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

O Problema de Alocação de Servidores para Redes de Filas Markovianas

Celio Cesar Mendonça Costa

Ouro Preto-MG
2022

Celio Cesar Mendonça Costa

O Problema de Alocação de Servidores para Redes de Filas Markovianas

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: Prof. Dr. Anderson Ribeiro Duarte

Ouro Preto

2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C837p Costa, Celio Cesar Mendonca.
O problema de alocação de servidores para redes de filas Markovianas. [manuscrito] / Celio Cesar Mendonca Costa. - 2022.
77 f.: il.: color., gráf., tab..

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

1. Otimização. 2. Servidor da Web. 3. Produtividade. I. Duarte, Anderson Ribeiro. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004

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

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

FOLHA DE APROVAÇÃO

Célio César Mendonça Costa

O Problema de Alocação de Servidores para Redes de Filas Markovianas

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 14 de junho de 2022

Membros da banca

Dr. Anderson Ribeiro Duarte - Orientador - Universidade Federal de Ouro Preto
Dr. Rivert Paulo Braga Oliveira - Universidade Federal de Ouro Preto
Dr. Matheus Wanderley Romão - Universidade Federal de São João Del-Rei

Professor Dr. Anderson Ribeiro Duarte, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 14/06/2022

Documento assinado eletronicamente por **Anderson Ribeiro Duarte**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 16/06/2022, às 18:40, 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 **0339790** e o código CRC **50BB5121**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: (31)3559-1312 - www.ufop.br

Agradecimentos

Agradeço primeiramente ao Professor Dr. Anderson Ribeiro Duarte por abraçar a ideia e ser meu orientador. Por toda a dedicação e leveza me conduzindo na produção desse trabalho, além de toda a contribuição ao longo da minha graduação.

À UFOP, ao DEEST e todos os professores, pelo ensino de qualidade e oportunidades ao longo do curso.

À Estatis Jr. pela oportunidade de estar no mercado de trabalho ainda na faculdade.

A minha família, namorada e amigos, por todo apoio, paciência e incentivo. A graduação com certeza foi mais leve por ter vocês comigo.

Por fim, agradeço a República OxiGênios, meu eterno lar em Ouro Preto.

Resumo

Os processos produtivos geralmente precisam melhorar os níveis de produtividade para atender à demanda do cliente. Na fase de projeto desses processos, um dos problemas mais significativos é como alocar recursos (servidores) para dimensionar sistemas de filas eficientes. Redes de filas com vários servidores, topologias acíclicas arbitrárias com chegadas markovianas e serviços são consideradas neste estudo. Uma abordagem para alocação ótima de servidores em diversas topologias de redes de filas (série, fusão e divisão) é apresentada. A metodologia utiliza uma estratégia heurística multiobjetivo através do algoritmo *Simulated Annealing*. O desempenho da utilização do servidor através de uma métrica de produtividade é maximizado simultaneamente com a minimização do tempo geral esperado dos clientes para percorrer a rede de filas. Topologias diversas para as redes de filas são investigadas, assim como variações de roteamento dentro das topologias em estudo. Diversos resultados de experimentos computacionais mostram a eficácia da metodologia.

Palavras-chave: Otimização, Heurística, Rede de Filas, Produtividade, Alocação.

Abstract

Production processes often need to improve productivity levels in order to meet customer demand. When designing these processes, one of the most significant problems is how to allocate resources (servers) to develop efficient queueing systems. Queueing networks with multiple servers and arbitrary acyclic topologies with Markov arrivals and services are considered in this study. An approach for optimal server allocation in different queueing network topologies (series, merge, and split) is presented. The methodology uses a multi-objective heuristic strategy using the Simulated Annealing algorithm. Server utilization performance is maximized simultaneously with the minimization of the expected overall time of customers in the queueing network. Different topologies for the queueing networks are investigated, and variations in the routing probabilities within the topologies under study. Several results of computational experiments show the effectiveness of the methodology.

Keywords: Optimization, Heuristics, Queueing Network, Productivity, Allocation.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Ao abordar assuntos sobre filas é comum vir à mente processos desgastantes, demorados, burocráticos, entre outros dissabores. De fato, são transtornos que estão presentes na vida de todos os indivíduos, a todo momento. Sejam estes, momentos ocorridos em espera no supermercado, no trânsito, no hospital, ou até mesmo em um atendimento virtual. Porém, não são apenas esses tipos clássicos de filas que pertencem ao nosso cotidiano, é possível extrapolar um pouco mais essa teoria. Todo processo que necessita seguir um fluxo específico até chegar em seu destino final, pode ser considerado uma fila. Como, por exemplo, o processo de montagem de brinquedos em uma fábrica, ou o tráfego de dados *online*. Entender como essas filas se comportam com o intuito de melhorar seu funcionamento é um desafio antigo, bastante complexo e desafiador. Diversos pesquisadores já se propuseram a solucionar problemas dessa natureza com diferentes abordagens [4–28].

Existem várias maneiras de se estruturar um sistema de filas, seja pela forma que os clientes chegam, como são atendidos, pela dimensão, volume de servidores, se existe a possibilidade ciclos, cultura de *feedback*, entre outros parâmetros. Uma rede de filas, usualmente é denominada como um grafo de múltiplos vértices (nós). Consiste em um conjunto de filas e servidores interconectados. As entidades que percorrem a rede, transitam entre as filas para receber algum tipo de serviço. Uma rede de filas é dita fechada quando não se tem abertura externa no sistema, ou seja, o número total de usuários não varia, os individuos apenas permutam entre si suas posições. Quando existe essa abertura, seja com entrada ou saída de usuários, a rede é denominada aberta. Nesse caso, não possuí um número fixo total de individuos. Em uma rede, uma única fila que leva a um conjunto de servidores é também representada como um nó do grafo e os possíveis caminhos que um usuário pode percorrer na rede, como rota. A Figura 1 exemplifica uma rede de filas aberta que possui seis nós e três possíveis rotas.

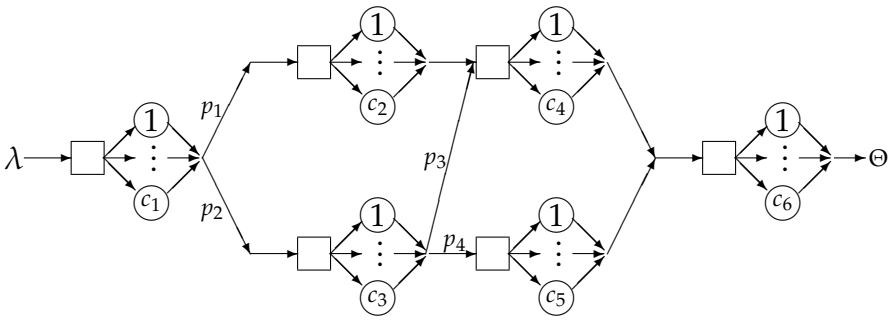

Figura 1 – Um exemplo de uma rede complexa de filas adapatado de MacGregor Smith e Cruz (2005) [1].

Para a rede de filas da Figura 1, em uma das possíveis rotas, os usuários entram no sistema no primeiro nó com taxa de chegada λ e são atendidos pelos primeiros c_1 servidores em seguida são atendidos por c_2 servidores em uma segunda tarefa, com probabilidade p_1, e por c_4 servidores em uma terceira tarefa. Por fim são atendidos por c_6 servidores, deixando a rede em seguida. Entretanto, após a tarefa dos c_1 servidores, a tarefa inicial, com probabilidade p_2, o nó subsequente pode ser a tarefa com c_3 servidores. Em seguida, com probabilidade p_3, os usuários se deslocam para o seu terceiro nó e são atendidos pelos c_4 servidores ou com probabilidade p_4 por c_5 servidores. Por fim, ambos os casos avançam para o último nó e são atendidos pelos c_6 servidores, deixando a rede em seguida.

Um problema de grande interesse é mensurar a eficiência do funcionamento em redes de filas. Existem diversas métricas capazes de aferir os níveis de qualidade da operação de tais redes. Cada uma desses métricas se direciona para algum interesse próprio. Diante disso, é necessário estabelecer algum funcional objetivo particular, associado à alguma, ou algumas dessas métricas. Em particular, uma investigação desafiadora ao lidar com redes multiservidores é estabelecer uma melhor maneira de alocar esses servidores em cada fila da rede, de forma que a performace do sistema seja melhorada.

Em geral, uma maneira de se obter a melhor performace é modelar matematicamente o sistema proposto e otimizar na busca computacional para a alocação mais adequada dos servidores. Observado por uma lente mais ingênua, um problema de otimização pode ser tratado como algo trivial, mas está longe disso. Investigações e problemas de otimização são recorrentes em nosso dia a dia. De acordo com Yang (2010) [2], a recorrência não é indicativo de trivialidade quando o assunto são os problemas de otimização. Um exemplo de grande destaque na literatura é o clássico problema do caixeiro viajante. Iniciamente é visto com um problema de definição simples, bem como sua compreensão. Por outro lado, é de certa forma surpreendente que não se conheça ainda um algoritmo eficiente para ele. A literatura apresenta uma vasta gama de mecanismos e algoritmos eficientes para problemas de otimização. Em particular, para uma versão do problema de alocação de servidores em redes de filas, Duarte (2022) [29] apresentou uma abordagem que aplica o clássico algoritmo de otimização *Simulated Annealing* (SA). Essa abordagem será discutida com mais detalhes neste estudo.

O SA é um algoritmo meta-heurístico de otimização. Uma meta-heurística é um método heurístico para resolver de forma genérica problemas de otimização. As meta-heurísticas são técnicas de otimização usuais principalmente em problemas de otimização combinatória. Em geral, para problemas dos quais não se conhece um método eficiente para obtenção de uma solução ótima global, as técnicas heurísticas e meta-heurísticas são aplicáveis. As meta-heurísticas combinam aleatoriedade de

conhecimento prévio no procedimento de busca por soluções no espaço das variáveis, com interesse em obter as melhores soluções possíveis no espaço dos objetivos de otimização. As meta-heurísticas utilizam mecanismos para evitar soluções ótimas locais na busca da direção de soluções ótimas globais.

O SA tem sua formulação conceitual baseada na ideia de recozimento de metais, em que a priori, se aquece algum material a uma elevada temperatura em um curto período de tempo, em seguida, o resfria lentamente até que se solidifique. Ao resfriar o material de forma lenta e controlada, é criado um ambiente em que os átomos possam se movimentar mais livremente e se organizarem de forma ótima naquele mínimo local, esse processo traz o menor número possível de imperfeições ao material. Nesse procedimento, o material passa por diversas mudanças, em cada estado gerado por determinada temperatura se pode observar uma quantidade de energia gasta, e para o objetivo final se almeja gastar o mínimo possivel de energia (baixa temperatura), de forma que se solidifique o material sem gerar imperfeições. Analogamente, cada estado do material representa o conjunto de soluções possíveis do algoritmo, a energia gasta para chegar em cada solução representa a função objetivo e a energia mínima corresponde a solução ótima local, possivelmente global, que o algoritmo retorna.

Como mencionado anteriormente, teoria de filas pode ser aplicada em diversos situações do cotidiano. Neste trabalho, o interesse é modelar redes de filas multiservidoras, em que cada uma das filas o processo de chegadas e atendimento é markoviano. O termo markoviano será melhor elucidado na parte metodológica desse estudo, mas trata-se de uma terminologia bastante usual para esta área de estudo. As filas sob investigação são multiservidores homogêneas, isto é, seus servidores desempenham a mesma função. Usualmente estas filas são denominadas filas $M/M/C$, de acordo com a clássica notação de Kendall [3]. Os indivíduos (ou entidades) entram no sistema proposto de acordo com um processo de Poisson de taxa λ e o tempo de duração do serviço executado é uma variável aleatória que segue uma distribuição exponencial com taxa μ. A priori, não existem restrições para a quantidade de indivíduos aguardando o serviço em filas no sistema, a disciplina de atendimento é do tipo: primeiro a chegar é o primeiro a ser atendido (*first come first served - FCFS*).

Redes de filas similares à rede proposta podem ser comparadas com a maneira que uma indústria de manufatura trabalha. Com essa abordagem, otimizar a alocação dos servidores, restrito a uma quantidade máxima e área de ocupação, com o objetivo de minimizar os tempos de serviço e maximizar a produtividade do sistema, pode trazer grandes contribuições tanto para o mundo coorporativo como para diversas outras áreas.

1.1 Motivação

Nessa área de estudo, o problema da alocação ideal de equipamentos em redes de filas, que é conhecido como Problema de Alocação de Servidores (SAP), ou Problema de Alocação de Recursos, tem sido intensamente estudado ao longo dos anos [30–33]. O SAP já foi bastante pesquisado. Diversos resultados sobre a alocação ideal de servidores já foram apresentados. Estudos foram feitos considerando nós únicos, redes abertas e fechadas, *buffers* infinitos e finitos e serviço exponencial. Um revisão bastante abrangente acerca do assunto foi apresentada por Smith et al. (2010) [34]. No entanto, a formulação aqui apresentada é inovadora, sua primeira aparição na literatura foi apresentada por Duarte (2022) [29]. Essa novidade é uma justificativa para essa nova discussão sobre o assunto.

Diversos exemplos de aplicação mostram a grande necessidade de investigações nessa área. Entre vários, suponha por exemplo o estudo em uma fila de atendimento de hospital, em que qualquer paciente entrante precisa passar pela triagem e, em sequência, ser direcionado para o médico que mais adequado para o atendimento. Posteriormente, o paciente pode sair do hospital (deixar a rede). Este é um exemplo de rede de filas que se inicia com todos os usuários se deslocando para a mesma fila da rede para serem atendidos pelos mesmos grupos de servidores. Depois, se deslocam para outra fila dependendo da sua necessidade de atendimento. Supondo que um mesmo médico seja capaz de realizar o atendimento primário na triagem e também possíveis atendimentos posteriores, qual é a melhor forma de alocar esses profissionais (servidores) para que o sistema ofereça o atendimento mais ágil possível mas também reduza a quantidade de médicos ociosos? Nessa situação, o funcional objetivo estaria associado à redução do ócio dos médicos e aumento da velocidade do atendimento. Será que a maioria dos médicos deveriam atender na triagem? Talvez dessa forma, fosse mais ágil inicialmente, mas e depois, poderia ocorrer um atraso no tempo de atendimento? Qual a melhor forma de balancear a distribuição desses médicos ao longo das tarefas de atendimento para que as filas do sistema tenham a menor retenção e os profissionais a menor ociosidade?

Uma alocação inadequada dos servidores poderiam levar à perda significativa da eficiência e rentabilidade do processo, dados os altos custos envolvidos. Para este cenário as abordagens de otimização são relevantes e neste conjunto de informações, alguns objetivos para este estudo podem ser delimitados.

1.2 Objetivos

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

1.2.1 Objetivos Gerais

i. apresentação de uma revisão bibliográfica na área de teoria de filas e problemas de otimização em filas;

ii. apresentação de propostas de formulação do problema de otimização multiobjetivo de alocação de servidores SAP;

iii. apresentação da inovadora formulação do problema de otimização multiobjetivo de alocação de servidores SAP apresentada por Duarte (2022) [29], uma formulação que garante alta produtividade dos servidores, baixo tempo de clientes na rede de filas restrito à um número total fixo de servidores.

1.2.2 Objetivos Específicos

i. adaptação específica da meta-heurística SA para o SAP em questão.

ii. obter configurações factíveis quanto ao total de servidores alocados, com elevada produtividade desses servidores, e com baixo tempo de permanência dos clientes na rede de filas;

iii. investigar a existência de padrões de alocação decorrentes de alterações nas probabilidades de roteamento nas redes de filas.

Este texto é organizado da seguinte forma, inicialmente um capítulo introdutório que aborda aspectos de pesquisa discutidos durante a concepção desse estudo, bem como o delineamento prévio de objetivos a serem trabalhados. Em seguida, o segundo capítulo apresenta uma relevante e atualizada 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 o algoritmo de otimização utilizado. O quarto capítulo apresenta de forma mais detalhada todo o conjunto de resultados alcançados. Por fim, o último capítulo apresenta as conclusões alcançadas através dessa investigação e também propostas de continuidade desse estudo.

2 Fundamentação Teórica

2.1 Teoria das Filas

As investigações associadas à teoria de filas buscam dar tratamento matemático para sistemas e redes que funcionam baseado na dinâmica de filas. Os sistemas de filas estão presentes em diversas situações de fluxo cuja demanda de entrada e velocidade de atendimento tenham um caráter estocástico.

Os sistemas de filas demandam investigação associadas à otimização da distribuição dos recursos envolvidos para a maximização da eficiência do atendimento. Ao abordar os recursos envolvidos, em geral, grandezas associadas com a quantidade de servidores e de espaços de espera estão envolvidas como variáveis importantes. Uma notação descritiva para modelos de filas enormemente difundida é a conhecida notação de Kendall [3].

2.1.1 Notação de Kendall

As filas são descritas em uma forma resumida $A/B/m/k$ em que: A representa a distribuição do tempo entre chegadas; B descreve a distribuição do tempo de serviço; m apresenta o número de servidores em paralelo; e k denota o espaço total disponibilizado para os clientes em espera além do número de clientes em atendimento. A Tabela 1, apresentada por Souza (2020) [35] esquematiza essa representação.

Tabela 1 – Parametrização para filas de acordo com a notação de Kendall [3].

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

Um exemplo particular seria considerar uma fila $M/D/2/\infty$ que indica um processo de fila com tempo entre chegadas markoviano, ou seja, com distribuição exponencial, tempo de serviço determinístico, dois servidores em paralelo, e sem restrição no tamanho máximo da capacidade do sistema. A omissão do termo k, na notação de

Kendall, implica que a referida fila tem capacidade infinita. Usualmente o símbolo M é utilizado para a distribuição exponencial. Isso acontece devido a associação ao termo markoviano para mencionar a distribuição exponencial.

A Tabela 1 apresenta algumas distribuições usuais em investigações para filas. Obviamente não são as únicas distribuições aplicáveis. É importante salientar que o interesse é ilustrar a estrutura da notação de Kendall e não delimitar toda e qualquer possibilidade de modelagem em filas. Em particular, este estudo investiga redes de filas do tipo $M/M/c$, ou seja, filas com chegadas e atendimentos markovianos com c servidores e sem limitação de áreas de espera. As investigações para otimização de redes de filas interessa em diversos setores do cotidiano.

Estudos baseados em redes de filas são de grande interesse em diversos problemas relevantes para a sociedade atual. Entre os potenciais usuários para modelos de otimização baseados em redes de filas diversas incluem cientistas da computação, engenheiros de produção entre outros pesquisadores. É muito clara a utilidade e o fato de que tais modelos tendem a auxiliar na compreensão e melhoria de vários sistemas reais, incluindo sistemas de manufatura [36], de produção [34] e de saúde [37], sistemas de tráfego de veículos e de pedestres [38–40], sistemas de computação e de comunicação [41–43], aplicações baseadas na *web*, configuradas em camadas [44] e com requisitos de qualidade de serviço (QoS) definidos em termos de tempo de resposta, taxa de atendimento (ou *throughput*), disponibilidade e segurança [45], entre outros [46, 47].

A proposição de um estudo dessa natureza requer uma descrição acerca da estruturação dos problemas de otimização e uma associação dessa descrição com a aplicação específica proposta nesta investigação.

2.2 Problemas de Otimização

De uma forma generalista, otimizar, no contexto de problemas de otimização, consiste em encontrar um conjunto solução que representa o conjunto dos melhores valores possíveis para satisfazer um dado problema. A medida de qualidade acerca das soluções candidatas é dada por um ou mais funcionais objetivo, cujas imagens são avaliados quanto à sua adequabilidade ao problema sob investigação. A Figura 2 proposta por Yang (2010) [2] representa um diagrama morfológico para as variantes associada com um problema clássico de otimização.

Em particular, a investigação objeto desse estudo se posiciona entre os problemas de otimização multiobjetivo, restrito com funções não-lineares, variáveis decisórias discretas. Diante disso, o conceito de dominância multiobjetivo é de extrema importância neste estudo.

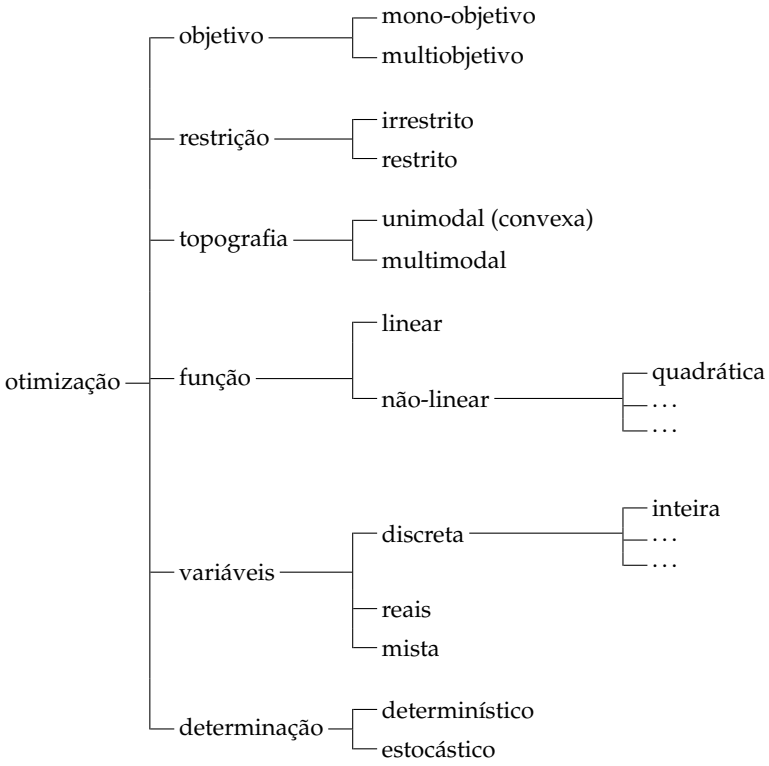

Figura 2 – Classificação dos problemas de otimização segundo Yang (2010) [2]

Como mencionado anteriormente, todo problema de otimização demanda uma função objetivo com interesse na busca por maximizar, ou minimizar, algum vetor de parâmetros dependende de variáveis decisórias. O desafio maior surge quando se possui um problema de otimização multiobjetivo, isto é, quando a natureza do problema acarreta em otimizar uma função objetivo no espaço multidimensional. Em geral, os objetivos unidimensionais, vistos de forma individual podem ser conflitantes entre si.

Diversos problemas do dia a dia podem seguir esse comportamento. Por exemplo, suponha uma pessoa que deseja comprar o melhor telefone celular possível dentre os existentes no mercado. Porém, também queira gastar o mínimo possível em recurso financeiro para atingir esse objetivo. O conflito nos objetivos deste problema fica explícito, telefone celulares superiores são, normalmente, mais caros também. Dentre todas as opções disponíveis no mercado, se o cliente optar por um telefone celular de custo mais limitado, necessariamente existirá um celular com melhor desempenho, ou seja, nenhuma solução que tenha menor custo, tende a ser também aquela que oferece a melhor performace. Por outro lado, existem aparelhos de telefonia celular com custos similares, mas desempenhos distintos entre si, e é nessa área cinza que é possível encontrar uma opção de compra que talvez se ajuste com a melhor compra possível.

Essa ideia intuitiva pode ser extrapolada para as mais diversas áreas de estudo

que possuem dois ou mais objetivos conflitantes. O problema de otimização multiobjetivo com m objetivos pode ser definido como:

Definição 2.2.1. *Dado um conjunto de variáveis de decisão $x = \{x_1, \ldots, x_n\}$ em um espaço específico, denominado espaço de busca $\mathcal{X}$, se deseja encontrar algum elemento $x^* = (x_1^*, \ldots, x_n^*)$ específico, pertencente a conjunto $\mathcal{X}$ que minimize simultaneamente as m funções objetivas associadas funcional objetivo $F(x_1, \ldots, x_n) = \big[f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\big]$. O problema de otimização fica descrito como:*

$$\text{minimizar } F(x_1, \ldots, x_n) = \big[f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\big], \quad (2.1a)$$

sujeito a:

$$\begin{aligned} & x^* \in \mathcal{X} \subset \mathbb{R}^n, \\ & \xi(x^*) = 0, \\ & \zeta(x^*) \leqslant 0, \end{aligned} \quad (2.1b)$$

em que $f_j : \mathbb{R}^n \longrightarrow \mathbb{R}$, $j \in \{1, ..., m\}$ são as funções objetivas, que algumas das vezes são conflitantes e $\xi : \mathbb{R}^n \longrightarrow \mathbb{R}$ e $\zeta : \mathbb{R}^n \longrightarrow \mathbb{R}$ são funções que impõem restrições específicas ao problema.

Em geral, $x^* = (x_1^*, \ldots, x_n^*)$ é definido por um conjunto de restrições e limites de especificação para as variáveis de decisão, e atender estas condições pode ser garantir a factibilidade de um candidato x^* a ser solução do problema de otimização.

A definição 2.2.1 remete para um problema de minimização, sem qualquer perda de generalidade pode ser adaptada para descrever um problema de maximização. Para os problemas de otimização multiobjetivo, a comparação entre soluções não é uma abordagem completamente trivial. Para problemas mono-objetivo, a comparação entre soluções se restinge à ordem de grandeza comparativa entre números no espaço dos reais, já para problemas multiobjetivo, este conceito precisa ser melhor elucidado. É o conceito de dominância.

2.2.1 Dominância Multiobjetivo

O conceito de solução ótima para um problema de otimização multiobjetivo remonta do seculo XIX, foi apresentado inicialmente por Vilfredo Pareto [48]. A ideia original surgiu ao pensar no bem estar populacional frente à distribuição de recursos. Observe que, em um contexto social com bens e recursos limitados (ou até mesmo escassos), necessariamente para um indivíduo ter mais recursos, e por consequencia, estar mais satisfeito, outro indivíduo precisa ter algum tipo de perda em sua situação de vida. Soluções que tendem a minimizar a piora do bem estar social condicionadas a uma distribuição mínima de recursos, são chamadas de soluções ótimas de Pareto.

Quando o problema de otimização leva todas as funções objetivas para a direção de minimização, pode-se descrever as soluções ótimas de Pareto pelas seguintes definições:

Definição 2.2.2. *Uma solução x é dita ser* dominante *com respeito a alguma solução y, isso de acordo com a formulação multiobjetivo descrita nas equações 2.1a e 2.1b se, e somente se, $f_i(x) \leqslant f_i(y)\ \forall i; i \in \{1, ..., n\}$ e ainda, $f_j(x) < f_j(y)$ para pelo menos um valor j tal que $j \in \{1, ..., n\}$. Usualmente afirma-se que x é uma solução não-dominada com respeito à solução y.*

A Figura 3 ilustra três situações comparativas entre soluções para um espaço bi-dimensional, em todos os cenários a solução x é não-dominada com respeito à solução y.

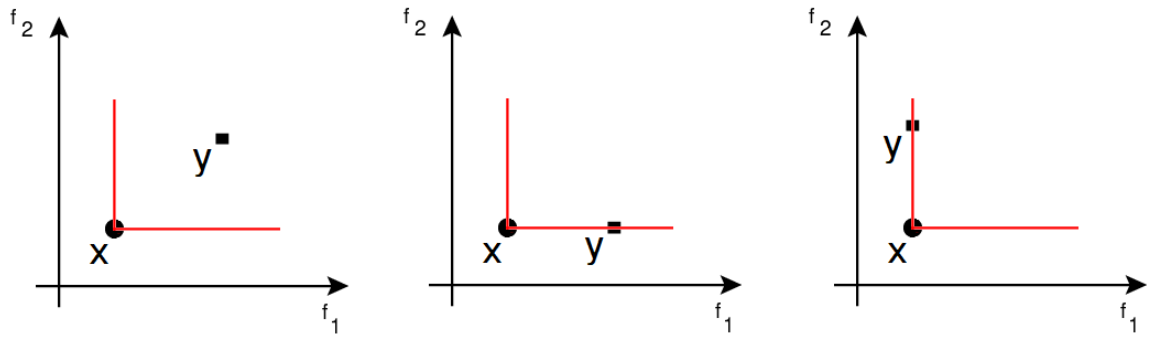

Figura 3 – Soluções dominadas (▪) e soluções não-dominados (●).

Definição 2.2.3. *Seja $\mathcal{X}$ o espaço de busca, ou seja, o espaço de todas as soluções a serem investigadas, uma solução x é dita ser* eficiente, *ou solução* Pareto-ótimo, *isso de acordo com a formulação multiobjetivo descrita nas equações 2.1a e 2.1b se, e somente se, x é não dominada com respeito a todas as soluções factíveis no espaço $\mathcal{X}$. O conjunto de todas as soluções não-dominadas, com respeito a um dado espaço sob investigação, é chamado usualmente de* fronteira, *ou* Pareto-front.

A Figura 4 apresenta um vasto conjunto de possíveis soluções que representa o espaço de busca $\mathcal{X}$ e separa as soluções não-dominadas das demais. Dessa forma, a representação ilustra a Fronteira de pareto em um espaço bi-dimensional. É importante salientar que ao comparar soluções distintas pertencentes à fronteira, não se estabelece uma relação de dominância. Entre si, as soluções da fronteira não dominam, tampouco são dominadas. Em outras palavras, restrito somente às soluções pertencentes à fronteira, elas estão em situação de igualdade quanto à relação de dominância.

