filas com fusão

Figura 8: Topologias testadas.

Na figura 8 é apresentada topologias em série de 3 vértices. Também uma topologia de divisão com 3 vértices sendo 1 vértice se dividindo em 2 através de um vetor com duas probabilidades de roteamento. E por fim, uma topologia de fusão com 3 vértices sendo 2 vértices se fundindo em 1.

Todos os algoritmos de otimização discutidos anteriormente estão implementados em FORTRAN e foram gentilmente cedidos pelos autores (CRUZ et al., 2012; CRUZ; DUARTE; SOUZA, 2018; SOUZA et al., 2020). 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.

As redes de filas apresentadas na Figura 8 foram analisadas com os métodos de otimização discutidos. São redes bastante adequadas para os experimentos em

estudo, isso porque incluem as situações topológicas mais recorrentes nos problemas reais. Foram analisados três valores distintos para os quadrados do coeficiente de variação $CV^2 = \{0{,}5; 1{,}0; 1{,}5\}$ para caracterizar sistemas que são hipoexponenciais, exponenciais (markovianos) e hiperexponenciais, respectivamente. A taxa de chegada no sistema de filas foi sempre fixada em $\Lambda = 5{,}0$. Os experimentos abordaram as duas formulações matemáticas multiobjetivos propostas, a primeira, referente as equações 5 e 6, será aqui nominada formulação 1. A segunda formulação, referente as equações 8 e 9, será aqui denominada formulação 2. O objetivo central dessa investigação é avaliar as soluções no espaço das variáveis do problema, ou seja, comparar as alocações das áreas de circulação (*buffers*) e as taxas de serviços associadas aos servidores ao longo da rede de filas dentre as diversas soluções obtidas.

4.1 AVALIAÇÃO DE RESULTADOS NO ESPAÇO DAS VARIÁVEIS ENTRE AS FORMULAÇÕES 1 E 2 VIA ALGORITMO NSGA-II

A presente seção tem intuito de avaliar os resultados obtidos pelo algoritmo NSGA-II, para o espaço das variáveis resultantes nas formulações 1 e 2. Porém, é de extrema importância deixar claro que as soluções investigadas não são comparáveis no espaço dos objetivos. O interesse comparativo se restringe ao espaço das variáveis (recursos em *buffers* e servidores). Em cada uma das formulações propostas, a otimização ocorreu para funcionais objetivos distintos. A formulação 1, o algoritmo foi desenvolvido para otimização voltada ao *throughput* Θ. Já na formulação 2, a otimização foi voltada na minimização da soma das probabilidades de bloqueios P_K ao longo de todo o sistema.

No espaço das variáveis, os resultados são de fato comparáveis. Os resultados obtidos no espaço dos objetivos mensuram o grau de eficiência das soluções segundo os objetivos avaliados. Já a análise no espaço das variáveis tenta comparar o custo das soluções fornecidas por meio das duas formulações, isso quanto ao consumo de recursos para o funcionamento das filas do sistema. Este tipo de análise permite identificar se uma das formulações é capaz de fornecer soluções interessan-

tes, porém com menor consumo de recursos.

A figura 9 apresenta graficamente os resultados obtidos pelo algoritmo NSGA-II, para cada uma das duas formulações. Por colunas são representados os três valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Já por linhas, são representadas as três filas da rede investigada. A rede em estudo é representada no esquema que pode ser visualizado na figura 8 (a).

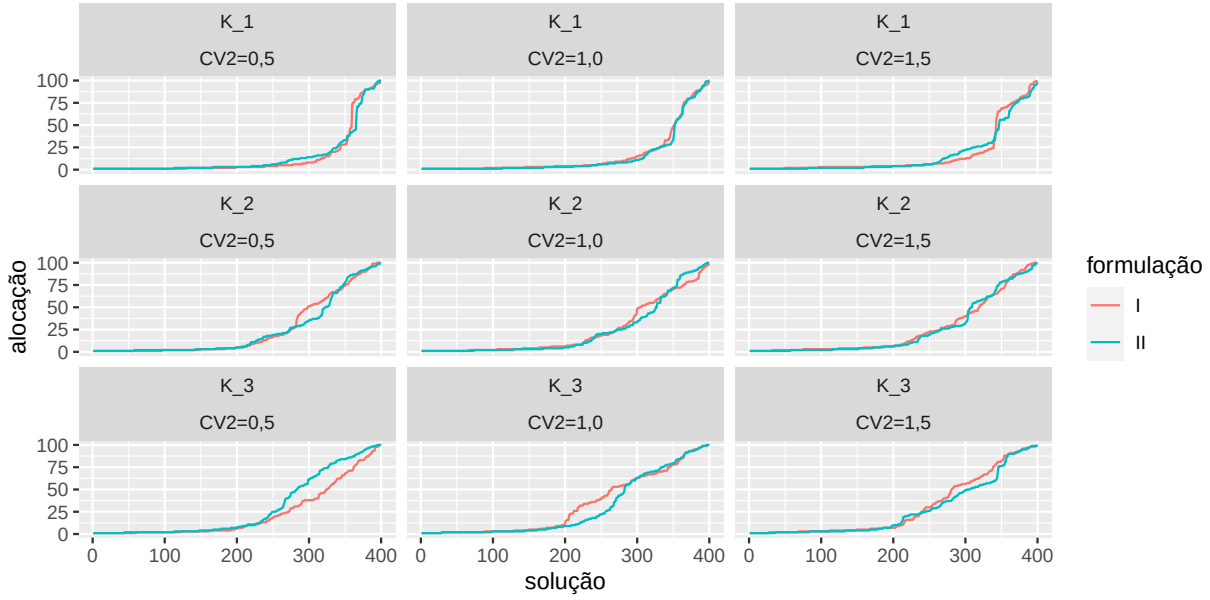


Figura 9: Alocação de áreas de circulação entre filas através do NSGA-II para a topologia (série) da figura 8 (a).

Para sistemas hipoexponenciais, com $CV^2 = 0{,}5$; verifica-se similaridade nas soluções com menos espaços alocados. Em particular para a primeira fila da rede, a partir de um determinado ponto, as soluções passam a ter uma alocação de *buffers* mais representativa. Entre as soluções 250 e 350, a soluções da formulação 1 são menos onerosas. Este efeito se inverte entre as soluções 350 e 375 e o equilíbrio retorna para as últimas soluções mais onerosas na primeira fila do sistema. Para a segunda fila da rede de filas, as soluções passam a ter uma alocação de *buffers* mais representativa a partir da solução 200, mas as diferenças se tornam significativas entre as duas formulações somente após a solução 250. Entre as soluções 250 e 350, as soluções da formulação 1 são bem mais onerosas que as da formulação 2 e posteriormente retornam para uma condição de equilíbrio. Já para a terceira fila

do sistema, as soluções passam a ter uma alocação de *buffers* mais representativa a partir da solução 200, as diferenças se tornam significativas entre as duas formulações somente após a solução 250. deste ponto em diante, as soluções da formulação 1 são bem menos onerosas que as da formulação 2.

Para sistemas markovianos, com $CV^2 = 1{,}0$; novamente verifica soluções com menos espaços alocados. Em particular para a primeira fila da rede de filas, as soluções têm volume de recursos alocados semelhante em todas as soluções com uma ligeira superioridade para poucas soluções fornecidas pela formulação 2. Para a segunda fila do sistema, novamente as soluções são similares e passam a ter uma alocação de *buffers* mais representativa a partir da solução 200, mas as diferenças se tornam significativas entre as duas formulações somente após a solução 300. Entre as soluções 300 e 330, as soluções da formulação 1 são mais custosas que as da formulação 2, o equilíbrio aparece entre as soluções 330 e 350 quando as soluções da formulação 1 se tornam economicamente mais interessantes. Já para a terceira fila da rede, as soluções passam a ter uma alocação de *buffers* mais representativa a partir da solução 200, neste ponto, as soluções da formulação 1 se tornam bem mais onerosas até a solução 270 quando novamente voltam para condições de equilíbrio com ligeiras flutuações favoráveis à formulação 1.

Para sistemas hiperexponenciais, com $CV^2 = 1{,}5$; a similaridade das soluções com menos espaços alocados é visualizada. Para a primeira fila da rede, a partir de um determinado ponto, as soluções passam a ter uma alocação de *buffers* mais representativa. Entre as soluções 260 e 340, a soluções da formulação 1 são menos onerosas. Este efeito se inverte entre as soluções 340 e 400. Para a segunda fila do sistema, as soluções passam a ter uma alocação de *buffers* mais representativa a partir da solução 200, aparentemente existe equilíbrio entre as soluções das duas formulações, com pequenas flutuações favoráveis ora para uma formulação ora para a outra. Já para a terceira fila do sistema, as soluções passam a ter uma alocação de *buffers* representativa a partir da solução 200, as diferenças se tornam favoráveis

para a formulação 1 entre as soluções 200 e 230 e se invertem para a formulação 2 sendo mais viável economicamente entre as soluções 230 e 350 quando o equilíbrio é reestabelecido.

A figura 9 deixa nítido para os três valores de CV^2 soluções similares até a solução 200, para as três filas da rede. O consumo de recursos é bem próximo entre as duas formulações. Entre as soluções 200 e 300 o comportamento da formulação 2 se mostrou superficialmente mais econômico na segunda fila da rede de filas para todos os valores de CV^2 investigados. Já na terceira fila do sistema de filas, existem diferenças notórias entre os sistemas hipoexponenciais e os demais avaliados. A formulação 1 somente revelou uma destacada superioridade nos sistemas hipoexponenciais.

A figura 10 representa a alocação total consumida nas três filas da rede. A formulação 1 se mostrou superior para os sistemas hipoexponenciais. A formulação 2 apresentou resultados notoriamente menos onerosos para os sistemas markovianos. Quanto aos sistemas hiperexponencial, existe um grande equilíbrio no consumo de recursos em *buffers*, mas uma ligeira desvantagem pode ser observada para a formulação 1 com um pouco mais de recursos consumidos.

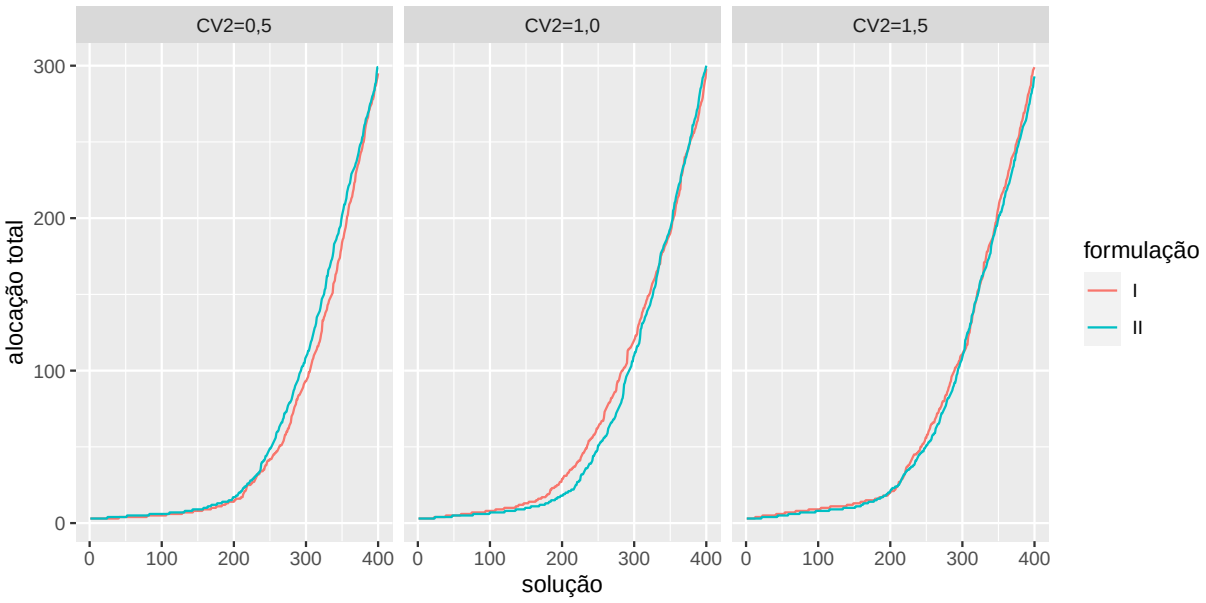


Figura 10: Alocação total de áreas de circulação através do NSGA-II para a topologia (série) da figura 8 (a).

A figura 11 traz a representação gráfica para os resultados obtidos pelo algoritmo NSGA-II, para cada uma das duas formulações. Nas colunas são representados os valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Nas linhas, as taxas de serviços para cada uma das três filas da topologia apresentada na figura 8 (a).

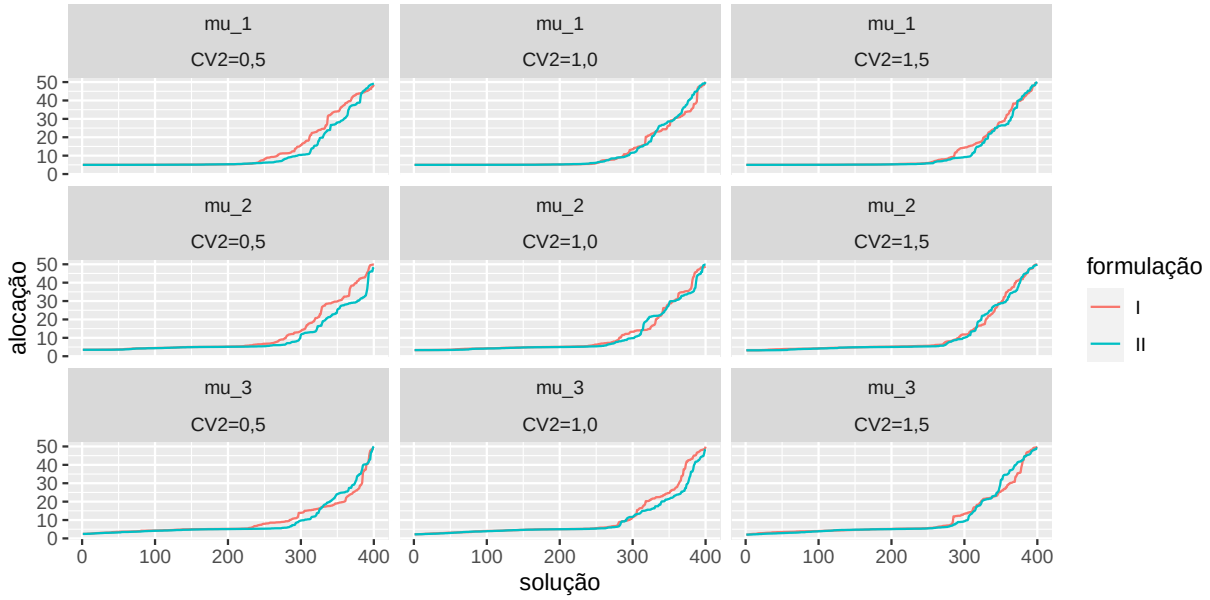


Figura 11: Recurso gasto em taxas de serviço entre filas através do NSGA-II para a topologia (série) da figura 8 (a).

Nos sistemas de filas hipoexponenciais, $CV^2 = 0{,}5$; pode ser observada uma semelhança nos resultados obtidos com as três taxas de serviços, com baixo gasto de recursos em grande parte do estudo. Entre as soluções 200 e 250 há uma mudança no comportamento dos resultados obtidos, em que a formulação 2 apresentou resultados melhores para as taxas de serviços da primeira e segunda rede de filas. Já para a terceira fila da rede, a formulação 2 se apresentou mais eficiente para as soluções entre 300 e 350. Após este momento, a formulação 1 revelou resultados melhores até próximo da solução 400.

Para sistema markovianos, com $CV^2 = 1{,}0$; pode ser verificado novamente soluções com baixo gasto em taxas de serviços, com um visível equilíbrio entre os resultados obtidos para as três filas da rede, pelas duas formulações até a solução 300. Para a primeira fila da rede da figura 8 (a), uma pequena oscilação entre os

resultados obtidos para as duas formulações propostas é verificada. Existe uma discreta vantagem para as soluções obtidas pela formulação 1. Já para a segunda fila da topologia, as soluções são semelhantes às demonstradas anteriormente. Após a solução 300, uma estreita vantagem para os resultados obtidos pela formulação 2 é observada. Na terceira fila do sistema, o comportamento das respostas também é semelhante as demais, porém é mais visível o melhor desempenho das resoluções obtidas pela formulação 2, após a solução 300.

Já nos sistemas hiperexponenciais, com $CV^2 = 1{,}5$; existe semelhança entre as soluções com menor gasto de recursos em taxas de serviço para as três filas da topologia proposta na figura 8 (a). Visualmente é possível notar o equilíbrio entre as soluções para ambas as formulações, entretanto após a solução 300, ocorre uma mudança no comportamento das soluções encontradas, com um mínimo ganho para os resultados gerados pela formulação 2. Já para a segunda fila da rede de filas, há um equilíbrio maior entre os resultados encontrados nas duas formulações, com a formulação 2 se mostra ligeiramente superior. Na terceira fila do sistema, também existe um equilíbrio entre as soluções obtidas. Porém é perceptível uma oscilação entre os resultados para as duas formulações após a solução 300. Com um pequeno ganho em performance nas resoluções obtidas pela formulação 1.

Para as três proposições utilizadas com interesse em verificar o desempenho do algoritmo NSGA-II. Nota-se um padrão nas soluções obtidas entre as três filas da topologia da figura 8 (a). É apresentado um baixo gasto em taxas de serviços próximo à solução 300, com pequenas variações entre o desempenho das duas formulações utilizadas. É notório um comportamento similar de aumento do gasto de recursos, com pequenas oscilações entre as formulações.

A figura 12 apresenta o recurso total gasto em taxas de serviço para as três filas da rede. Para os sistemas hipoexponeciais, a formulação 2 exibiu melhor performance. Nas redes de filas markovianas, o equilíbrio é mais latente, com uma mínima vantagem para a formulação 2. Já nos sistemas hiperexponencias, também ocorre

uma constância entre as soluções, da mesma forma observada para os sistemas de filas markovianas, em que a formulação 2 apresenta visualmente uma pequena vantagem para os resultados obtidos.

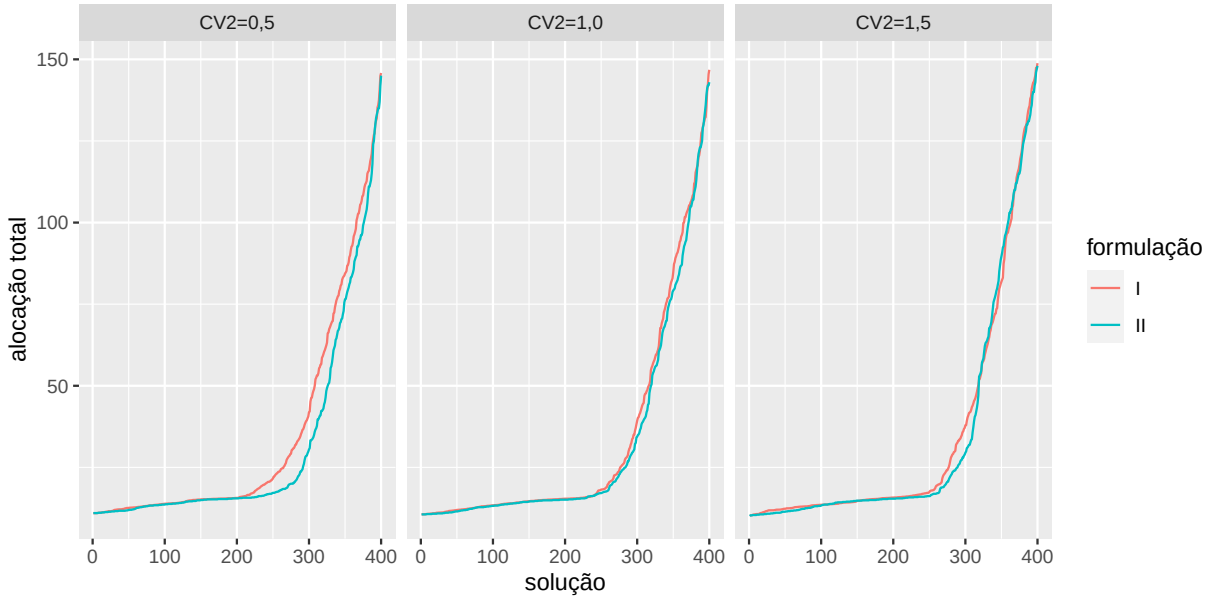


Figura 12: Recurso total gasto em taxas de serviço através do NSGA-II para a topologia (série) da figura 8 (a).