Uma estratégia de comparação pode ser considerar a métrica de hipervolume (HV) [49]. Da Fronteira de pareto é possível se obter o hipervolume de Pareto. Na

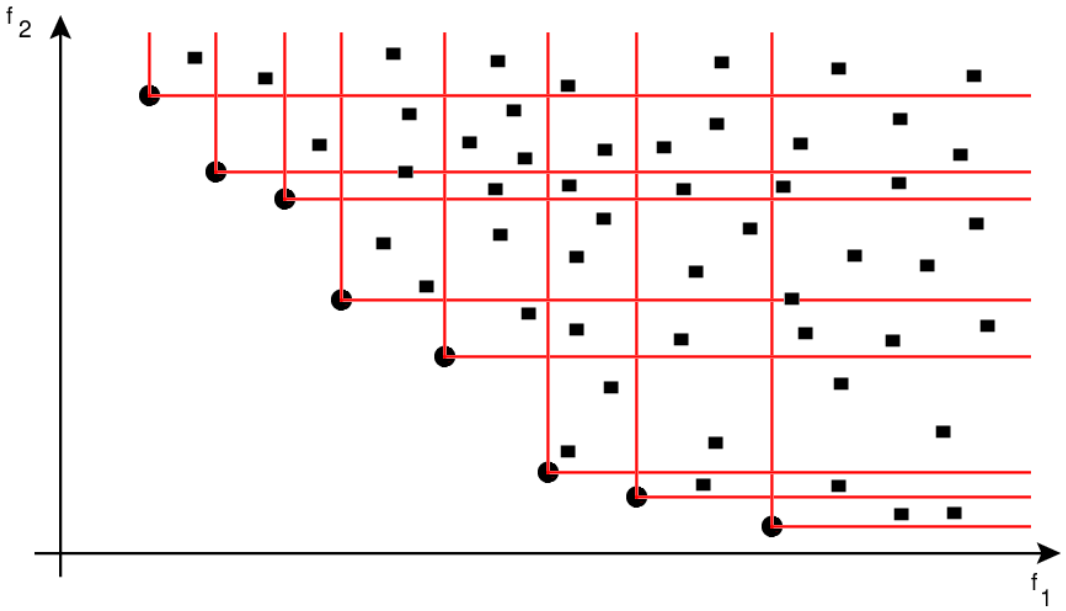

Figura 4 – Fronteira de Pareto das soluções não-dominados com soluções dominadas (▪) e soluções não-dominados (•).

Figura 4 o hipervolume é representado pela área em branco abaixo da fronteira. O hipervolume é uma métrica de avaliação para mensurar cobertura das soluções na fronteira. O hipervolume entre a frente de Pareto um ponto fixo máximo nos dois objetivos será considerado nesse estudo. Assim, a frente de Pareto superior possui o menor hipervolume verificado. De forma resumida, o Hipervolume representa a área de possível melhoria da fronteira. Quanto maior o hipervolume, mais as soluções teoricamente ainda podem melhorar.

2.3 O Problema de Alocação de Servidores (SAP)

Diversas investigações para alocação adequada de servidores já foram apresentadas na literatura. Novamente este estudo será abordado aqui, porém com uma formulação bastante recente apresentada por Duarte (2022) [29]. A inovação dessa formulação é uma justificativa para a relevância dessa discussão sobre o assunto. Diante disso, uma revisão de literatura acerca do tema se torna de extrema importância.

Yechiali et al. (1971) [4] apresentam um dos primeiros estudos neste campo, um sistema markoviano de dois níveis do modelo de filas $M/M/1$. Boxma et al. (2001) [5] discutem uma fila $M/G/1$ cuja velocidade do servidor varia entre dois valores máximo e mínimo. Baykal-Gürsoy et al. (2004) [6] e Baykal-Gürsoy et al. (2009) [7] discutem sistemas de filas dependentes do estado em estudos de transportes. Os estudos abordam a influência de incidentes de tráfego para formação de congestionamentos de veículos em partes de uma rodovia. Gao et al. (2013) [8] abordam uma fila $M/G/1$ com férias

únicas de trabalho e interrupção de férias com a programação Bernoulli.

Heidemann (2001) [9] investigaram o comportamento transiente em filas $M/M/1$. Vandaele et al. (2000) [10] utilizaram filas $M/M/1$, $M/G/1$ e $GI/G/1$ para modelar o fluxo de tráfego. Van Woensel & Vandaele (2006) [11] além de Van Woensel (2006) [12] validaram o uso de modelos de filas para descrever o fluxo de tráfego empiricamente e via simulação, respectivamente. Van Woensel & Cruz (2009) [13] investigaram custos de congestionamento em ambientes complexo, dinâmico e estocástico através de filas $M/M/1$ e $M/G/1$.

Hanukov et al. (2019) [14] apresentam um sistema de serviço de dois servidores $M/M/2$ cujos servidores ociosos produzem e armazenam serviços preliminares para reduzir o tempo de permanência dos clientes. Wang et al. (2010) [15] discutem a introdução de um modelo *QuickPass* para melhorar o sistema de filas de bancos modelado como $M/M/c$ através de um algoritmo guloso ajustado. Liu et al. (2017) [16] propuseram um modelo de avaliação de praças de pedágio baseado em autômatos celulares e teoria de filas $M/M/c$. Khodemani-Yazdi et al. (2019) [17] apresentam um problema de localização de *hub* hierárquico bi-objetivo cujos objetivos são minimizar o custo total e o comprimento máximo da rota, simultaneamente. As estruturas de filas para esses tipos de instalações são consideradas como $M/M/c$ e $M/M/1$. Goodarzi et al. (2021) [18] apresentam uma formulação $M/M/c$ para modelar um ambiente de *cross-docking*, no qual o plano de despacho dos veículos para iniciar o processo de coleta deverá ser determinado.

Hillier et al. (1995) [19] e Spinellis et al. (2000) [20] estudaram redes de filas Markovianas multi-servidor finitas. As redes são dispostas sempre em série. Hillier et al. (1995) [19] utilizam uma estratégia de enumeração completa para otimizar redes de filas Markovianas multi-servidores. No entanto, os resultados apresentados são apenas para redes de filas relativamente simples e pequenas. Spinellis et al. (2000) [20] combinam uma estratégia de *Simulated Annealing* e o método de expansão generalizada (GEM), a conhecida técnica desenvolvida por Kerbache & MacGregor Smith (1987) [21] para otimizar o desempenho das linhas de produção.

A literatura apresenta uma vasta gama de estudos que utilizam sistemas de filas multi-servidor, isto garante até o presente momento o interesse neste assunto [22–28]. Particularmente este estudo volta na abordagem desse tema para apresentar uma formulação multiobjetivo inovadora de Duarte (2022) [29]. Essa nova formulação apresentou resultados promissores mesmo através de uma heurística de otimização de custo computacional baixo.

Na literatura já existem trabalhos focados na minimização do tempo de permanência dos clientes no sistema de filas. Essa proposição é conflitante com o custo total alocado em servidores. O objetivo dessa investigação é abordar uma proposição

capaz de minimizar o tempo de permanência dos clientes no sistema de filas e simultaneamente maximizar a produtividade dos servidores das filas do sistema por meio da realocação dos servidores entre as filas da rede. A medida de produtividade dos servidores utilizada neste estudo é a soma da razão entre o número esperado de clientes em serviço pelo número de servidores para cada fila da rede.

Uma razão entre o número esperado de clientes em serviço pelo número de servidores próxima da unidade implica que o nível de ociosidade dos servidores é menor. Porém, do ponto de vista econômico, minimizar essa razão tende a aumentar o tempo de permanência dos clientes. Isso deixa clara a natureza conflitante dos dois objetivos. Este estudo apresenta uma abordagem heurística para fornecer uma fronteira sub-ótima de soluções para estes dois objetivos conflitantes através do clássico algoritmo *Simulated Annealing*.

2.4 O algoritmo *Simulated Annealing*

O algoritmo *Simulated Annealing* (SA) é uma meta-heurística de otimização ideal para encontrar soluções satisfatórias para problemas de otimização de grande complexidade. Proposto por Kirkpatrick et al. (1983) [50] e posteriormente por Černỳ (1985) [51], o algoritmo vem recebendo muita atenção de diversos estudiosos, uma vez que apresenta grande eficiência para solucionar problemas de otimização combinatorial com baixo custo computacional.

A SA teve sua origem baseada em métodos de análise das propriedades dos átomos presentes em substâncias solidas e líquidas da mecânica estatística. Por existir uma quantidade extrema de átomos, a análise do comportamento de uma substância com relação a variação da temperatura é feita a partir de seu comportamento mais provável, devido a forma aleatória que os átomos se comportam. Uma questão fundamental da área veio da seguinte pergunta: o que acontece quando um solido tem sua temperatura elevada e em seguida reduzida ao extremo? Se essa redução de temperatura ocorrer de maneira específica, o material pode atingir o que chamam de estado fundamental ou estado da energia mínima. Como esse estado não é o estado natural da substância, só pode ser obtido por meio de experimentos, esse processo para chegar na energia mínima é chamado de *annealing* (recozimento) e segue dois passos fundamentais:

i. deixar a temperatura do matérial suficientemente alta e, em seguida;

ii. reduzir a temperatura de forma lenta e gradual em banho térmico.

O material em processo de redução de temperatura controlada, lenta e gradual permite que os átomos tenham tempo para se reorganizarem da melhor forma possível formando

uma estrutura sólida e uniforme, com o mínimo gasto de energia. Caso esse processo aconteça de maneira mais brusca, os átomos se organizam de forma aleatória, formando uma estrutura irregular e mais fraca, com alto gasto de energia devido ao grande esforço interno necessário durante a solidificação.

Metropolis et al. (1953) [52] apresentou um algoritmo que simulava de maneira simples a evolução da temperatura de um sólido em banho quente até o equilíbrio térmico. Kirkpatrick et al. (1983) [50] e Černỳ (1985) [51] então mostraram que esse algoritmo poderia ser extrapolado para problemas de otimização em geral.

O funcionamento do *Simulated Annealing* é basicamente uma extensão de um método de busca local padrão de problemas de otimização combinatorial. Um algoritmo de busca local padrão precisa apenas de uma função objetivo e um esquema de vizinhança de soluções definido, com isso, o método procura dentro de suas opções qual é a solução que retorna melhoria para a função objetivo. Apesar desse esquema sempre retornar uma solução final melhor ou igual a solução inicial, essa resposta pode ser muito pobre comparada a solução global ótima. Isso ocorre pois os métodos de busca muito simplistas caem facilmente em uma armadilha do ótimos locais, isto é, dado todas as soluções vizinhas, nenhuma é melhor que a atual, mas em uma visão do todo, aquela não é a solução ótima.

A Figura 5 ilustra a situação das referidas armadilhas de máximos locais. Uma busca por soluções candidatas à minimizar cuja vizinhança investigada seja restrita ao intervalo (a, b) observam o ponto x como solução ótima. Por outro lado, é fácil ver que se fosse possível investigar a vizinhança (c, d), a solução ótima seria o ponto y.

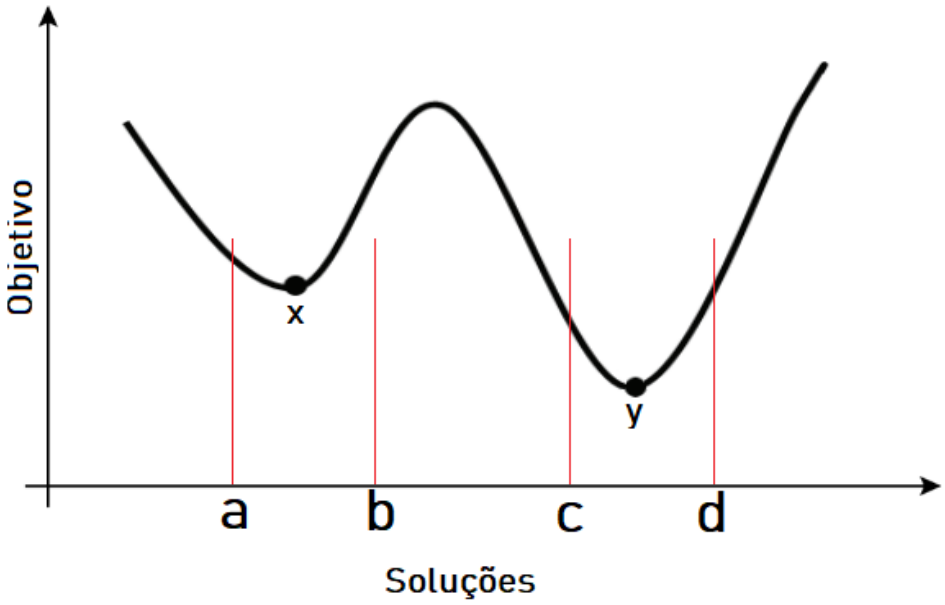

Figura 5 – Análise visual com solução ótima local e solução ótima global.

O SA, assim como um método de busca padrão, também trabalha analisando

soluções vizinhas e caso observe alguma possibilidade de melhoria, substitui a solução corrente. Entretanto, seu diferencial principal é oferecer uma forma de escapar dos ótimos locais aceitando soluções menos eficientes condicionadas a determinada probabilidade controlada e/ou parâmetro de aceitação. A probabilidade de aceitar uma solução que piore a função objetivo, isto é, dê um passo atrás na otimização, é chamada de função de aceitação. Esta por sua vez, usualmente representada por $g(\Delta, T)$, em que Δ é uma diferença, sob alguma métrica, entre as soluções analisadas e T um parâmetro de controle. Em analogia ao processo de recozimento, T representa a temperatura. A função de aceitação precisa retornar um número entre 0 e 1, ou seja, uma medida de probabilidade. Este valor é utilizado como parâmtero decisório para aceitar soluções inferiores. De forma similar ao processo físico, a função $g(\Delta, T)$ implica em:

i. a probabilidade retornada pela função é inversamente proporcional ao tamanho de Δ;

ii. quando T é muito alto, a função retorna grande chance de aceitação de soluções inferiores, quando T se aproxima de 0, a maioria das soluções piores são rejeitadas.

Por exemplo, seja $g(\Delta, T) = e^{-\Delta/T}$, quanto maior o valor de Δ entre as soluções, menos provavel é aceitar uma solução pior, ou seja, soluções com discrepância muito grande em sua função objetivo são menos prováveis de serem aceitas. Quanto mais próximo de 0 é o argumento T, também é menos provável aceitar soluções inferiores, pois significa que a solução atual está próxima do ótimo global.

O SA é iniciado de forma aleatória, preferencialmente com um valor de temperatura T relativamente alto, para evitar ficar prematuramente preso em um mínimo local. O algoritmo parte dessa solução inicial, procura outra solução vizinha (isso para algum critério de vizinhança pré-fixado) e as compara quanto a função objetivo se esta for unidimensional, ou alguma quantidade unidimensional para funções multiobjetivos. Se a solução comparada é melhor, descarta-se a solução anterior e adota a nova solução como corrente. Quando a solução comparada é pior, o algoritmo aceita ou rejeita a possibilidade de ainda assim vasculhar através dessa solução de acordo com a função de aceitação. O processo se repete até que algum critério de parada seja satisfeito. Após a parada, dentre todas as soluções investigadas o algoritmo retorna a melhor solução observada ao longo de toda a busca. Apesar do método não garantir convergência para o ótimo global, em muitos casos essa solução é suficientemente aceitável, devido seu baixo custo computacional e de implementação e também à qualidade das soluções usualmente fornecidas.

3 Abordagem do Problema e Aspectos Metodológicos

O interesse deste estudo é a investigação acerca de algoritmos eficazes para obter a alocação ótima de servidores. Essa tarefa é executada para otimizar o desempenho de uma rede de filas multi-servidor markoviana (chegadas conforme distribuição Poisson e tempos de atendimento com distribuição exponencial). Vários algoritmos têm sido propostos para resolver esse tipo de problema. Existe uma forte dependência entre os algoritmos e a formulação de programação matemática utilizada. A introdução deste tema se dará através da discussão de uma formulação de objetivo único do SAP, descrita a seguir.

3.1 Formulação Mono-objetivo

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 SAP, em sua formulação primal pode ser descrito da seguinte forma:

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

sujeito a:

$$\begin{aligned} &W(\mathbf{C}) \leqslant W_{\max}, \\ &c_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{3.1b}$$

que minimiza o custo total de alocação de servidores para uma rede com m filas, sujeito a um limiar $W_{\max}$ (tempo total esperado de clientes na rede de filas) e que a quantidade de servidores c_i seja um inteiro positivo.

Uma outra proposição de formulação bastante relacionada com a anterior, é uma proposta de SAP dual, que busca minimizar o tempo médio de permanência dos clientes na rede de filas, $W(\mathbf{C})$, restrito a um limite máximo para a alocação total de servidores ao longo da rede de filas $C_{\max}$, descrito na formulação seguintes. A dualidade dessa proposição com respeito a proposição anterior decorre do fato de $W(\mathbf{C})$ estar associado ao funcional objetivo, e $\sum_{i=1}^{m} c_i$ estar associado à restrição e isso é invertido para

a formulação primal anterior. A proposta dual de formulação é uma clara analogia para o clássico problema da mochila estocástico [1], que pode ser definido da seguinte forma:

$$\text{minimize } W(\mathbf{C}), \tag{3.2a}$$

sujeito a:

$$\begin{aligned} &\sum_{i=1}^{m} c_i \leqslant C_{\max}, \\ &c_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{3.2b}$$

que minimiza o tempo total esperado pelos clientes na rede de filas, $W(\mathbf{C})$, sujeito a um limiar máximo, $C_{\max}$, para a alocação total de servidores ao longo da rede, e que a quantidade de servidores c_i seja um inteiro positivo.

Embora as duas formulações do SAP 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

As formulações descritas anteriormente tendem a buscar uma alta alocação de servidores para reduzir o tempo total dos clientes na rede de filas. Por outro lado, essa estratégia não se preocupa com o custo adicional resultante do acréscimo de servidores que permanecem ociosos por uma fração de tempo significativa. Uma formulação adequada deve considerar o tempo geral dos clientes, mas também alguma medida da produtividade do servidor. Normalmente, a fração não ociosa do servidor em uma fila é avaliada como a proporção entre o número esperado de clientes em serviço pelo número de servidores. Quando essa relação está mais próxima da unidade, maior será a produtividade dos servidores naquela fila.

O problema de otimização de redes filas $M/M/c$, descrito, pode ser reformulado para uma versão multiobjetivo que compreende a maximização da produtividade dos servidores envolvidos e a minimização do tempo total de permanência dos clientes na rede de filas. Essa formulação foi descrita por Duarte (2022) [29] da seguinte maneira:

$$\text{maximizar } F(\mathbf{C}) = \Big[f_1(\mathbf{C}), f_2(\mathbf{C}) \Big], \tag{3.3a}$$

sujeito a:

$$\begin{aligned} &c_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\sum_{i=1}^{m} c_i = C_{tot}, \end{aligned} \tag{3.3b}$$

que maximiza a produtividade dos servidores e simultaneamente minimiza o tempo total esperado dos clientes na rede de filas. Então, $f_1(\mathbf{C}) = -W(\mathbf{C}) = -\sum_{i=1}^{m} W(c_i)$ está associado ao tempo total esperado dos clientes na rede de filas enquanto a função $f_2(\boldsymbol{\mu}) = P(\mathbf{C}) = \sum_{i=1}^{m} P(c_i)$ representa a produtividade dos servidores.

O problema foi transformado em um problema somente de maximização ao multiplicar por -1 o objetivo associado ao tempo total de permanência dos clientes na rede de filas. Dessa forma ele se transforma em um objetivo de maximização.

A produtividade dos servidores na fila única com c_i servidores é dada por $P(c_i) = [L(c_i) - L_q(c_i)]/c_i$, em que $L(c_i)$ é o número esperado de clientes na i-ésima estação de serviço (incluindo clientes em serviço e clientes esperando na fila), $L_q(c_i)$ é o número esperado de clientes esperando na fila da i-ésima estação de serviço e c_i é o número de servidores na i-ésima estação de serviço.

A restrição $\sum_{i=1}^{m} c_i = C_{tot}$ garante que o problema proposto preserva o total de sevidores alocados. O interesse está na realocação de servidores para melhorar o desempenho da rede de filas. Diante disso, nós temos aqui uma clara analogia ao problema da mochila estocástico (mais detalhes em [53]).

3.2.1 Obtenção de Medidas de Desempenho em Filas $M/M/c$

As expressões para $L(c_i)$ e $L_q(c_i)$ devem ser detalhadas para utilizar a formulação multiobjetivo anterior. Para uma fila simples $M/M/c_i$, com taxa de chegada λ e taxa de atendimento μ (igual para todos os servidores), seja p_j a probabilidade de j clientes (incluindo clientes em serviço e clientes esperando na fila). Para tanto, p_j é dado por:

$$p_j = \begin{cases} \dfrac{1}{j!}\left(\dfrac{\lambda}{\mu}\right)^j p_0 & \text{se } j \leqslant c_i \\ \dfrac{1}{c_i!}\left(\dfrac{\lambda}{\mu}\right)^{c_i}\left(\dfrac{\lambda}{c_i\mu}\right)^{j-c_i} p_0 & \text{se } j > c_i\,. \end{cases} \tag{3.4}$$

Diante disso,

$$\sum_{j=0}^{\infty} p_j = p_0\left[\sum_{j=0}^{c_i}\frac{1}{j!}\left(\frac{\lambda}{\mu}\right)^j + \sum_{j=c_i+1}^{\infty}\frac{1}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i}\right] = 1\,. \tag{3.5}$$

Note que p_0 é a probabilidade de 0 clientes, ou seja, nenhum cliente na fila e todos os servidores ociosos. Acaba por funcionar como uma constante normalizadora para garantir que as probabilidades p_j configurem uma função de probabilidade.

A existência de distribuição invariante está condicionada à convergência da quantidade $\sum_{j=c_i+1}^{\infty}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i}$, essa convergência ocorre se, e somente se, $\lambda < c_i\mu$, para este caso, $\sum_{j=c_i+1}^{\infty}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i} = \frac{\lambda}{c_i\mu-\lambda}$.

Dessa forma,

$$p_0 = \frac{1}{\left[\sum_{j=0}^{c_i}\frac{1}{j!}\left(\frac{\lambda}{\mu}\right)^j + \frac{1}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}\left(\frac{\lambda}{c_i\mu-\lambda}\right)\right]}. \tag{3.6}$$

Considerando a expressão para p_0, as expressões para $L_q(c_i)$ and $L(c_i)$ são:

$$\begin{aligned} L_q(c_i) &= \sum_{j=c_i+1}^{\infty}(j-c_i)\frac{1}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i}p_0 \\ &= \frac{1}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}p_0\sum_{j=c_i+1}^{\infty}(j-c_i)\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i} \\ &= \frac{1}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}\left[\frac{\lambda c_i\mu}{(c_i\mu-\lambda)^2}\right]p_0 \end{aligned} \tag{3.7}$$

e

$$\begin{aligned} L(c_i) &= \sum_{j=0}^{c_i} j\frac{1}{j!}\left(\frac{\lambda}{\mu}\right)^j p_0 + \sum_{j=c_i+1}^{\infty} j\frac{1}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i}p_0 \\ &= \sum_{j=0}^{c_i}\frac{p_0}{(j-1)!}\left(\frac{\lambda}{\mu}\right)^j + \frac{p_0}{(c_i-1)!}\left(\frac{\lambda}{\mu}\right)^{c_i}\sum_{j=c_i+1}^{\infty}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i} \\ &\quad + \sum_{j=c_i+1}^{\infty} j\frac{1}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i}p_0 \\ &= \sum_{j=0}^{c_i}\frac{p_0}{(j-1)!}\left(\frac{\lambda}{\mu}\right)^j + \frac{p_0}{(c_i-1)!}\left(\frac{\lambda}{\mu}\right)^{c_i}\sum_{j=c_i+1}^{\infty}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i} + L_q(c_i) \end{aligned} \tag{3.8}$$

A produtividade dos servidores de uma única fila com c_i servidores é dada por:

$$\begin{aligned} P(c_i) &= \frac{L(c_i)-L_q(c_i)}{c_i} \\ &= \sum_{j=0}^{c_i}\frac{p_0}{c_i(j-1)!}\left(\frac{\lambda}{\mu}\right)^j + \frac{p_0}{c_i!}\left(\frac{\lambda}{\mu}\right)^{c_i}\sum_{j=c_i+1}^{\infty}\left(\frac{\lambda}{c_i\mu}\right)^{j-c_i} \end{aligned} \tag{3.9}$$

Dado que $L(c_i)$ é o número esperado de clientes na i-ésima estação de serviço (incluindo clientes em serviço e clientes esperando na fila). Usando a clássica lei de Little [54], o tempo esperado dos clientes na i-ésima estação de serviço é dado por:

$$W(c_i) = \frac{L(c_i)}{\lambda} \tag{3.10}$$

3.3 Detalhamento do Algoritmo *Simulated Annealing*

Como mencionado na seção 2.4, o algoritmo SA depende da definição de uma função objetivo, e também de um critério de vizinhança entre as soluções candidatas. O propósito é utilizar uma cadeia de Markov com espaço de estados composto por um conjunto de soluções possíveis para o problema de otimização. O n-ésimo estado da cadeia de Markov é uma solução corrente $\texttt{sol}_{curr}$, uma solução vizinha $\texttt{sol}_{neig}$ é escolhida aleatoriamente. Se a solução vizinha domina a solução corrente então a solução vizinha será $n+1$-ésimo estado da cadeia de Markov.

Por outro lado, mesmo que a solução vizinha não domine a solução corrente, ela poderá ser o $n+1$-ésimo estado da cadeia de Markov p, caso contrário a cadeia permanecerá no estado referente à solução corrente. Este é exatamente o caráter heurístico do algoritmo que permite a possibilidade de escapar de bacias atrativas de soluções locais. A escolha de p geralmente depende de o número de passos já realizadas pela cadeia de Markov e/ou pela diferença da função objetivo entre as soluções corrente e vizinha. Na nomenclatura de definição do algoritmo, a probabilidade p é a função $g(\Delta, T)$. Em um problema de função objetivo m-dimensional, uma escolha usual é $p = g(\Delta, T) = e^{-\Delta/T}$, em que

$$\Delta = \max_{i \in \{1,2,\ldots,m\}} \left| F_i(\texttt{sol}_{neig}) - F_i(\texttt{sol}_{curr}) \right| . \tag{3.11}$$

Para a formulação específica do problema sob investigação (ver Equações 3.3a e 3.3b), uma solução qualquer $\texttt{sol}_1$, ao gerar N passos sucessivos da cadeia de Markov com espaço de estados $\Omega = \{\texttt{sol}_1, \texttt{sol}_2, \ldots, \texttt{sol}_N\}$, é possível estimar a solução subótima $\texttt{sol}^*$. O objetivo aqui é a busca entre possíveis alocações $C = \{c_1, c_2, \ldots, c_m\}$ para uma rede de m filas que sejam ótimas de acordo com a formulação multiobjetivo em estudo, restrito ao conjunto Ω. Para realizar tal tarefa, é necessário definir um conceito de vizinhança entre as possíveis alocações. Este estudo utiliza dois formatos distintos. Dados dois valores aleatórios $a, b \in \{1, 2, \ldots, m\}$ tal que $c_a > 1$, uma possível solução vizinha para ser considerada a nova alocação é dada por:

$$\texttt{PerturbSolution}_1 = (c_1, ..., c_{a-1}, c_a - 1, c_{a+1}, ..., c_{b-1}, c_b + 1, c_{b+1}, ..., c_m) . \tag{3.12}$$

Para a segunda estrutura de vizinhança, dados dois valores aleatórios $a, b \in \{1, 2, \ldots, m\}$, uma possível solução vizinha a ser considerada para a nova alocação é dada por:

$$\texttt{PerturbSolution}_2 = (c_1, ..., c_{a-1}, c_b, c_{a+1}, ..., c_{b-1}, c_a, c_{b+1}, ..., c_m)\,. \tag{3.13}$$

A probabilidade para utilização da primeira estrutura de vizinhança foi 0,7, e 0,3 para a segunda estrutura de vizinhança. Diversos valores foram testados (os testes não são apresenteados aqui), a configuração utilizada foi a de melhor comportamento nos testes. É importante observar que este critério preserva a restrição $\sum_{i=1}^{m} c_i = C_{tot}$, ou seja, preserva a estrutura do problema da mochila estocástico. Uma versão resumida pode ser verificada no Algoritmo 1.

Algoritmo 1 Algoritmo multiobjetivo *Simulated Annealing*

```
/* inicializar temperatura T_1 */
/* gerar conjunto de soluções iniciais {C_1, C_2, ..., C_w} */
para j = 1; j ⩽ w, j = j + 1 faça
    i ← 1
    repita
        se i = 1 então
            sol_max ← sol_i ← C_j
        U ← uniforme(0, 1)
        se U > 0.3 então
            sol_aux ← PerturbSolution_1(sol_i)
        senão
            sol_aux ← PerturbSolution_2(sol_i)
        se sol_aux domina sol_i então
            sol_{i+1} ← sol_aux
            se sol_aux domina sol_max então
                sol_max ← sol_aux
        senão
            U ← uniform(0, 1)
            Δ ← max_{i∈{1,2}} |F_i(sol_i) − F_i(sol_aux)|
            se U < e^{−Δ/T_i} então
                sol_{i+1} ← sol_aux
        i ← i + 1
        /* atualiza temperatura, T_i */
    até i > i_max ou T_i < Tε
    sol_final_j ← sol_max
imprime sol_final
```

De volta ao assunto das perturbações, para a primeira perturbação, o algoritmo seleciona duas filas aleatoriamente, em seguida, subtrai um servidor de uma delas e o soma na outra. Basicamente muda um servidor de uma fila aleatória e o aloca em

outra também aleatória. Já para a segunda perturbação de vizinhança, o algoritmo simplesmente troca a quantidade total de servidores de uma fila por outra. Também seleciona essas duas filas de forma aleatória.

4 Resultados Alcançados

O algoritmo de otimização discutido anteriormente foi implementado em R *statistical software* (R Core Team, 2021 [55]). O ambiente de execução para realização dos experimentos computacionais foi um Intel (R) Core (TM) i7-8565U 1,80GHz, executando Windows 10 Pro 64 bits, com 16,00 GB de memória RAM.

Diferentes topologias de redes complexas de filas foram utilizadas experimentalmente. A rede de filas apresentada no capítulo 1, na Figura 1 foi utilizada como uma rede mista que inclui situações de filas em série, com fusão e com divisão. Outras topologias mais enxutas para testar os efeitos de série, fusão e divisão são apresentados nas Figuras 6, 7 e 8, respectivamente.

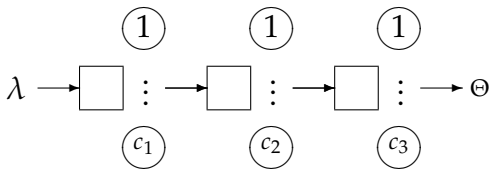

Figura 6 – Rede complexa de filas com topologia série.

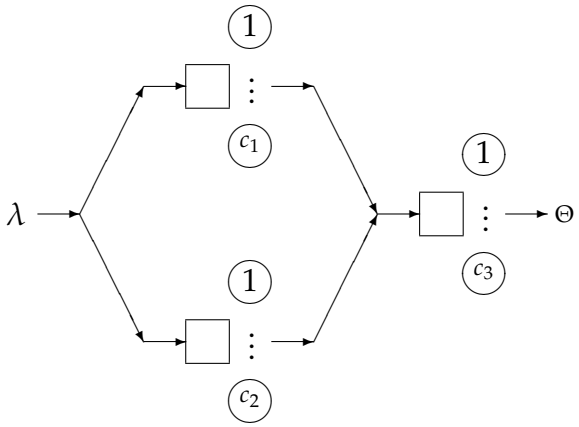

Figura 7 – Rede complexa de filas com topologia fusão.

Experimentos computacionais foram executados para as quatro topologias distintas. Para as topologias com entrada única (ver Figuras 1, 6, 8), o parâmetro λ foi sempre fixado em 5. Para a topologia fusão, a taxa de entrada λ é dividida entre duas filas (ver Figura 7), várias possibilidades de proporção foram investigadas e serão detalhadas na discussão de resultados.

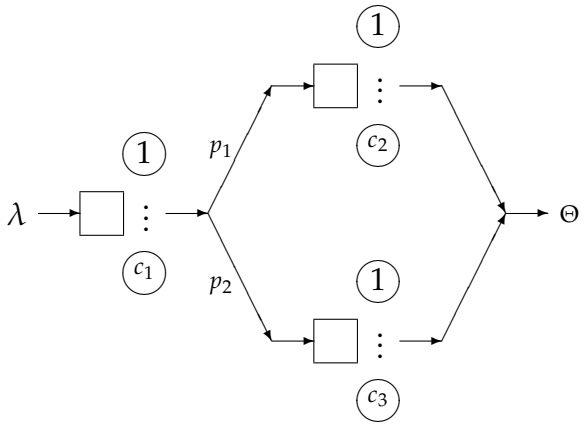

Figura 8 – Rede complexa de filas com topologia divisão.

As topologias com divisão e a topologia mista (ver Figuras 8 e 1) possuem probabilidades de roteamento ligadas ao seu funcionamento. Diversas configurações de roteamento foram investigadas e serão detalhadas nos resultados. Em todas as situações, as taxas de atendimento de cada servidor foram fixadas em $\mu = 10$, trata-se de uma condição de problema específico no qual os servidores podem ser intercambiados entre as diferentes filas da rede. Essa é uma condição específica deste problema sob investigação.

Para todas as execuções experimentais, as soluções iniciais foram geradas de forma aleatória, mas sob a garantia de que cada fila rede tenha no máximo 50 servidores. Uma vez estabelecida uma solução inicial, o número total de servidores C_{tot} dessa solução fica fixado, o algoritmo trabalha em analogia ao problema da mochila estocástico, ou seja, realoca servidores em diferentes filas, mas preserva C_{tot}.

O algoritmo teve número máximo de iterações fixado em $i_{max} = 1000$ com função de aceitação condicionada a $T_i = 1/\log(1+i)$, com $T_\varepsilon = 10^{-2}$. O número de soluções iniciais para cada fila foi fixado em 5000, todas geradas aleatoriamente. Devido a natureza do SA possibilitar evolução de uma solução dominada para uma não-dominada dentre as soluções finais, o algoritmo foi aplicado para todas as soluções iniciais, independente de serem ou não soluções não-dominadas. Posteriormente serão apresentados resultados específicos associados com cada topologia sob investigação.

4.1 Rede de Filas em Série

A rede em estudo é representada pela Figura 6. A imagem ilustra uma rede com filas em série, trata-se de uma sequência de filas em que o usuário só tem um caminho a seguir e, por consequência, precisa passa por todos os nós da rede.

Como mencionado anteriormente, o algoritmo foi aplicado para todas as 5000 soluções iniciais, geradas aleatoriamente. A Figura 9 apresenta as soluções iniciais e finais no espaço de soluções do problema para a rede com filas em série. São apresentadas todas as soluções e também uma representação somente com as soluções não dominadas.

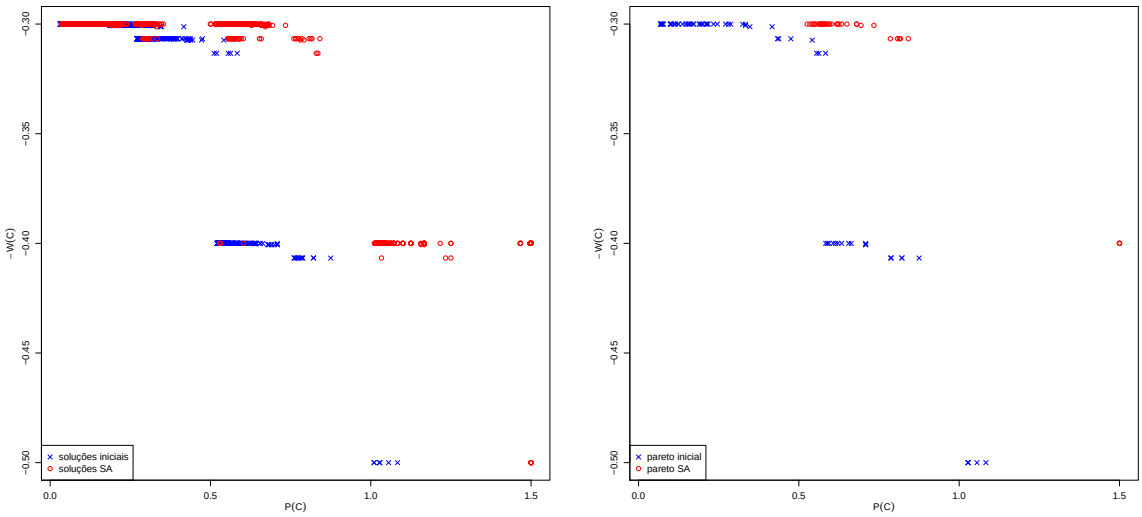

Figura 9 – Representação gráfica do espaço de soluções para filas em serie (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida).

As cruzes azuis representam as soluções iniciais aleatórias e os pontos em vermelho a solução final obtida pelo algoritmo SA. Na Figura 9, à esquerda, observa-se o conjunto completo das soluções iniciais geradas, à direita, apenas as soluções não dominadas. Como se trata de uma maximação nos dois objetivos, o espaço das melhores soluções se encontra no canto superior direito do gráfico.

Apesar da Figura 9 revelar uma cobertura pequena no espaço de objetivos, aparentemente existe um aumento representativo da produtividade, mantendo o tempo total em níveis mais reduzidos, com a melhoria obtida via algoritmo SA. Vale ressaltar que é uma rede pequena, com apenas três filas. O espaço de melhoria é reduzido e, mesmo assim, o método se mostra eficiente.

A Figura 10 ilustra como os servidores foram alocados em cada uma das filas. Em azul, a solução inicial, com uma alocação uniforme de servidores. Em vermelho, a alocação proposta pelo algoritmo SA. Na avaliação de redes com filas em série, o algoritmo retorna padrão bem similar de alocação de servidores para as três filas. É importante frisar que por se tratar de uma rede mais simples, com taxa de atendimento fixa entre os servidores, não existem muitas possibilidades de variação no padrão de alocação dos servidores entre as filas da rede. É bastante plausível que o algoritmo encontre soluções com padrão de alocação de servidores similar.

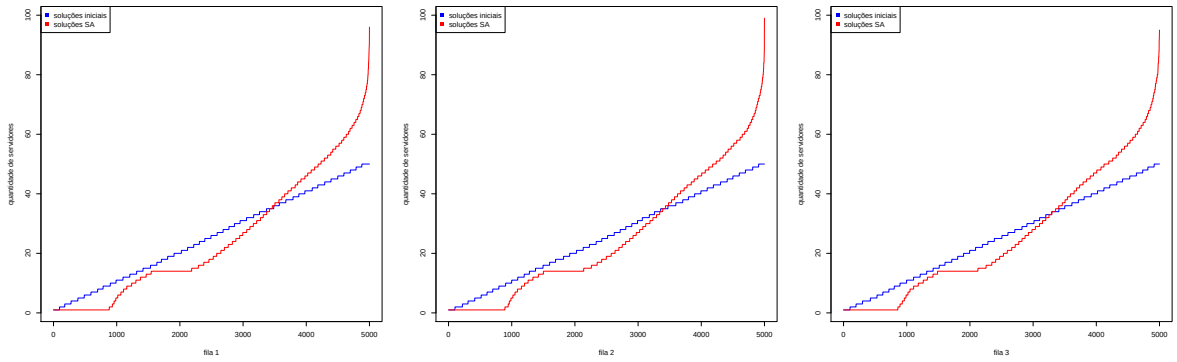

Figura 10 – Alocação de servidores nas filas da rede em série.

As Tabelas 2 e 3 apresentam resultados comparativos entre as soluções iniciais e as soluções fornecidas pelo algoritmo SA.

Tabela 2 – Melhoria obtida em produtividade média das soluções para redes de filas em série.

soluções iniciais	soluções SA	percentual de aumento
$\overline{P(\mathbf{C})}$ inicial (d.p.)	$\overline{P(\mathbf{C})}$ SA (d.p.)	
0,399037 (0,297687)	0,723576 (0,287536)	81,3305%

d.p. - desvio padrão

As soluções SA apresentaram, em média, um ganho substancial em aumento de produtividade sem uma alteração significativa na variabilidade entre as soluções.

Tabela 3 – Melhoria obtida em tempo médio de percurso das soluções para redes de filas em série.

soluções iniciais	soluções SA	percentual de redução
$\overline{W(\mathbf{C})}$ inicial (d.p.)	$\overline{W(\mathbf{C})}$ SA (d.p.)	
0,340079 (0,062025)	0,311746 (0,031233)	8,3313%

d.p. - desvio padrão

Já para a avaliação do tempo médio para percorrer a rede de filas, as soluções SA apresentaram ganho, porém não tão expressivo quanto o verificado paara a produtividade. Por fim, a Tabela 4 apresenta uma comparação entre o hipervolume do Pareto solução inicial e do Pareto solução SA. De fato as soluções fornecidas apresentam ganho. É importante salientar que, em particular, para as redes de filas em série com apenas três filas, as margens de melhoria não são muito largas.

Tabela 4 – Melhorias obtidas em hipervolume para redes de filas em série.

hipervolume inicial	hipervolume SA	percentual de redução
0,07349054	0,06661866	9,3507%

4.2 Rede de Filas com Fusão

A rede em estudo é representada na Figura 7. Trata-se uma rede de filas com fusão, existem duas entradas distintas na rede, através das filas 1 e 2 e todos os clientes atendidos, após passar por uma dessas filas, seguem para a fila 3. Dessa forma, o usuário da rede possui diferentes maneiras de percurso. No caso em estudo, ao entrar no sistema, o usuário opta por ser atendido pelos c_1 servidores que atendem a fila 1 ou pelos c_2 servidores que atendem a fila 2. Como observado para a rede de filas em série, a Figura 11 mostra as soluções iniciais e finais no espaço de soluções do problema para a rede com filas. Neste caso particular, $\lambda_1 = \lambda_2 = 5/2$.

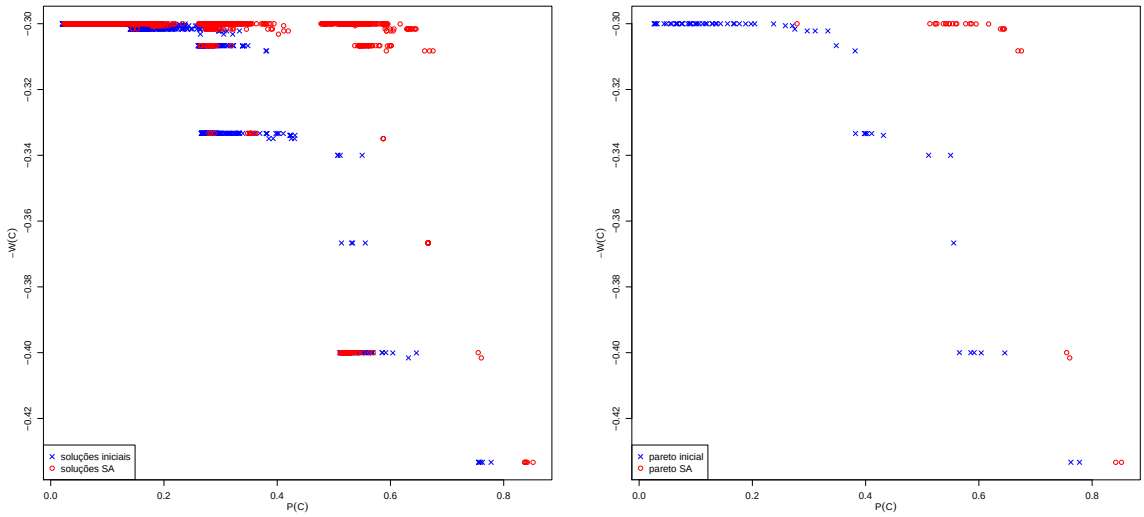

Figura 11 – Representação gráfica do espaço de soluções para filas com fusão e λ para fila 1 igual ao λ para fila 2 (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

Com interesse em verificar padrões nas soluções obtidas, diferentes configurações de divisões entre as proporções de entradas nas filas 1 e 2 foram verificadas. A estratégia foi produzir um vetor λ de duas coordenadas, tal que a coordenada λ_1 representa a taxa de entrada na fila 1 e a coordenada λ_2 representa a taxa de entrada na fila 2. Sete configurações distintas foram avaliadas, em todas elas, a soma $\lambda_1 + \lambda_2 = 5$ foi preservada e apenas a proporção de distribuição dessa taxa entre λ_1 e λ_2 foi alterada.

A Figura 11 revelou uma cobertura maior quanto ao espaço de objetivos, isso

em comparação com a rede de filas em serie. Possivelmente isso decorre da maior complexidade da rede e por consequência, mais opções de configurações. Aparentemente existe um aumento representativo da produtividade e manutenção do tempo total em níveis mais reduzidos, com a melhoria obtida via algoritmo SA.

A Figura 12 ilustra como os servidores foram alocados em cada uma das filas. Da mesma forma da rede com filas em serie e, para as seções em sequência, em azul, a solução inicial, com uma alocação uniforme de servidores. Em vermelho, a alocação proposta pelo algoritmo SA.

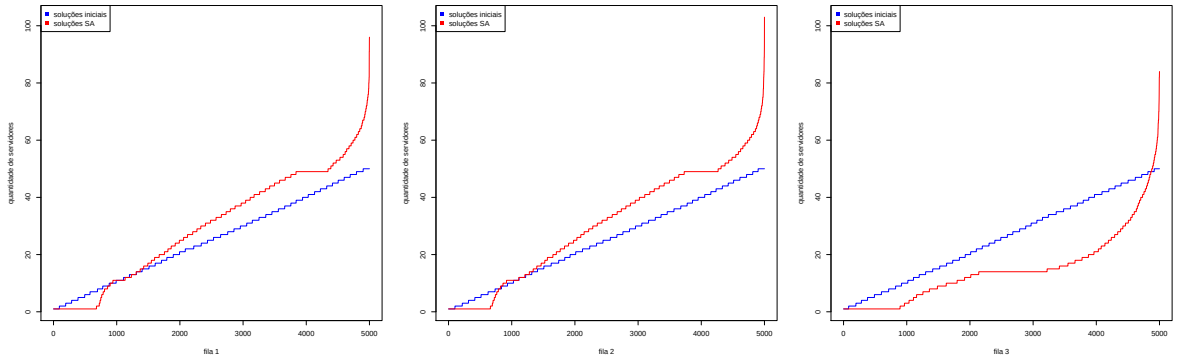

Figura 12 – Alocação de servidores nas filas da rede com fusão e λ para fila 1 igual ao λ para fila 2.

Para as filas 1 e 2, o algoritmo SA retorna uma alocação dos servidores bastante semelhante, o que faz muito sentido, dado que $\lambda_1 = \lambda_2 = 5/2$. A fila 3 tem uma alocação via algoritmo SA com consumo de recursos notoriamente inferior.

Em sequência, da Figura 13 até a Figura 18 são apresentadas as soluções no espaço de objetivos para os diferentes vetores λ investigados. Na Figura 13, o vetor utilizado para definir as taxas de entrada foi $\lambda = (5/3, 10/3)$, na Figura 14, o vetor utilizado para definir as taxas de entrada foi $\lambda = (1, 4)$, na Figura 15, o vetor utilizado para definir as taxas de entrada foi $\lambda = (5/9, 40/9)$, na Figura 16, o vetor utilizado para definir as taxas de entrada foi $\lambda = (10/3, 5/3)$, na Figura 17, o vetor utilizado para definir as taxas de entrada foi $\lambda = (4, 1)$. Por fim, na Figura 18, o vetor utilizado para definir as taxas de entrada foi $\lambda = (40/9, 5/9)$.

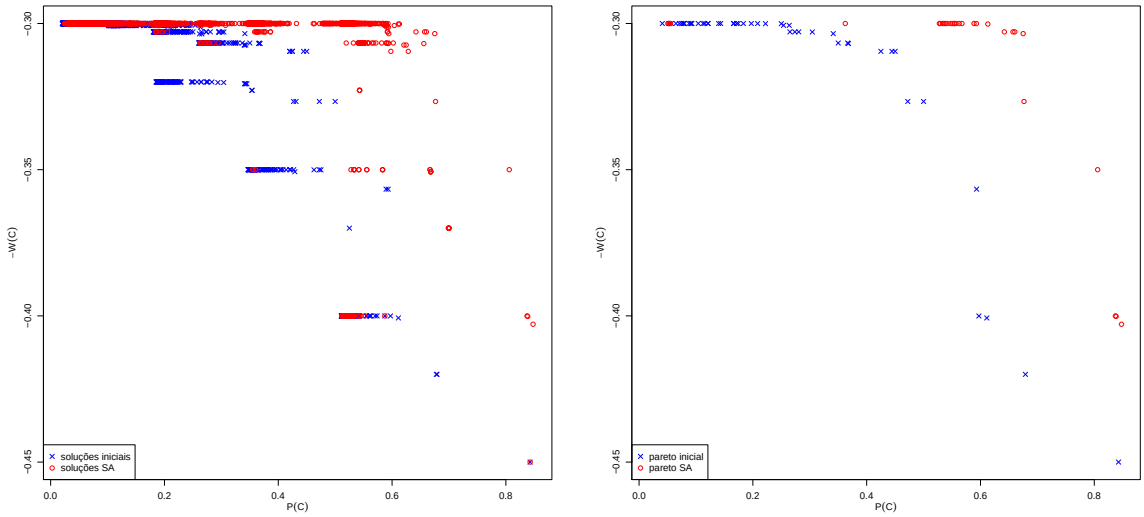

Figura 13 – Representação gráfica do espaço de soluções para filas com fusão e λ para fila 2 o dobro do λ para fila 1 (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

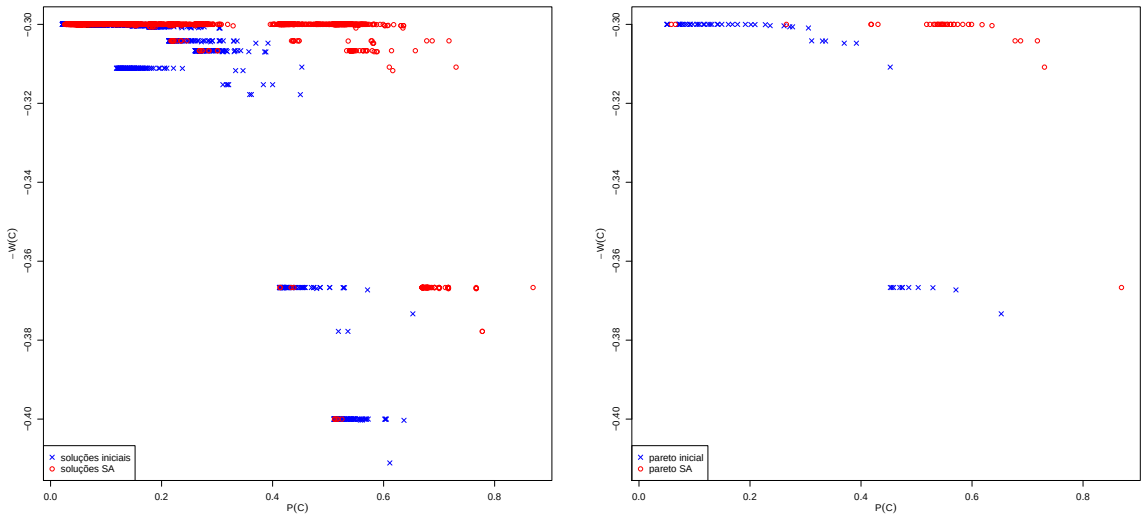

Figura 14 – Representação gráfica do espaço de soluções para filas com fusão e λ para fila 2 quatro vezes maior que o λ para fila 1 (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

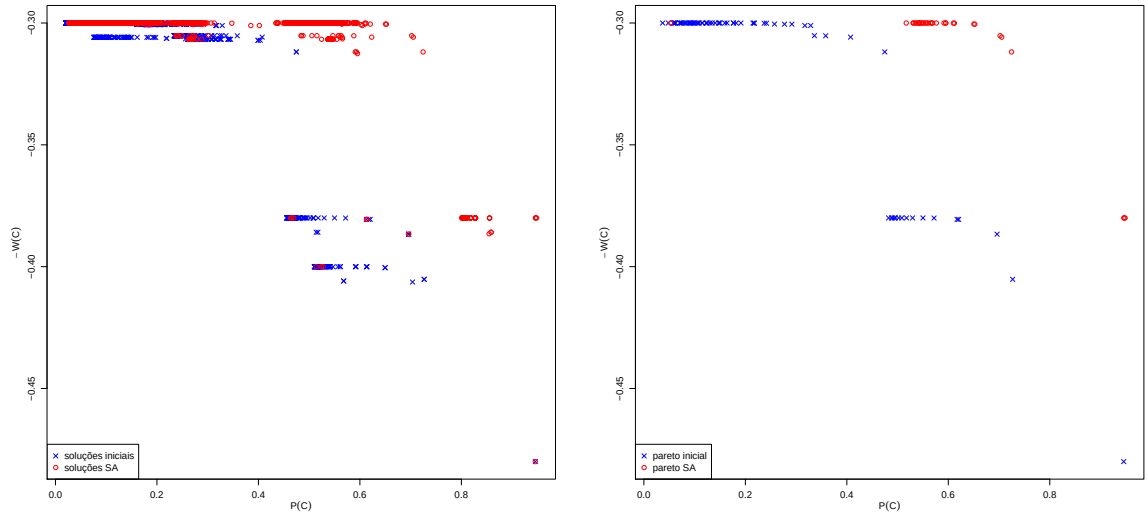

Figura 15 – Representação gráfica do espaço de soluções para filas com fusão e λ para fila 2 oito vezes maior que o λ para fila 1 (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

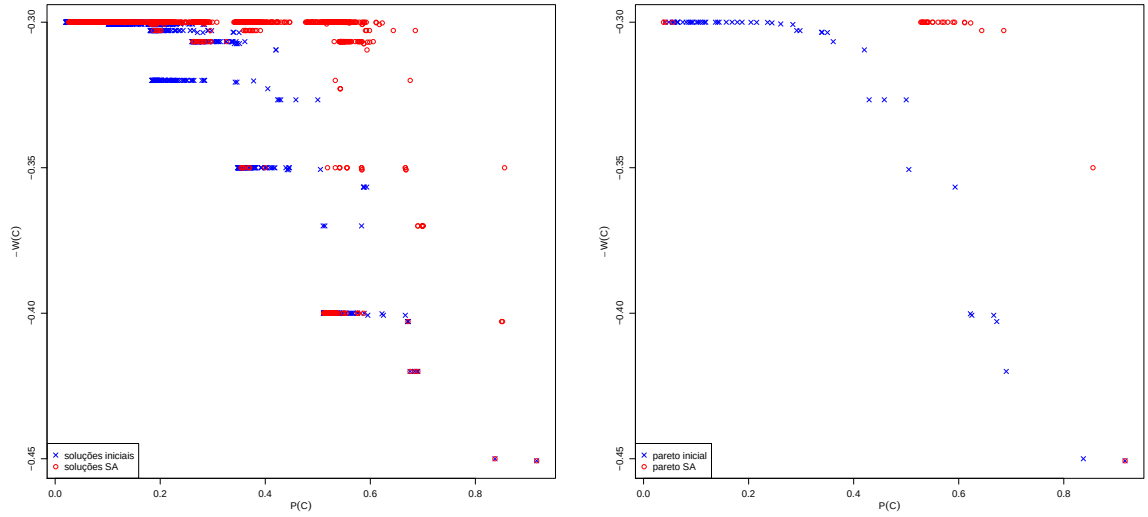

Figura 16 – Representação gráfica do espaço de soluções para filas com fusão e λ para fila 1 o dobro do λ para fila 2 (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

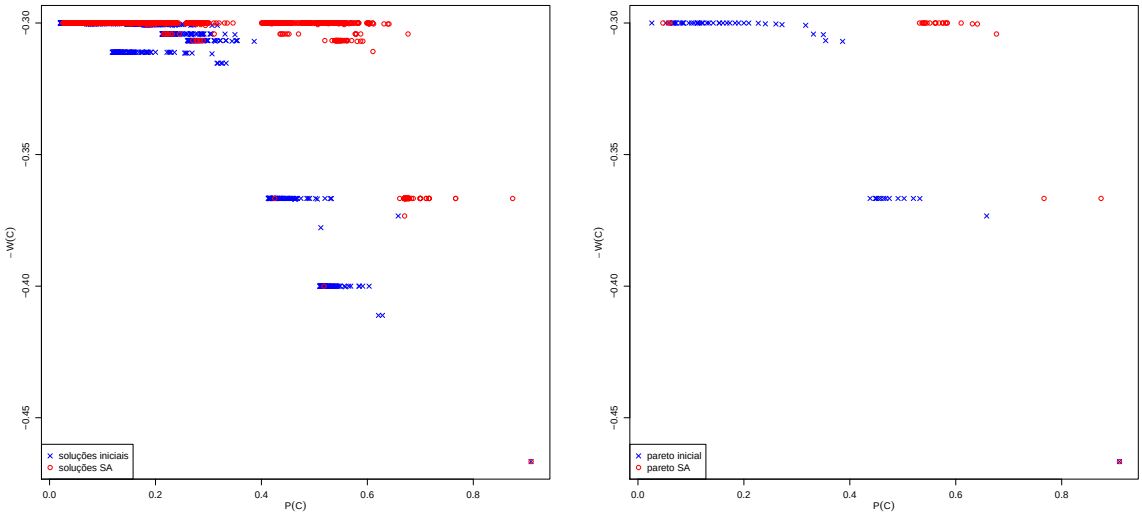

Figura 17 – Representação gráfica do espaço de soluções para filas com fusão e λ para fila 1 quatro vezes maior que o λ para fila 2 (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

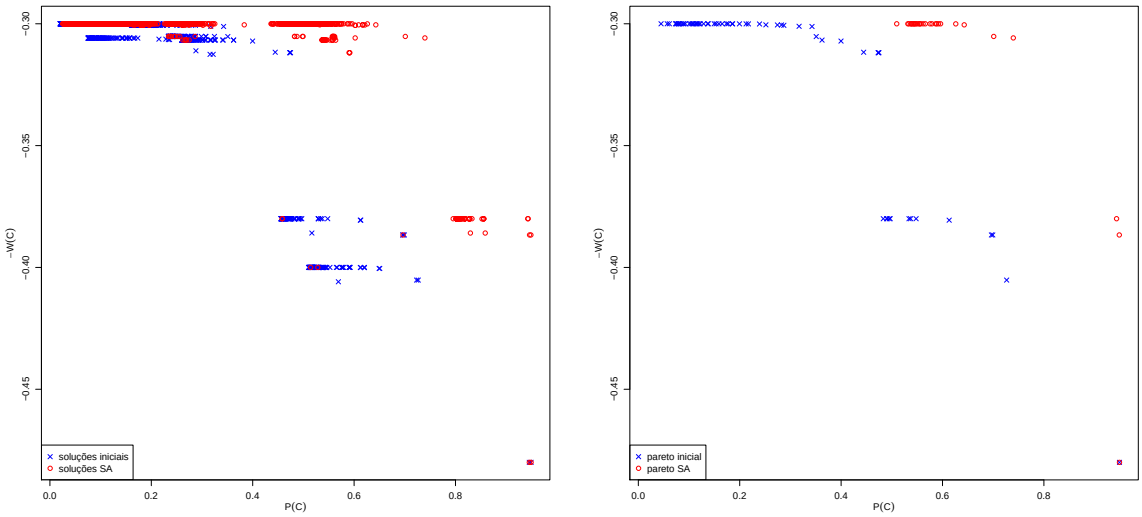

Figura 18 – Representação gráfica do espaço de soluções para filas com fusão e λ para fila 1 oito vezes maior que o λ para fila 2 (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

De forma similar a rede com fusão com $\lambda_1 = \lambda_2$, para todos as diferentes configurações do vetor λ, os experimentos apresentaram um aumento significativo da produtividade dos servidores da rede de filas e a manutenção dos tempos totais de percurso em níveis aceitáveis.

Da mesma forma de apresentação anterior, sequencialmente da Figura 19 até a Figura 24, são apresentados os padrões de alocação para as diferentes distribuições da

taxa de entrada. Novamente, na Figura 19, o vetor utilizado para definir as taxas de entrada foi $\lambda = (5/3, 10/3)$ e nas Figuras 20 e 21, respectivamente os vetores $\lambda = (1, 4)$ e $\lambda = (5/9, 40/9)$.

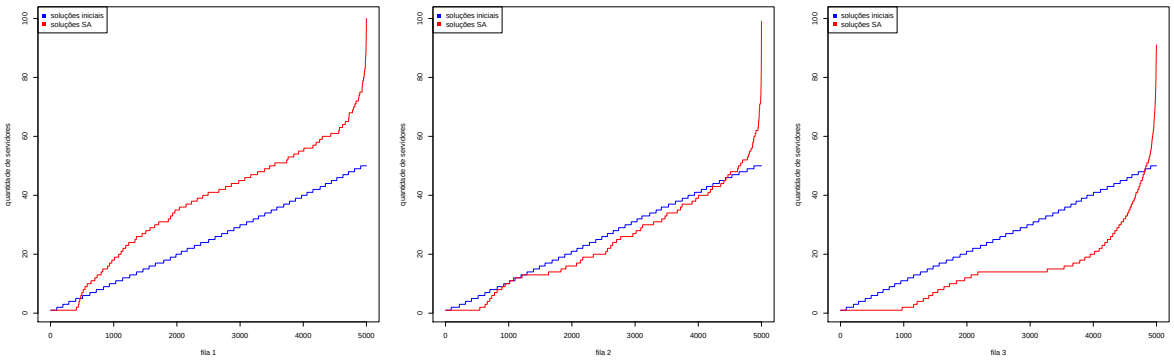

Figura 19 – Alocação de servidores nas filas da rede de filas com fusão e λ para fila 2 o dobro do λ para fila 1.