A figura 13 representa graficamente os resultados obtidos pelo algoritmo NSGA-II, para cada uma das duas formulações.

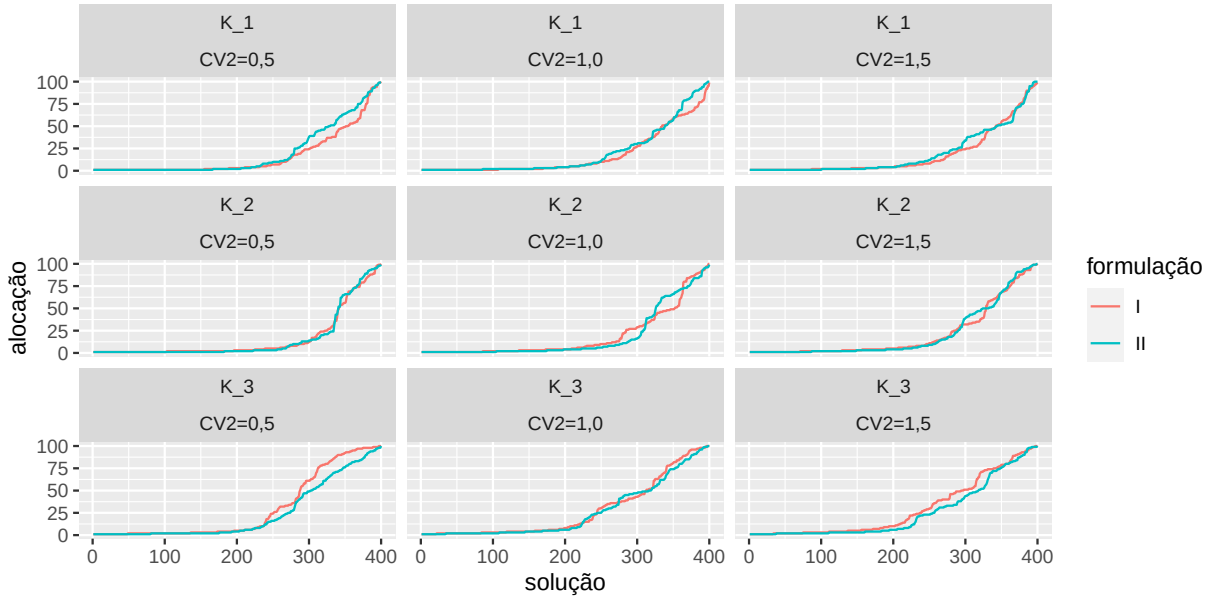


Figura 13: Alocação de áreas de circulação entre filas através do NSGA-II para a topologia (divisão) da figura 8 (b).

Novamente por colunas são representados os três valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Já por linhas, são representadas as três filas da rede investigada. A rede em estudo é representada no esquema que pode ser visualizado na figura 8 (b).

Em sistemas de filas hipoexponenciais, com $CV^2 = 0{,}5$; observa-se igualdade nas soluções para alocações com menor consumo de recursos. Em especial, na primeira fila da rede, é notório o equilíbrio e a partir das proximidades da solução 250 há uma alocação melhor para os *buffers* sugeridos através da formulação 1, para as soluções mais caras. Já na segunda fila do sistema, há uma grande estabilidade nos resultados obtidos duas formulações até próximo da solução 350, com uma leve superioridade para os resultados gerados pela formulação 1. Já para a terceira fila da rede de filas, é novamente notório o equilíbrio entre as formulações entre as soluções com baixo gasto de área de circulação, até a solução 250. A partir deste ponto, é aparente que os melhores resultados são obtidos pela formulação 2, com menor despendimento de recursos.

No sistema de filas markovianos, com $CV^2 = 1{,}0$; novamente é possível notar soluções com menor gasto em área de circulação. Para a primeira fila da rede, o equilíbrio é presente nas soluções até 250. A formulação 1 forneceu soluções mais eficazes, com menor consumo de recursos. Na segunda fila dp sistema, o equilíbrio também está presente no baixo gasto de recursos para os *buffers*, demonstrando uma oscilação um pouco superior, para soluções com maior gasto em recursos, com um melhor desempenho para a formulação 1. Na terceira fila, o desempenho dos resultados é muito semelhante entre as duas formulações em todo o estudo, com uma mínima vantagem para a formulação 2 a partir da solução 300.

A respeito do sistema hiperexponencial, com $CV^2 = 1{,}5$; a paridade das soluções com baixo gasto de espaço alocado é visível. Na primeira fila da rede, também é equilibrado o consumo dos recursos, com um melhor desempenho nas soluções da formulação 1. Na segunda fila, o equilíbrio é maior em todo o estudo, porém a partir

da solução 300 a formulação 1 apresentou soluções minimamente mais eficientes. Na terceira fila, é presente o desempenho semelhante nas soluções até a solução 200. A partir deste ponto, com os maiores gastos de recursos de alocação dos *buffers*, a formulação 2 performou melhor, com melhores soluções.

A figura 14 representa graficamente a alocação total consumida para as três filas da rede. Tanto para os sistemas de hipoexponenciais, markovianos e hiperexponenciais é predominante o equilíbrio entre as soluções obtidas por ambas as formulações. Para as filas hipoexponenciais o desempenho é minimamente melhor para as soluções obtidas pela formulação 2, para soluções com maior gasto em espaços de alocação. Já para os sistemas de filas markovianos, nas soluções com maior gasto de recursos, a formulação 1 apresentou resultados discretamente mais eficientes. Nos sistemas hiperexponenciais é visível a vantagem para os resultados obtidos para a formulação 2.

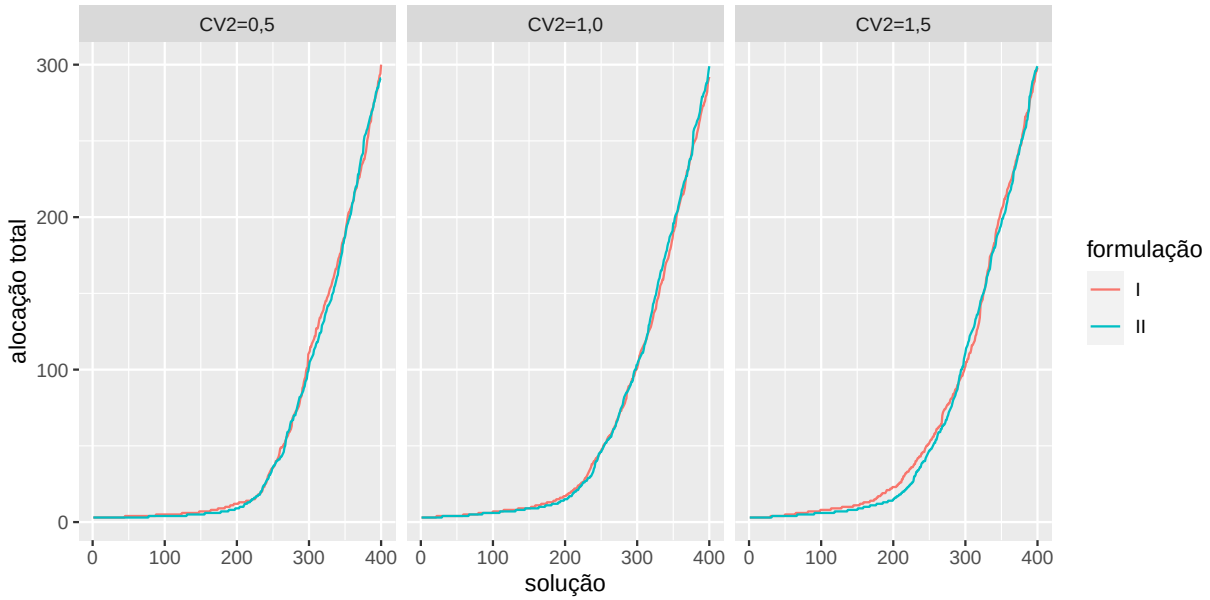


Figura 14: Alocação total de áreas de circulação através do NSGA-II para a topologia (divisão) da figura 8 (b).

A figura 15 apresenta os gráficos para os resultados obtidos pelo algoritmo NSGA-II, para as duas formulações. Nas colunas são representados os valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Nas linhas, as taxas de serviços para cada uma das três topolo-

gias descritas na figura 8 (b).

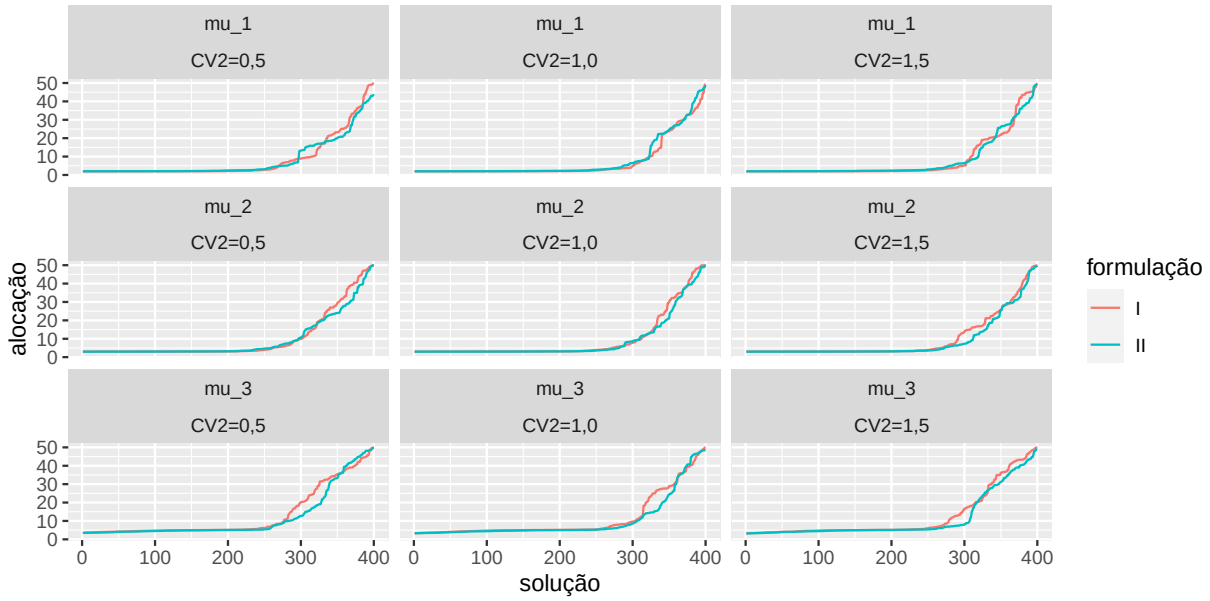


Figura 15: Recurso gasto em taxas de serviço entre filas através do NSGA-II para a topologia (divisão) da figura 8 (b).

Nos sistemas de filas hipoexponenciais, $CV^2 = 0{,}5$; pode se observar uma equiparidade nos resultados obtidos com baixo gasto de taxas de serviços. Entre as soluções 250 e 300 ocorre uma mudança no comportamento das soluções. A formulação 1 se mostrou menos eficaz que a formulação 2. Para a primeira fila da rede o equilíbrio é superior nos maiores gastos de recursos com taxas de serviço, porém a partir da solução 300 a formulação 2 é mais eficiente. Já para a segunda fila, o equilíbrio é ainda mais evidente. A formulação 1 tem resultados piores que os da formulação 2 após a solução 300. Na terceira fila, a formulação 2 se mostrou mais eficiente entre as soluções 280 e 350, a partir deste momento a formulação 1 apresentou resultados melhores. Porém no geral, a formulação 2 apresentou melhores resultados em taxas de serviços durante o estudo.

Com os sistemas de filas markovianas, as três filas da rede apresentam soluções muito semelhantes até próximo da solução 300, ou seja, as soluções de menor custo. Entretanto, na primeira fila da rede, um pouco antes da solução 300, a formulação 1 apresenta resultados minimamente melhores. Na segunda fila, o comportamento das soluções é também parecido que ao anterior, porém a formulação 1 tem

um desempenho inferior. Já para a terceira fila, é perceptivo o comportamento semelhante às demais redes de filas descritas na figura 8 (b). Entretanto, após a solução 300, a formulação 2 mostrou-se uma superioridade.

Para os sistemas de filas hiperexponenciais, com $CV^2 = 1{,}5$; também é notório o comportamento similar aos demais valores dos quadrado dos coeficientes de variação utilizados nas soluções “mais baratas” quanto ao gasto em taxas de serviço. Para a primeira fila da rede, as soluções encontradas por ambas formulações são semelhantes até a solução 250. Após este momento, existe alguma oscilação, predominando a formulação 2. Já para a segunda fila, também há um equilíbrio até próximo da solução 300, em que é um pouco mais visível que a formulação 2 trouxe melhores resultados. Para a terceira fila, o baixo gasto de recursos ocorreu até próximo da solução 300, em que é mais visível a vantagem da formulação 2 entre as soluções 270 e 320, e após este instante as duas formulações ficam mais equilibradas, mas ainda com ligeira superioridade para a formulação 1. Na figura 16, como previsto, após análise dos gráficos da figura 15, a formulação 2 se mostrou discretamente melhor nos três valores do coeficiente de variação.

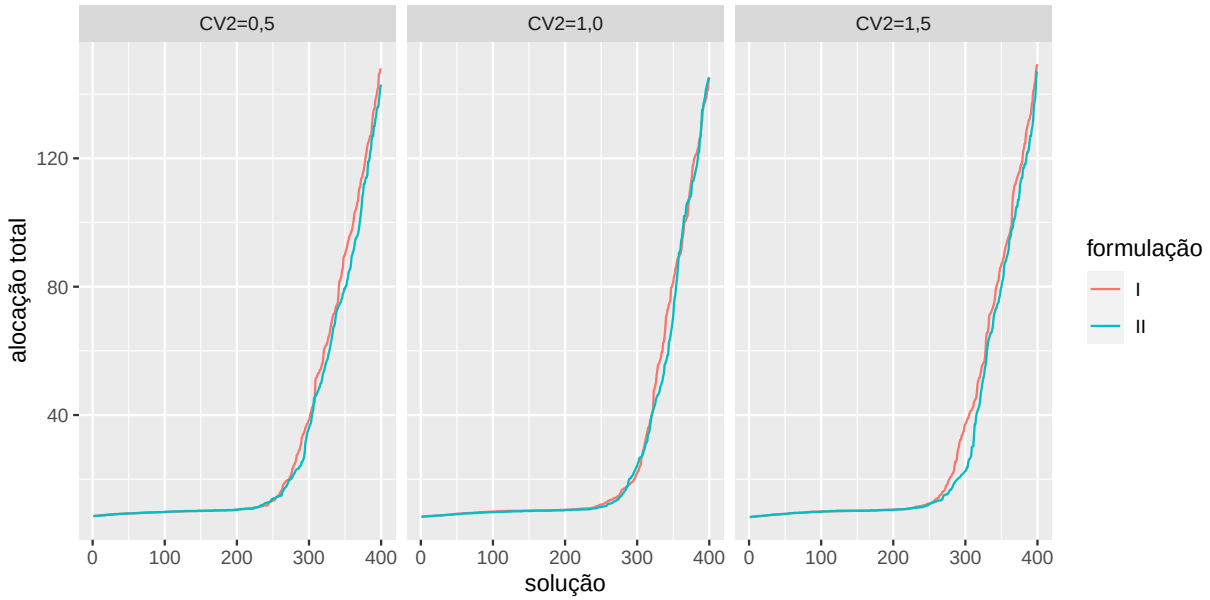


Figura 16: Recurso total gasto em taxas de serviço através do NSGA-II para a topologia (divisão) da figura 8 (b).

O efeito ocorre principalmente em situações com baixos gastos em taxas de

serviços. Até próximo da solução 250 o consumo de recursos é similar nas três redes de filas. Após este ponto, o gasto de recursos aumenta significativamente. Contudo, nos sistemas hipoexponenciais e markovianos o desempenho é similar, com discreta vantagem para soluções obtidas pela formulação 2. Já nos sistemas hiperexponenciais, o consumo de recursos também é semelhante entre os resultados gerados nas duas formulações, mas é visível que soluções encontradas pela formulação 2 são superiores.

A figura 17 representa os resultados obtidos pelo algoritmo NSGA-II, para cada uma das duas formulações. Por colunas são representados os três valores para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Já por linhas, são representadas as três filas da rede investigada. A rede em estudo é representada no esquema que pode ser visualizado na figura 8 (c).

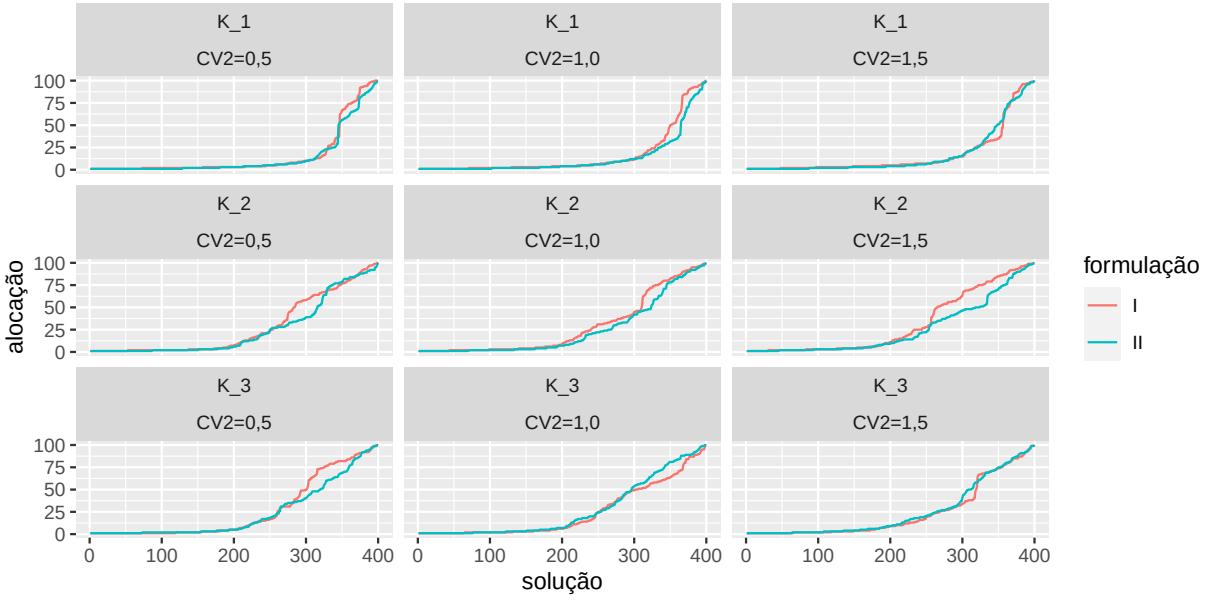


Figura 17: Alocação de áreas de circulação entre filas através do NSGA-II para a topologia (fusão) da figura 8 (c).

Nos sistemas de filas hipoexponenciais, com $CV^2 = 0{,}5$; observa-se similaridade nos resultados gerados com alocações menos onerosas, até a solução 250, para os três sistemas de filas. Para a primeira fila da rede, é visivelmente a semelhança nas soluções encontradas até a solução 300, após este momento a formulação 1 apresenta soluções superiores até a solução 330, contudo a partir deste momento,

a formulação 2 resultou em soluções melhores. Para a segunda fila, há constância nos resultados obtidos nas duas formulações até próximo da solução 260, após este ponto, os resultados gerados pela formulação 2 são melhores. Já para a terceira fila do sistema, é novamente notório o equilíbrio entre as soluções geradas por ambas as formulações, com baixo gasto de espaços (*buffers*), isso até a solução 260, a partir deste ponto, aparentemente os melhores resultados obtidos pela formulação 2 são visualmente superiores.

Em sistemas de filas markovianos, novamente é possível verificar equilíbrio nas soluções mais baratas. Para a primeira fila da rede, o equilíbrio é presente nas soluções até a solução 300, daí em diante, a formulação 1 trouxe soluções mais onerosas. Já para a segunda fila, a semelhança nos resultados também está presente no baixo gasto em recursos para os *buffers*. A partir da solução 200, as melhores soluções são as geradas pela formulação 2, em que há menor gasto em *buffers*. Para a terceira fila, o desempenho dos resultados até a solução 280 é muito semelhante entre as duas formulações, porém é visível uma vantagem para a formulação 1 nas soluções geradas a partir desse ponto.