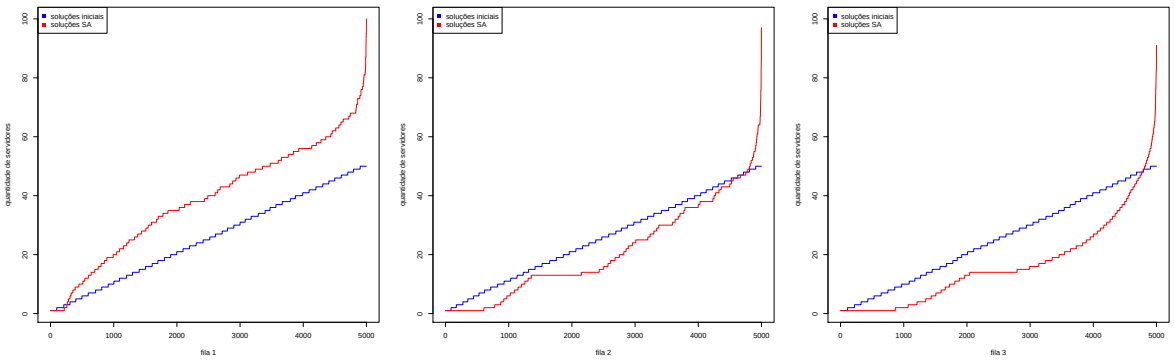

Figura 20 – Alocação de servidores nas filas da rede de filas com fusão e λ para fila 2 quatro vezes maior que o λ para fila 1.

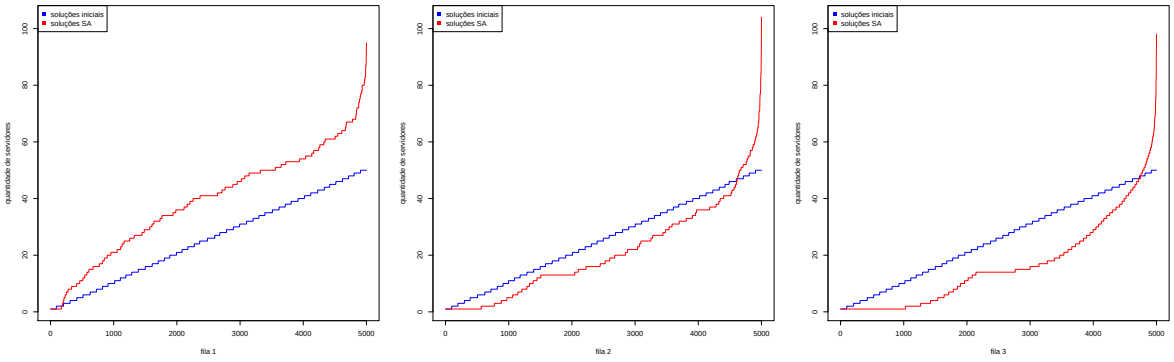

Figura 21 – Alocação de servidores nas filas da rede de filas com fusão e λ para fila 2 oito vezes maior que o λ para fila 1.

Já na Figura 22, o vetor utilizado foi $\lambda = (10/3, 5/3)$, na Figura 23, o vetor utilizado foi $\lambda = (4, 1)$ e finalmente na Figura 24, o vetor utilizado para definir as taxas de entrada foi $\lambda = (40/9, 5/9)$.

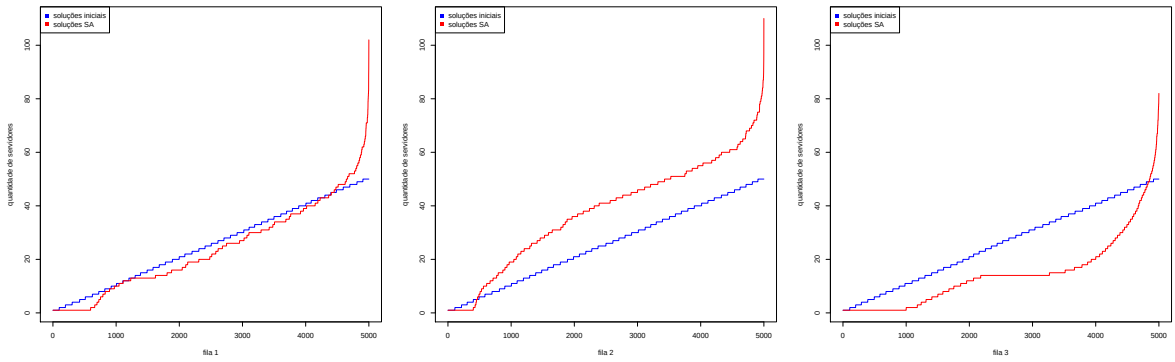

Figura 22 – Alocação de servidores nas filas da rede de filas com fusão e λ para fila 1 o dobro do λ para fila 2.

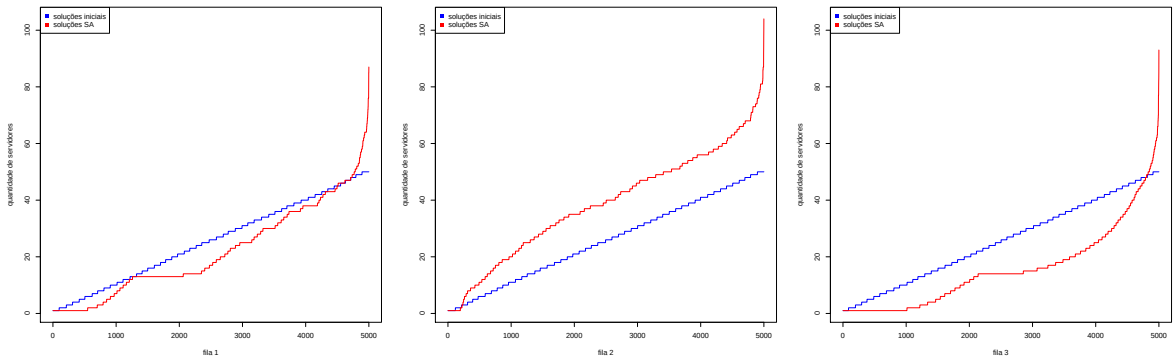

Figura 23 – Alocação de servidores nas filas da rede de filas com fusão e λ para fila 1 quatro vezes maior que o λ para fila 2.

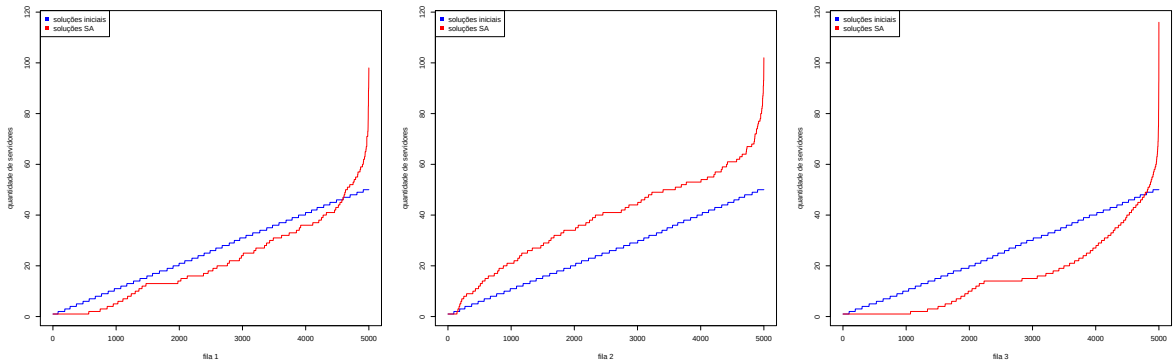

Figura 24 – Alocação de servidores nas filas da rede de filas com fusão e λ para fila 1 oito vezes maior que o λ para fila 2.

As variações investigadas nas taxas de entrada deixam evidenciado o algoritmo SA propõe alocações de servidores personalizadas para cada configuração da rede de filas. É fácil ver que quanto maior a taxa de chegada de uma fila em relação a outra, mais lentamente o método parece alocar os servidores. Quando a taxa de entrada λ de uma fila é o dobro da outra, como na Figura 19, a alocação encontrada cresce de maneira similar a alocação inicial uniforme. A terceira fila da rede de filas permaneceu com alocações semelhantes em todo o processo. A fila 3, por se tratar da mesma em todos os experimentos, não sofre alteração marcantes, mas existe uma pequena variação. É possível observar que o algoritmo SA consegue consumir menos recursos na fila 3 para as redes cujas taxas de entrada λ_1 e λ_2 são menos discrepantes.

As Tabelas 5 e 6 apresentam resultados comparativos entre soluções iniciais e soluções fornecidas pelo algoritmo SA, para todas as variações de λ investigadas.

Tabela 5 – Melhoria obtida em produtividade média das soluções para redes de filas com fusão.

λ	soluções iniciais	soluções SA	percentual de aumento
	$\overline{P(\mathbf{C})}$ inicial (d.p.)	$\overline{P(\mathbf{C})}$ SA (d.p.)	
$(5/2, 5/2)$	0,234851 (0,200626)	0,601302 (0,112198)	156,0360%
$(5/3, 10/3)$	0,249766 (0,189987)	0,536943 (0,220636)	114,9781%
$(1, 4)$	0,229235 (0,165523)	0,536724 (0,150877)	134,1375%
$(5/9, 40/9)$	0,248701 (0,204830)	0,598568 (0,153858)	140,6778%
$(10/3, 5/3)$	0,271259 (0,226962)	0,535472 (0,195746)	97,4021%
$(4, 1)$	0,244131 (0,187619)	0,546474 (0,201507)	123,8451%
$(40/9, 5/9)$	0,271430 (0,207993)	0,610941 (0,124045)	125,0824%

d.p. - desvio padrão

As soluções SA apresentaram, em média, um ganho substancial em aumento de produtividade. Com exceção de $\lambda = (10/3, 5/3)$, a produtividade mais que dobrou.

Na tabela 5 ainda é possível verificar que apenas para $\lambda = (5/2, 5/2)$ e ainda para $\lambda = (40/9, 5/9)$, a variabilidade entre as soluções teve uma alteração significativa, mas com redução da variabilidade e não um aumento. Já para a avaliação do tempo médio para percorrer a rede de filas com fusão (ver Tabela 6), as soluções SA apresentaram ganho, com exceção em $\lambda = (5/2, 5/2)$, que apresentou aumento no tempo. Porém, esse aumento não foi tão expressivo quanto o ganho verificado para a produtividade.

Por fim, a Tabela 7 apresenta uma comparação entre o hipervolume do Pareto solução inicial e do Pareto solução SA. De fato, as soluções fornecidas apresentam ganho. Por se tratar de uma rede mais complexa que a rede vista anteriormente, pode-se observar uma melhoria mais significativa quando comparada à melhoria verificada nas redes de filas em série.

Tabela 6 – Melhoria obtida em tempo médio de percurso das soluções para redes de filas com fusão.

λ	soluções iniciais	soluções SA	percentual de redução
	$\overline{W(\mathbf{C})}$ inicial (d.p.)	$\overline{W(\mathbf{C})}$ SA (d.p.)	
$(5/2, 5/2)$	0,317522 (0,035384)	0,317546 (0,0419003)	−0,0073%
$(5/3, 10/3)$	0,313181 (0,033423)	0,313065 (0,0314828)	0,0371%
$(1, 4)$	0,313626 (0,026574)	0,302443 (0,011047)	3,5656%
$(5/9, 40/9)$	0,3200701 (0,038568)	0,307997 (0,023246)	3,7721%
$(10/3, 5/3)$	0,3201294 (0,041429)	0,30739 (0,029626)	3,9795%
$(4, 1)$	0,319176 (0,034586)	0,311291 (0,035757)	2,4702%
$(40/9, 5/9)$	0,319840 (0,038536)	0,311938 (0,038100)	2,4705%

d.p. - desvio padrão

Tabela 7 – Melhorias obtidas em hipervolume para redes de filas com fusão.

λ	hipervolume inicial	hipervolume SA	percentual de redução
$(5/2, 5/2)$	0,03395144	0,01965590	42,1058%
$(5/3, 10/3)$	0,04231959	0,00994771	76,4939%
$(1, 4)$	0,01500969	0,00048620	96,7608%
$(5/9, 40/9)$	0,06217051	0,01827819	70,5999%
$(10/3, 5/3)$	0,05239663	0,00867745	83,4389%
$(4, 1)$	0,06122329	0,01332812	78,2303%
$(40/9, 5/9)$	0,06286776	0,01729592	72,4884%

4.3 Rede de Filas com Divisão

A rede em estudo é representada na Figura 8. Trata-se uma rede de filas com divisão, em que clientes podem optar por escolher uma fila i ou uma fila j em determinado ponto da rede de filas. Nesta topologia específica, os usuários entram na rede através da fila 1 e todos são atendidos por c_1 servidores. Após o atendimento, os clientes optam, com probabilidade p_1, em ir para a fila 2, ou, para a fila 3, com probabilidade $p_2 = 1 - p_1$. Como observado para a rede de filas em série e de fusão, a Figura 25 mostra as soluções iniciais e finais no espaço de soluções do problema para a rede de filas com divisão determinada pelo vetor de roteamento (p_1, p_2), neste caso particular, $p_1 = p_2 = 1/2$.

De forma similar ao método aplicado a rede de filas com fusão, a Figura 25 revelou uma cobertura maior quanto ao espaço de objetivos, em comparação com a rede de filas em série. Possívelmente isso decorre da maior complexidade da rede e por consequência, mais opções de configurações. O método aparentemente também foi eficaz para esse tipo de topologia. Há evidências de um aumento representativo da produtividade e manutenção do tempo total em níveis mais reduzidos. A Figura 26

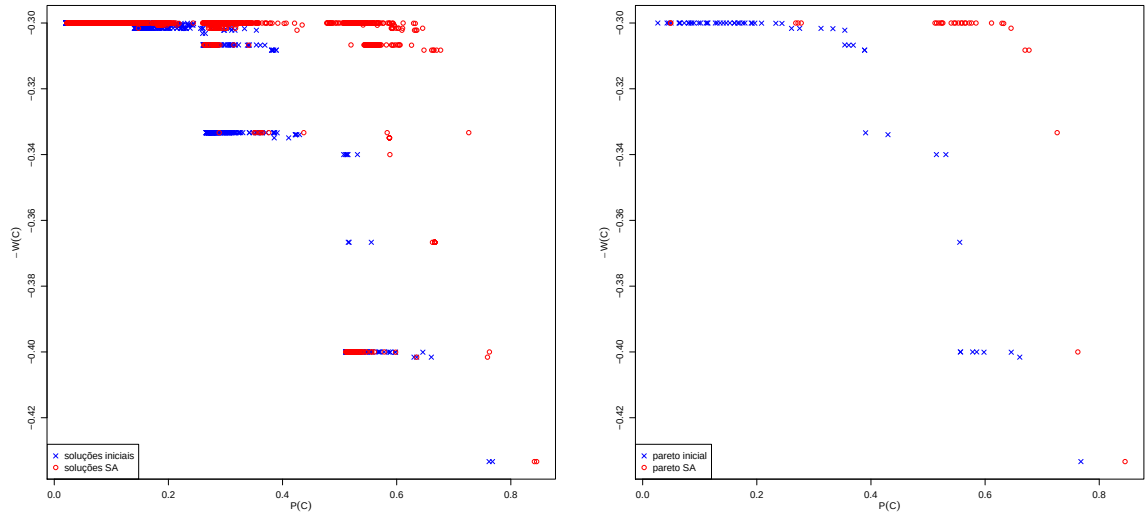

Figura 25 – Representação gráfica do espaço de soluções para rede de filas com divisão e $p_1 = p_2$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

ilustra como os servidores foram alocados em cada uma das filas para esse caso em específico, com $p_1 = p_2 = 1/2$.

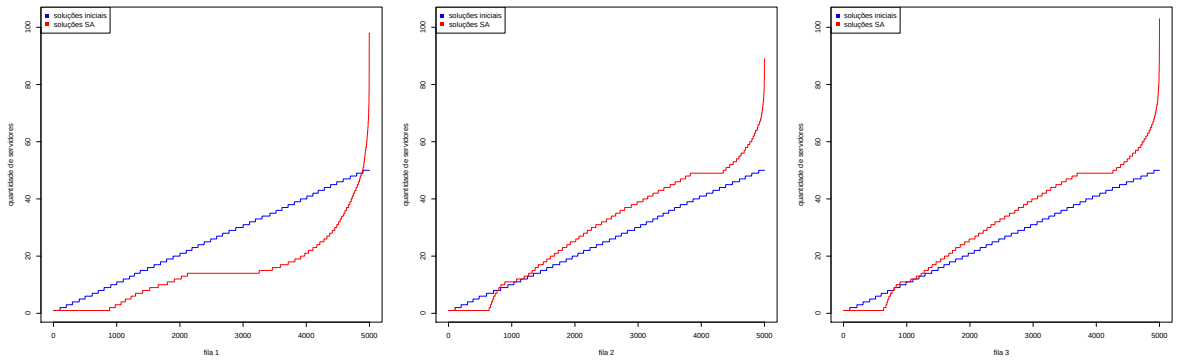

Figura 26 – Alocação de servidores nas filas da rede com divisão e $p_1 = p_2$ no roteamento.

Para as filas 2 e 3, o algoritmo SA retorna uma alocação dos servidores bastante semelhante, o que faz muito sentido, dado que $p_1 = p_2 = 1/2$. A fila 1, além de receber uma alocação parecida com as filas 3 da rede de fusão, tem uma alocação via algoritmo SA com consumo de recursos inferior.

Em sequência, da Figura 27 até a Figura 32 são apresentadas as soluções no espaço de objetivos para os diferentes vetores de roteamentos p investigados. Na Figura 27, o roteamento utilizado para definir a probabilidade de escolha de fila foi $p_1 = 1/3$ e $p_2 = 2/3$, na Figura 28, o roteamento utilizado para definir a probabilidade de escolha de fila foi $p_1 = 1/5$ e $p_2 = 4/5$, na Figura 29, o roteamento utilizado para definir a

probabilidade de escolha de fila foi $p_1 = 1/9$ e $p_2 = 8/9$, na Figura 30, o roteamento utilizado para definir a probabilidade de escolha de fila foi $p_1 = 2/3$ e $p_2 = 1/3$, na Figura 31, o roteamento utilizado para definir a probabilidade de escolha de fila foi $p_1 = 4/5$ e $p_2 = 1/5$. Por fim, na Figura 32, o roteamento utilizado para definir a probabilidade de escolha de fila foi $p_1 = 8/9$ e $p_2 = 1/9$.

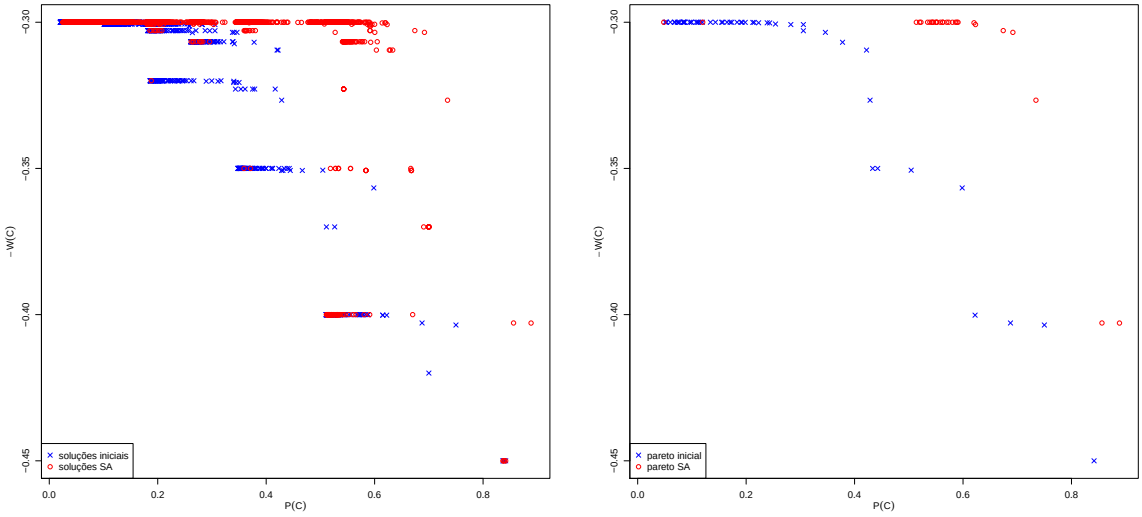

Figura 27 – Representação gráfica do espaço de soluções para rede de filas com divisão e $p_2 = 2 \times p_1$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

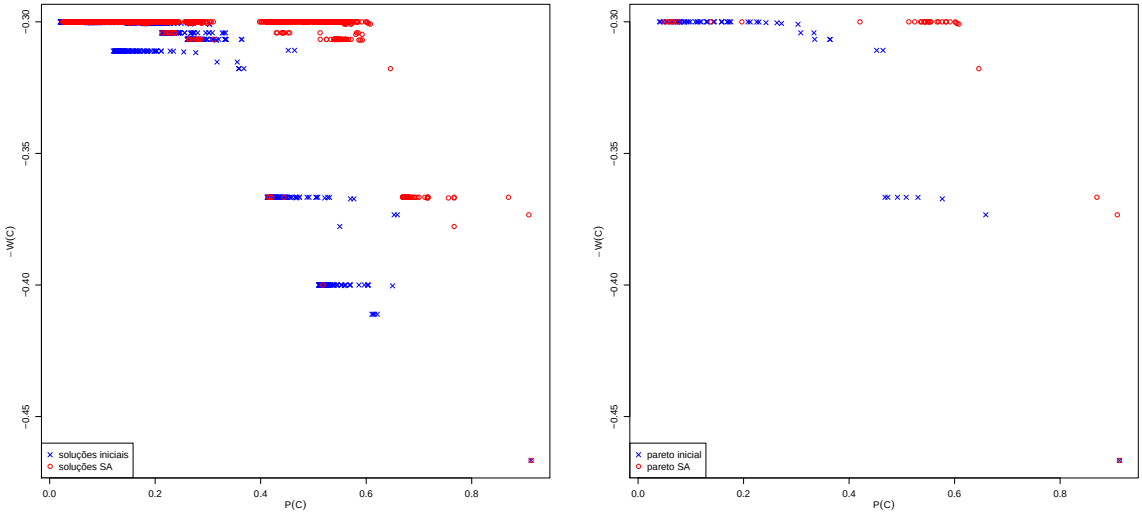

Figura 28 – Representação gráfica do espaço de soluções para rede de filas com divisão e $p_2 = 4 \times p_1$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

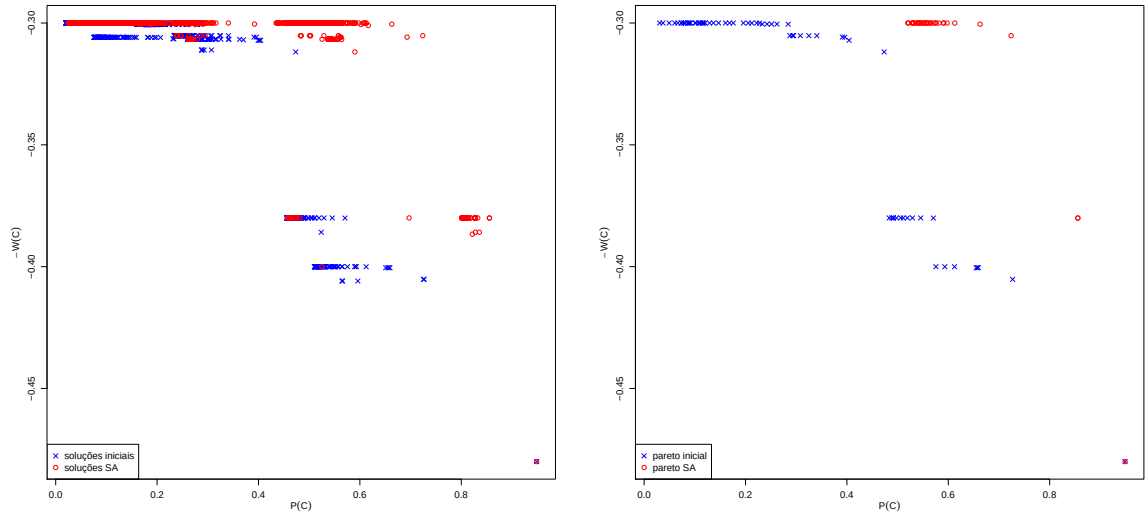

Figura 29 – Representação gráfica do espaço de soluções para rede de filas com divisão e $p_2 = 8 \times p_1$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

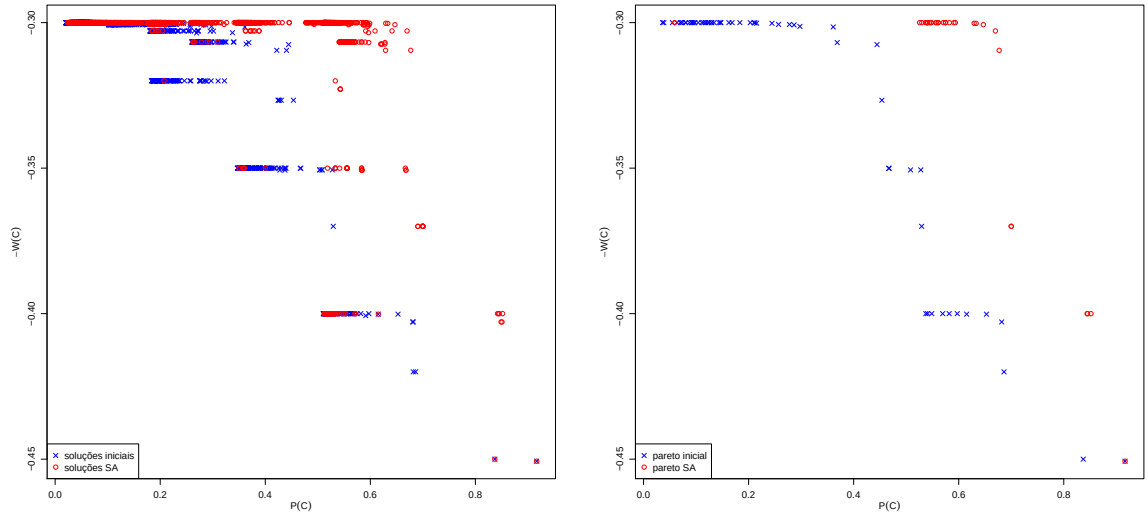

Figura 30 – Representação gráfica do espaço de soluções para rede de filas com divisão e $p_1 = 2 \times p_2$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

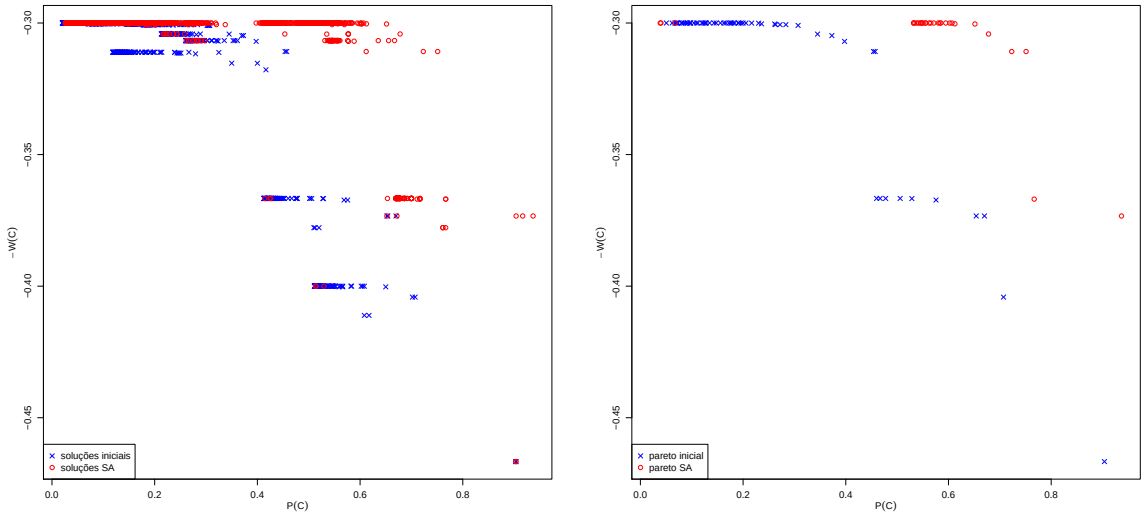

Figura 31 – Representação gráfica do espaço de soluções para rede de filas com divisão e $p_1 = 4 \times p_2$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

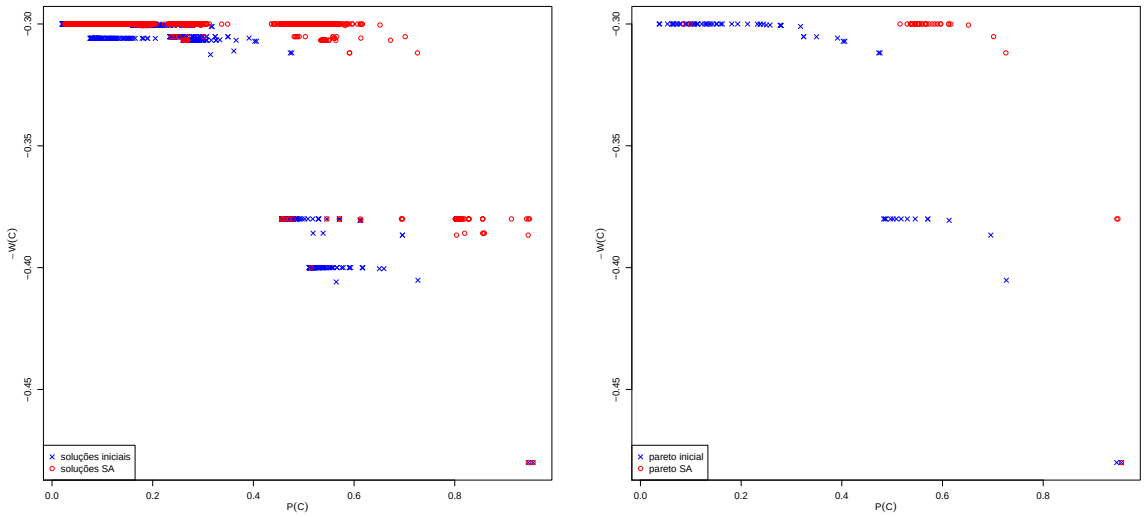

Figura 32 – Representação gráfica do espaço de soluções para rede de filas com divisão e $p_1 = 8 \times p_2$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

Semelhante ao verificado nas análises anteriores, a rede com divisão para os variados vetores de roteamento, as soluções fornecidas pelo algoritmo SA apresentaram um aumento significativo da produtividade dos servidores da rede de filas e a manutenção dos tempos totais de percurso em níveis aceitáveis.