Para os sistemas hiperexponenciais, a equiparidade nas soluções com baixo gasto de espaço alocado novamente prevalece. A primeira fila da rede, tem um equilíbrio maior no gasto dos recursos, com pequena vantagem em desempenho pela formulação 1, a partir da solução 330. Já para a segunda fila do sistema, o equilíbrio é menor entre as formulações nas soluções mais caras (a partir da solução 200), neste ponto, o desempenho da formulação 2 é superior. Na terceira fila, as soluções são bastante semelhantes, com leve superioridade para a formulação 2 nas soluções entre 300 e 330.

Apesar de observar na figura 17 resultados superiores para a formulação 1, ao observar os gráficos da figura 18, na alocação total para as filas hipoexponenciais é visível para a formulação 2 gerou resultados melhores após a solução 200. Já na rede de filas markovianas, o equilíbrio entre as soluções é maior, entretanto também na

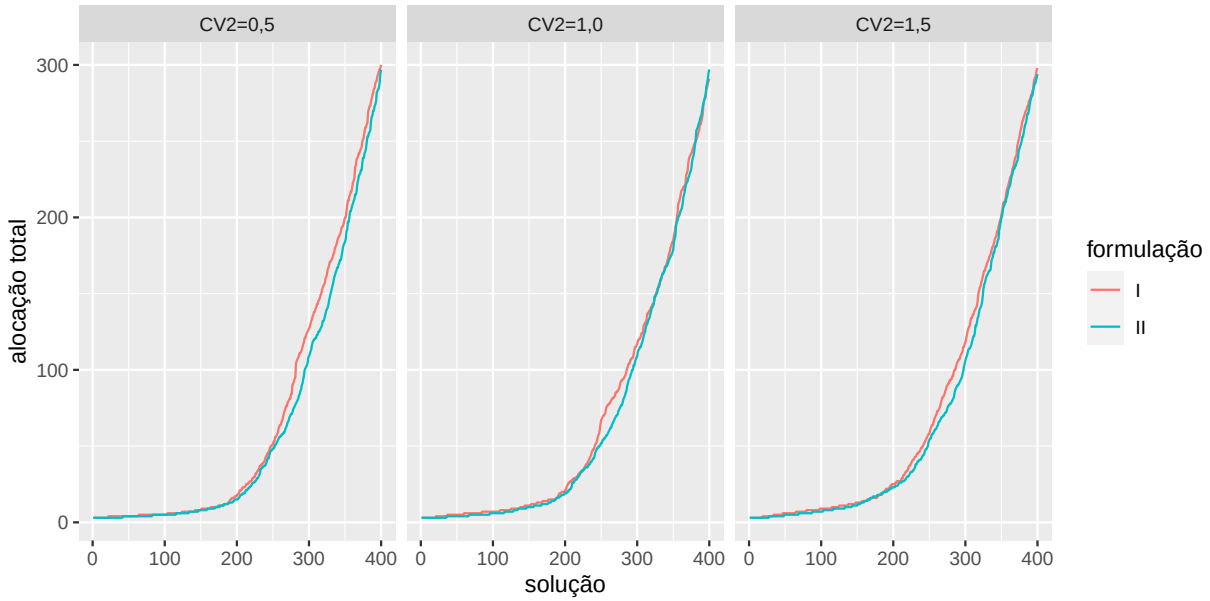


Figura 18: Alocação total de áreas de circulação através do NSGA-II para a topologia (fusão) da figura 8 (c).

alocação total das áreas de circulação, a formulação 2 também apresenta ganho em algumas soluções. Já para as filas hiperexponencias o comportamento é semelhante ao do sistema markoviano, com melhor desempenho para as soluções resultantes da formulação 2.

A figura 19, apresenta os gráficos para os resultados obtidos pelo algoritmo NSGA-II, para as duas formulações. Nas colunas são representados os valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Nas linhas, as taxas de serviços para cada uma das três topologias descritas na figura 8 (c). Para os sistemas de filas hipoexponenciais, $CV^2 = 0{,}5$; observa-se um equilíbrio entre os resultados obtidos com baixo gasto de taxas de serviços. Para a primeira fila, o consumo em taxas de serviços sofre uma oscilação de comportamento para as soluções obtidas em ambas as formulações, no geral as melhores soluções foram originadas pela formulação 2. Já para a segunda fila da rede, ocorre maior equilibrio em consumo de recursos para taxas de serviço, com discreta vantagem para a formulação 2 após a solução 320. Para a terceira fila da rede, a formulação 2 também foi mais eficiente após a solução 300.

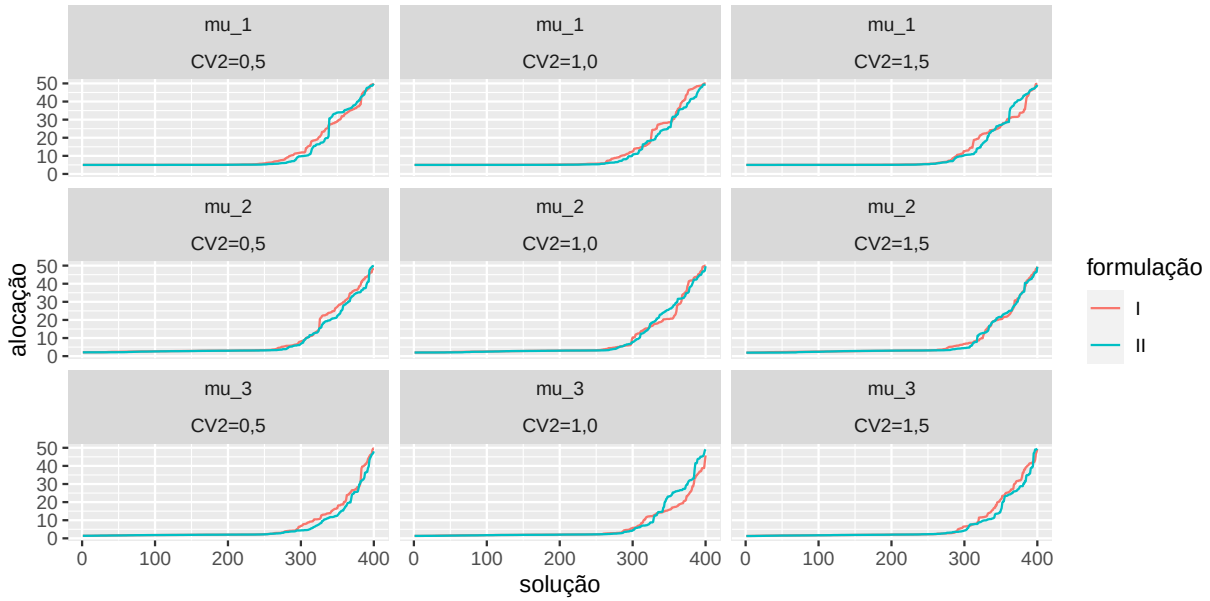


Figura 19: Recurso gasto em taxas de serviço entre filas através do NSGA-II para a topologia (fusão) da figura 8 (c).

Com os sistemas de filas markovianas, as três filas da rede também apresentam soluções semelhantes. Porém, na primeira fila da rede, um pouco antes da solução 300 a formulação 1 demonstrou resultados menos vantajosos. Já na segunda fila, o comportamento das soluções é também parecido com o anterior, novamente a formulação 2 resulta em soluções mais eficientes. Já para a terceira fila do sistema, é perceptivo o comportamento semelhante as demais filas da rede, entretanto após a solução 300, a formulação 1 apresentou melhores soluções.

Para os sistemas de filas hiperexponenciais, com $CV^2 = 1{,}5$; é notório um comportamento similar para os valores dos quadrados dos coeficientes de variação investigados, com gasto praticamente igual quando há o baixo gasto de recursos em taxas de serviço, entre as filas. Porém, para a primeira fila da rede, as melhores soluções encontradas por ambas as formulações propostas, para as taxas de serviços oscilam a partir da solução 300. Com uma pequena vantagem para as soluções resultantes pela formulação 2, e depois da solução 360, a formulação 1 parece mais vantajosa. Já para a segunda fila do sistema, o equilíbrio ocorre em todas as soluções. Na terceira fila, as soluções são similares até a solução 300, depois disso, a formulação 2 parece um pouco mais adequada.

A figura 20 apresenta o desempenho total dos recursos gastos em taxas de serviços, para a topologia da figura 8 (c).

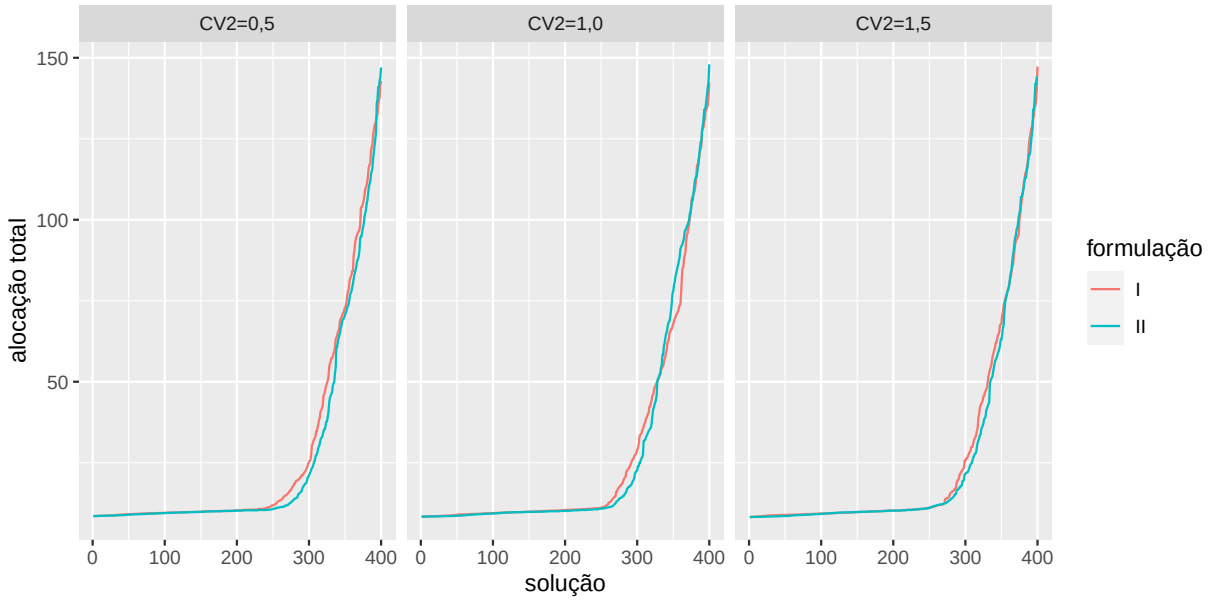


Figura 20: Recurso total gasto em taxas de serviço através do NSGA-II para a topologia (fusão) da figura 8 (c).

É visível o comportamento similar entre os três valores utilizados para quadrados dos coeficientes de variação. A formulação 2 é superior em todas as situações quando os consumo em recursos começa a atingir os patamares mais elevados.

4.2 AVALIAÇÃO DA EVOLUÇÃO VIA PÓS-PROCESSAMENTO COM *SIMULATED ANNEALING* PARA FORMULAÇÃO 1

Nesta seção, o objetivo é discutir os resultados gerados pelo procedimento de pós-processamento através do algoritmo *Simulated Annealing*, para as soluções encontradas incialmente pela formulação 1, utilizando o algoritmo NSGA-II. O interesse novamente comparativo se restringe ao espaço das variáveis (recursos em *buffers* e servidores), para o procedimento do pós-processamento com o *Simulated Annealing*, as alterações efetivas em busca de novas soluções, ocorre somente para os recursos em espaços de alocação (*buffers*), impactando nas taxas de serviços.

A figura 21 apresenta graficamente as soluções obtidas para as áreas de circulação entre filas antes (NSGA-II) e após o pós-processamento via *Simulated Annealing*

(NSGA-II+SA) para a formulação 1 aplicado a topologia da figura 8 (a).

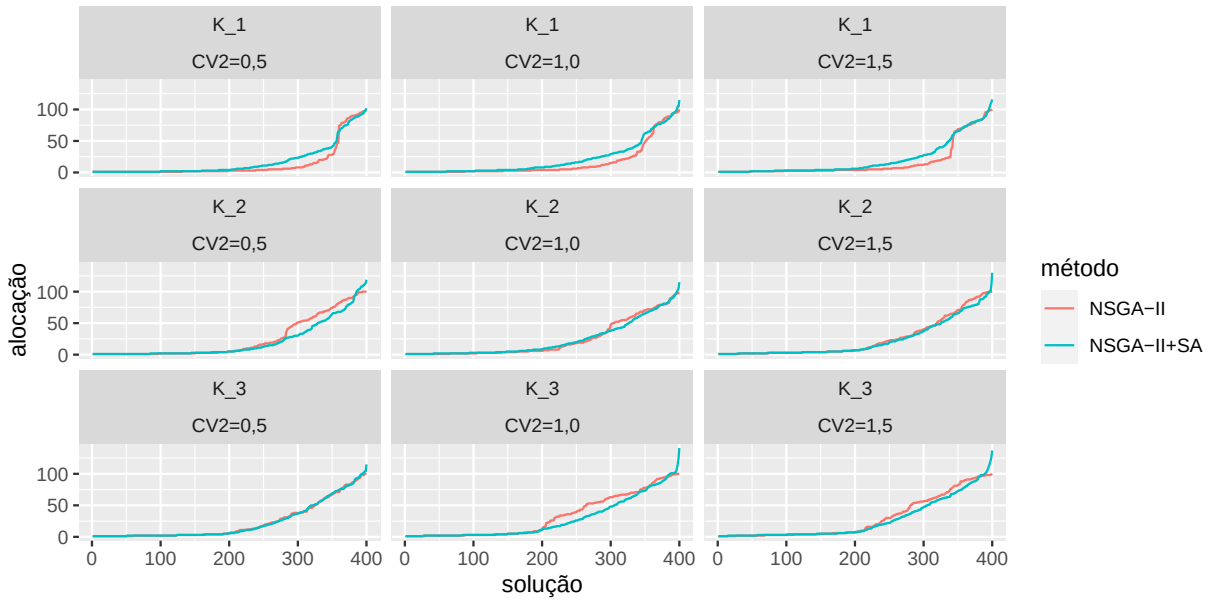


Figura 21: Alocação de áreas de circulação entre filas antes e após o pós-processamento via *Simulated Annealing* para a topologia (série) da figura 8 (a).

Para os sistemas hipoexponenciais, o comportamento das soluções obtidas em ambas as metodologias são semelhantes nas três redes de filas apresentadas na figura 8 (a), em geral até a solução 200. As 200 primeiras soluções, são situações com consumo de recursos em alocação de *buffers* visualmente baixo e parecido em todas as filas do sistema. Já a partir da solução 200, em particular, na primeira fila da rede, observa-se que até a solução 350, o pós-processamento levou para soluções com maior consumo de recursos.

Entretanto, soluções com maior gasto em recurso, posteriores à solução 350 ocorreu alguma melhora em economia de recursos decorrente do pós-processamento. Já para a segunda fila do sistema, diferente da primeira fila observada anteriormente, o equilíbrio encontrado nas soluções de ambos os métodos alcança a solução 250, em seguida, observa-se que o algoritmo *Simulated Anneling* trouxe ganhos reais para as soluções. Na terceira fila, visualmente o desempenho das duas metodologias é semelhante para todas as 400 soluções.

Em sistemas markovianos, novamente verifica-se bastante similaridade entre

as primeiras soluções, com um menor consumo de recursos em espaço de alocação. Como na verificação do caso hipoexponencial, para a primeira fila da rede de filas, o pós processamento propôs alocações menos onerosas. Na segunda fila da rede de filas, as soluções são bastante semelhantes mesmo após o pós-processamento, mas um pequeno ganho marginal já é observado para a estratégia de pós-processamento. Na terceira fila, o pós-processamento revela ganhos mais significativos entre as soluções 200 e 350.

Para sistemas hiperexponenciais, com $CV^2 = 1{,}5$; a similaridade das soluções com menos consumo de recurso em alocação de espaços é mantida. Na primeira fila da rede, destaca-se novamente que o pós-processamento levou a redução do consumo de recurso, na prática, fica clara uma tendência de que o pós-processamento consumiu mais recursos na primeira fila do sistema. Já na segunda fila, existe novamente uma regularidade em todo o período. Para a terceira fila do sistema, a partir da solução 200 é visível o menor consumo em recursos nas soluções pós-processadas. A figura 22 apresenta a alocação total de áreas de circulação antes e após o pós-processamento através do algoritmo *Simulated Annealing* para a topologia da figura 8 (a).

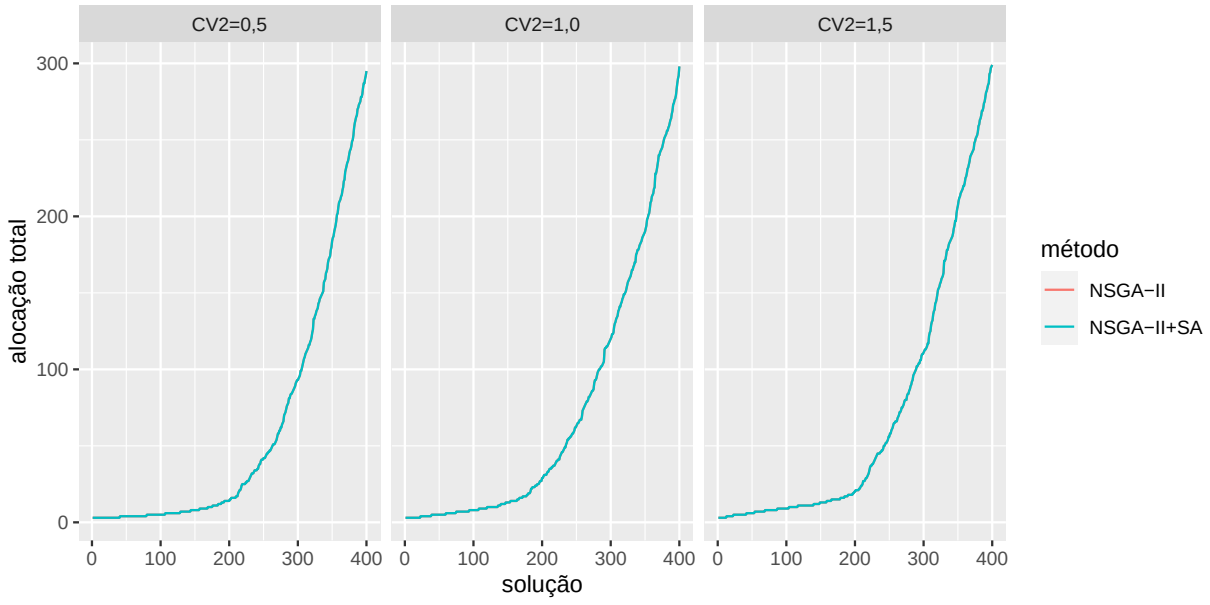


Figura 22: Alocação total de áreas de circulação antes e após o pós-processamento via *Simulated Annealing* para a topologia (série) da figura 8 (a).

Não existe distinção para as curvas entre os quadrados dos coeficientes de variação pelas duas metodologias, pois o algoritmo *Simulated Annealing*, não aumenta ou reduz o número de recursos gastos entre os *buffers*, somente explora o reposicionamento dos espaços já existentes, não interferindo o desempenho do espaço total do sistema já definido.

Por este motivo, as curvas entre as soluções dos dois métodos são coincidentes e são iguais a curva exibida na figura 10. A figura 22 tem um caráter muito mais de validação da proposição que propriamente um caráter informativo. Em virtude disso, a representação gráfica para a alocação total de áreas de circulação nas demais topologias será apresentada para a comparação entre as soluções NSGA-II e NSGA-II+SA.

A estratégia de pós-processamento via *Simulated Annealing* altera as taxas de serviço, portanto o mesmo efeito detectado na figura 22 se repete para as análises de consumo de recursos em taxas de serviço. As figuras 23 e 24 serão apresentadas meramente com caráter validativo e a análise similar será omitida para as demais topologias em investigação.

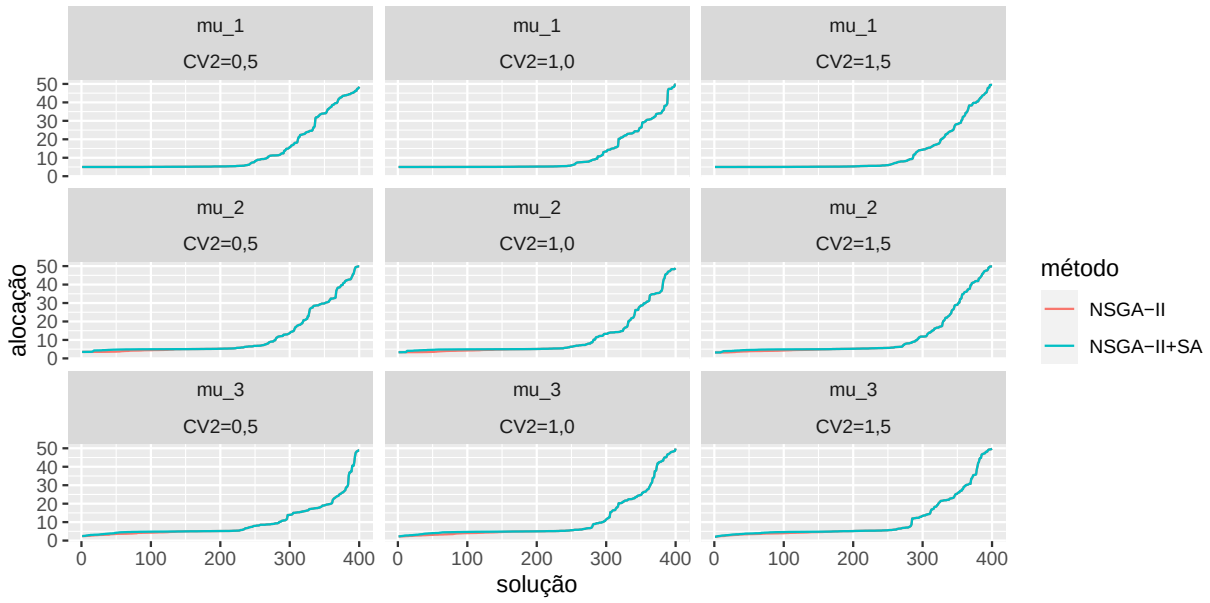


Figura 23: Recurso gasto em taxas de serviço entre filas antes e após o pós-processamento via *Simulated Annealing* para a topologia (série) da figura 8 (a).

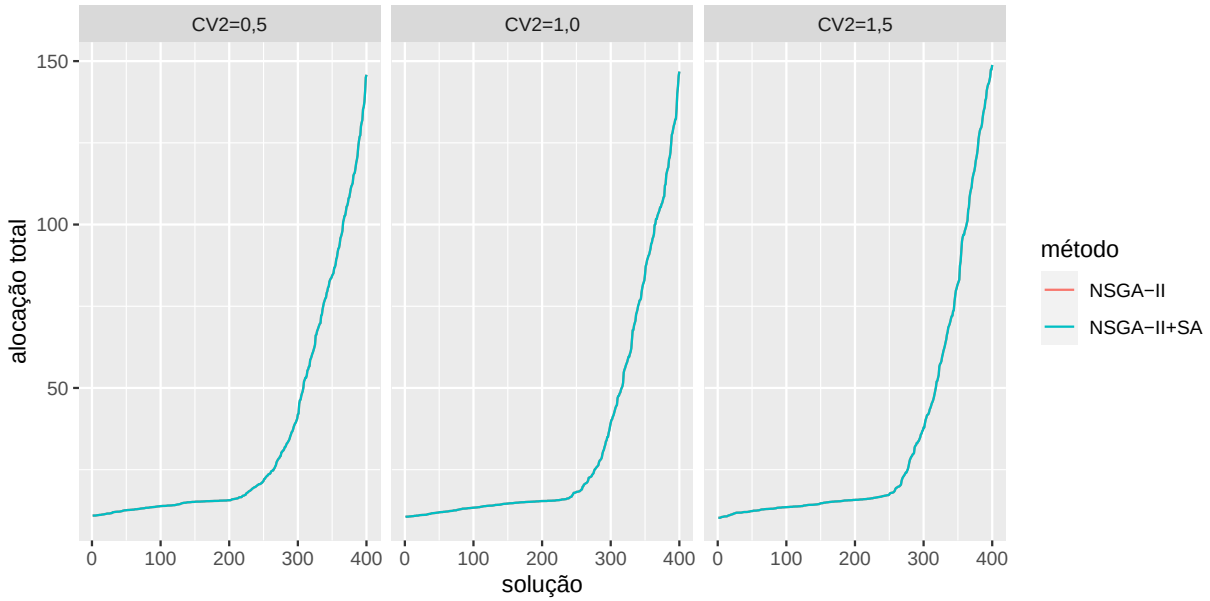


Figura 24: Recurso total gasto em taxas de serviço antes e após o pós-processamento via *Simulated Annealing* para a topologia (série) da figura 8 (a).

A figura 25 apresenta graficamente as soluções obtidas para as áreas de circulação entre filas antes (NSGA-II) e após o pós-processamento via *Simulated Annealing* (NSGA-II+SA) para a formulação 1 aplicado a topologia da figura 8 (b).

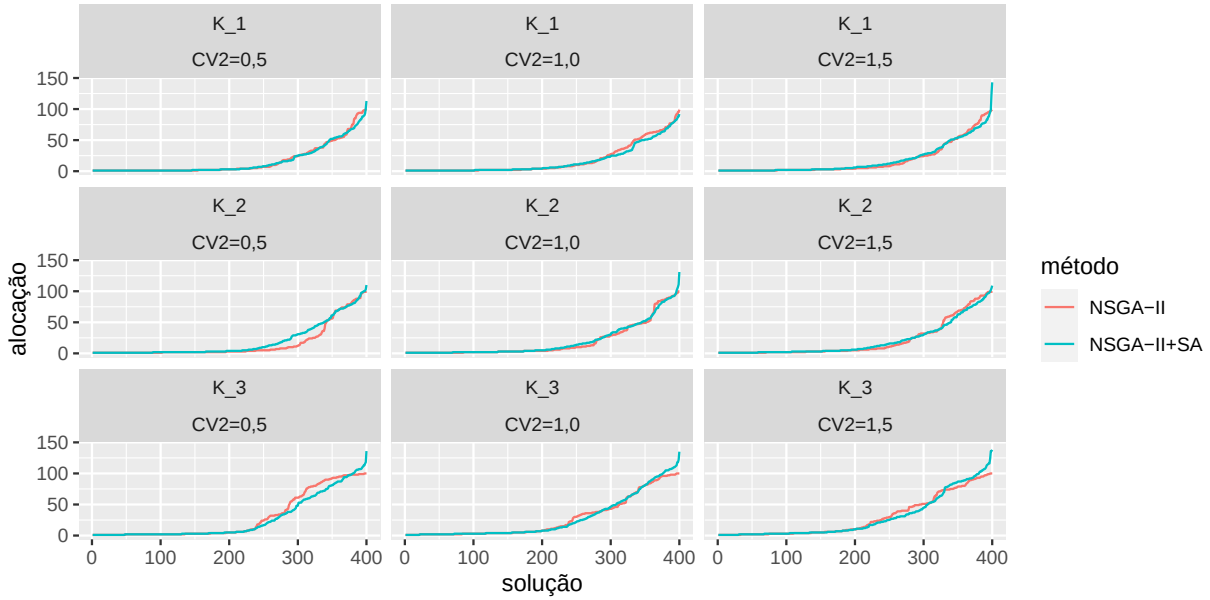


Figura 25: Alocação de áreas de circulação entre filas antes e após o pós-processamento via *Simulated Annealing* para a topologia (divisão) da figura 8 (b).

Em sistemas hipoexponenciais, o desempenho das soluções obtidas apresenta uma troca de posicionamento dos recursos de espaço de alocação entre as filas

2 e 3 na rede apresentada na figura 8 (b). Para a primeira fila da rede, observa-se soluções bastante similares antes e após o pós-processamento. Já na segunda fila da rede de filas, diferente da primeira, entre as soluções 200 e 350 as soluções NSGA-II consomem menos recursos em circulação que as soluções pós-processadas. Na terceira fila, visualmente o pós-processamento acarretou na redução de espaço de circulação alocado para uma grande parte das soluções.

Nos sistemas markovianos, observa-se uma grande semelhança entre alocação proposta para as três filas via NSGA-II e após o pós-processamento. As variações são bem sutis e existem alterações significativas. Para os sistemas hiperexponenciais, novamente uma grande similaridade é vista. As variações são tão sutis quanto no caso markoviano, mas são ainda pequenas. Como mencionado anteriormente, as demais representações gráficas utilizadas neste estudo se aplicam às análises na comparação antes e após o pós-processamento via *Simulated Annealing*.

A figura 26 apresenta graficamente as soluções obtidas para as áreas de circulação entre filas antes (NSGA-II) e após o pós-processamento via *Simulated Annealing* (NSGA-II+SA) para a formulação 1 aplicado a topologia da figura 8 (c).

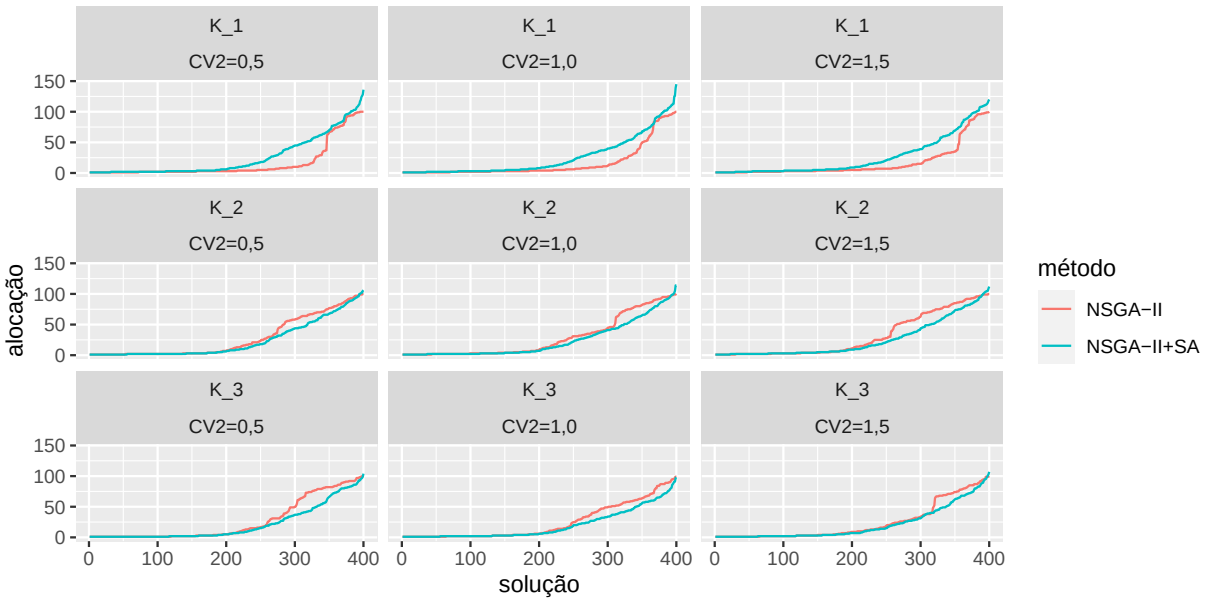


Figura 26: Alocação de áreas de circulação entre filas antes e após o pós-processamento via *Simulated Annealing* para a topologia (fusão) da figura 8 (c).

O desempenho das soluções obtidas parece revelar uma troca de posicionamento dos recursos de espaço de alocação da fila 1 para as filas 2 e 3 na rede apresentada na figura 8 (c). Para a primeira fila da rede, a soluções NSGA-II, antes do pós processamento sempre consomem menos recursos em áreas de circulação. Este efeito é mais evidente no sistema hipoexponencial, mas está presente para todos os coeficientes de variação investigados. Para as filas 2 e 3, a solução pós-processada utiliza menos recursos que a solução prévia, mostrando o efeito de realocação de recursos proposto pelo *Simulated Annealing*. O efeito é notório para todos os coeficientes de variação em estudo. Análogo as investigações anteriores sobre o pós-processamento através do *Simulated Annealing*, as demais representações gráficas utilizadas neste estudo se aplicam aqui.

4.3 AVALIAÇÃO DA EVOLUÇÃO VIA PÓS-PROCESSAMENTO COM *PARTICLE SWARM OPTIMIZATION* PARA FORMULAÇÃO 2

A presente seção apresenta resultados para a formulação 2 do problema de otimização. Uma comparação entre soluções geradas através do algoritmo NSGA-II e posteriormente pós-processadas via algoritmo *Particle Swarm Optimization*.

A figura 27 retrata os resultados obtidos para a alocação de áreas de circulação entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia da figura 8 (a). Nos sistemas hipoexponenciais, com $CV^2 = 0{,}5$; o desempenho das soluções obtidas em ambas as metodologias são semelhantes nas três filas da rede até a solução 200 aproximadamente. À medida que o consumo de recursos de alocação dos espaços (*buffers*) cresce as soluções ficam distintas. Na primeira fila da rede, é visível que a similaridade se mantém até próximo da solução 300, a partir daí as soluções pós-processadas tendem a consumir mais recursos. Este efeito se repete na primeira fila da rede também para sistemas markovianos e hiperexponenciais, porém a vantagem das soluções obtidas através do algoritmo NSGA-II já aparece a partir da solução 250. Para as demais filas, a alocação proposta nas soluções pós processadas se mostra mais vantajosa a partir da solução 200 para todos

os valores de coeficientes de variação. Na terceira fila, os casos hipoexponencial e markoviano revelam uma economia de recursos maior que o caso hiperexponencial.

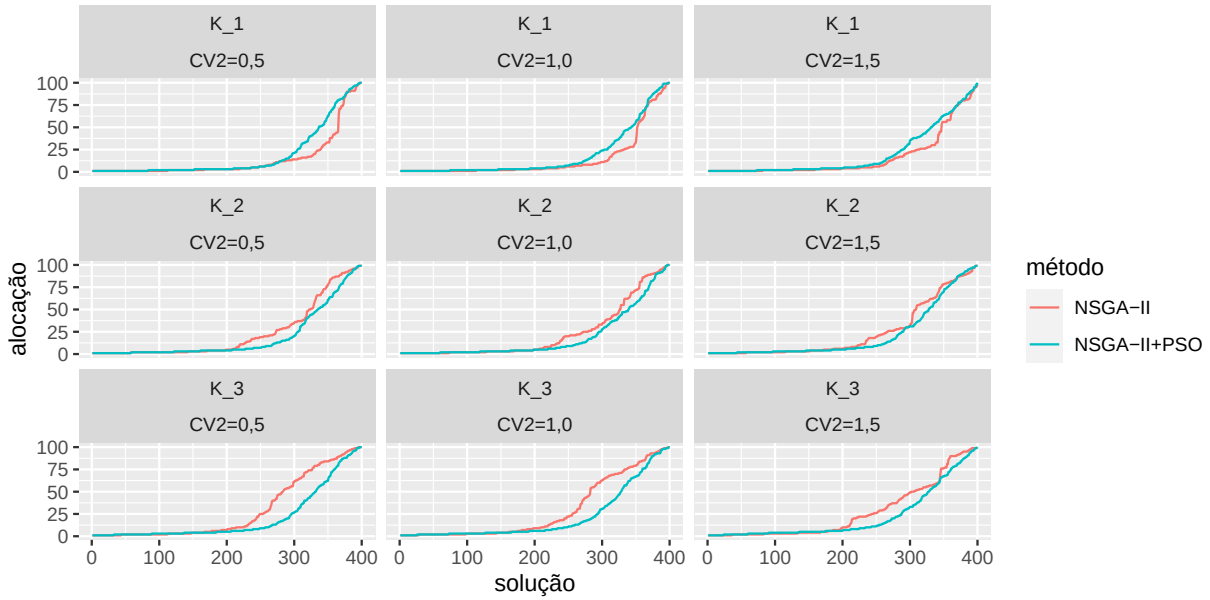


Figura 27: Alocação de áreas de circulação entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

A figura 28 exibe os gráficos para a alocação total para as áreas de circulação antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia da figura 8 (a).

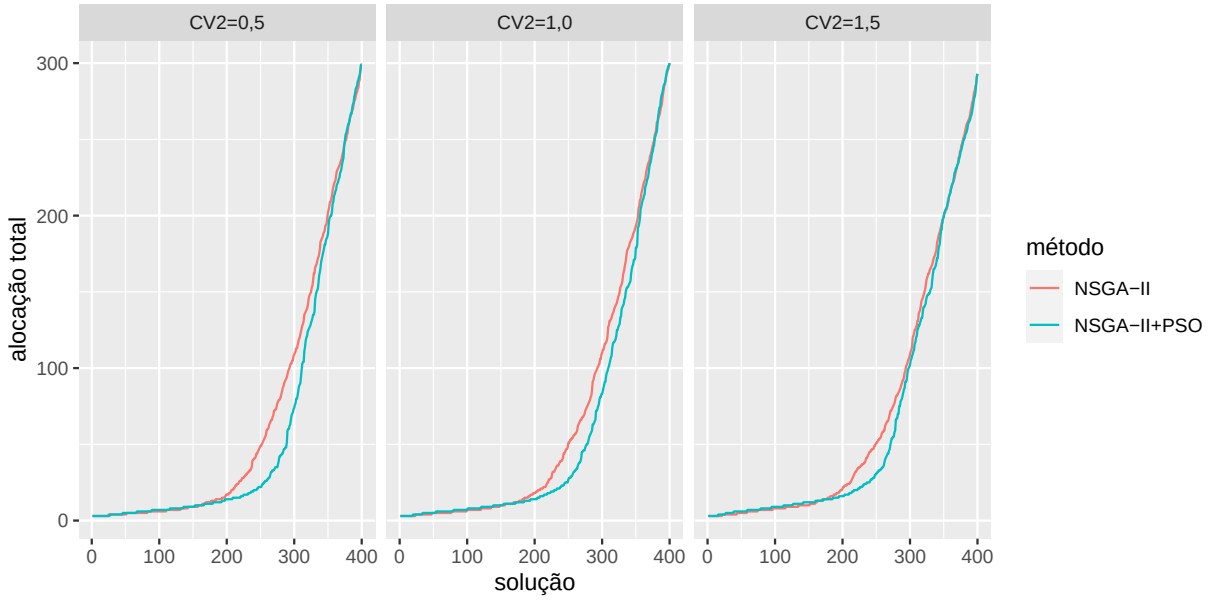


Figura 28: Alocação total de áreas de circulação antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

A avaliação fila a fila, observada na figura 27, fica evidenciada na alocação total exibida na figura 28. Para as soluções que exigem maior consumo de recursos em alocação de *buffers*, o pós-processamento se mostra bem adequado. Para uma análise ligada ao consumo de recursos em taxas de serviço, a figura 29 avalia graficamente as soluções encontradas para o recurso gasto antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia da figura 8 (a).

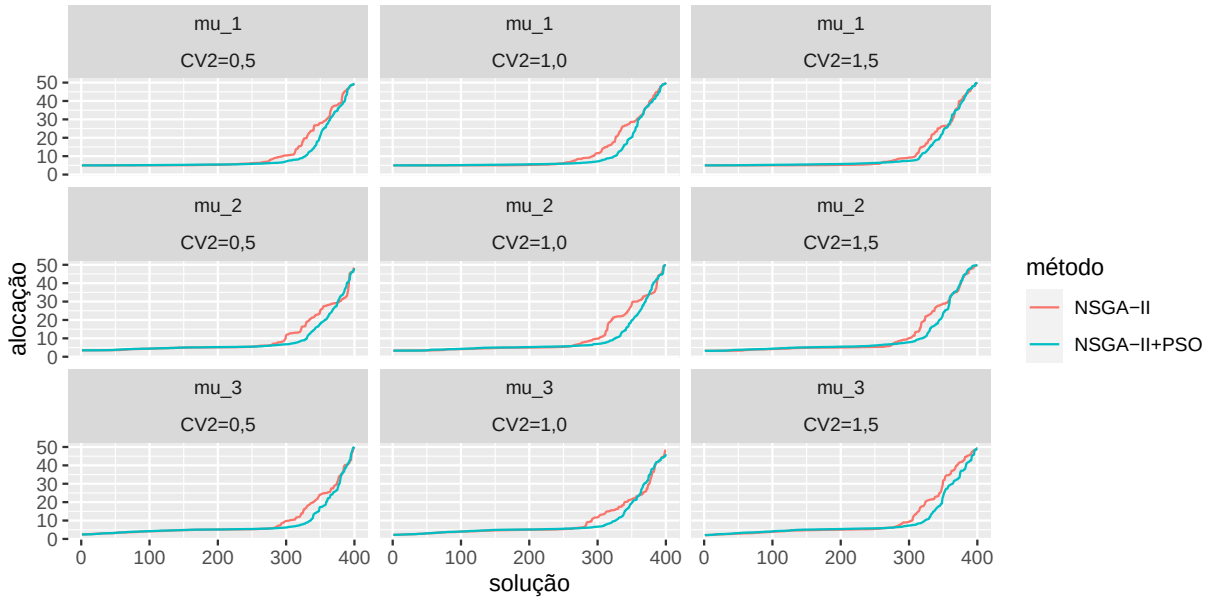


Figura 29: Recurso gasto em taxas de serviço entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

Novamente parece claro que o pós-processamento agrega mudanças efetivas para soluções com baixo consumo em taxas de serviço (este efeito é similar ao verificado na alocação de áreas de circulação). Porém, para a análise das taxas de serviço, em todas as filas da rede, e para todos os coeficientes de variação, a estratégia de pós-processamento apresenta soluções com menor consumo de recursos em taxas de serviço. Em geral, este efeito aparece a partir da solução 300. Para a segunda fila da rede, aproximadamente nas 30 soluções com maior consumo de recurso, as soluções originais NSGA-II são superiores por uma pequena margem.