Novamente com o formato de apresentação anterior, sequencialmente da Figura 33 até a Figura 38, são apresentados os padrões de alocação para os diferentes vetores de

roteamento. Novamente, na Figura 33, o vetor utilizado para definir as probabilidades de escolha foi $p = (1/3, 2/3)$ e nas Figuras 34 e 35, respectivamente os vetores $p = (1/5, 4/5)$ e $p = (1/9, 8/9)$.

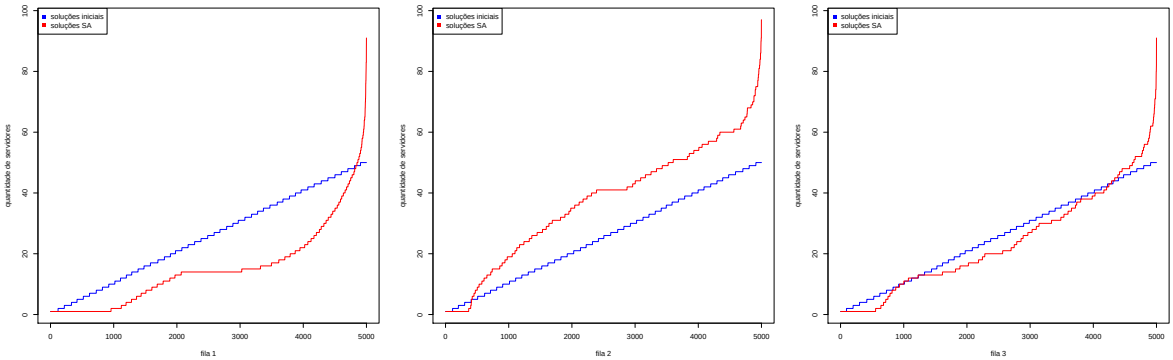

Figura 33 – Alocação de servidores nas filas da rede com divisão e $p_2 = 2 \times p_1$ no roteamento.

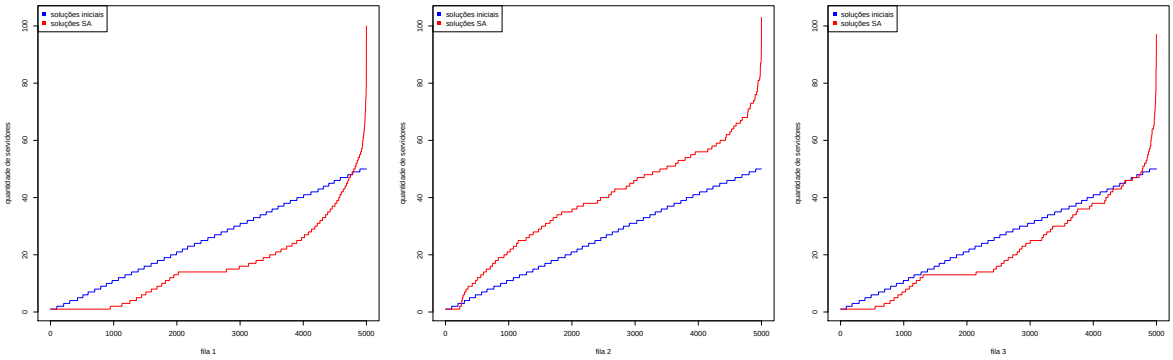

Figura 34 – Alocação de servidores nas filas da rede com divisão e $p_2 = 4 \times p_1$ no roteamento.

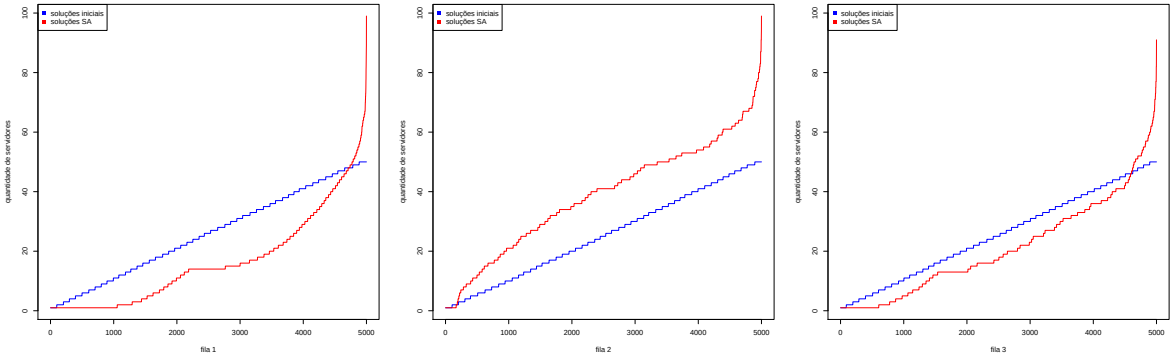

Figura 35 – Alocação de servidores nas filas da rede com divisão e $p_2 = 8 \times p_1$ no roteamento.

Já na Figura 36, o vetor utilizado foi $p = (2/3, 1/3)$, na Figura 37, o vetor utilizado foi $p = (4/5, 1/5)$ e finalmente na Figura 38, o vetor utilizado para definir as probabilidades de escolha de servidor foi $p = (8/9, 1/9)$.

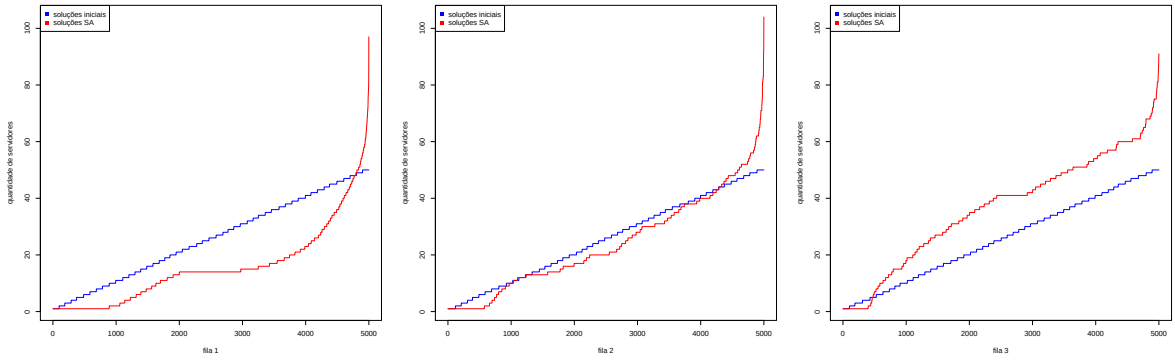

Figura 36 – Alocação de servidores nas filas da rede com divisão e $p_1 = 2 \times p_2$ no roteamento.

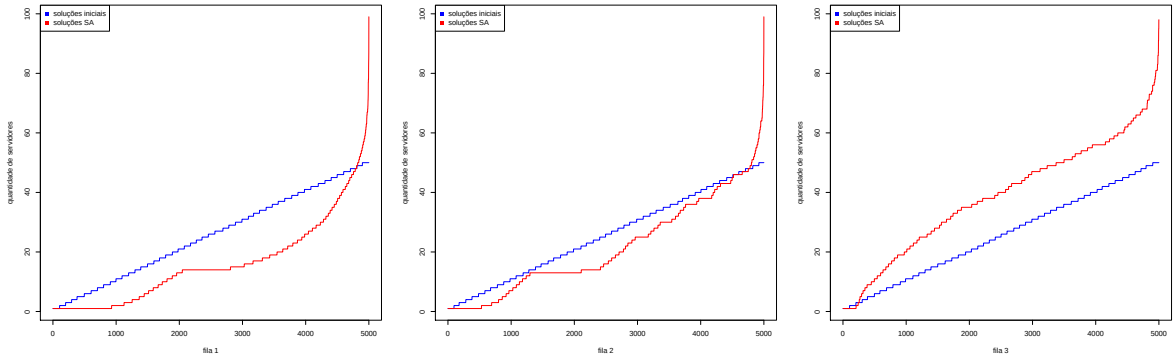

Figura 37 – Alocação de servidores nas filas da rede com divisão e $p_1 = 4 \times p_2$ no roteamento.

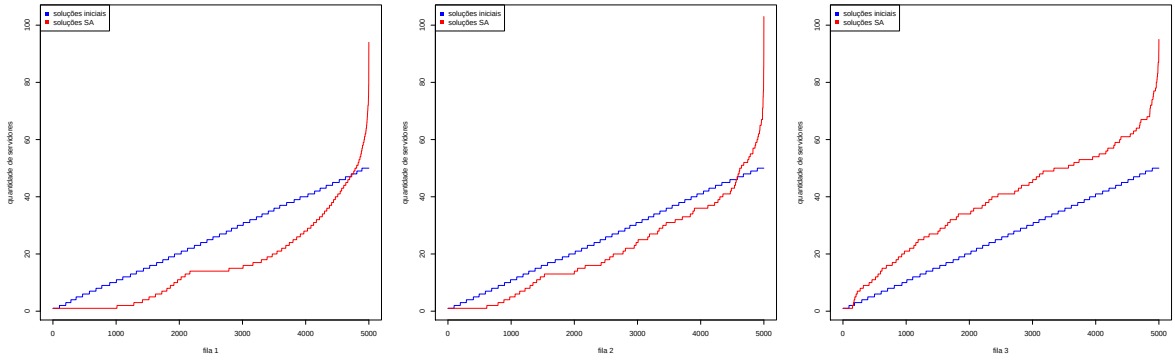

Figura 38 – Alocação de servidores nas filas da rede com divisão e $p_1 = 8 \times p_2$ no roteamento.

As formas variadas utilizadas para o vetor de roteamento novamente ilustram que o algoritmo SA é capaz de fornecer alocações de servidores específicas para cada configuração da rede de filas de divisão. Nota-se que quanto menor é a probabilidade de escolha de uma fila em relação a outra, mais lentamente o método tende a alocar os servidores. Quando a probabilidade p de escolha de uma fila é o dobro da outra, como na Figura 33, a alocação encontrada cresce de maneira similar a alocação inicial uniforme. A primeira fila da rede de filas permaneceu com alocações semelhantes em todo o processo, o que era bem previsível.

As Tabelas 8 e 9 apresentam resultados comparativos entre as soluções iniciais e as soluções fornecidas pelo algoritmo SA, para todas as especificações testadas.

Tabela 8 – Melhoria obtida em produtividade média das soluções para redes de filas com divisão.

roteamento	soluções iniciais	soluções SA	percentual de aumento
	$\overline{P(\mathbf{C})}$ inicial (d.p.)	$\overline{P(\mathbf{C})}$ SA (d.p.)	
$(1/2, 1/2)$	0,258845 (0,194120)	0,529157 (0,177738)	104,4302%
$(1/3, 2/3)$	0,228123 (0,191079)	0,563595 (0,161453)	147,0572%
$(1/5, 4/5)$	0,218196 (0,180592)	0,483330 (0,237071)	121,5122%
$(1/9, 8/9)$	0,288451 (0,213272)	0,591827 (0,100656)	105,1745%
$(2/3, 1/3)$	0,284986 (0,226991)	0,602746 (0,145644)	111,4999%
$(4/5, 1/5)$	0,238876 (0,188544)	0,550009 (0,191657)	130,2489%
$(8/9, 1/9)$	0,269222 (0,215780)	0,580107 (0,165220)	115,4748%

d.p. - desvio padrão

As soluções SA apresentaram, em média, um ganho representativo em aumento de produtividade. Para todos os diferentes roteamentos testados, a produtividade mais que dobrou. Apenas para $p = (1/9, 8/9)$ e $p = (8/9, 1/9)$, a variabilidade entre as soluções teve uma alteração significativa, mas novamente com queda na variabilidade.

Tabela 9 – Melhoria obtida em tempo médio de percurso das soluções para redes de filas com divisão.

roteamento	soluções iniciais	soluções SA	percentual de redução
	$\overline{W(\mathbf{C})}$ inicial (d.p.)	$\overline{W(\mathbf{C})}$ SA (d.p.)	
$(1/2, 1/2)$	0,318559 (0,036674)	0,309193 (0,029669)	2,9401%
$(1/3, 2/3)$	0,313025 (0,032247)	0,308266 (0,026669)	1,5203%
$(1/5, 4/5)$	0,311553 (0,029775)	0,309063 (0,031605)	0,7993%
$(1/9, 8/9)$	0,326532 (0,043044)	0,309880 (0,035073)	5,0997%
$(2/3, 1/3)$	0,326913 (0,045080)	0,318885 (0,040639)	2,4559%
$(4/5, 1/5)$	0,313643 (0,032314)	0,305561 (0,017792)	2,5769%
$(8/9, 1/9)$	0,321588 (0,041960)	0,309933 (0,034565)	3,6242%

d.p. - desvio padrão

Já para a avaliação do tempo médio para percorrer a rede de filas com divisão, as soluções SA apresentaram ganho, porém, não tão expressivo quanto o verificado para a produtividade. Finalmente, a Tabela 10 apresenta uma comparação entre o hipervolume do Pareto solução inicial e do Pareto solução SA. De fato as soluções fornecidas apresentam ganho.

Tabela 10 – Melhorias obtidas em hipervolume para redes de filas com divisão.

roteamento	hipervolume inicial	hipervolume SA	percentual de redução
$(1/2, 1/2)$	0,03245202	0,00554718	82,9065%
$(1/3, 2/3)$	0,03953189	0,01386989	64,9147%
$(1/5, 4/5)$	0,05726457	0,01843619	67,8052%
$(1/9, 8/9)$	0,06522433	0,01086952	83,3351%
$(2/3, 4/3)$	0,05535512	0,01694901	69,3813%
$(4/5, 1/5)$	0,05263673	0,00197770	96,2427%
$(8/9, 1/9)$	0,06397849	0,01836799	71,2904%

4.4 Rede de Filas Mista

A rede de filas mista em estudo é a rede representada na Figura 1. Trata-se uma rede de filas com topologia complexa, existem divisões, fusões e filas em série, daí a nomenclatura rede mista. O usuário que percorre a rede possui diferentes maneiras de percurso. Quanto mais nós de fusão e divisão existem na rede, mais complexo e variado o caminho se torna.

No caso em estudo, ao entrar no sistema, os usuários obrigatoriamente passam pela fila 1 com taxa $\lambda = 5$ fixa. Após, os usuários optam com uma probabilidade p_1 para seguir na fila 2, ou, com uma probabilidade $p_2 = 1 - p_1$, na fila 3. Os clientes que seguiram para a fila 2, obrigatoriamente se direcionam para a fila 4. O restante, segue para a fila 4 a uma probabilidade p_3, ou, para a fila 5 a uma probabilidade $P_4 = 1 - p_3$. Em seguida, todos se juntam novamente por meio de uma fusão na fila 6, que finaliza a rede.

Como verificado anteriormente, para as redes de filas em série, fusão e divisão, a Figura 39 mostra as soluções iniciais e finais no espaço de soluções do problema para a rede de filas mista determinada pelo vetor de roteamento (p_1, p_2, p_3, p_4), neste caso particular, $p_1 = p_2 = p_3 = p_4 = 1/2$. È possível verificar uma cobertura bem maior do espaço de objetivos, em comparação com ad reded de filas anteriores. Possívelmente isso decorre da maior complexidade da rede e por consequência, mais opções de distribuição dos servidores. O método aparentemente também foi eficaz para esse tipo de topologia.

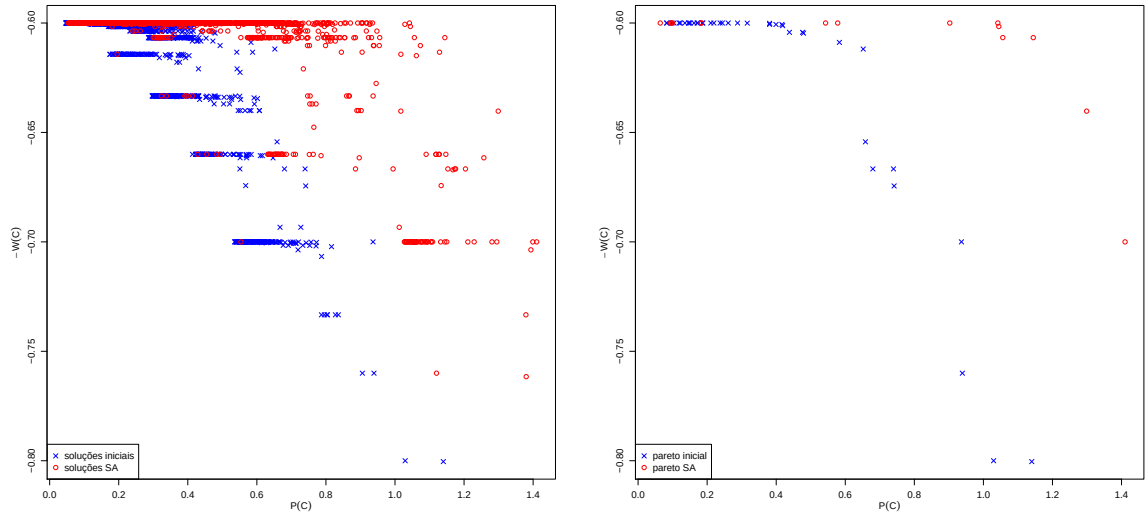

Figura 39 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = p_2 = p_3 = p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

Existem evidências de um aumento representativo da produtividade e manutenção do tempo total em níveis mais reduzidos. A Figura 40 ilustra como os servidores foram alocados em cada uma das filas para esse caso em específico, com $p_1 = p_2 = p_3 = p_4 = 1/2$.

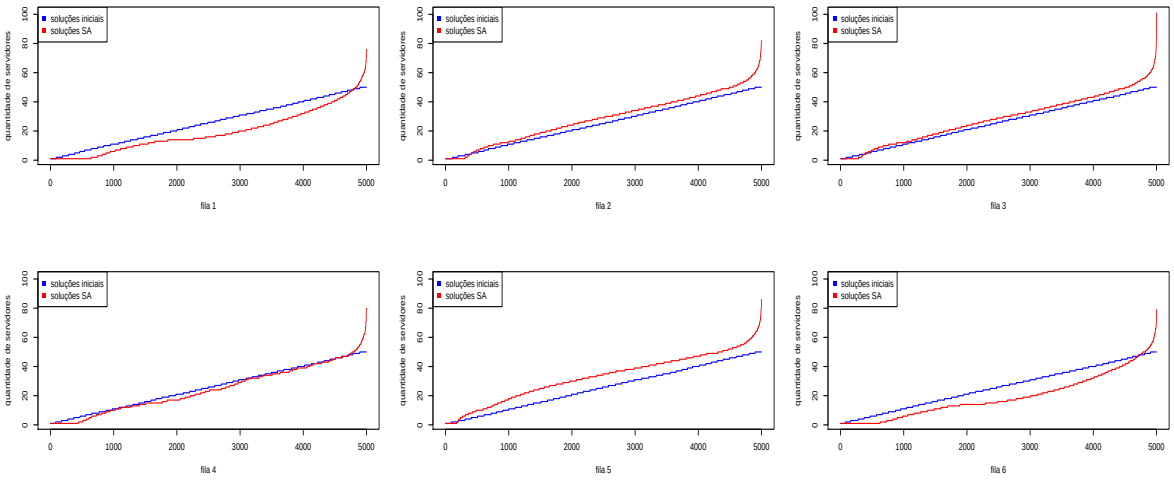

Figura 40 – Alocação de servidores nas filas da rede mista com $p_1 = p_2 = p_3 = p_4$ no roteamento.

Para as filas 1 e 6, o algoritmo SA retorna uma alocação dos servidores bastante semelhante, o que faz sentido, são as únicas duas filas que recebem todos os clientes obrigatoriamente. As filas 2 e 3 também se comportam de forma semelhante, se deve ao fato de que ambas possuem probabilidade de roteamento iguais, isto é, $p_1 = p_2 = 1/2$,

ou seja, chegam clientes para as duas de forma homogenea. Para fila 4 o algoritmo aloca os servidores de forma semelhante as soluções iniciais. Já para a fila 5, a alocação via SA retorna um consumo de recursos superior a alocação inicial.

Em sequência, da Figura 41 até a Figura 64 são apresentadas as soluções no espaço de objetivos para os diferentes vetores de roteamentos p investigados.

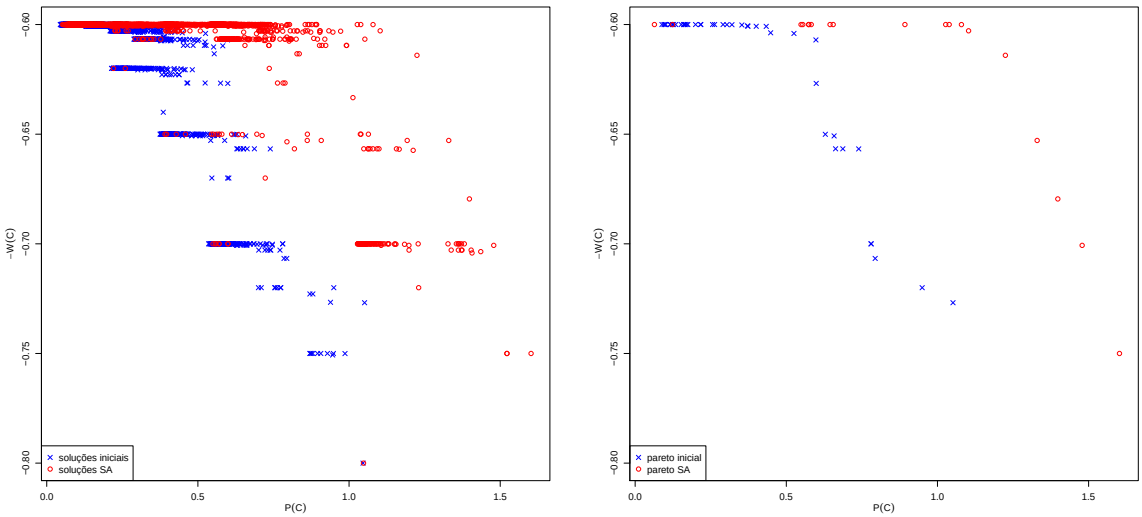

Figura 41 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 2 \times p_1$, $p_3 = p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

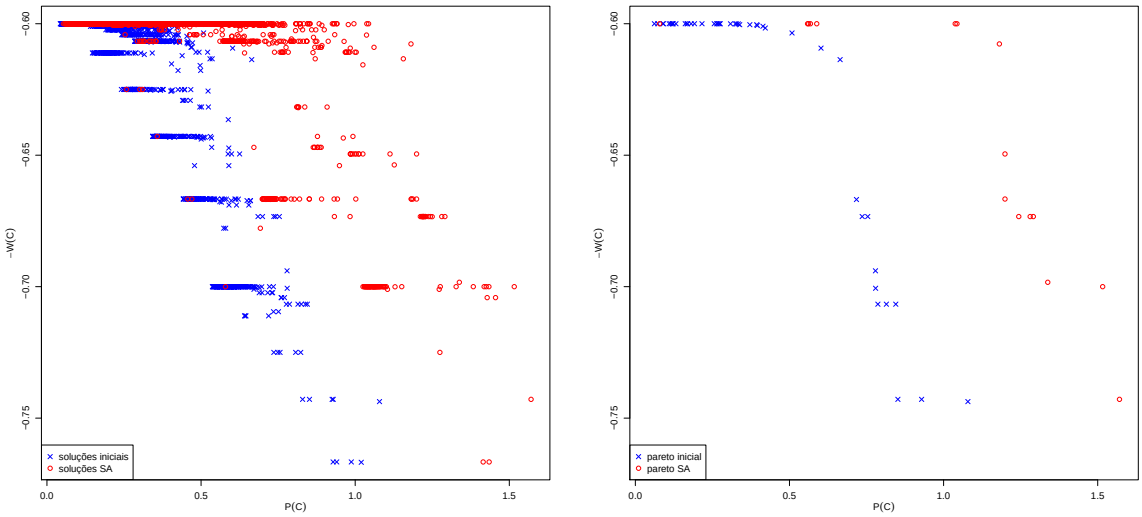

Figura 42 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 4 \times p_1$, $p_3 = p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

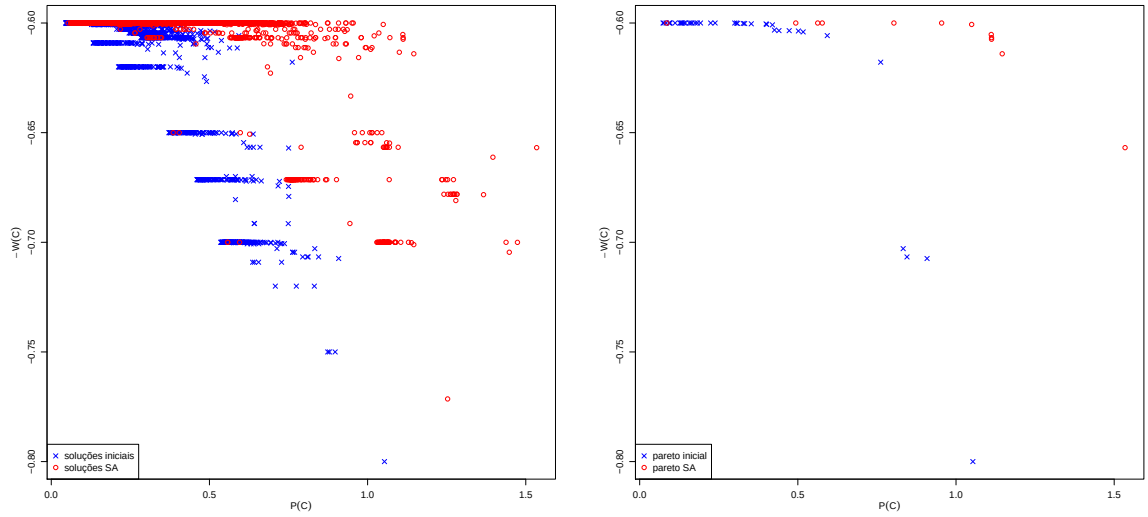

Figura 43 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 2 \times p_2$, $p_3 = p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

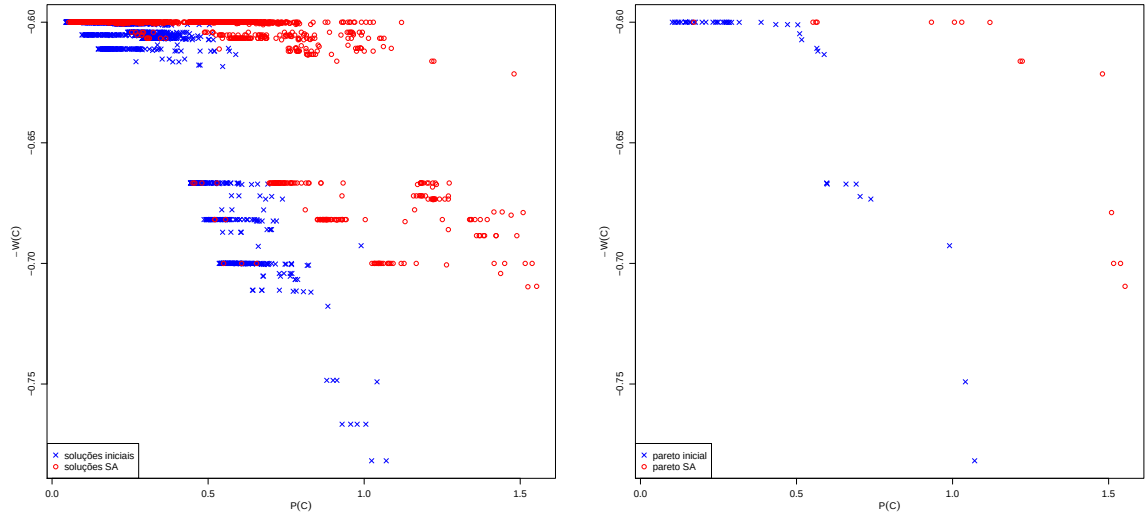

Figura 44 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 4 \times p_2$, $p_3 = p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

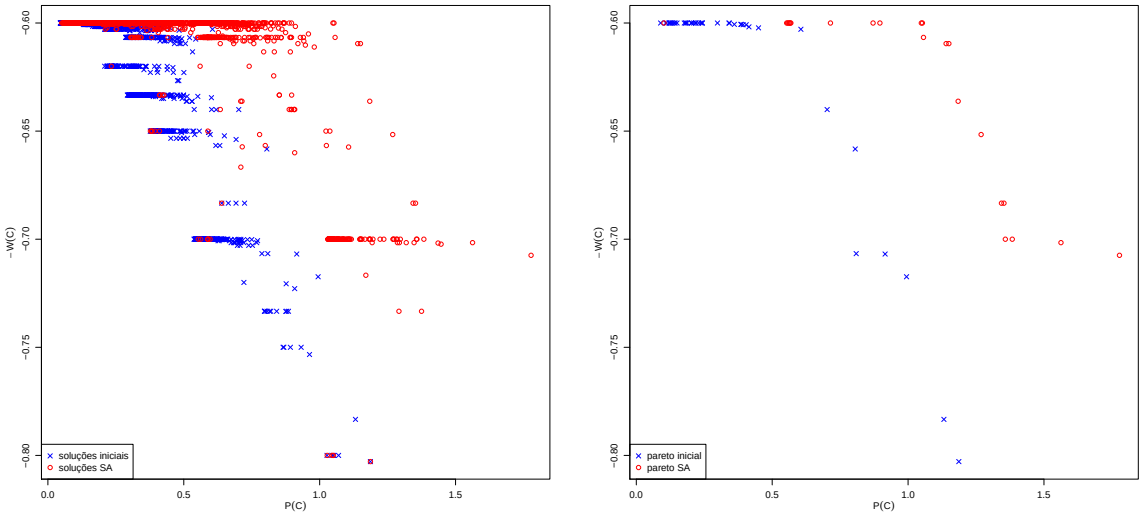

Figura 45 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = p_2$, $p_4 = 2 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

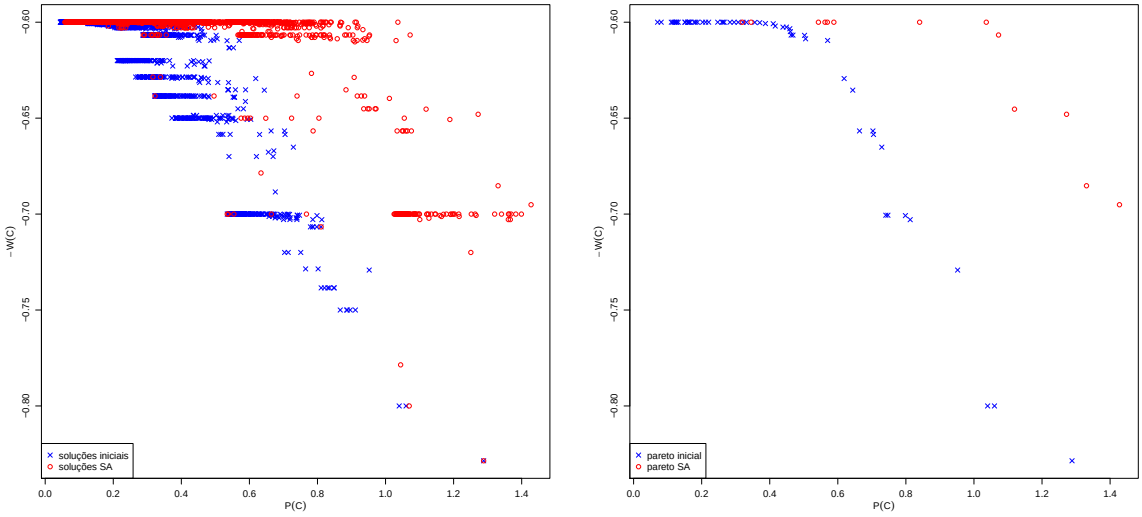

Figura 46 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 2 \times p_1$, $p_4 = 2 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

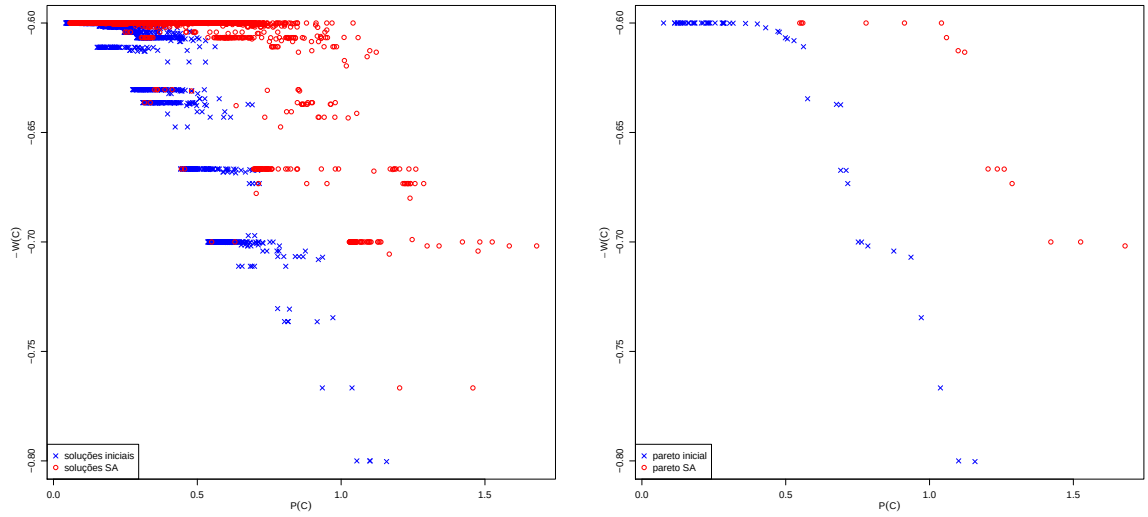

Figura 47 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 4 \times p_1$, $p_4 = 2 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

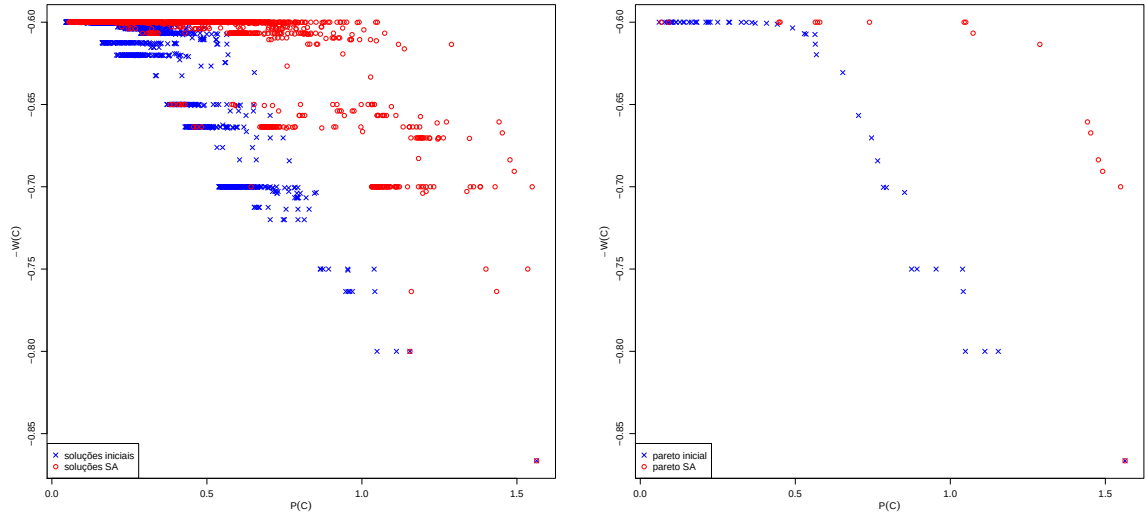

Figura 48 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 2 \times p_2$, $p_4 = 2 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

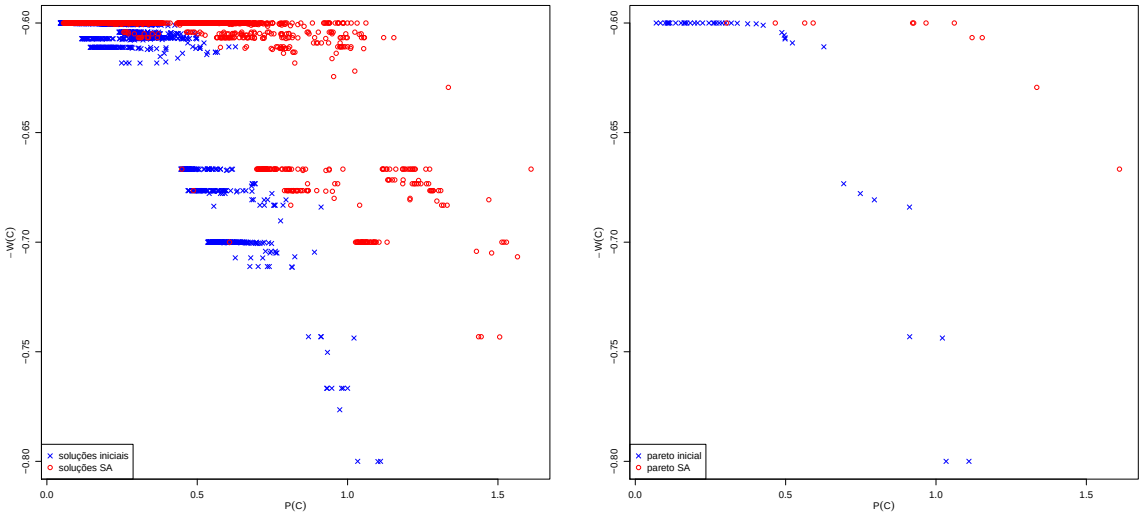

Figura 49 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 4 \times p_2$, $p_4 = 2 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

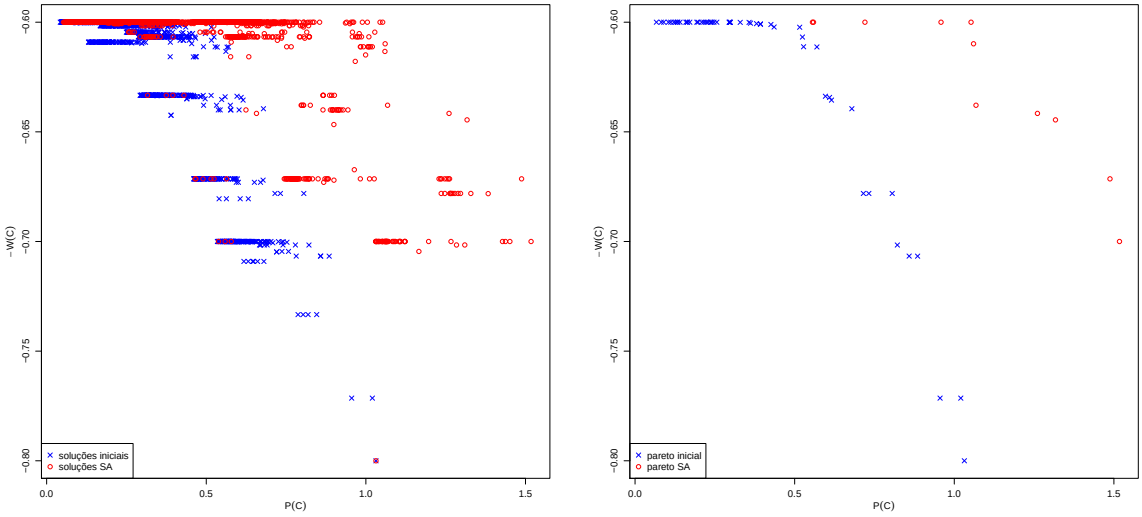

Figura 50 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = p_2$, $p_3 = 2 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

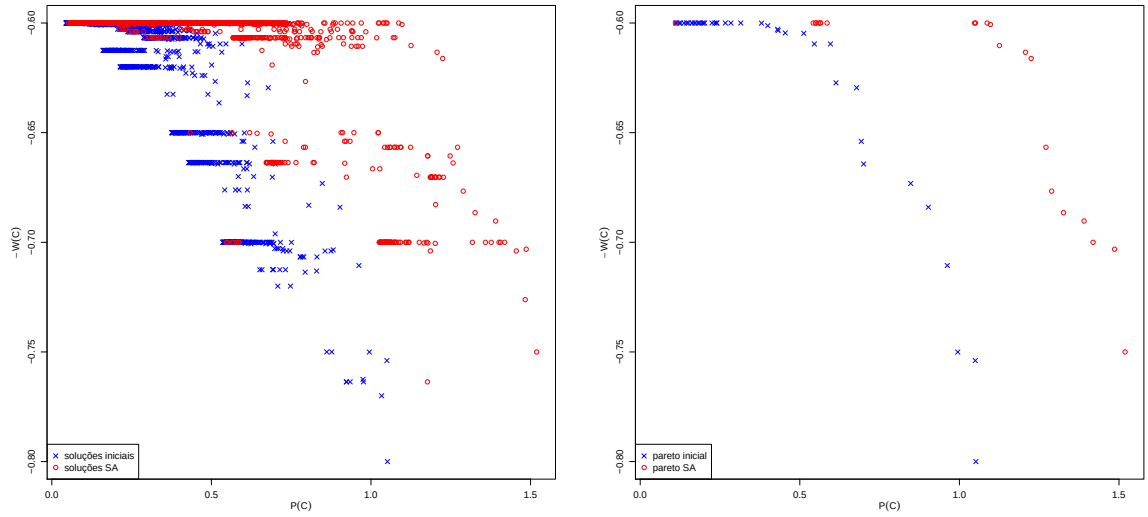

Figura 51 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 2 \times p_1$, $p_3 = 2 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

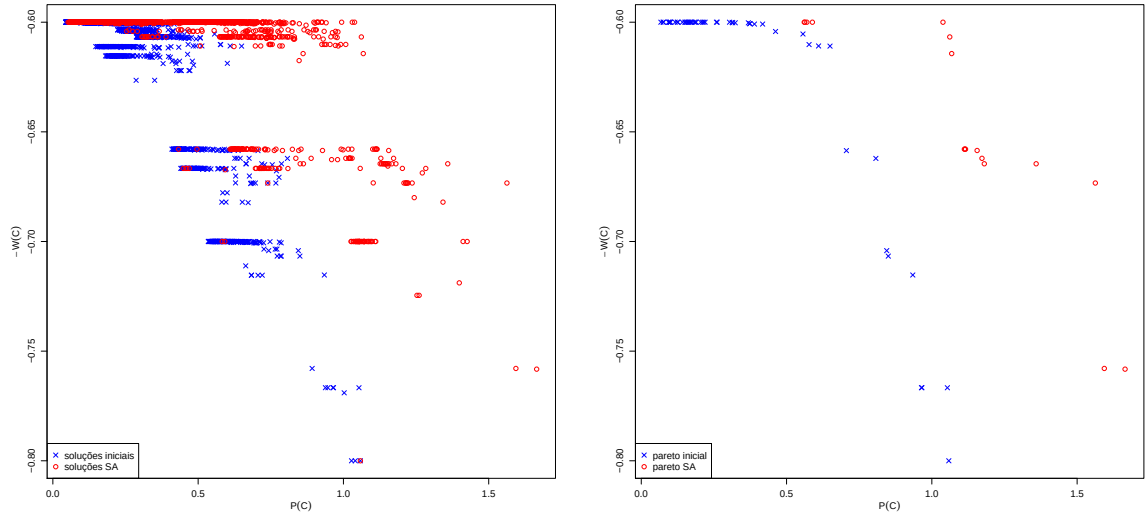

Figura 52 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 4 \times p_1$, $p_3 = 2 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

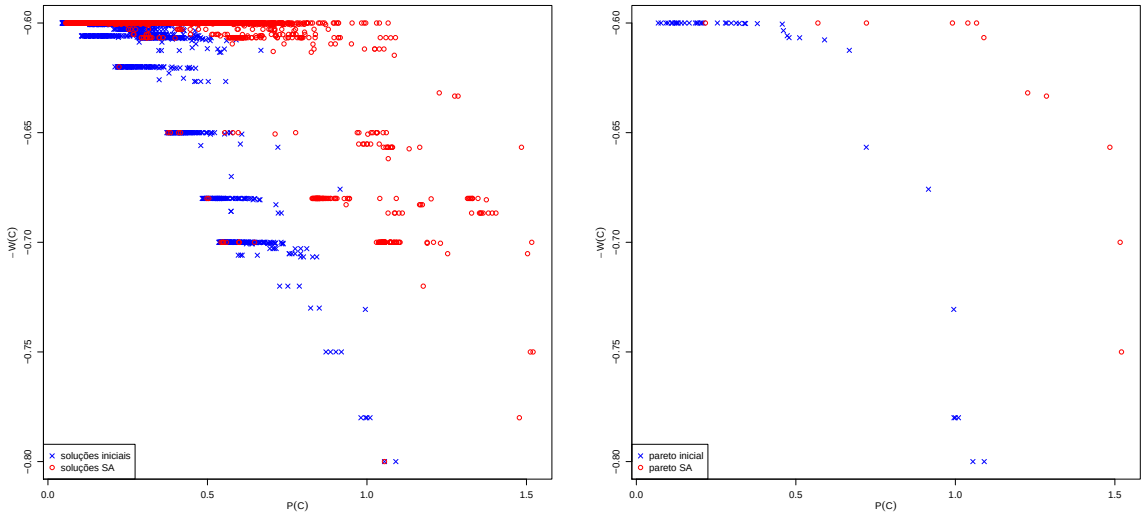

Figura 53 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 2 \times p_2$, $p_3 = 2 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

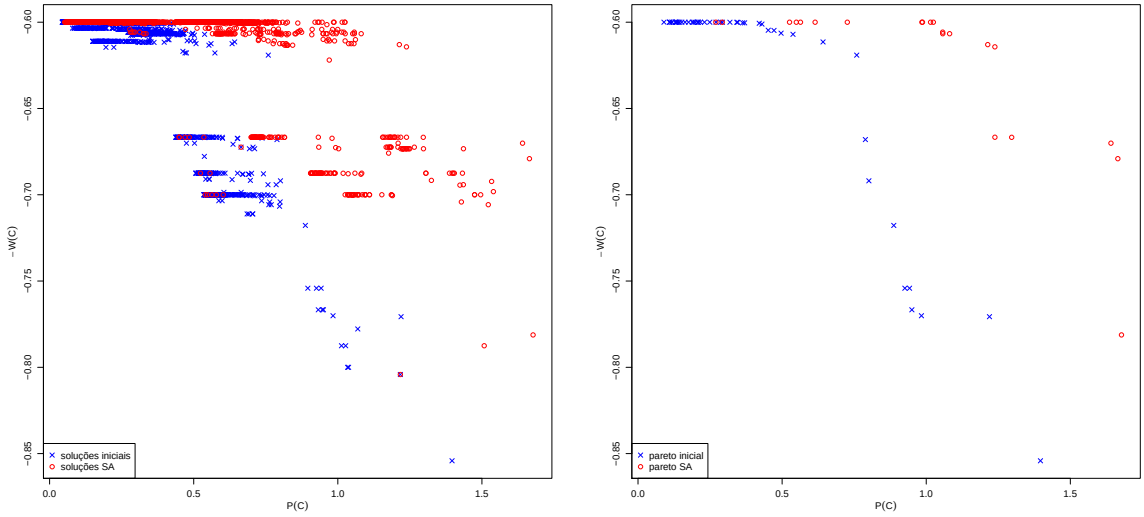

Figura 54 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 4 \times p_2$, $p_3 = 2 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

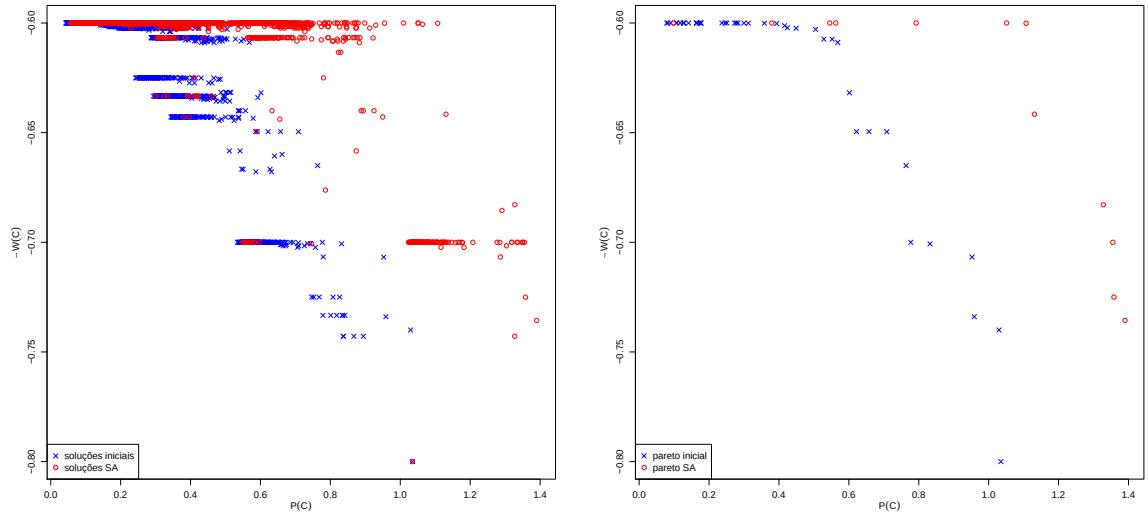

Figura 55 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = p_2$, $p_4 = 4 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

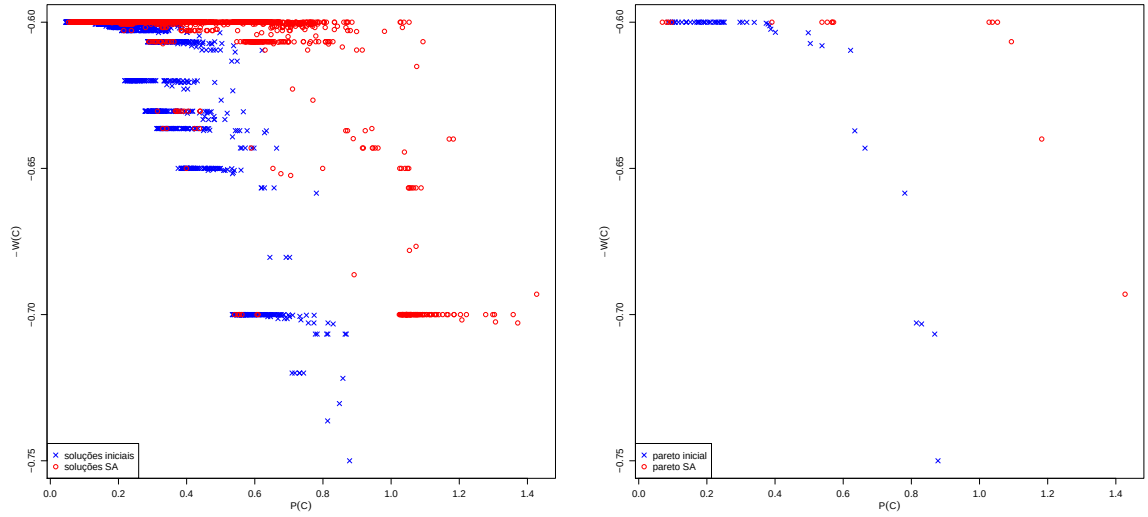

Figura 56 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 2 \times p_1$, $p_4 = 4 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

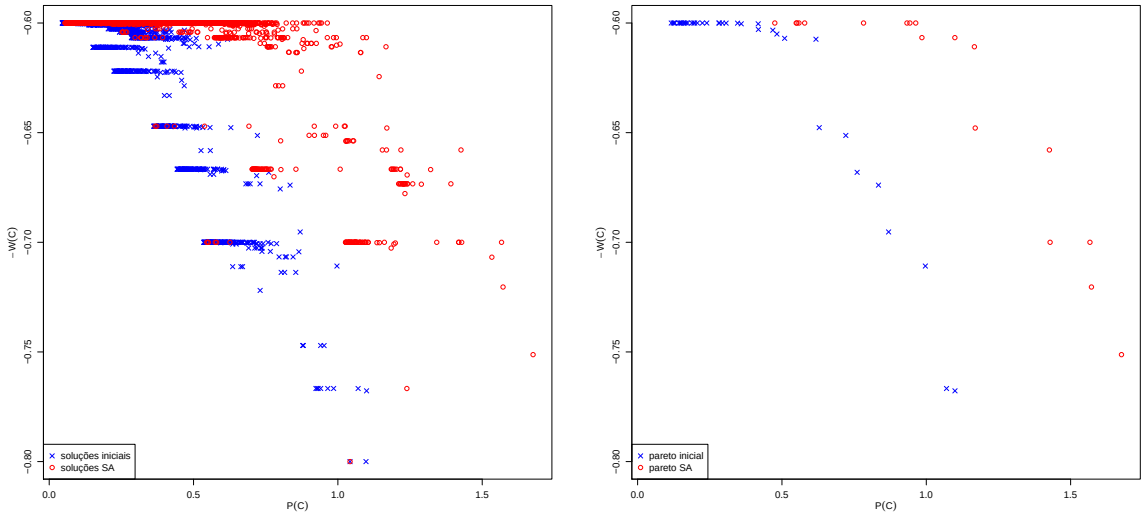

Figura 57 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 4 \times p_1$, $p_4 = 4 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

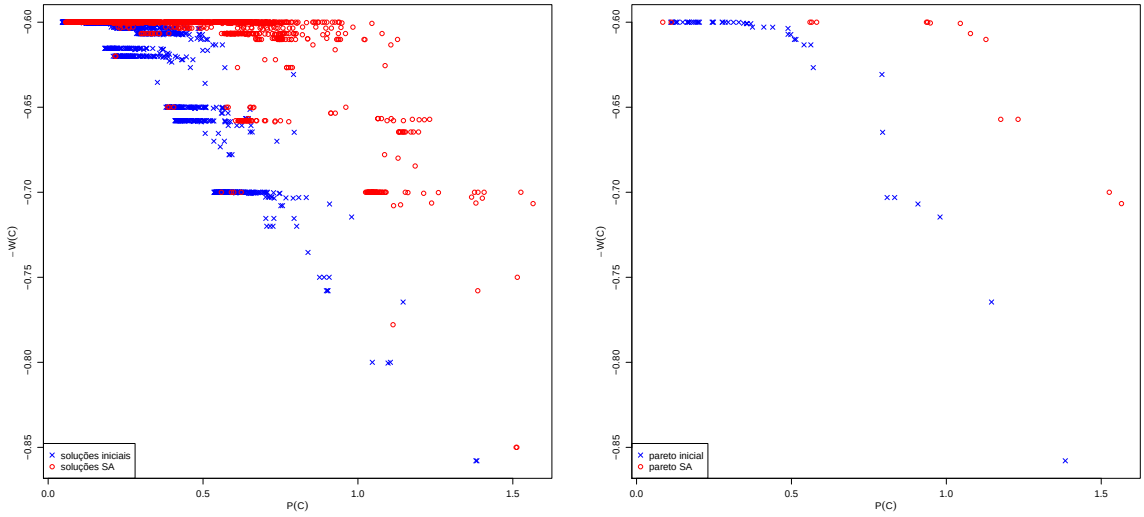

Figura 58 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 2 \times p_2$, $p_4 = 4 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

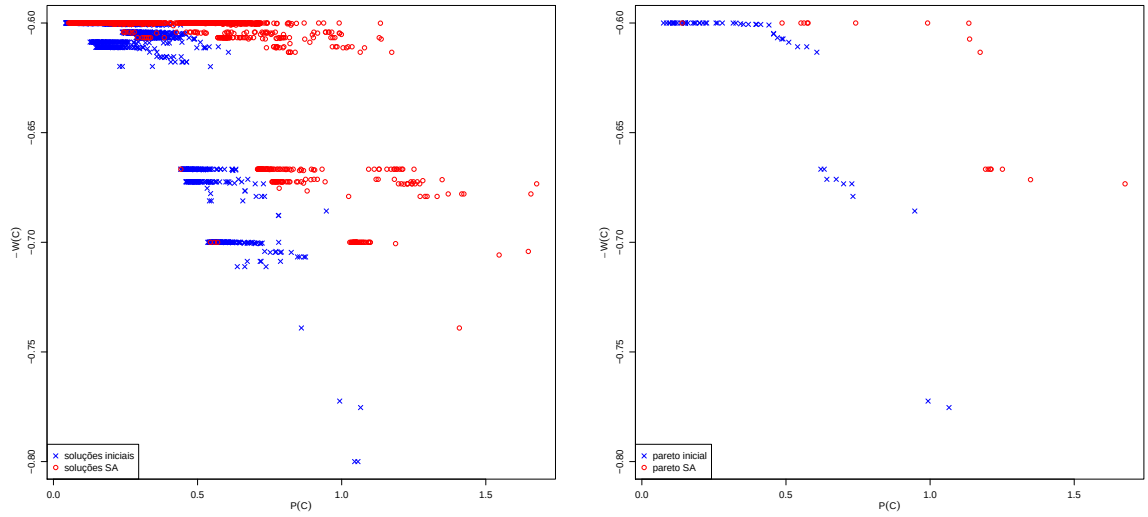

Figura 59 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 4 \times p_2$, $p_4 = 4 \times p_3$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

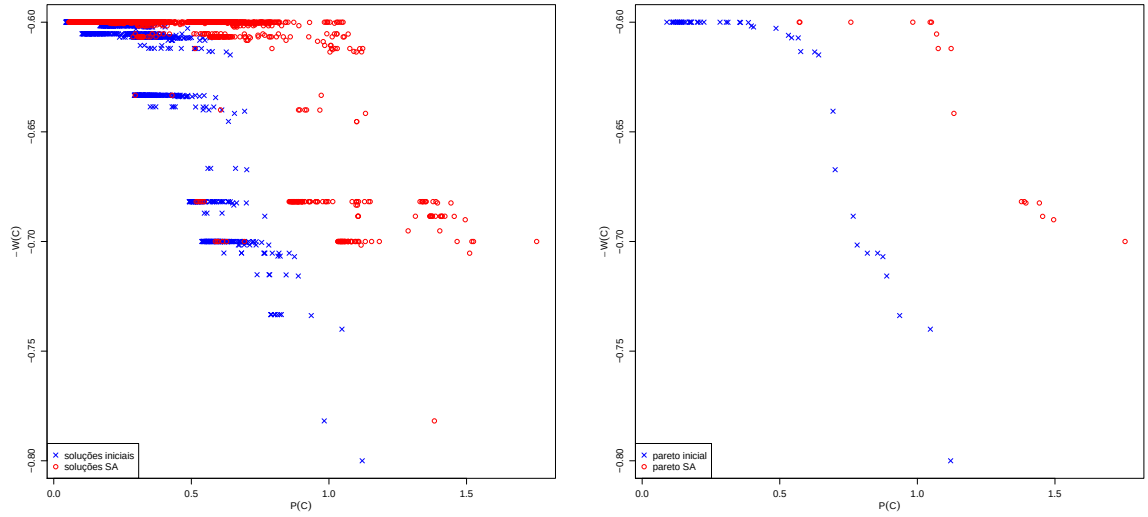

Figura 60 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = p_2$, $p_3 = 4 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