A figura 30 apresenta o recurso total gasto em taxas de serviço antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia da figura 8 (a), para os três sistemas de filas somadas.

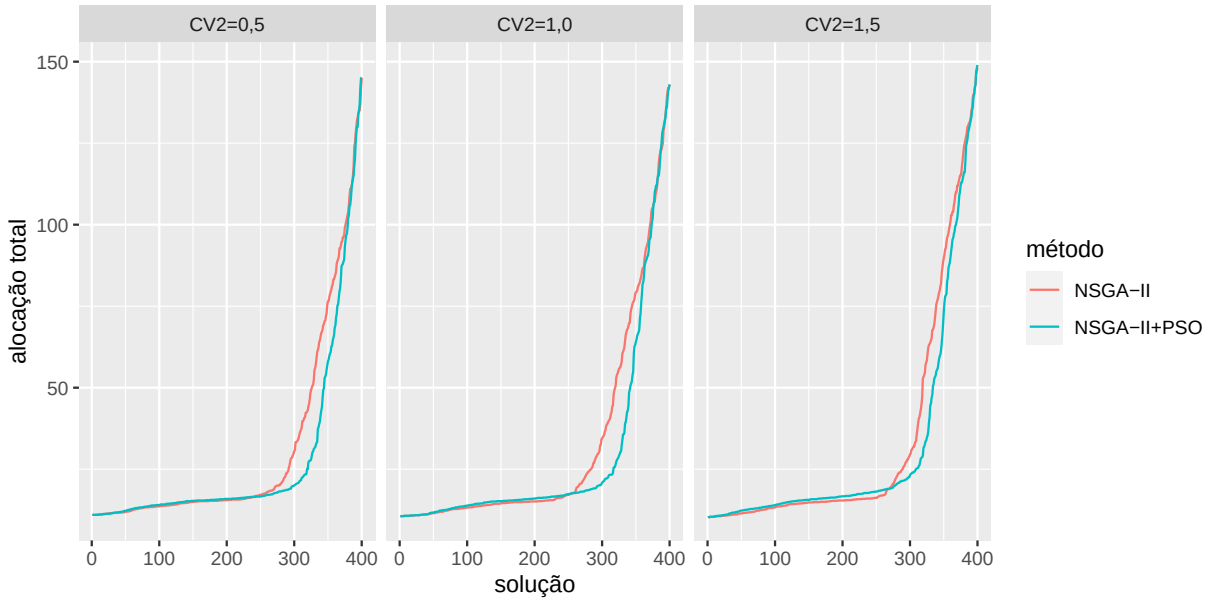


Figura 30: Recurso total gasto em taxas de serviço antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

A superioridade das soluções pós-processados, quando considerado o cenário de maior gasto em recursos é notória. Aparentemente, os ganhos são mais efetivos em sistemas hipoexponenciais e markovianos, mas ainda assim existem ganhos relevantes em sistemas hiperexponenciais. As figuras 28 e 30, em conjunto, ilustram que o pós-processamento é capaz de fornecer soluções com menor consumo de recursos.

A figura 31 repete a análise fila a fila para a alocação de *buffers* agora para a topologia da figura 8 (b). Aqui, o pós-processamento forneceu mudanças efetivas para a soluções de baixa alocação de recursos. Entretanto, para soluções com maior alocação, o pós-processamento pareceu fornecer menor consumo de áreas de circulação nos sistemas markovianos. Este efeito apareceu nas três filas da rede, sendo claramente maior na terceira fila do sistema. Alguma oscilação foi observada na segunda fila da rede, mas ainda sim a estratégia de pós-processamento parece mais eficaz. Para os sistemas hipoexponenciais, o pós-processamento pareceu vantajoso na alocação de espaços nas filas 1 e 3, porém desvantajoso na segunda fila. Por fim, para os sistemas hiperexponenciais, apesar da presença de alguma oscilação, globalmente as soluções parecem se equipararem.

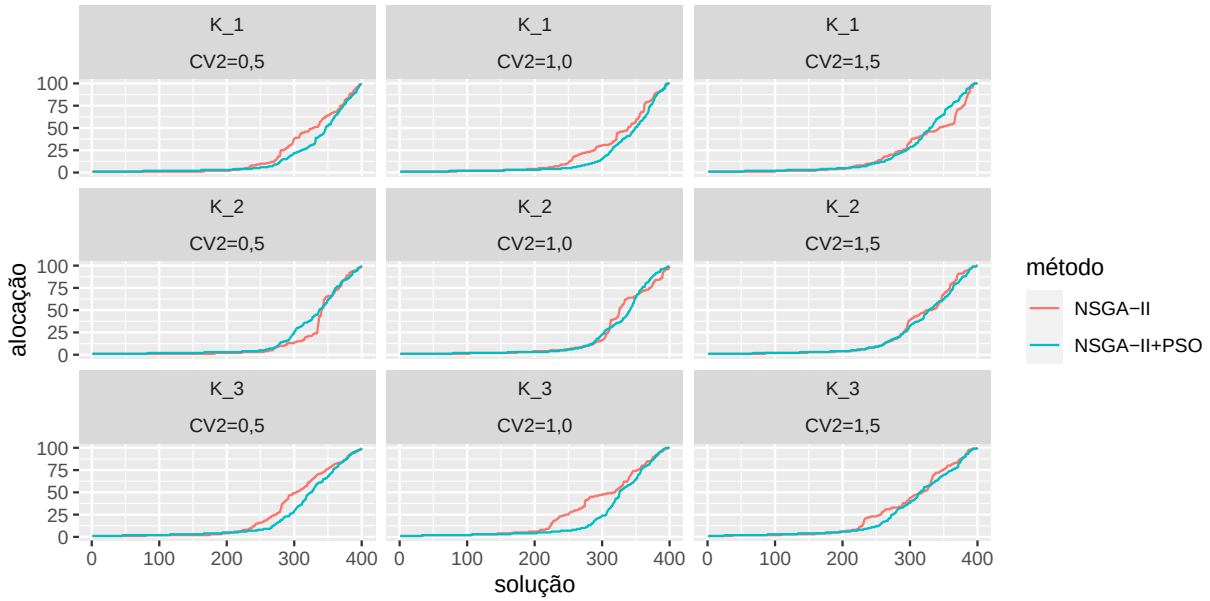


Figura 31: Alocação de áreas de circulação entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

A análise com a soma dos espaços de circulação alocados em todo o sistema é apresentada na figura 32.

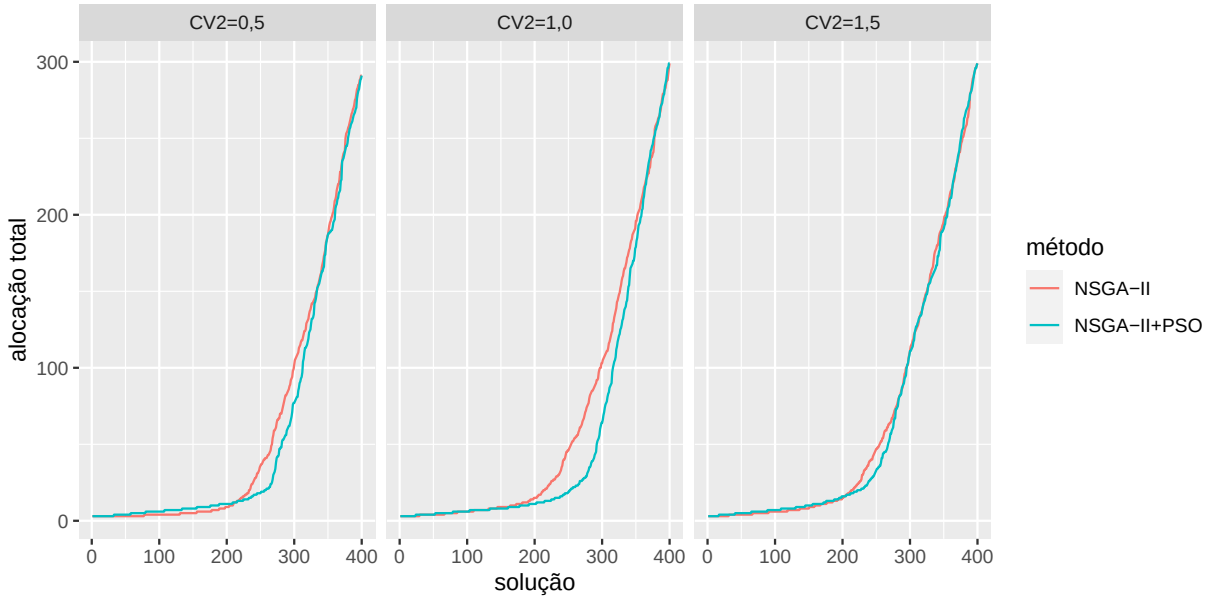


Figura 32: Alocação total de áreas de circulação antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

A constatação de maior efetividade para redução de alocação em áreas de circulação para sistemas markovianos fica bastante evidente aqui. Assim como a verificação de que nos sistemas hipoexponenciais ainda existe um ganho representativo

e que existe equilíbrio entre as soluções quando o sistema possui atendimento hiperexponencial. Entretanto, na análise da alocação total é possível observar que mesmo para sistemas hiperexponenciais existe um ganho marginal favorável às soluções pós-processadas.

A figura 33 mostra a análise gráfica do consumo de recurso em taxas de serviço para as três filas da rede. São apresentadas as soluções antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia da figura 8 (b).

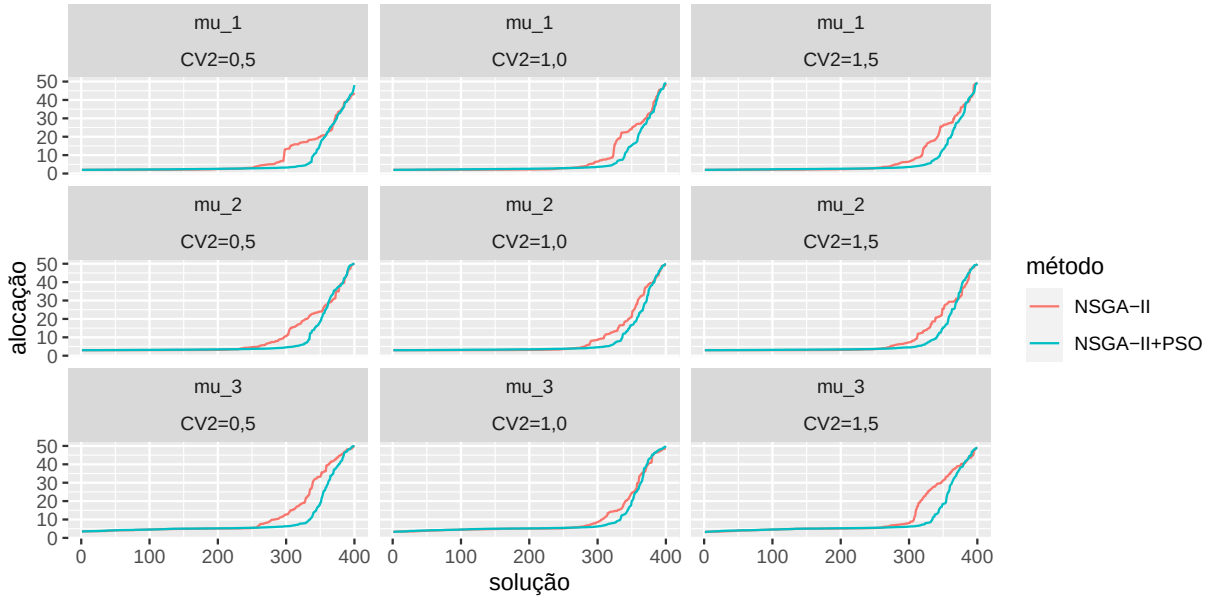


Figura 33: Recurso gasto em taxas de serviço entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

Como avaliado anteriormente, o pós-processamento influenciou as soluções de baixo custo em taxas de serviço. Contudo, em todas as filas da rede, e para todos os coeficientes de variação, nas soluções com maior alocação, o pós-processamento reduziu o gasto em taxas de serviço. E efeito foi menos significativo no sistema markoviano, um contraponto ao efeito verificado para as áreas de circulação. Tanto os sistemas hipoexponenciais quanto os hiperexponenciais apresentaram ganhos efetivos em redução de gastos com taxas de serviço.

A superioridade das soluções pós-processados, com respeito às taxas de serviço é evidenciada através de análise da soma do recurso total consumido em taxa de

serviços para as três filas da rede para a topologia da figura 8 (b). A figura 34 mostra este efeito nas soluções pós-processadas para redes de filas com atendimentos hipoexponencial, markoviano e hiperexponencial.

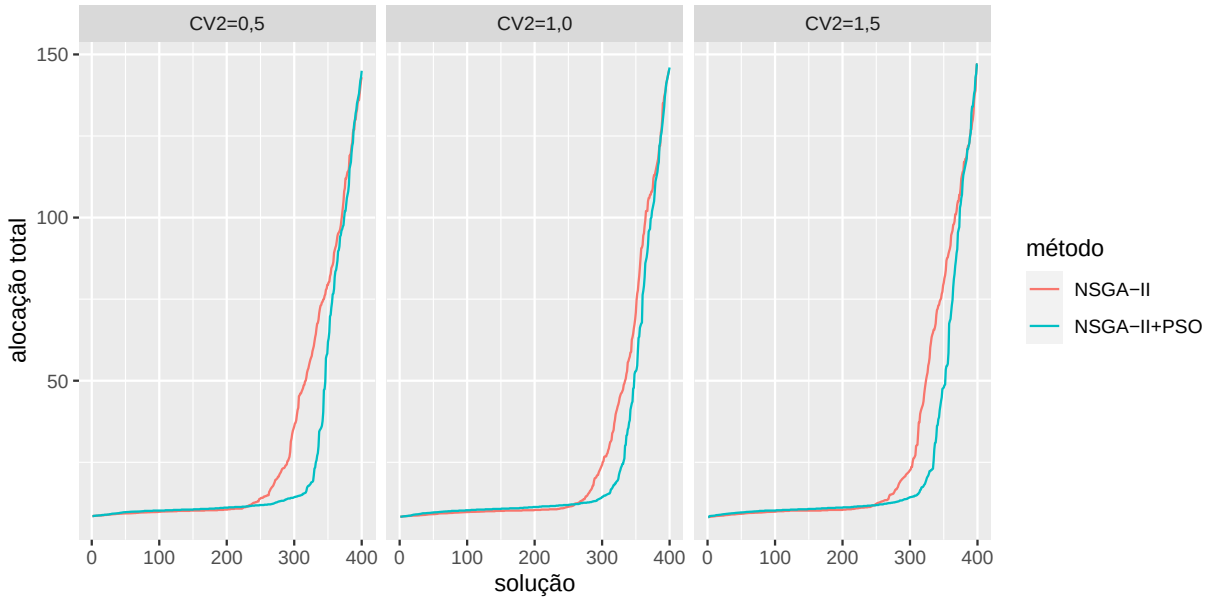


Figura 34: Recurso total gasto em taxas de serviço antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

As soluções pós-processados novamente refletem uma redução no consumo de recursos em taxas de serviço. O maior ganho aparece em sistemas de atendimento hipoexponencial, enquanto a menor redução aparece em sistemas markovianos. Uma análise conjunta das figuras 32 e 34 demonstram que efetivamente a estratégia de pós-processamento produz soluções com menor consumo de recursos que ainda são bastante efetivas para o problema em estudo.

A figura 35 apresenta os resultados obtidos para a alocação de áreas de circulação entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia da figura 8 (c) com fusão de duas filas. As soluções fornecidas pelo NSGA-II, na primeira fila do sistema, são sempre menos onerosas que as soluções pós-processadas. Já as outras duas filas da rede apresentam soluções pós-processadas menos onerosas. O desempenho é similar para todos os coeficientes de variação investigados, entretanto existe um equilibrio maior nas soluções das filas 2 e 3 quando o atendimento é hiperexponencial.

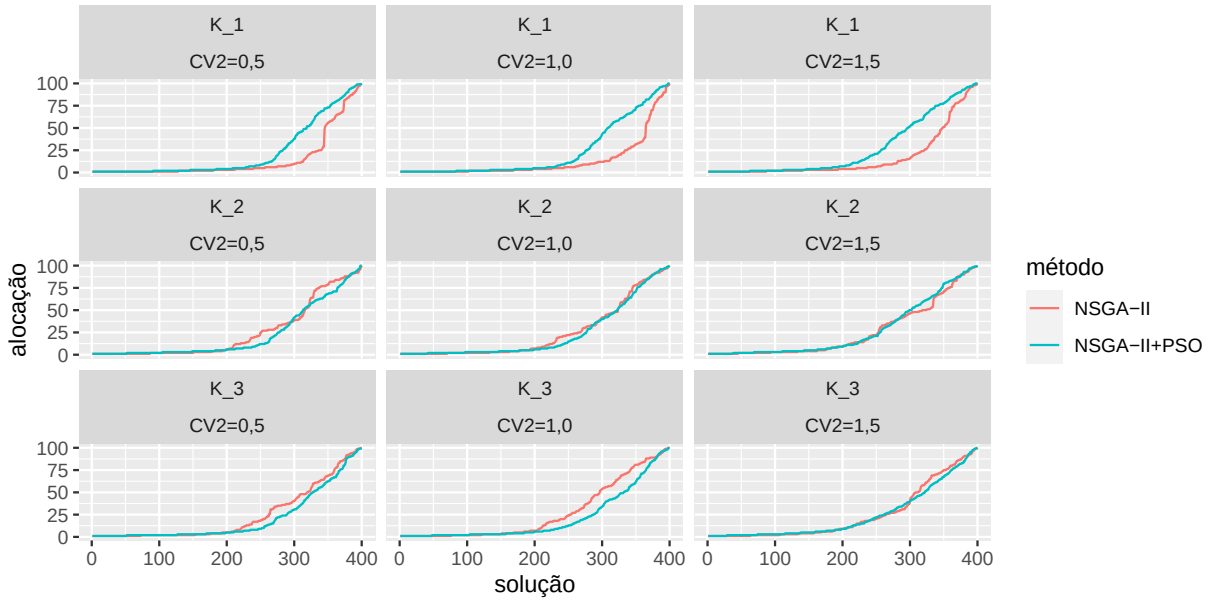


Figura 35: Alocação de áreas de circulação entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

A análise com a soma dos espaços de circulação alocados em todo o sistema é apresentada na figura 36. Existe algum equilíbrio entre as soluções fornecidas pelo NSGA-II e as soluções pós-processadas. Mas alguma vantagem para as soluções do NSGA-II (antes do pós-processamento) podem ser observadas. Isso ocorre principalmente no cenário de atendimento hiperexponencial.