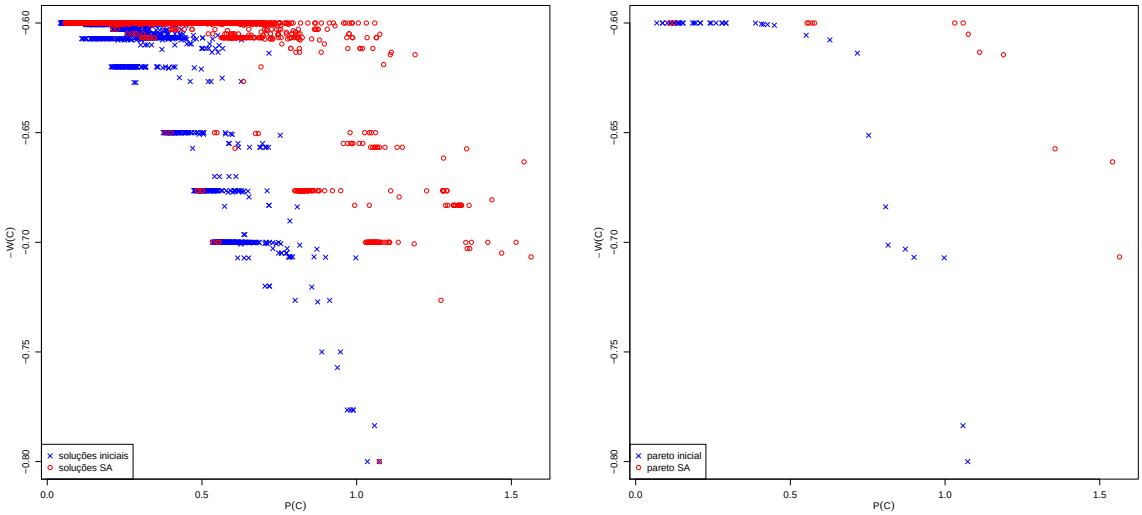

Figura 61 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 2 \times p_1$, $p_3 = 4 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

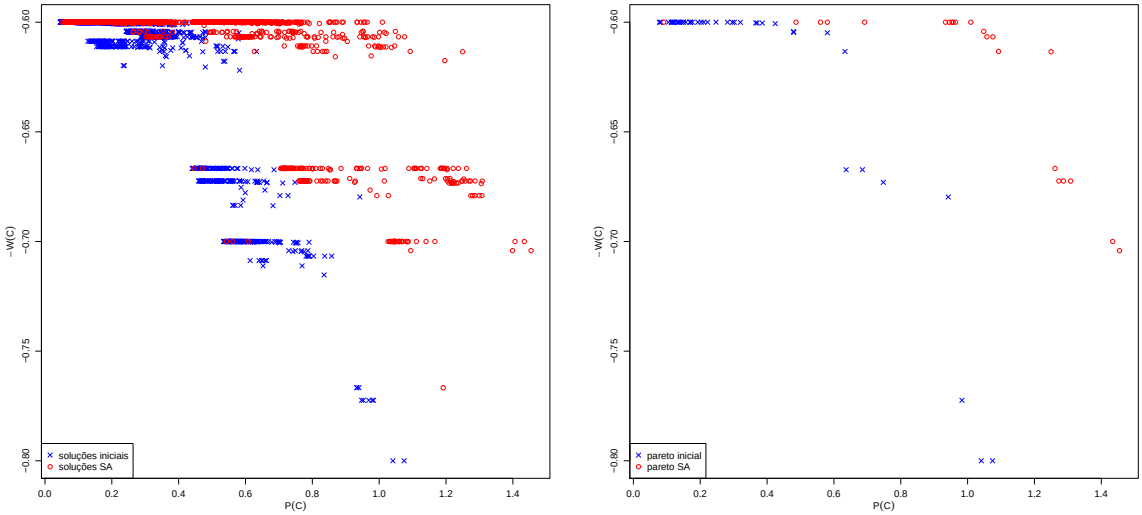

Figura 62 – Representação gráfica do espaço de soluções para rede de filas mista com $p_2 = 4 \times p_1$, $p_3 = 4 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

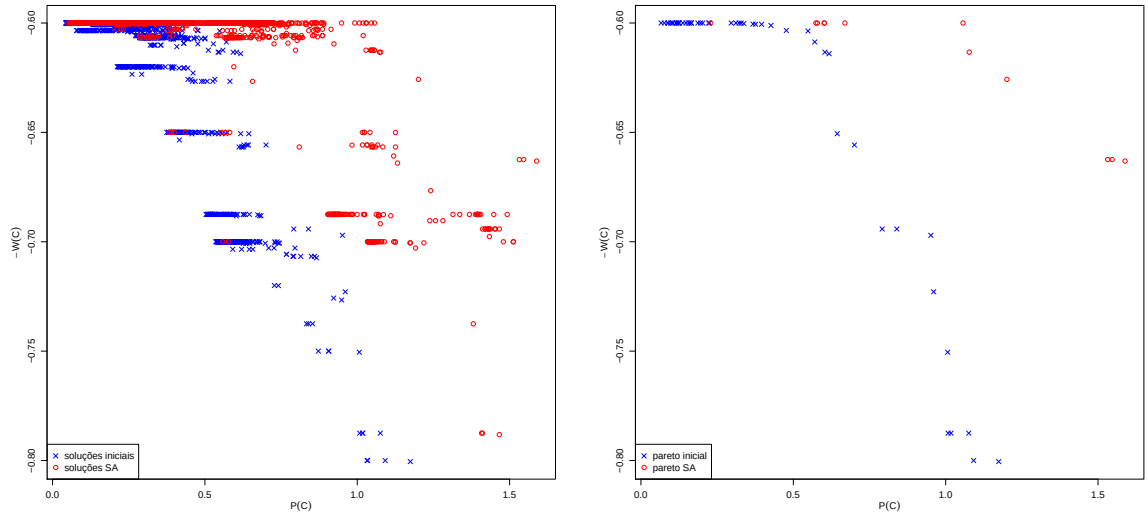

Figura 63 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 2 \times p_2$, $p_3 = 4 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

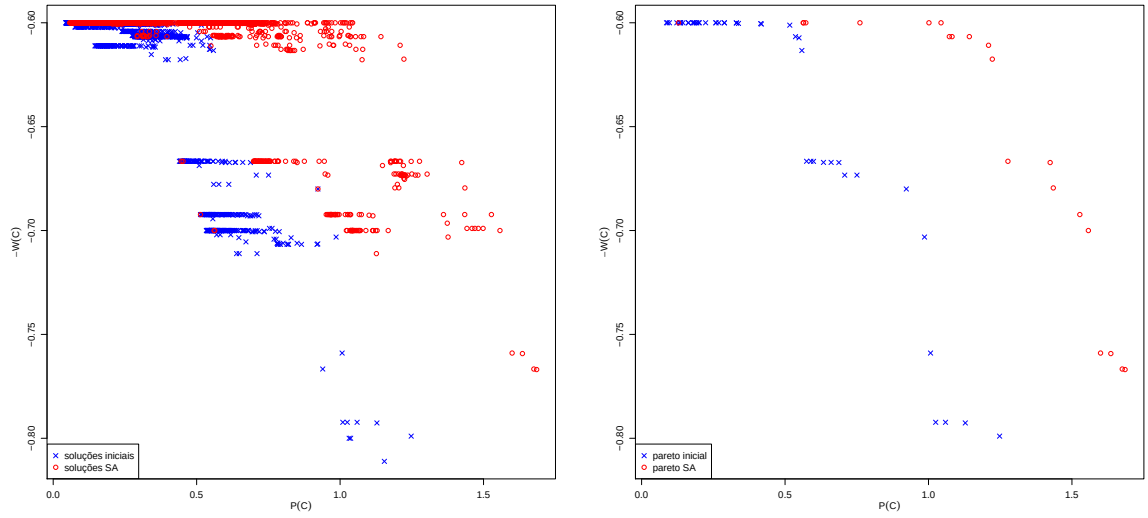

Figura 64 – Representação gráfica do espaço de soluções para rede de filas mista com $p_1 = 4 \times p_2$, $p_3 = 4 \times p_4$ no roteamento (à esquerda conjunto completo de soluções e à direita fronteira Pareto obtida.

Semelhante ao verificado nas análises anteriores, para os variados vetores de roteamento da rede de filas mista em estudo, as soluções fornecidas pelo algoritmo SA apresentaram um aumento significativo da produtividade dos servidores da rede de filas e a manutenção dos tempos totais de percurso em níveis aceitáveis.

Novamente com o formato de apresentação anterior, sequencialmente da Figura 65 até a Figura 88, são apresentados os padrões de alocação para os diferentes vetores

de roteamento.

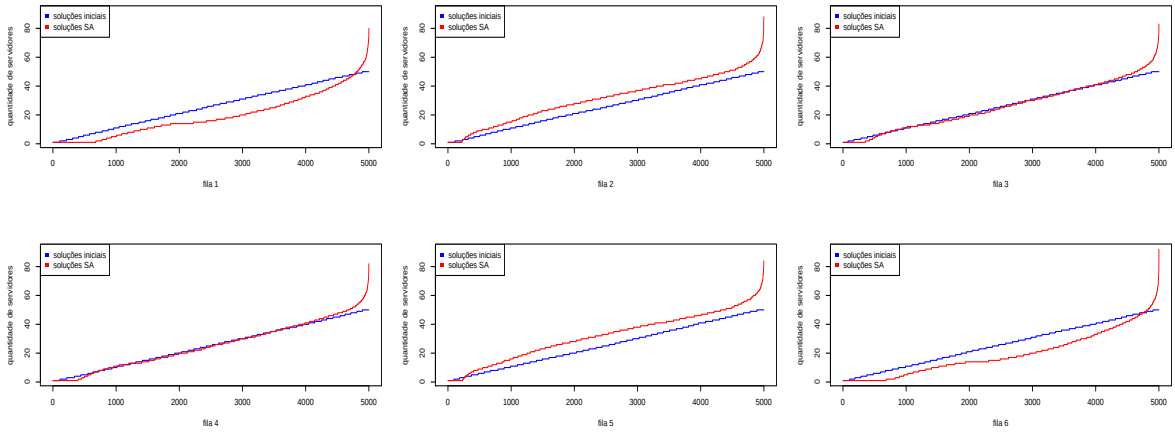

Figura 65 – Alocação de servidores nas filas da rede mista com $p_2 = 2 \times p_1$, $p_3 = p_4$ no roteamento.

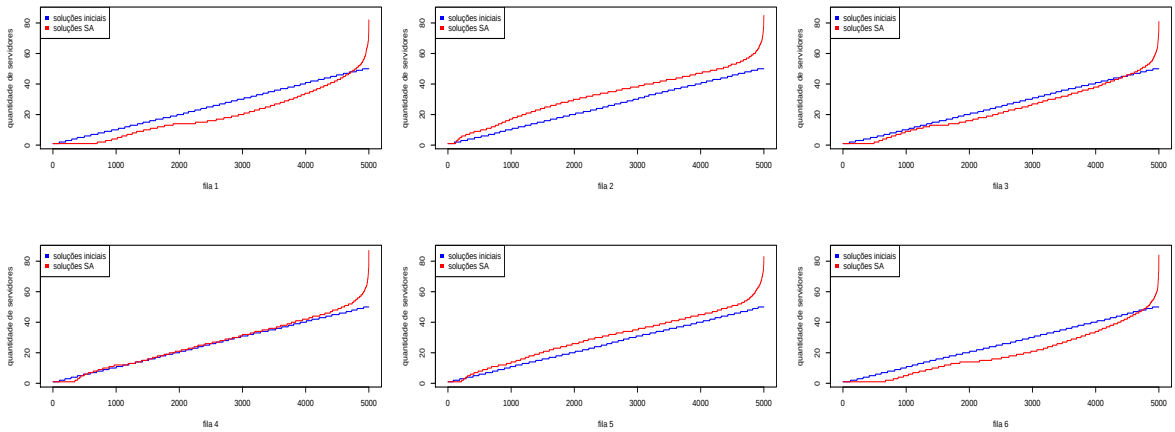

Figura 66 – Alocação de servidores nas filas da rede mista com $p_2 = 4 \times p_1$, $p_3 = p_4$ no roteamento.

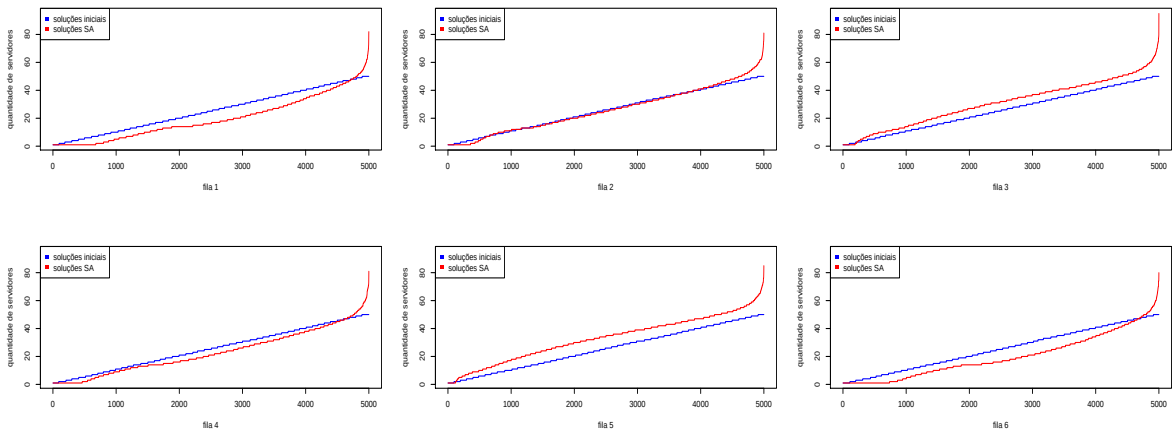

Figura 67 – Alocação de servidores nas filas da rede mista com $p_1 = 2 \times p_2$, $p_3 = p_4$ no roteamento.

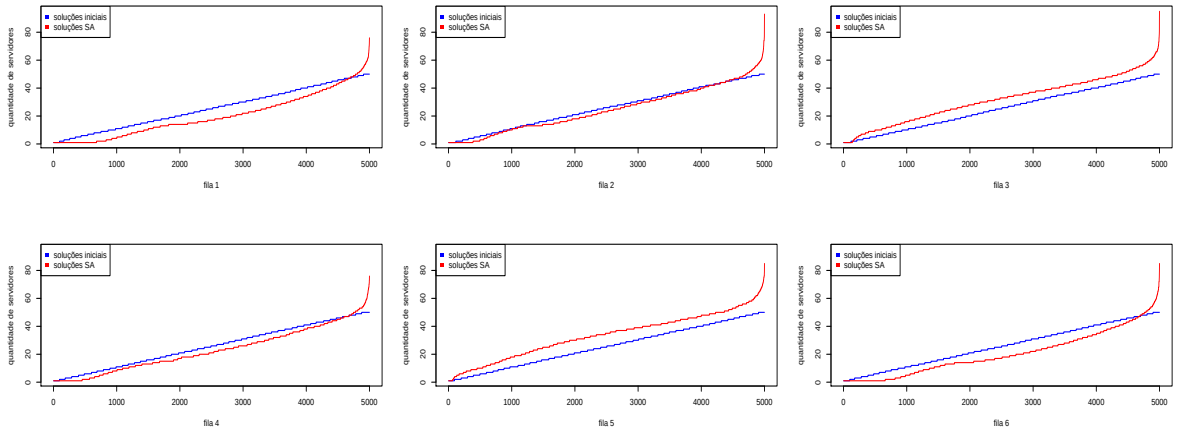

Figura 68 – Alocação de servidores nas filas da rede mista com $p_1 = 4 \times p_2$, $p_3 = p_4$ no roteamento.

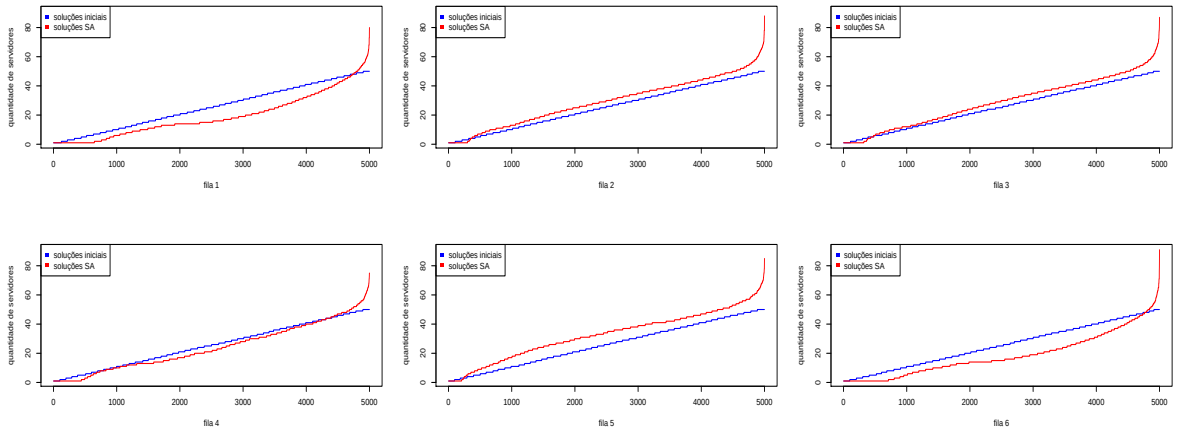

Figura 69 – Alocação de servidores nas filas da rede mista com $p_1 = p_2$, $p_4 = 2 \times p_3$ no roteamento.

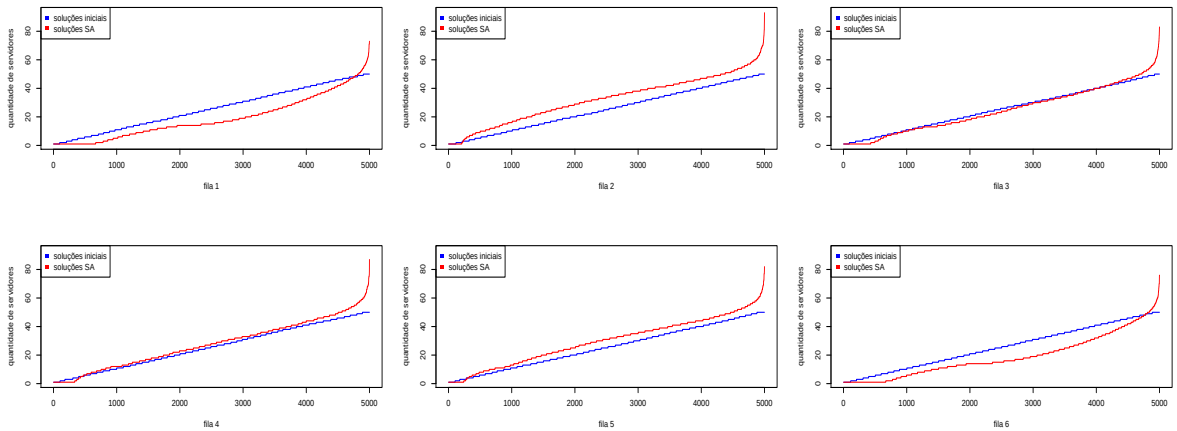

Figura 70 – Alocação de servidores nas filas da rede mista com $p_2 = 2 \times p_1$, $p_4 = 2 \times p_3$ no roteamento.

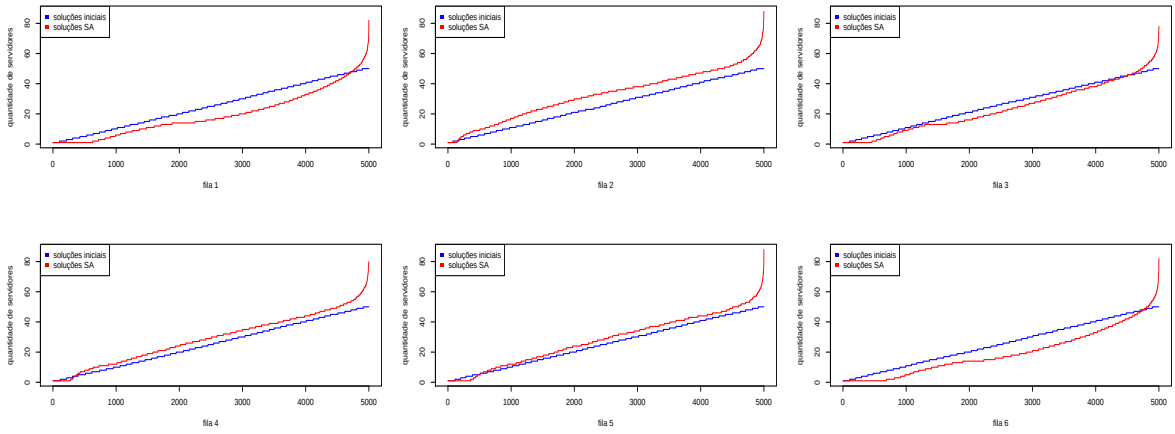

Figura 71 – Alocação de servidores nas filas da rede mista com $p_2 = 4 \times p_1$, $p_4 = 2 \times p_3$ no roteamento.

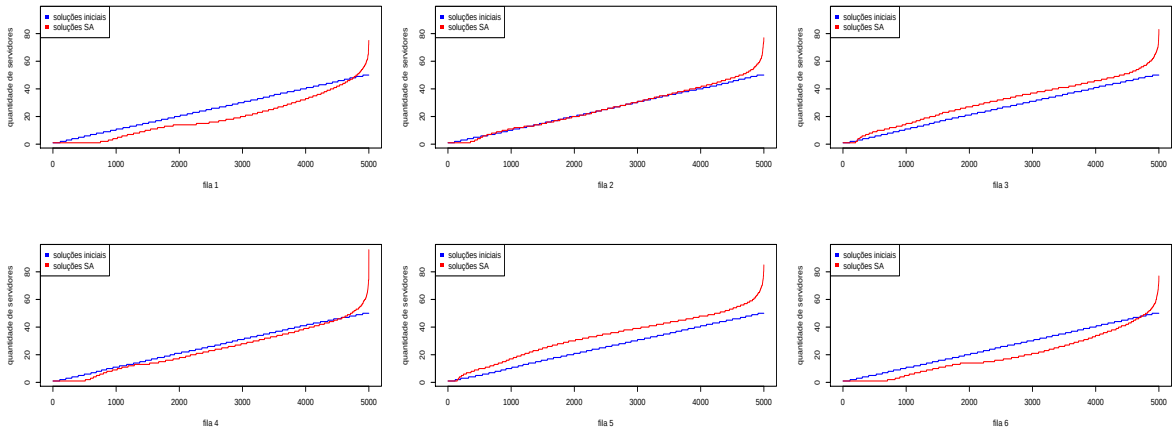

Figura 72 – Alocação de servidores nas filas da rede mista com $p_1 = 2 \times p_2$, $p_4 = 2 \times p_3$ no roteamento.

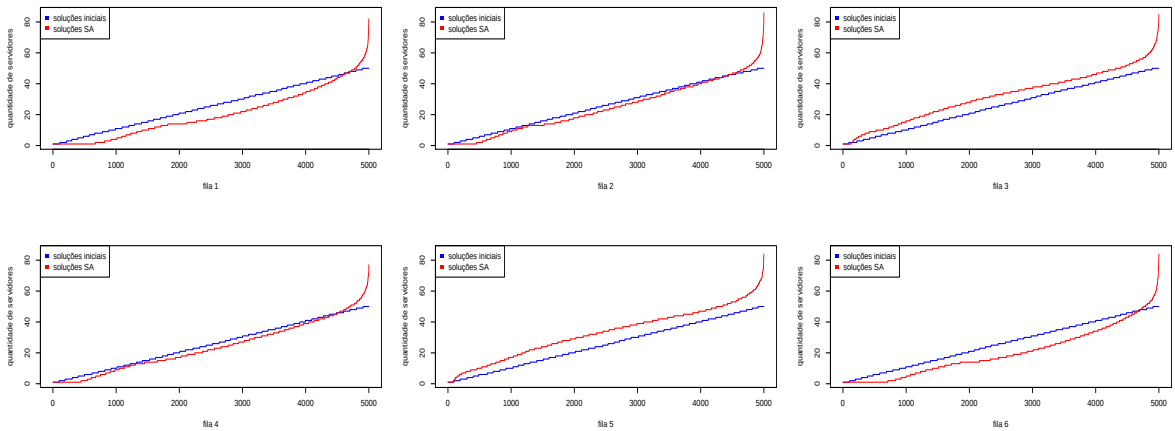

Figura 73 – Alocação de servidores nas filas da rede mista com $p_1 = 4 \times p_2$, $p_4 = 2 \times p_3$ no roteamento.

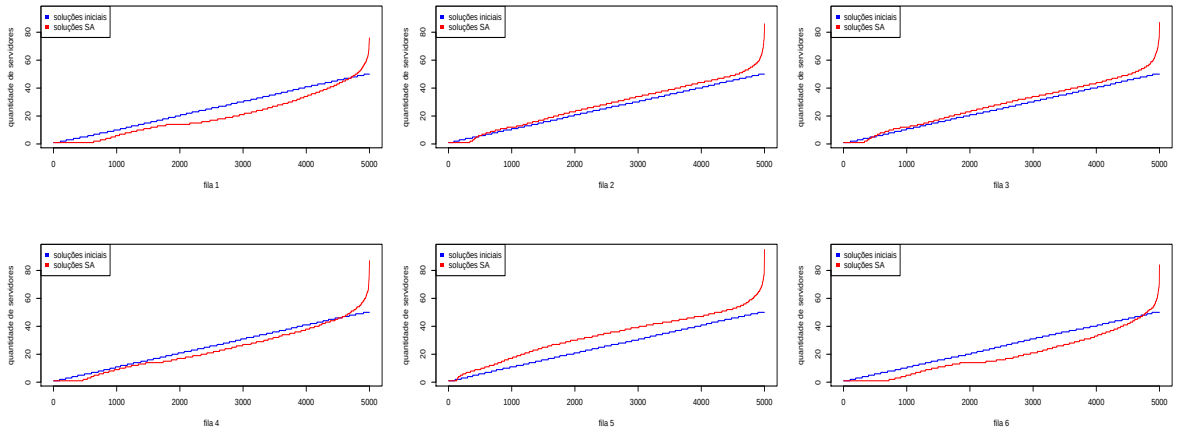

Figura 74 – Alocação de servidores nas filas da rede mista com $p_1 = p_2$, $p_3 = 2 \times p_4$ no roteamento.

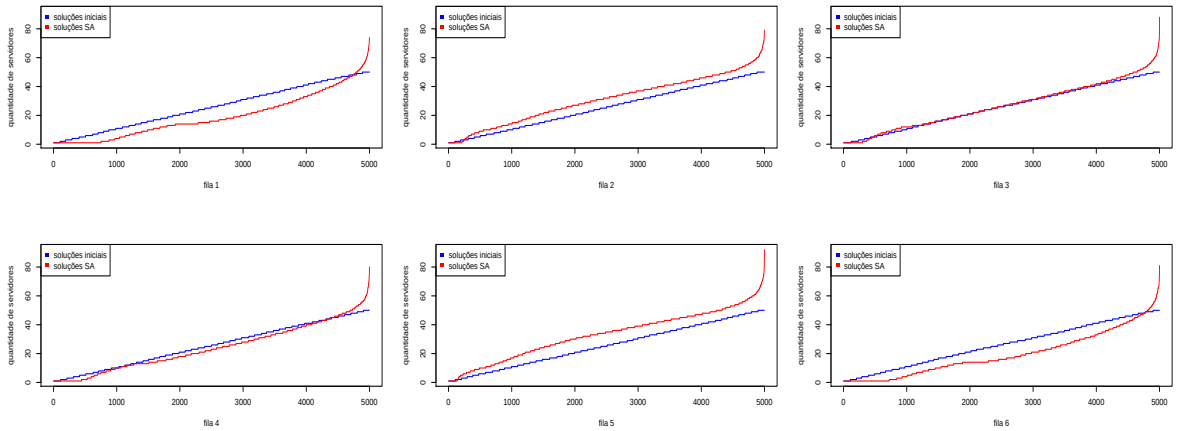

Figura 75 – Alocação de servidores nas filas da rede mista com $p_2 = 2 \times p_1$, $p_3 = 2 \times p_4$ no roteamento.

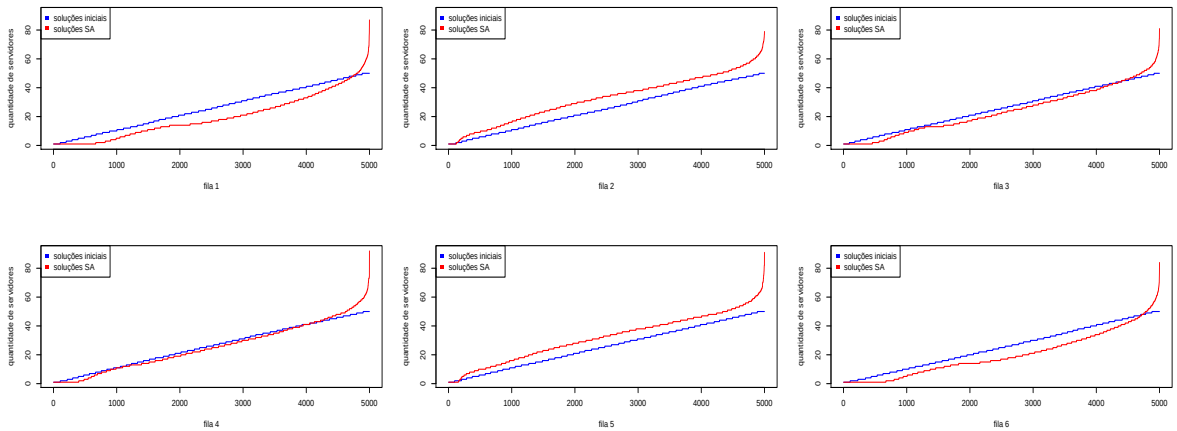

Figura 76 – Alocação de servidores nas filas da rede mista com $p_2 = 4 \times p_1$, $p_3 = 2 \times p_4$ no roteamento.