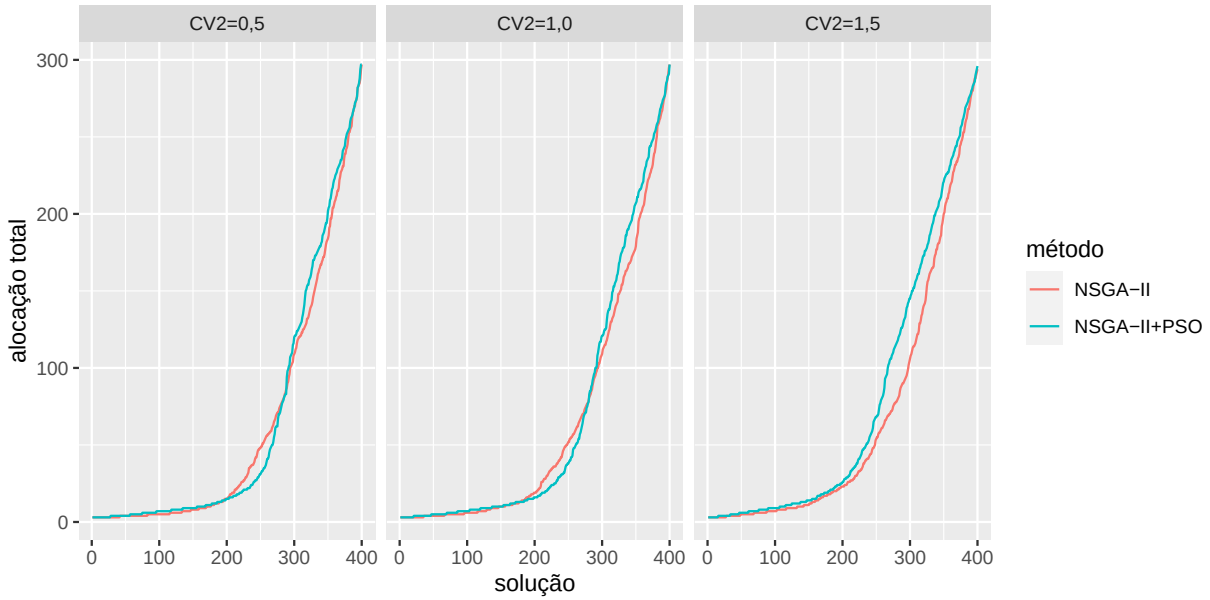


Figura 36: Alocação total de áreas de circulação antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

A figura 37 ilustra graficamente o gasto em recurso com taxas de serviço para as três filas da rede da topologia da figura 8 (c).

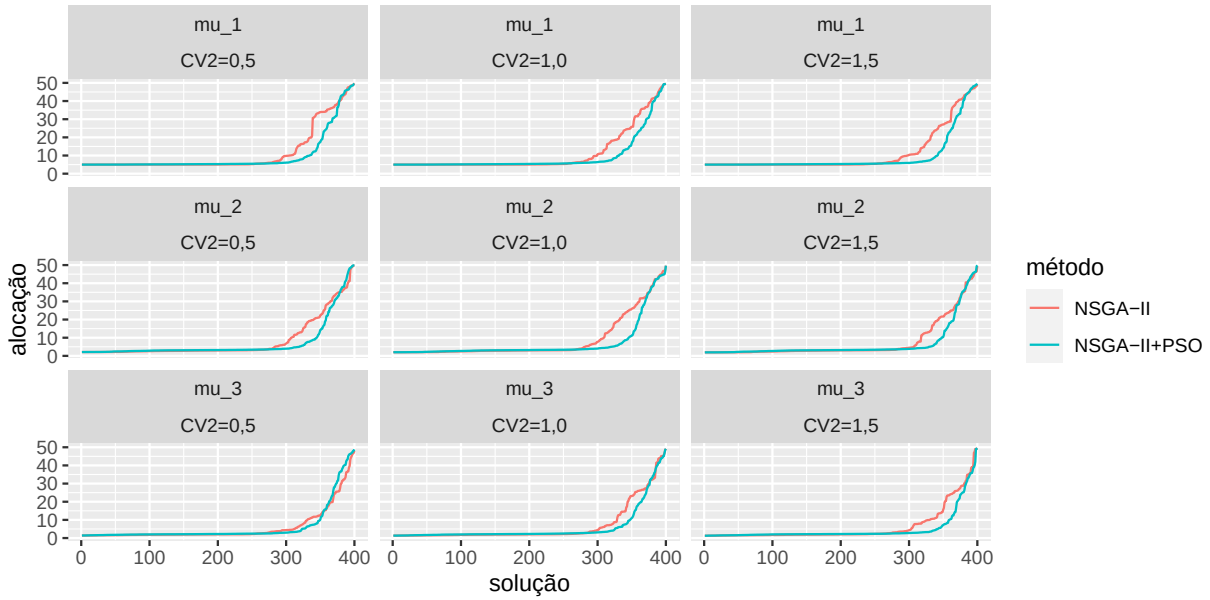


Figura 37: Recurso gasto em taxas de serviço entre filas antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

Como observado nas situações anteriores, o pós-processamento influenciou as soluções de menor custo quanto às taxas de serviço. Por outro lado, a partir da solução 300 aproximadamente, o consumo de recursos em taxas de serviço começa a aumentar e os efeitos do pós-processamento começam a ficar aparentes. Para quase todas as situações é verificado uma redução no custo em recursos das soluções pós-processadas. Apenas para a terceira fila da rede, e em particular quando o atendimento é hipoexponencial esta superioridade é clara. Existe, para essa situação particular, uma oscilação entre as vantagens das soluções fornecidas via NSGA-II e das soluções pós-processadas.

A referida superioridade das soluções pós-processados, em relação às taxas de serviço é bem ilustrada na análise da soma do recurso total consumido em taxa de serviços para as três filas da rede para a topologia da figura 8 (c). A figura 38 apresenta esta constatação nas soluções pós-processadas para redes de filas com os três formatos atendimentos analisados (hipoexponencial, markoviano e hiperexponencial).

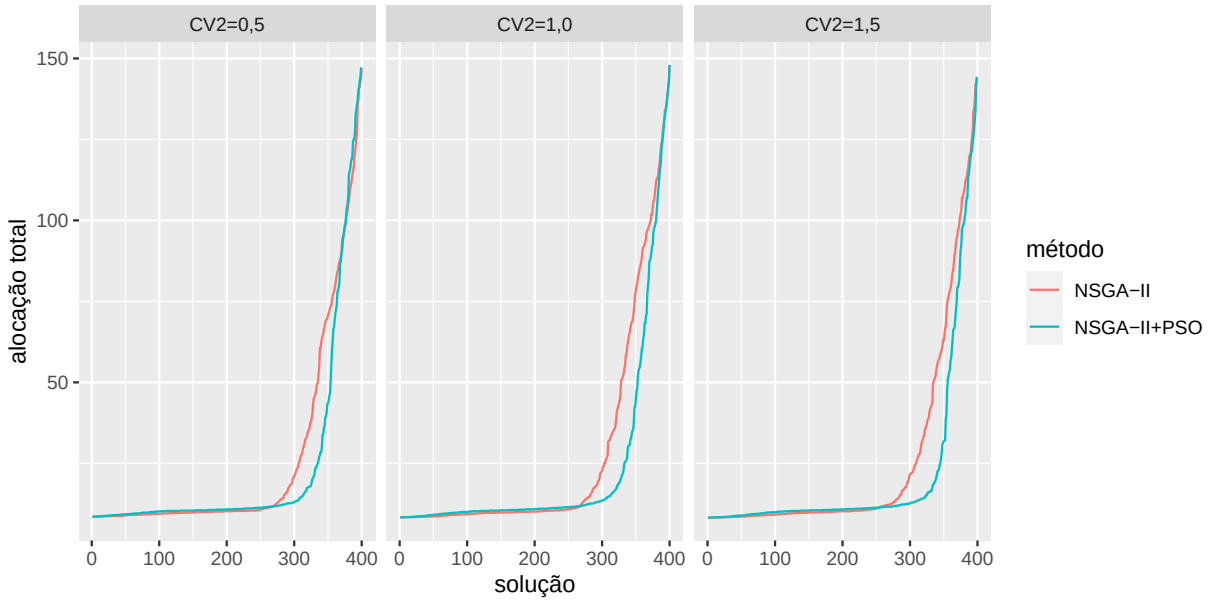


Figura 38: Recurso total gasto em taxas de serviço antes e após o pós-processamento via *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

As reduções de custo, agregadas pelo pós-processamento, parecem se tornar mais efetivas com o aumento do coeficiente de variação.

4.4 COMPARAÇÃO ENTRE SOLUÇÕES PÓS-PROCESSADAS DAS FORMULAÇÕES 1 VIA *SIMULATED ANNEALING* E 2 VIA *PARTICLE SWARM OPTIMIZATION*

A presente seção tem interesse em investigar as soluções obtidas pelos algoritmos NSGA-II+PSO e NSGA-II+SA para os espaços das variáveis resultantes nas formulações 1 e 2. Vale lembrar que a implementação apresentada por Cruz, Duarte e Souza (2018) utiliza os algoritmos NSGA-II+SA para solucionar o problema de otimização da formulação 1. Por outro lado, a implementação apresentada por Souza et al. (2020) utiliza os algoritmos NSGA-II+PSO para solucionar o problema de otimização da formulação 2.

Novamente é importante ressaltar que que as soluções não são comparáveis no espaço dos objetivos. O intuito comparativo se limita ao espaço das variáveis (recursos em *buffers* e servidores). Em cada uma das formulações propostas, a otimização ocorreu para funcionais objetivos distintos. A formulação 1, o algoritmo foi

desenvolvido para otimização voltada ao *throughput*. Já na formulação 2, a otimização foi voltada na minimização da soma das probabilidades de bloqueios P_K ao longo de todo o sistema.

Restrito ao espaço das variáveis, as soluções são de fato comparáveis. Os resultados, quando avaliados no espaço dos objetivos, medem o grau de eficiência das soluções de acordo com os objetivos de otimização. Já quando analisados no espaço das variáveis, a comparação ocorre para o custo das soluções fornecidas por meio dos dois métodos. O interesse fica em verificar as soluções quanto ao consumo de recursos em taxas de serviço e em áreas de circulação. Este tipo de análise permite identificar se uma das formulações é capaz de fornecer soluções menos onerosas.

A figura 39 apresenta os gráficos com os resultados obtidos através dos algoritmos NSGA-II+PSO e NSGA-II+SA. Nas colunas são representados os três valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Já nas linhas, são representadas as três filas da rede investigada. Novamente a rede em estudo é apresentada no esquema que pode ser visualizado na figura 8 (a).

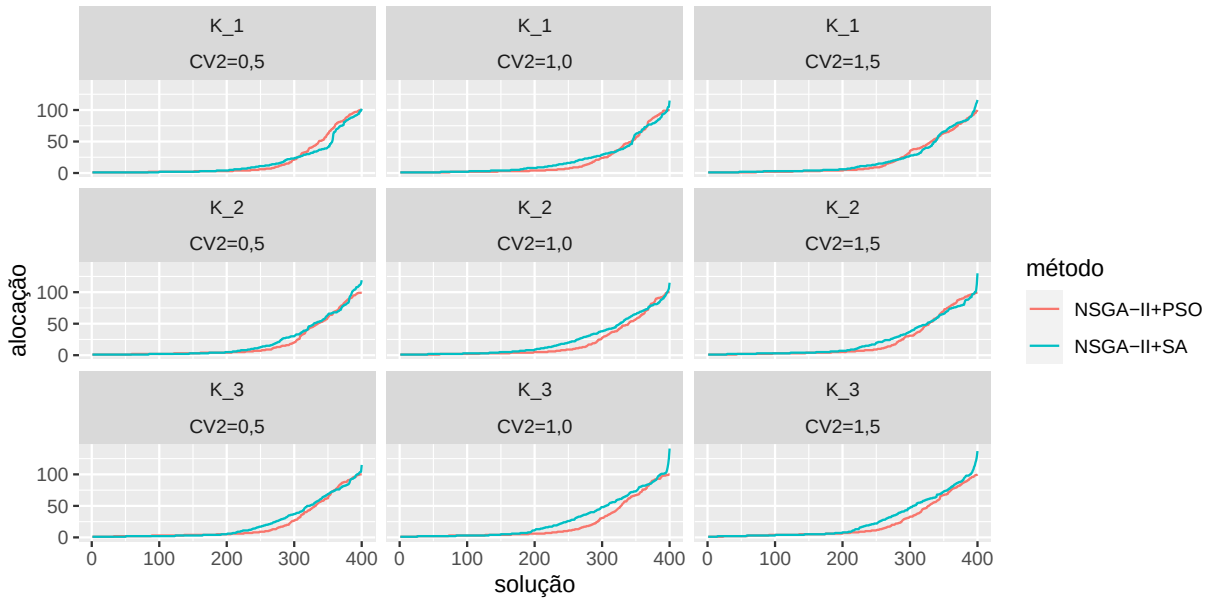


Figura 39: Alocação de áreas de circulação entre filas após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

As soluções de baixo consumo de recursos em áreas de circulação (aproximadamente as 200 primeiras soluções) apresentam diferenças entre os resultados via NSGA-II+PSO e via NSGA-II+SA. Porém, para as demais soluções alguma oscilação é verificada. Na primeira fila da topologia série, independente do atendimento (hipoexponencial, markoviano e hiperexponencial), parece haver alguma técnica notoriamente superior. Já para as duas filas seguintes da topologia série, para todos os coeficientes de variação, as soluções pós-processadas através do algoritmo *Particle Swarm Optimization* parecem consumir menos recursos em áreas de circulação. Este efeito parece ser mais significativo na situação de atendimento markoviano.

A análise com a soma dos espaços de circulação alocados em todo o sistema de filas é apresentada na figura 40. A avaliação da alocação total deixa claro que as soluções pós-processadas via algoritmo *Particle Swarm Optimization* são menos onerosas que as soluções pós-processadas via algoritmo *Simulated Annealing*. Este efeito ocorre para todos os coeficientes de variação, mas é notoriamente mais evidente para sistemas markovianos.

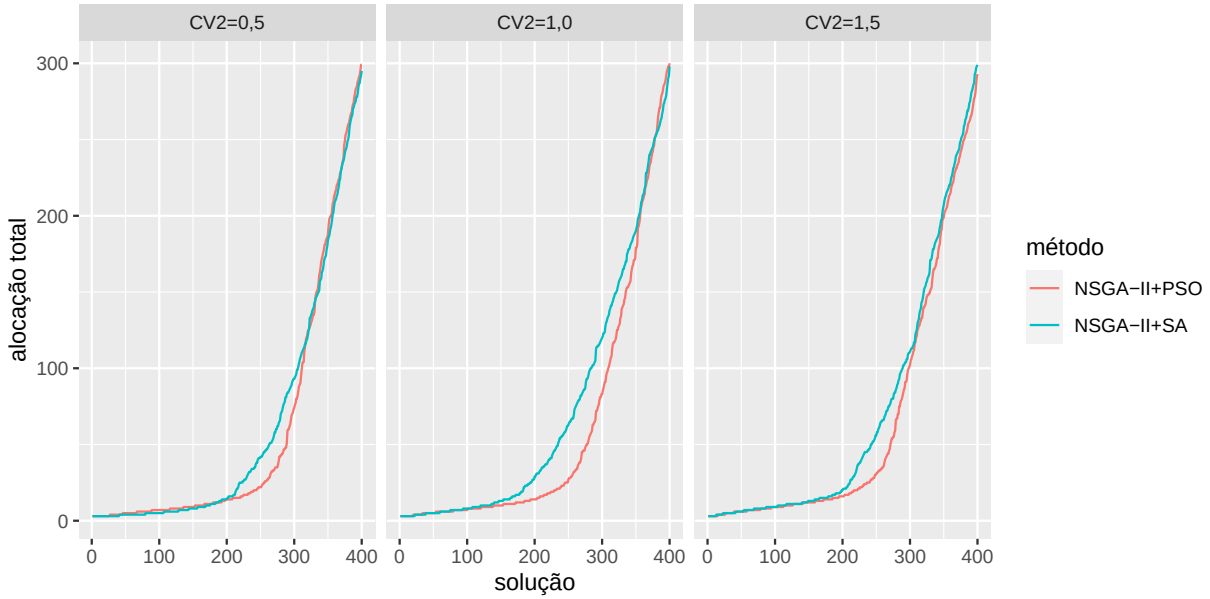


Figura 40: Alocação total de áreas de circulação após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

A figura 41 ilustra a representação gráfica para os resultados obtidos via NSGA-II+PSO e NSGA-II+SA para os três valores distintos para o quadrado dos co-

eficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$ e para as três filas da rede investigada de acordo com a topologia da figura 8 (a).

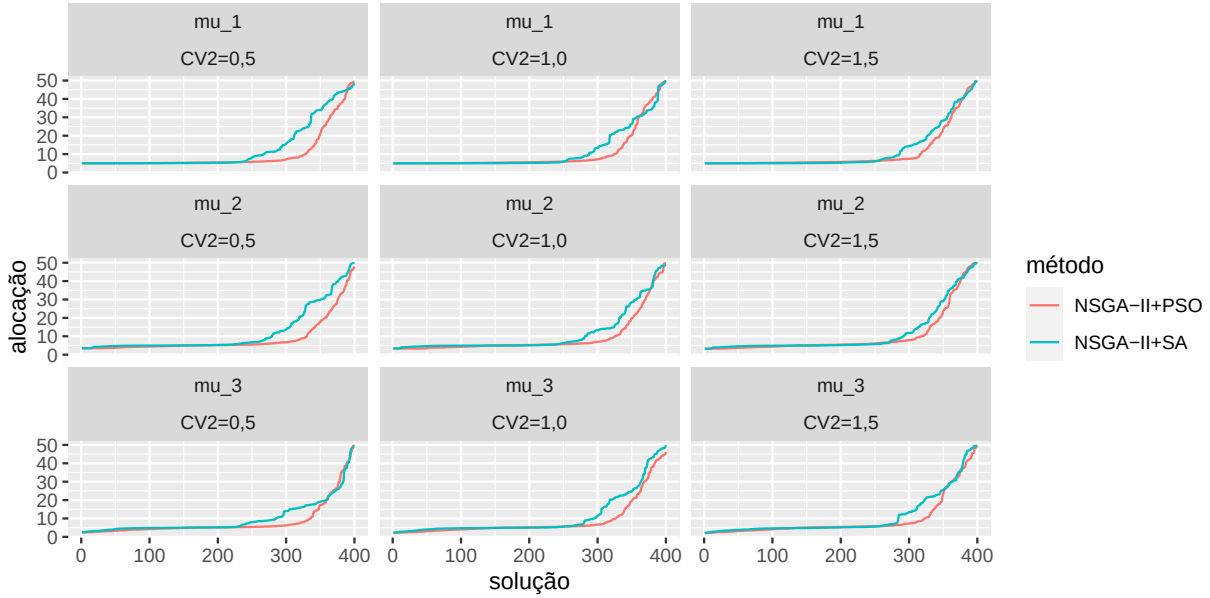


Figura 41: Recurso gasto em taxas de serviço entre filas após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

Novamente, para as soluções de baixo consumo de recursos (aproximadamente as 250 primeiras soluções) apresentam diferenças entre os resultados via NSGA-II+PSO e via NSGA-II+SA. Porém, para as demais soluções, para soluções pós-processadas através do algoritmo *Particle Swarm Optimization* existe uma evidente superioridade, na maioria das situações e soluções o consumo de recursos é inferior. Esta redução de consumo parece ser maior quanto menor for o coeficiente de variação associado aos atendimentos.

A avaliação através da soma dos recursos consumidos com as taxas de serviço nas três filas da rede deixa essa avaliação mais clara. Esta análise pode ser feita através dos gráficos apresentados na figura 42. A avaliação da alocação total evidência que as soluções obtidas por meio do pós-processamento através do algoritmo *Particle Swarm Optimization* são menos onerosas que as soluções pós-processadas através do algoritmo *Simulated Annealing*. Este efeito ocorre para todos os coeficientes de variação, mas é notoriamente mais evidente quanto menor for o coeficiente de variação dos tempos de atendimento.

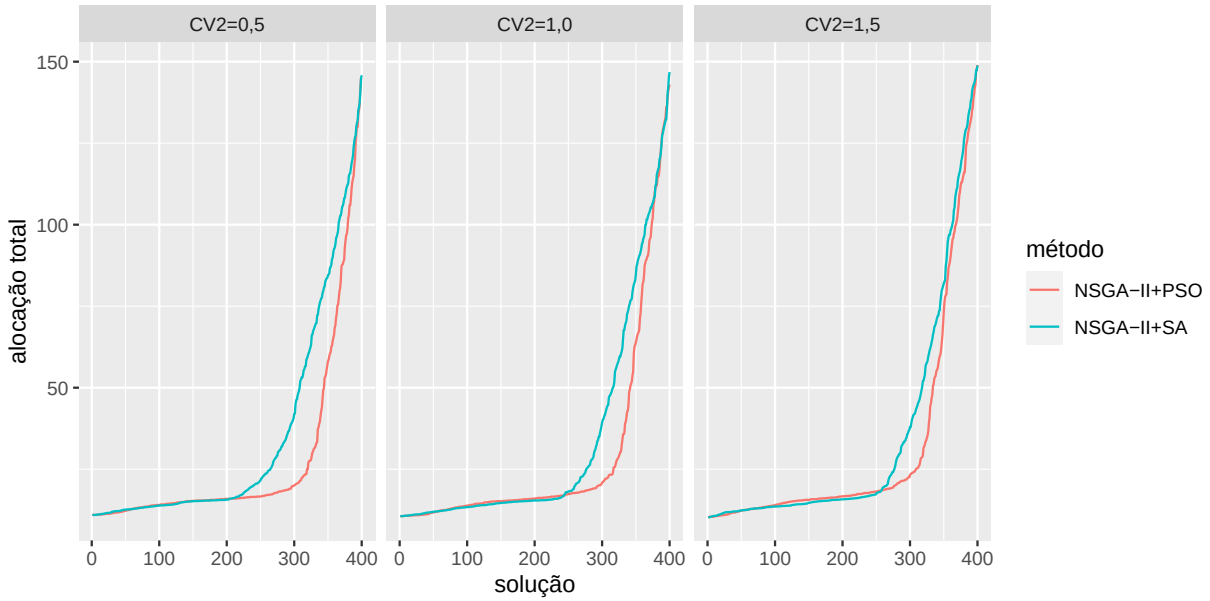


Figura 42: Recurso total gasto em taxas de serviço após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (série) da figura 8 (a).

A figura 43 mostra resultados por meio dos algoritmos NSGA-II+PSO e NSGA-II+SA. Nas colunas são representados os três valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Já nas linhas, são representadas as três filas da rede investigada que segue a topologia com divisão apresentada na figura 8 (b).

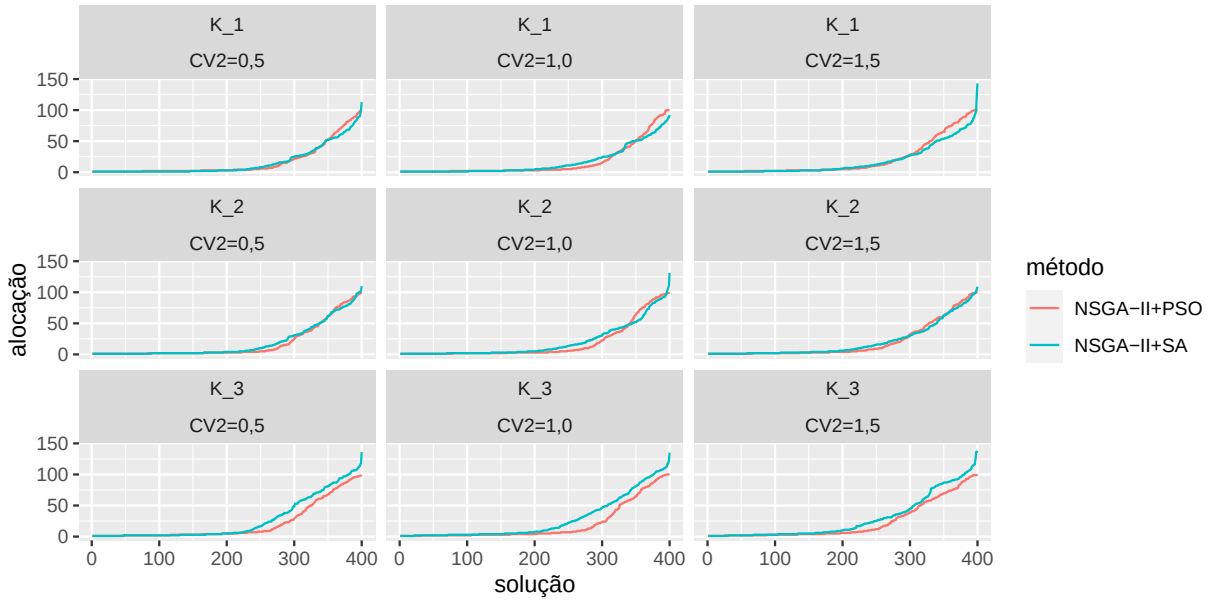


Figura 43: Alocação de áreas de circulação entre filas após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

As soluções de baixo consumo de recursos em áreas de circulação (aproximadamente as 200 primeiras soluções) apresentam diferenças entre os resultados via NSGA-II+PSO e via NSGA-II+SA. Porém, para as demais soluções alguma oscilação é verificada. Nas filas 1 e 2 existe um grande equilíbrio entre os dois conjuntos de resultados. Para a terceira fila da topologia divisão, conforme figura 8 (b), o pós-processamento através do algoritmo *Particle Swarm Optimization* parecem conduzir para soluções menos onerosas independentemente do coeficiente de variação.

A soma dos espaços em área de circulação alocados em todo o sistema de filas é apresentada na figura 44. Com respeito a alocação total, as soluções pós-processadas via algoritmo *Particle Swarm Optimization* consomem menos recursos onerosas que as soluções pós-processadas via algoritmo *Simulated Annealing*. Isto ocorre para todos os coeficientes de variação, porém o ganho é maior para sistemas markovianos.

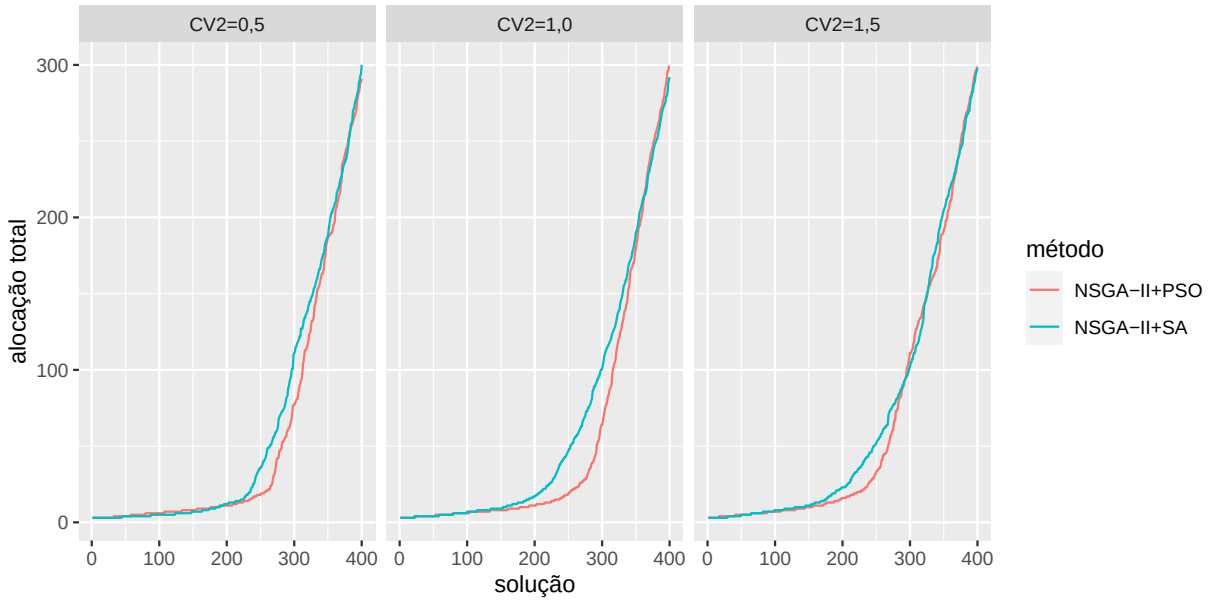


Figura 44: Alocação total de áreas de circulação após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

A figura 45 mostra gráficos das soluções obtidas pelos algoritmos NSGA-II+PSO e NSGA-II+SA. Por colunas, são representados os valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$. Já por linhas, as taxas de serviços para cada uma das três filas da topologia apresen-

tada na figura 8 (b).

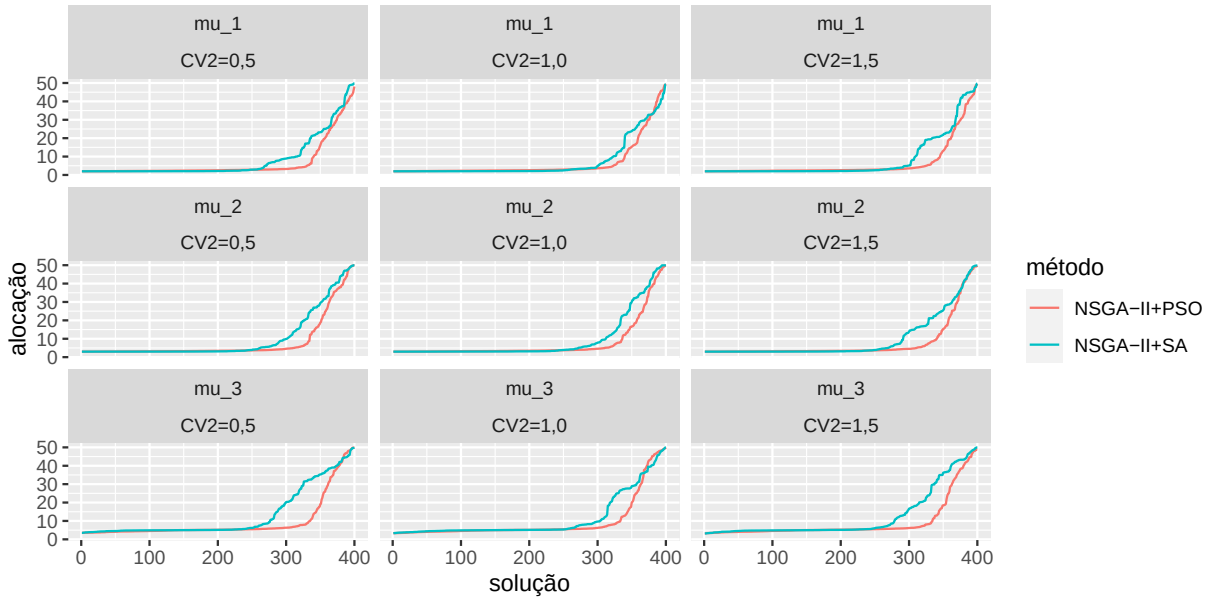


Figura 45: Recurso gasto em taxas de serviço entre filas após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

Para as soluções de baixo consumo de recursos em taxas de serviço, aproximadamente as 250 primeiras soluções, ocorrem diferenças evidentes entre os resultados via NSGA-II+PSO e via NSGA-II+SA. Para as demais soluções, as soluções pós-processadas através do algoritmo PSO são significativamente menos onerosas. Este efeito é mais claro em sistemas de atendimento hipoexponencial e hiperexponencial.

A avaliação por meio da soma dos recursos consumidos com as taxas de serviço, para as três filas da rede da topologia da figura 8 (b) deixa essa análise mais clara. Esta discussão pode ser feita através dos gráficos apresentados na figura 46. A avaliação da alocação total deixa claro que as soluções obtidas via pós-processamento com o algoritmo *Particle Swarm Optimization* são menos onerosas que as soluções via pós-processamento com o algoritmo *Simulated Annealing*. Este efeito ocorre para todos os coeficientes de variação, mas é notoriamente mais evidente essa redução de gastos para os sistemas de atendimento hipoexponencial e hiperexponencial. Este efeito ainda ocorre em sistemas de atendimento markoviano, mas em menor escala.

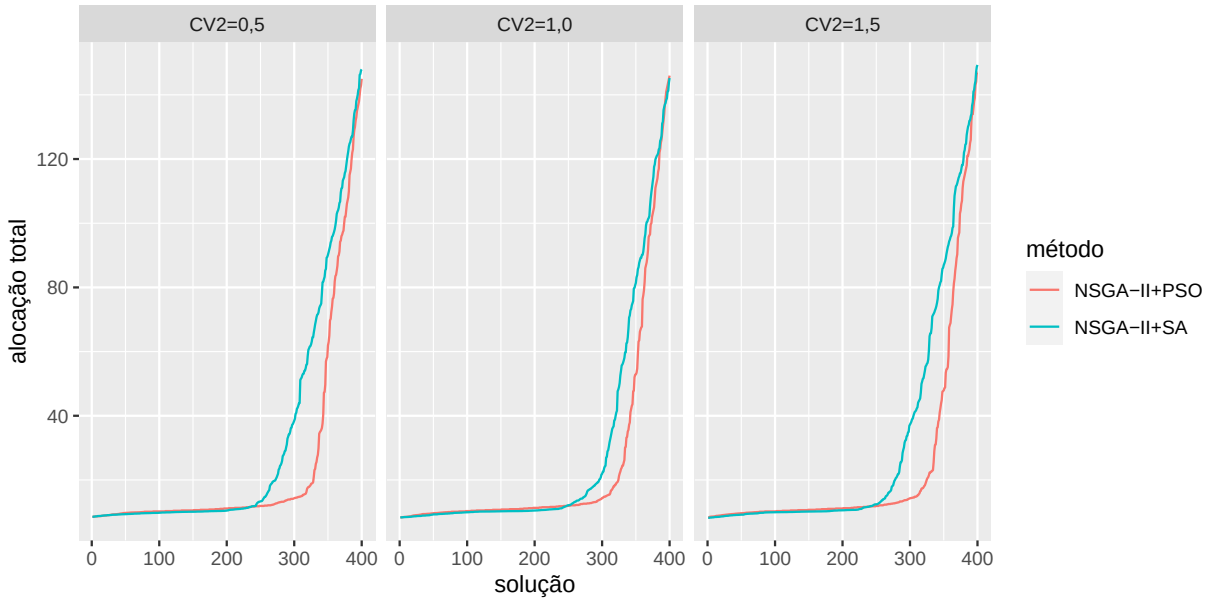


Figura 46: Recurso total gasto em taxas de serviço após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (divisão) da figura 8 (b).

A figura 47, apresenta os gráficos com os resultados obtidos pelos algoritmos NSGA-II+PSO e NSGA-II+SA para a topologia da figura 8 (c). Nas colunas são representados os três valores para o quadrado dos coeficientes de variação $CV^2 = 0{,}5; CV^2 = 1{,}0$ e $CV^2 = 1{,}5$, e nas linhas, são representadas as três filas da rede avaliada.

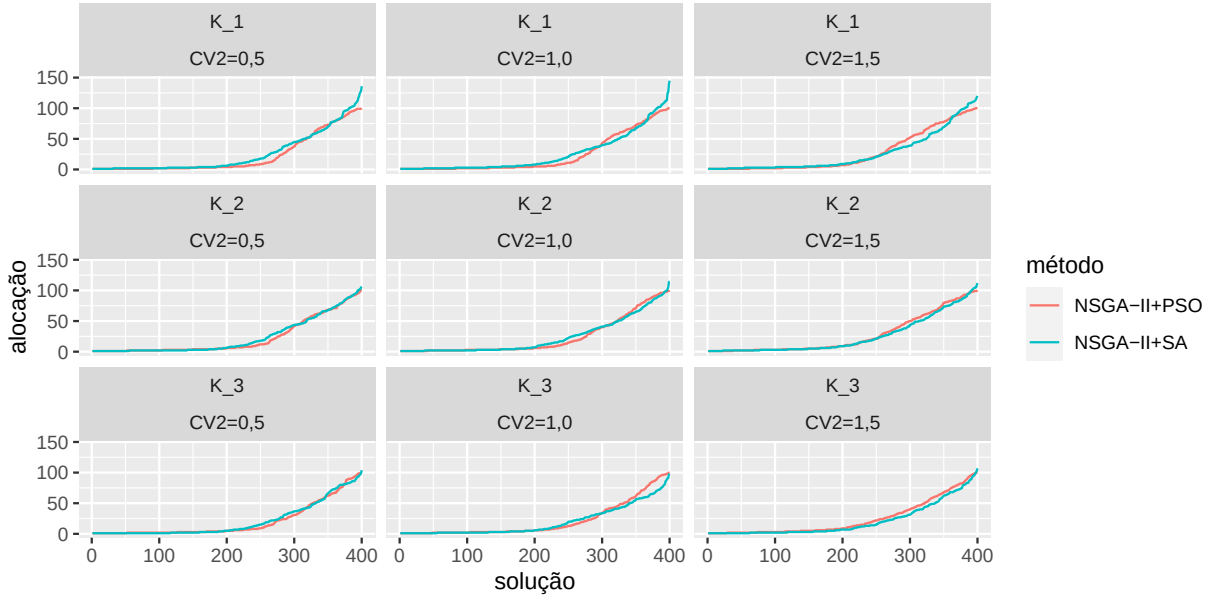


Figura 47: Alocação de áreas de circulação entre filas após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

Para a topologia representada na figura 8 (c) existe um equilibrio no consumo de recursos para as três filas do sistema e os diferentes coeficientes de variação em estudo. parece possível identificar algum efeito marcante de superioridade de um conjunto de soluções em comparação ao outro conjunto. A figura 48 avalia a alocação total consumida nas três filas da rede.

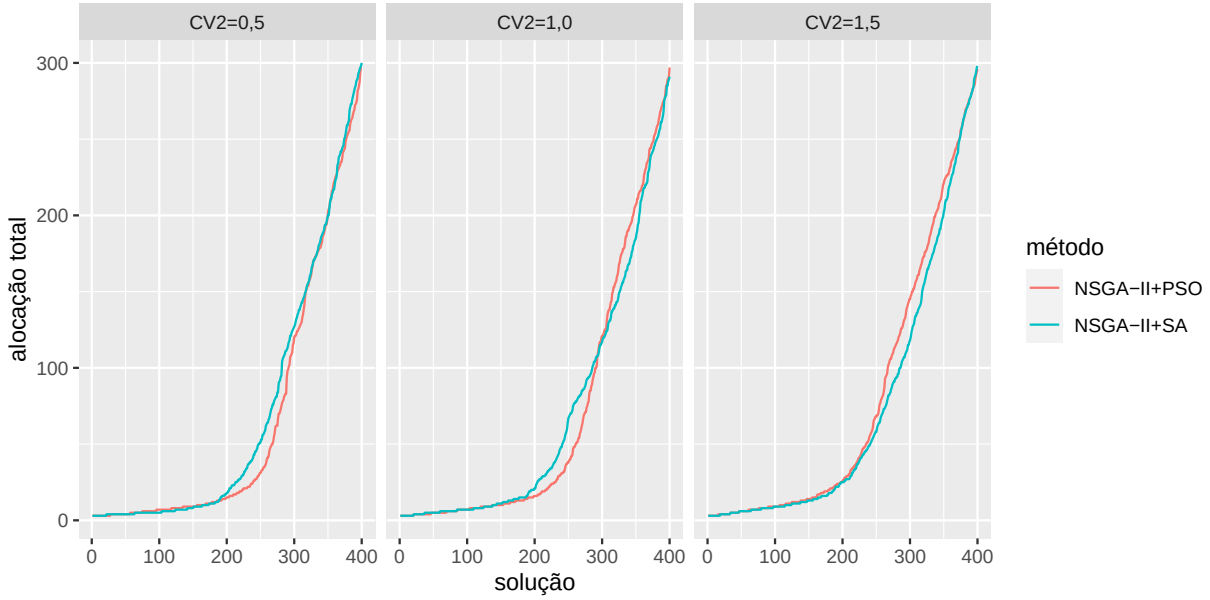


Figura 48: Alocação total de áreas de circulação após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

Ao analisar a soma dos recursos nas três filas do sistema da topologia da figura 8 (c), parece mais visual as diferenças das soluções entre NSGA-II+PSO e NSGA-II+SA. Na rede de filas com atendimento hipoexponencial as soluções via NSGA-II+PSO são menos onerosas, ao passo que na rede de filas com atendimento hiperexponencial as soluções via NSGA-II+SA são menos onerosas. Existe uma marcante oscilação para o sistema de atendimento markoviano, fica clara uma vantagem de um conjunto de soluções (NSGA-II+PSO) em relação ao outro conjunto de soluções (NSGA-II+SA).

A figura 49 apresenta os gráficos dos conjuntos de soluções obtidas pelos algoritmos NSGA-II+PSO e NSGA-II+SA. Nas colunas são representados os valores distintos para o quadrado dos coeficientes de variação utilizados $CV^2 = 0{,}5$ (atendimento hipoexponencial); $CV^2 = 1{,}0$ (atendimento markoviano) e $CV^2 = 1{,}5$ (aten-

dimento hiperexponencial). Nas linhas estão representadas as taxas de serviços de cada uma das três filas que compõem a topologia com fusão representada na figura 8 (c).

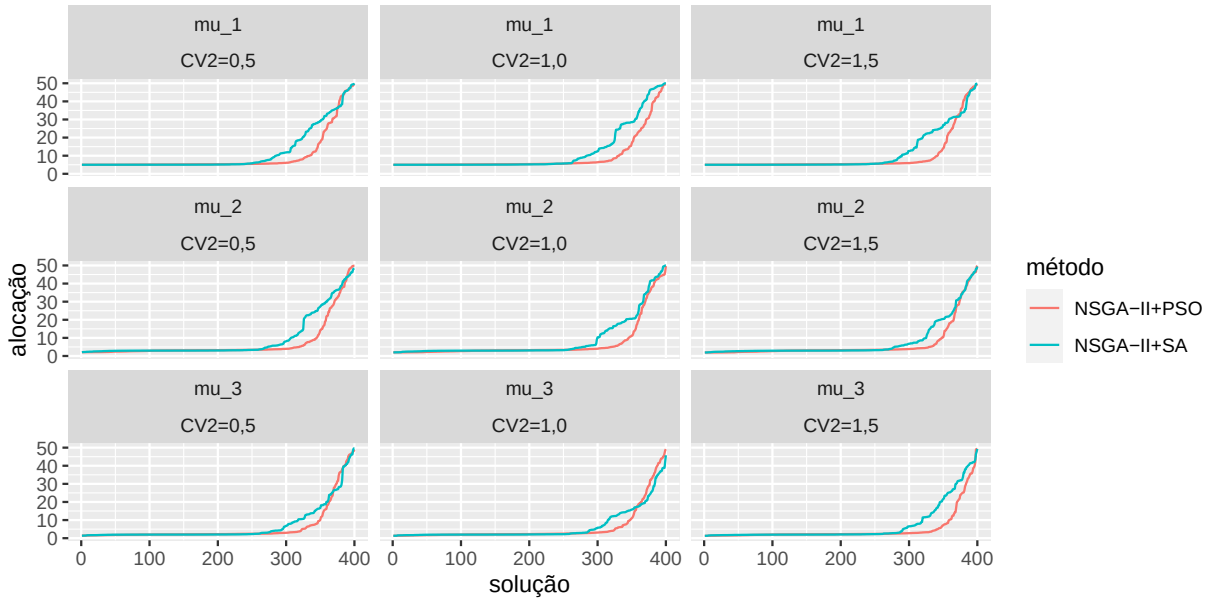


Figura 49: Recurso gasto em taxas de serviço entre filas após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

Para a topologia com fusão, apresentada na figura 8 (c), novamente, para as soluções de baixo consumo de recursos (aproximadamente as 270 primeiras soluções), ocorrem distinções muito evidentes entre os resultados obtidos através dos algoritmos NSGA-II+PSO e NSGA-II+SA. A partir daí, para soluções pós-processadas através do algoritmo *Particle Swarm Optimization* existe uma notória superioridade. Isso ocorre principalmente nas duas primeiras filas da rede. Na terceira fila da rede, para alocações maiores e atendimento markoviano, uma ligeira vantagem para as soluções pós-processadas através do algoritmo *Simulated Annealing* pode ser verificada.

Como verificado nas investigações anteriores, uma avaliação através da soma dos recursos consumidos com as taxas de serviço nas três filas da rede é bastante importante para interpretar essa comparação. Os gráficos podem ser visualizados na figura 50.

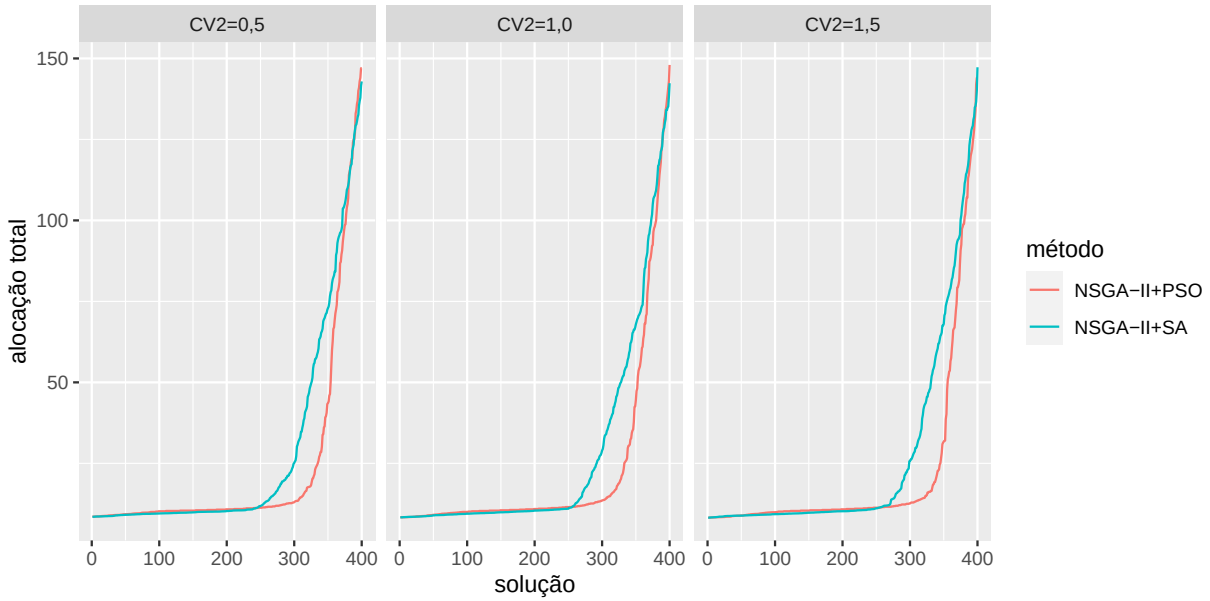


Figura 50: Recurso total gasto em taxas de serviço após o pós-processamento via *Simulated Annealing* e *Particle Swarm Optimization* para a topologia (fusão) da figura 8 (c).

A avaliação da alocação total de recursos em taxas de serviço deixa claro que as soluções obtidas por meio do pós-processamento através do algoritmo *Particle Swarm Optimization* são menos onerosas que as soluções pós-processadas através do algoritmo *Simulated Annealing*. Este efeito parece ocorre de forma completamente independente do modelo de atendimento, seja ele hipoexponencial, markoviano ou então hiperexponencial.

5 CONSIDERAÇÕES FINAIS

Este estudo propôs uma investigação comparativa das soluções fornecidas através dos trabalhos apresentados em Cruz, Duarte e Souza (2018) e Souza et al. (2020). Estes estudos tratam da formulação e análise do *problema de alocação de áreas de espera e capacidade de servidores* para redes de filas de atendimento geral e servidor único.

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

Além disso, a execução deste estudo permitiu aprofundamento na utilização da linguagem TEX. Trata-se de um editor de textos que é padrão na confecção de textos estatísticos e de outras áreas de estudo em vários níveis de pesquisa.

Foram apresentadas formulações de objetivo único e multiobjetivos para a discussão do problema alvo. Além disso. os algoritmos genéticos NSGA-II, *Simulated Annealing* e *Particle Swarm optimization* foram detalhados. O clássico método da expansão generalizado foi abordado para ser utilizado nos procedimentos de otimização.

O estudo exigiu ainda a capacitação para o planejamento de uma extensa gama de experimentos. Muitas execuções de simulação computacional foram necessárias para a realização da presente investigação.

A principal contribuição deste trabalho se refere à identificação de alocações ótimas em uma rede de filas complexa com diversas configurações topológicas com configurações em série, divisão e fusão.

Observa-se que, dos efeitos decorrentes das topologias de rede em estudo, a alocação de áreas de espera e recursos em servidores apresenta resultados com algum padrão específico para os diversos casos investigados. É fácil observar que os diversos algoritmos apresentados são úteis para a solução do problema, mas alguma superioridade é verificada no desempenho através da estratégia de pós-

processamento através do algoritmo *Particle Swarm optimization*.

5.1 PROPOSTAS DE CONTINUIDADE

Como trabalhos futuros, podem-se citar os seguintes:

- Análise dos padrões obtidos em uma única rede de filas complexas combinando as três topologias básicas;
- Investigação das soluções em redes de filas com tempos de chegada gerais e independentes, ou seja, em redes do tipo G/G/c/k, na notação de Kendall (1953);
- Alocação ótima em redes de filas com ciclos, que podem modelar o retrabalho, dentre diversas outras possibilidades.

Investigações futuras também incluem a avaliação da qualidade na estimação de outras medidas de desempenho das filas da rede, tais como a probabilidade da ociosidade do servidor, $P(M = 0)$, o tempo de espera no sistema, W, e o tempo médio de permanência na fila, W_q. Outras investigações com filas de estruturas distintas, tais como filas markovianas multi-servidoras infinitas, M/M/c, finitas, M/M/c/k, e assim por diante. Estes são apenas alguns tópicos para trabalhos futuros nesta instigante linha de pesquisa.

REFERÊNCIAS

ABENSUR, E. O. et al. Tendências para o auto-atendimento bancário brasileiro: um enfoque estratégico baseado na teoria das filas. *RAM. Revista de Administração Mackenzie*, Universidade Presbiteriana Mackenzie, v. 4, n. 2, p. 40–59, 2003.

AHMED, N. U.; OUYANG, X. H. Suboptimal red feedback control for buffered tcp flow dynamics in computer network. *Mathematical Problems in Engineering*, Hindawi, v. 2007, 2007.

ALVES, F. S. Q. et al. Upper bounds on performance measures of heterogeneous M/M/c queues. *Mathematical Problems in Engineering*, v. 2011, n. Article ID 702834, p. 18 pages, 2011.

ANDRIANSYAH, R. et al. Performance optimization of open zero-buffer multi-server queueing networks. *Computers & Operations Research*, v. 37, n. 8, p. 1472–1487, 2010.

AZIMI, P.; ASADOLLAHI, A. Developing a new bi-objective functions model for a hierarchical location-allocation problem using the queuing theory and mathematical programming. *Journal of Optimization in Industrial Engineering*, QIAU, v. 12, n. 2, p. 149–154, 2019.

BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. Handbook of evolutionary computation. *Release*, v. 97, n. 1, p. B1, 1997.

BRUIN, A. M. et al. Modeling the emergency cardiac in-patient flow: an application of queuing theory. *Health Care Management Science*, Springer, v. 10, n. 2, p. 125–137, 2007.

CAMELO, G. R. et al. Teoria das filas e da simulação aplicada ao embarque de minério de ferro e manganês no terminal marítimo de ponta da madeira. *Cadernos do IME - Série Estatística*, v. 29, n. 2, p. 1, 2010.

ČERNỲ, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, Springer, v. 45, n. 1, p. 41–51, 1985.

CHANGFU, L.; ZHENYU, L. Research of transaction request handling queueing system in the e-business environment based on queuing theory. In: IEEE. *2009 Asia-Pacific Conference on Information Processing*. [S.l.], 2009. v. 2, p. 589–592.

CHAUDHURI, K. et al. Server allocation algorithms for tiered systems. *Algorithmica*, v. 48, n. 2, p. 129–146, 2007.

CHEAH, J. Y.; MACGREGOR SMITH, J. Generalized M/G/c/c state dependent queueing models and pedestrian traffic flows. *Queueing Systems*, Springer, v. 15, n. 1-4, p. 365–386, 1994.

CHEN, J.; HU, C.; JI, Z. An improved ARED algorithm for congestion control of network transmission. *Mathematical Problems in Engineering*, Hindawi, v. 2010, 2010.

CHOWDHURY, S.; MUKHERJEE, S. P. Estimation of traffic intensity based on queue length in a single M/M/1 queue. *Communications in Statistics - Theory and Methods*, Taylor & Francis, v. 42, n. 13, p. 2376–2390, 2013.

COELLO COELLO, C. A.; LECHUGA, M. S. MOPSO: A proposal for multiple objective particle swarm optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. [S.l.: s.n.], 2002. v. 2, p. 1051–1056.

CRUZ, F. R. B. Optimizing the throughput, service rate, and buffer allocation in finite queueing networks. *Electronic Notes in Discrete Mathematics*, Elsevier, v. 35, p. 163–168, 2009.

CRUZ, F. R. B.; DUARTE, A. R.; SOUZA, G. L. Multi-objective performance improvements of general finite single-server queueing networks. *Journal of Heuristics*, Springer, v. 24, n. 5, p. 757–781, 2018.

CRUZ, F. R. B.; DUARTE, A. R.; VAN WOENSEL, T. Buffer allocation in general single-server queueing networks. *Computers & Operations Research*, v. 35, n. 11, p. 3581 –3598, 2008.

CRUZ, F. R. B. et al. Throughput maximization of queueing networks with simultaneous minimization of service rates and buffers. *Mathematical Problems in Engineering*, v. 2012, n. Article ID 692593, p. 19 pages, 2012.

CRUZ, F. R. B.; MACGREGOR SMITH, J.; QUEIROZ, D. C. Service and capacity allocation in m/g/c/c state-dependent queueing networks. *Computers & operations research*, Elsevier, v. 32, n. 6, p. 1545–1563, 2005.

CRUZ, F. R. B.; OLIVEIRA, P. C.; DUCZMAL, L. State-dependent stochastic mobility model in mobile communication networks. *Simulation Modelling Practice and Theory*, v. 18, n. 3, p. 348–365, 2010.

CRUZ, F. R. B. et al. On the system optimum of traffic assignment in M/G/c/c state-dependent queueing networks. *European Journal of Operational Research*, Elsevier, v. 201, n. 1, p. 183–193, 2010.

DEB, K.; AGRAWAL, R. B. Simulated binary crossover for continuous search space. *Complex systems*, Citeseer, v. 9, n. 2, p. 115–148, 1995.

DEB, K.; BEYER, H.-G. Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary computation*, MIT Press, v. 9, n. 2, p. 197–221, 2001.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 2, p. 182–197, 2002.

DIMITRIOU, I.; LANGARIS, C. A repairable queueing model with two-phase service, start-up times and retrial customers. *Computers & Operations Research*, Elsevier, v. 37, n. 7, p. 1181–1190, 2010.

DOY, F. E. et al. Simulação do serviço de correio eletrônico através de um modelo de filas. *Pesquisa Operacional*, SciELO Brasil, v. 26, n. 2, p. 241–253, 2006.

FAN, Z. et al. An improved multiobjective particle swarm optimization algorithm using minimum distance of point to line. *Shock and Vibration*, v. 2017, p. 1–16, 2017.

GROSS, D. et al. *Fundamentals of Queueing Theory*. 4th edition. ed. New York, NY: Wiley - Interscience, 2009.

HU, A. B.; MEERKOV, S. M. Lean buffering in serial production lines with bernoulli machines. *Mathematical Problems in Engineering*, Hindawi, v. 2006, 2006.

HU, X.-B.; DI PAOLO, E. An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem. In: *Multi-objective memetic algorithms*. [S.l.]: Springer, 2009. p. 71–89.

JIA, C.; ZHU, H. An improved multiobjective particle swarm optimization based on culture algorithms. *Algorithms*, v. 10, n. 2, p. 46, 2017.

KAMALI, S. H. et al. The monitoring of the network traffic based on queuing theory and simulation in heterogeneous network environment. In: IEEE. *2009 International Conference on Computer Technology and Development*. [S.l.], 2009. v. 1, p. 322–326.

KENDALL, D. G. Stochastic processes occurring in the theory of queues and their analysis by the method of embedded Markov chains. *Annals Mathematical Statistics*, v. 24, p. 338–354, 1953.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. [S.l.: s.n.], 1995. v. 4, p. 1942–1948.

KERBACHE, L.; MACGREGOR SMITH, J. The generalized expansion method for open finite queueing networks. *European Journal of Operational Research*, v. 32, p. 448–461, 1987.

KERBACHE, L.; MACGREGOR SMITH, J. Multi-objective routing within large scale facilities using open finite queueing networks. *European Journal of Operational Research*, v. 121, n. 1, p. 105–123, 2000.

KIMURA, T. A transform-free approximation for the finite capacity M/G/s queue. *Operations Research*, v. 44, n. 6, p. 984–988, 1996.

KIRKPATRICK, S. et al. Optimization by simmulated annealing. *science*, Washington, v. 220, n. 4598, p. 671–680, 1983.

LABETOULLE, J.; PUJOLLE, G. Isolation method in a network of queues. *IEEE Transactions on Software Engineering*, IEEE, v. 4, p. 373–381, 1980.

LI, J.; ENGINARLAR, E.; MEERKOV, S. M. Conservation of filtering in manufacturing systems with unreliable machines and finished goods buffers. *Mathematical Problems in Engineering*, Hindawi, v. 2006, 2006.

MACGREGOR SMITH, J. M/G/c/k blocking probability models and system performance. *Performance Evaluation*, v. 52, n. 4, p. 237–267, 2003.

MACGREGOR SMITH, J. Optimal design and performance modelling of M/G/1/k queueing systems. *Mathematical and Computer Modelling*, v. 39, n. 9-10, p. 1049–1081, 2004.

MACGREGOR SMITH, J.; CRUZ, F. R. B. The buffer allocation problem for general finite buffer queueing networks. *IIE Transactions*, v. 37, n. 4, p. 343–365, 2005.

MACGREGOR SMITH, J.; CRUZ, F. R. B.; VAN WOENSEL, T. Topological network design of general, finite, multi-server queueing networks. *European Journal of Operational Research*, v. 201, n. 2, p. 427–441, 2010.

MARTINS, H. S. R. et al. Modeling and optimization of buffers and servers in finite queueing networks. *OPSEARCH*, Springer, v. 56, n. 1, p. 123–150, 2019.

MENASCÉ, D. A. QoS issues in web services. *IEEE Internet Computing*, v. 6, n. 6, p. 72–75, 2002.

OSORIO, C.; BIERLAIRE, M. An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, v. 196, n. 3, p. 996–1007, 2009.

QI, D. et al. Weighted likelihood ratio chart for statistical monitoring of queueing systems. *Quality Technology & Quantitative Management*, Taylor & Francis, v. 14, n. 1, p. 19–30, 2017.

SOUZA, G. L. *Uma nova formulação para otimização multi-objetivo em redes de filas finitas gerais e com único servidor*. 59 p. Dissertação (Mestrado) — Universidade Federal de Ouro Preto, Ouro Preto, 2020.

SOUZA, G. L. et al. A novel formulation for multi-objective optimization of general finite single-server queueing networks. In: IEEE. *2020 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2020. p. 1–8.

SPIECKERMANN, S. et al. Simulation-based optimization in the automotive industry-a case study on body shop design. *Simulation*, SCS Society for Computer Simulation, v. 75, n. 5/6, p. 276–286, 2000.

SPINELLIS, D.; PAPADOPOULOS, C. T.; MACGREGOR SMITH, J. Large production line optimization using simulated annealing. *International Journal of Production Research*, v. 38 (3), p. 509–541, 2000.

SYSWERDA, G. Uniform crossover in genetic algorithms. In: *Proceedings of the 3rd international conference on genetic algorithms*. [S.l.: s.n.], 1989. p. 2–9.

TANG, L. et al. Efficient sensitivity analysis approach modeling and optimization of m/g/1-type queueing networks: An volume 2010, article id 130319, 20 pages. *Mathematical Problems in Engineering*, v. 2010, n. 2, 2010.

TRIVEDI, V.; VARSHNEY, P.; RAMTEKE, M. A simplified multi-objective particle swarm optimization algorithm. *Swarm Intelligence*, Springer, v. 14, p. 83–116, 2020.

VAN WOENSEL, T. et al. Allocation in general multi-server queueing networks. *International Transactions in Operational Research*, v. 17, n. 2, p. 257–286, 2010.

VAN WOENSEL, T.; CRUZ, F. R. B. A stochastic approach to traffic congestion costs. *Computers & Operations Research*, Elsevier, v. 36, n. 6, p. 1731–1739, 2009.

VAN WOENSEL, T.; CRUZ, F. R. B. Optimal routing in general finite multi-server queueing networks. *PLoS ONE*, Public Library of Science, v. 9, n. 7, p. e102075, 07 2014.

VAN WOENSEL, T. et al. Vehicle routing with dynamic travel times: A queueing approach. *European journal of operational research*, Elsevier, v. 186, n. 3, p. 990–1007, 2008.

WANG, Q. et al. Analysis of a linear walking worker line using a combination of computer simulation and mathematical modeling approaches. *Journal of manufacturing systems*, Elsevier, v. 28, n. 2-3, p. 64–70, 2009.

YANG, X. S. *Engineering Optimization: An Introduction with Metaheuristic Applications*. 1st. ed. [S.l.]: Wiley Publishing, 2010. ISBN 0470582464, 9780470582466.

YOUSSEF, A. M. A.; ELMARAGHY, H. A. Performance analysis of manufacturing systems composed of modular machines using the universal generating function. *Journal of Manufacturing Systems*, Elsevier, v. 27, n. 2, p. 55–69, 2008.

ZHAO, X. et al. An improved mixed-integer multi-objective particle swarm optimization and its application in antenna array design. In: IEEE. *2013 5th IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*. [S.l.], 2013. p. 412–415.