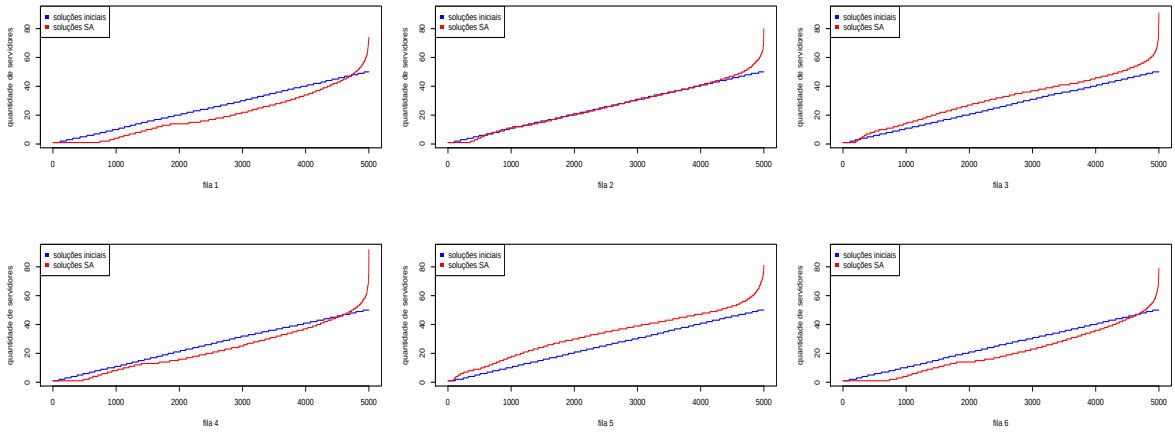

Figura 77 – Alocação de servidores nas filas da rede mista com $p_1 = 2 \times p_2$, $p_3 = 2 \times p_4$ no roteamento.

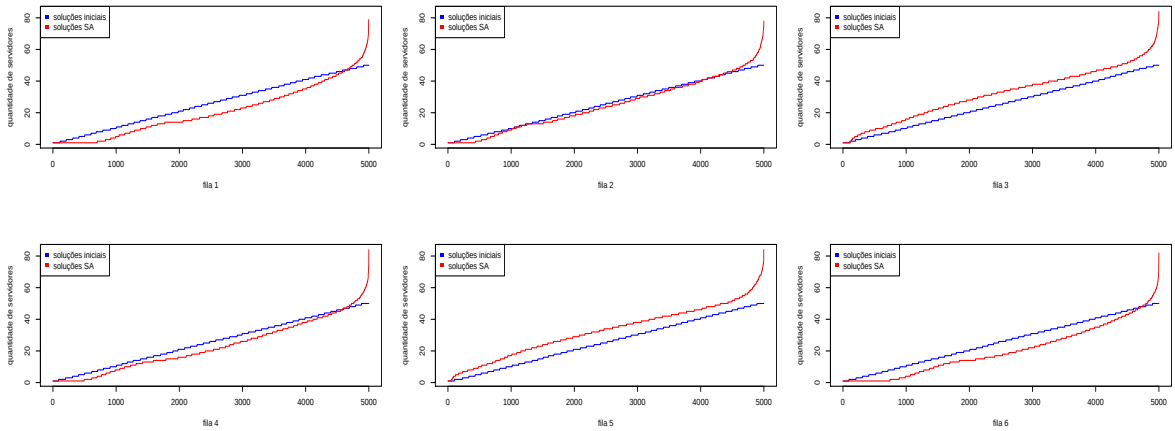

Figura 78 – Alocação de servidores nas filas da rede mista com $p_1 = 4 \times p_2$, $p_3 = 2 \times p_4$ no roteamento.

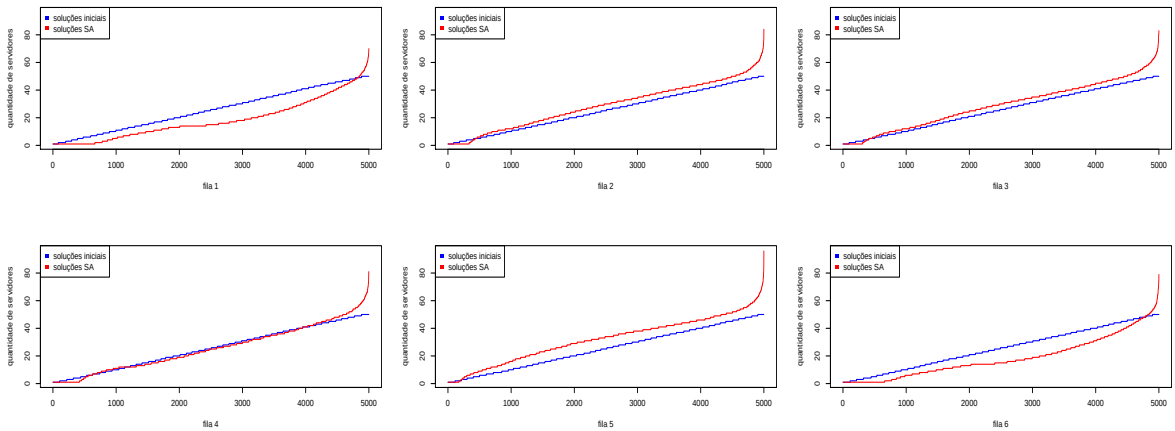

Figura 79 – Alocação de servidores nas filas da rede mista com $p_1 = p_2$, $p_4 = 4 \times p_3$ no roteamento.

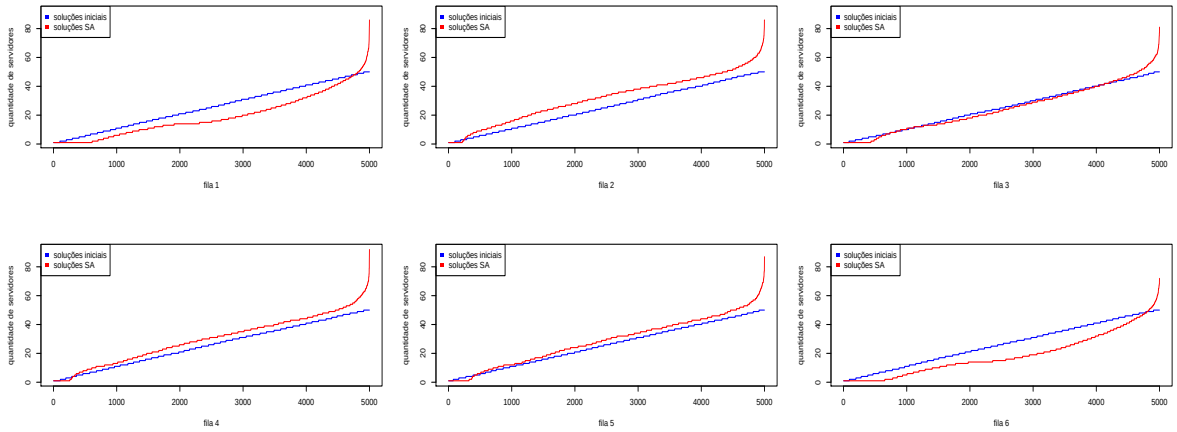

Figura 80 – Alocação de servidores nas filas da rede mista com $p_2 = 2 \times p_1$, $p_4 = 4 \times p_3$ no roteamento.

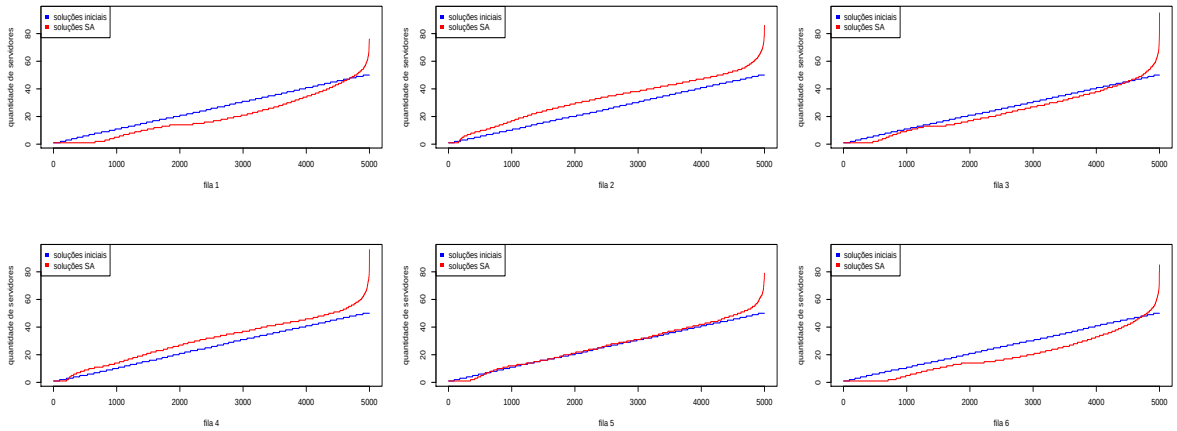

Figura 81 – Alocação de servidores nas filas da rede mista com $p_2 = 4 \times p_1$, $p_4 = 4 \times p_3$ no roteamento.

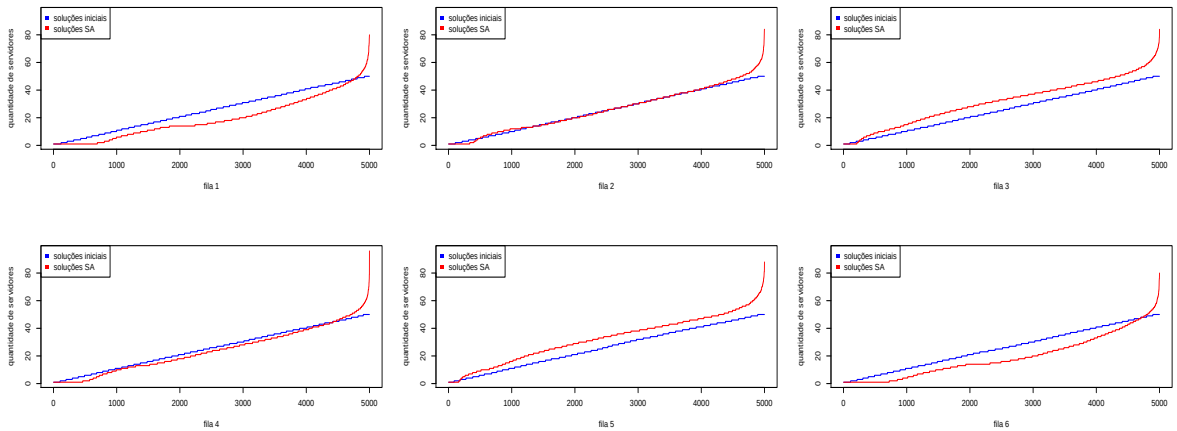

Figura 82 – Alocação de servidores nas filas da rede mista com $p_1 = 2 \times p_2$, $p_4 = 4 \times p_3$ no roteamento.

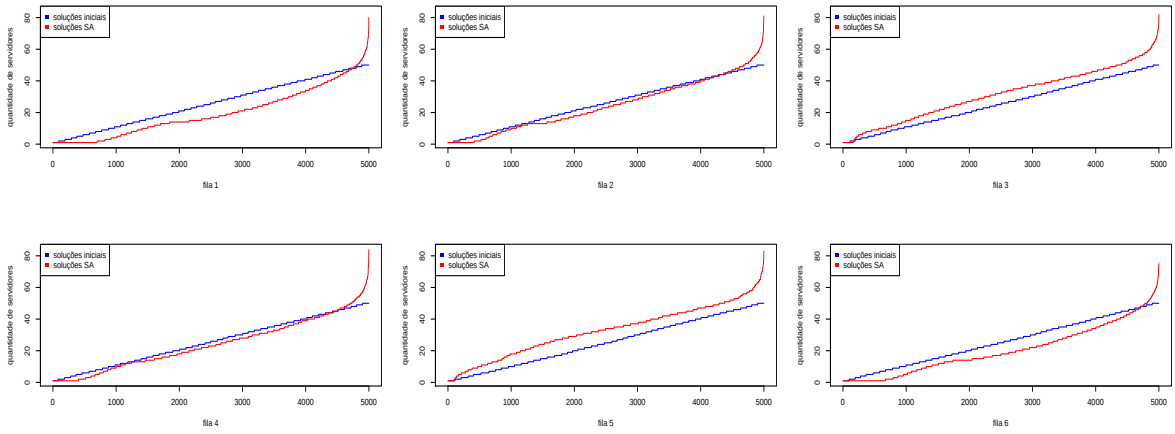

Figura 83 – Alocação de servidores nas filas da rede mista com $p_1 = 4 \times p_2$, $p_4 = 4 \times p_3$ no roteamento.

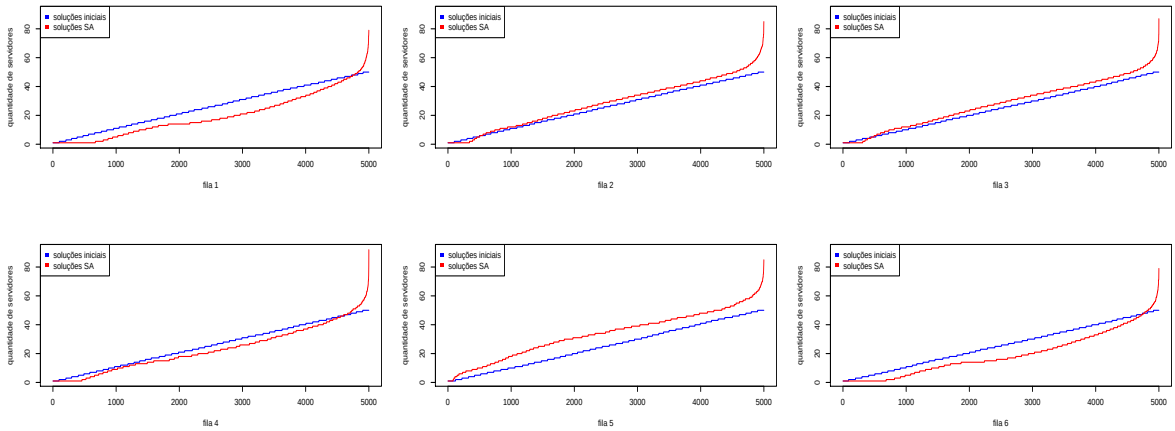

Figura 84 – Alocação de servidores nas filas da rede mista com $p_1 = p_2$, $p_3 = 4 \times p_4$ no roteamento.

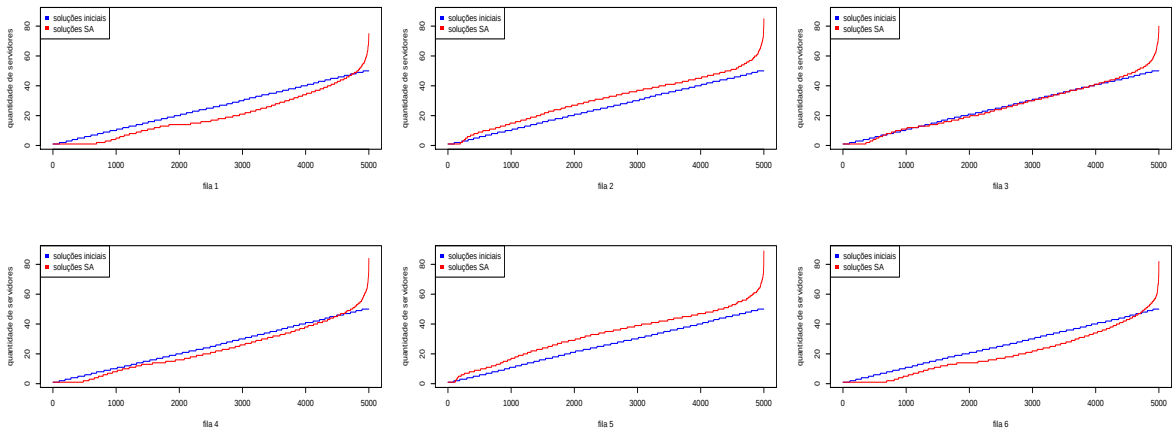

Figura 85 – Alocação de servidores nas filas da rede mista com $p_2 = 2 \times p_1$, $p_3 = 4 \times p_4$ no roteamento.

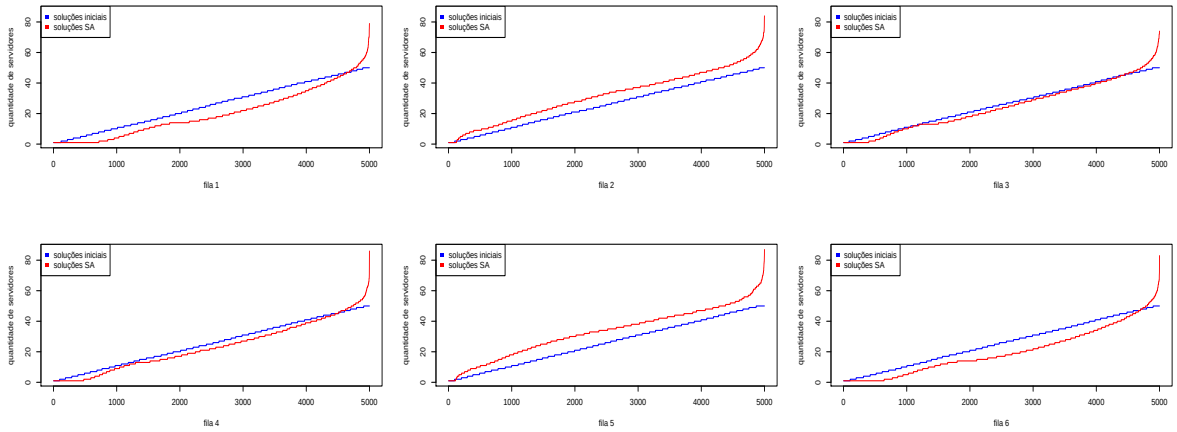

Figura 86 – Alocação de servidores nas filas da rede mista com $p_2 = 4 \times p_1$, $p_3 = 4 \times p_4$ no roteamento.

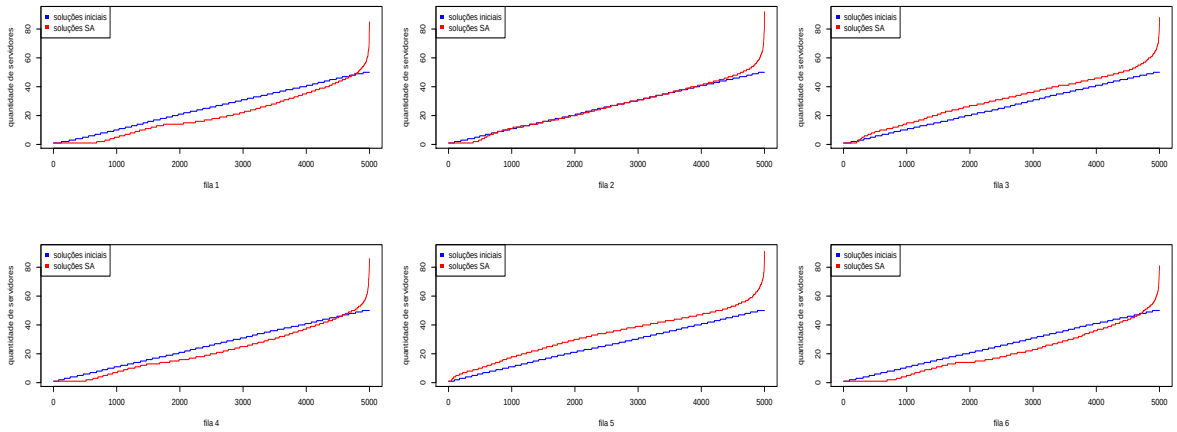

Figura 87 – Alocação de servidores nas filas da rede mista com $p_1 = 2 \times p_2$, $p_3 = 4 \times p_4$ no roteamento.

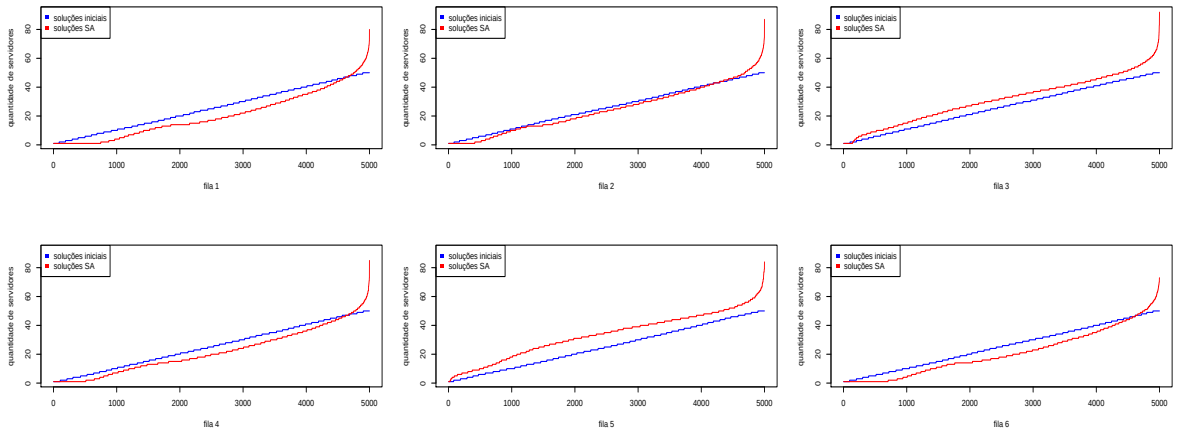

Figura 88 – Alocação de servidores nas filas da rede mista com $p_1 = 4 \times p_2$, $p_3 = 4 \times p_4$ no roteamento.

As formas variadas utilizadas para o vetor de roteamento novamente ilustram, agora para uma rede de filas com nível de complexidade mais avançado, que o algo-

ritmo implementado é capaz de fornecer alocações de servidores específicas para cada configuração da rede de filas. É possível verificar que o padrão de alocação proposto pelo algoritmo SA se adapta às modificações no roteamento e altera o padrão de alocação de servidores. Inclusive existe um padrão de simetria no processo de alocação e alteração da probabilidade de roteamento entre as filas 2 e 3. Para as filas 4 e 5, a identificação destes padrões não é tão imediata, uma vez que as quatro cordenadas do vetor de roteamento tem impacto nas filas 4 e 5.

A Tabela 11 apresenta resultados comparativos entre as soluções iniciais e as soluções fornecidas pelo algoritmo SA para os ganhos associados à medida de produtividade.

Tabela 11 – Melhoria obtida em produtividade média das soluções para redes de filas mista.

roteamento	soluções iniciais	soluções SA	percentual de aumento
	$\overline{P(\mathbf{C})}$ inicial (d.p.)	$\overline{P(\mathbf{C})}$ SA (d.p.)	
$(1/2, 1/2, 1/2, 1/2)$	0,383387 (0,286697)	0,649351 (0,499707)	69,3722%
$(1/3, 2/3, 1/2, 1/2)$	0,361656 (0,264378)	0,868158 (0,437063)	140,0507%
$(1/5, 4/5, 1/2, 1/2)$	0,395796 (0,279620)	1,016039 (0,418884)	156,7081%
$(2/3, 1/3, 1/2, 1/2)$	0,327778 (0,252956)	0,878586 (0,389970)	168,0434%
$(4/5, 1/5, 1/2, 1/2)$	0,371929 (0,254690)	1,064990 (0,436929)	186,3424%
$(1/2, 1/2, 1/3, 2/3)$	0,357934 (0,292059)	0,984065 (0,407756)	174,9293%
$(1/3, 2/3, 1/3, 2/3)$	0,400563 (0,281042)	0,848132 (0,382981)	111,7351%
$(1/5, 4/5, 1/3, 2/3)$	0,410163 (0,293012)	1,080609 (0,340666)	163,4586%
$(2/3, 1/3, 1/3, 2/3)$	0,453635 (0,363880)	0,941299 (0,511228)	107,5014%
$(4/5, 1/5, 1/3, 2/3)$	0,379143 (0,290075)	0,918230 (0,379827)	142,1859%
$(1/2, 1/2, 2/3, 1/3)$	0,386836 (0,266165)	1,050636 (0,335969)	171,5970%
$(1/3, 2/3, 2/3, 1/3)$	0,404501 (0,293230)	0,980315 (0,400647)	142,3517%
$(1/3, 2/3, 2/3, 1/3)$	0,358753 (0,293839)	1,087163 (0,352159)	203,0396%
$(2/3, 1/3, 2/3, 1/3)$	0,387521 (0,316464)	1,060509 (0,399186)	173,6647%
$(4/5, 1/5, 2/3, 1/3)$	0,399079 (0,326822)	0,986596 (0,415030)	147,2180%
$(1/2, 1/2, 1/5, 4/5)$	0,391390 (0,284282)	0,924829 (0,438733)	136,2933%
$(1/3, 2/3, 1/5, 4/5)$	0,309265 (0,221058)	0,646509 (0,442141)	109,0471%
$(1/5, 4/5, 1/5, 4/5)$	0,388563 (0,289877)	1,023723 (0,393760)	163,4636%
$(2/3, 1/3, 1/5, 4/5)$	0,395193 (0,287956)	0,898419 (0,442841)	127,3366%
$(4/5, 1/5, 1/5, 4/5)$	0,356165 (0,248043)	0,953317 (0,395513)	167,6619%
$(1/2, 1/2, 4/5, 1/5)$	0,392710 (0,295180)	1,158489 (0,327355)	194,9983%
$(1/3, 2/3, 4/5, 1/5)$	0,365098 (0,300553)	0,887334 (0,474817)	143,0402%
$(1/5, 4/5, 4/5, 1/5)$	0,329863 (0,277835)	0,987046 (0,335380)	199,2289%
$(2/3, 1/3, 4/5, 1/5)$	0,406211 (0,337587)	0,93810 (0,4583544)	130,9389%
$(4/5, 1/5, 4/5, 1/5)$	0,435255 (0,321877)	1,151448 (0,432965)	164,5454%

d.p. - desvio padrão

O aumento nos níveis de produtividade é expressivo. Não é possível identificar

algum padrão específico que conduz para uma melhoria mais acentuada em um vetor de roteamento em detrimento de outros. Talvez o componente aleatorio do algoritmo seja o único responsável por esta variação. Mas de forma clara todas as configurações apresentam melhora significativa.

A Tabela 12 mostra os resultados comparativos entre as soluções iniciais e as soluções fornecidas pelo algoritmo SA para os ganhos associados ao tempo de percurso da rede de filas.

Tabela 12 – Melhoria obtida em tempo médio de percurso das soluções para redes de filas mista.

roteamento	soluções iniciais	soluções SA	percentual de redução
	$\overline{W(\mathbf{C})}$ inicial (d.p.)	$\overline{W(\mathbf{C})}$ SA (d.p.)	
$(1/2, 1/2, 1/2, 1/2)$	0,623993 (0,053202)	0,610347 (0,026869)	2,1869%
$(1/3, 2/3, 1/2, 1/2)$	0,619323 (0,037111)	0,621053 (0,043168)	−0,2778%
$(1/5, 4/5, 1/2, 1/2)$	0,625840 (0,046989)	0,642817 (0,047494)	−2,7128%
$(2/3, 1/3, 1/2, 1/2)$	0,613692 (0,040690)	0,607569 (0,016145)	0,9977%
$(4/5, 1/5, 1/2, 1/2)$	0,619502 (0,040510)	0,629484 (0,043227)	−1,6113%
$(1/2, 1/2, 1/3, 2/3)$	0,619665 (0,049413)	0,629965 (0,042265)	−1,6622%
$(1/3, 2/3, 1/3, 2/3)$	0,625770 (0,053708)	0,621564 (0,034965)	0,6722%
$(1/5, 4/5, 1/3, 2/3)$	0,628140 (0,052759)	0,637991 (0,041932)	−1,5683%
$(2/3, 1/3, 1/3, 2/3)$	0,641434 (0,070433)	0,638262 (0,067864)	0,4945%
$(4/5, 1/5, 1/3, 2/3)$	0,623829 (0,053159)	0,609116 (0,019998)	2,3585%
$(1/2, 1/2, 2/3, 1/3)$	0,623149 (0,048330)	0,627752 (0,034460)	−0,7385%
$(1/3, 2/3, 2/3, 1/3)$	0,623969 (0,049043)	0,631988 (0,046751)	−1,2852%
$(1/3, 2/3, 2/3, 1/3)$	0,622929 (0,052244)	0,649050 (0,050222)	−4,1933%
$(2/3, 1/3, 2/3, 1/3)$	0,628351 (0,062543)	0,631544 (0,048563)	−0,5081%
$(4/5, 1/5, 2/3, 1/3)$	0,628646 (0,062140)	0,309193 (0,029669)	0,6923%
$(1/2, 1/2, 1/5, 4/5)$	0,623549 (0,046935)	0,640435 (0,054770)	−2,7081%
$(1/3, 2/3, 1/5, 4/5)$	0,611819 (0,032058)	0,608729 (0,024599)	0,4890%
$(1/5, 4/5, 1/5, 4/5)$	0,621266 (0,045098)	0,633420 (0,050358)	−1,9564%
$(2/3, 1/3, 1/5, 4/5)$	0,620594 (0,049867)	0,622590 (0,038168)	−0,3216%
$(4/5, 1/5, 1/5, 4/5)$	0,617798 (0,039308)	0,627715 (0,033529)	−1,6052%
$(1/2, 1/2, 4/5, 1/5)$	0,624987 (0,048528)	0,639888 (0,041781)	−2,3841%
$(1/3, 2/3, 4/5, 1/5)$	0,621978 (0,049692)	0,618590 (0,033113)	0,5446%
$(1/5, 4/5, 4/5, 1/5)$	0,619748 (0,050713)	0,624203 (0,036910)	−0,7189%
$(2/3, 1/3, 4/5, 1/5)$	0,632830 (0,063521)	0,618922 (0,027507)	2,1977%
$(4/5, 1/5, 4/5, 1/5)$	0,635348 (0,060306)	0,652660 (0,064186)	−2,7248%

d.p. - desvio padrão

Já na avaliação dos tempos de percurso. Existem oscilações, mas nenhuma alteração de grande impacto. Em geral, é possível verificar que o algoritmo SA foi capaz de aumentar a produtividade sem pedras nos tempos de percurso. Visto de outra forma, o trabalho foi melhor dividido entre os servidores. As configurações prévias exigiam

maior esforço de alguns servidores enquanto outros servidores permaneciam por mais tempo ociosos.

A Tabela 13 mostra os resultados comparativos entre as soluções iniciais e as soluções fornecidas pelo algoritmo SA para os ganhos associados à redução no hipervolume associado ao Pareto solução.

Tabela 13 – Melhorias obtidas em hipervolume para redes de filas mista.

roteamento	hipervolume inicial	hipervolume SA	percentual de redução
$(1/2, 1/2, 1/2, 1/2)$	0,06808295	0,00689435	89,8736%
$(1/3, 2/3, 1/2, 1/2)$	0,04569722	0,02087051	54,3287%
$(1/5, 4/5, 1/2, 1/2)$	0,05125041	0,03111546	39,2874%
$(2/3, 1/3, 1/2, 1/2)$	0,04823581	0,00086638	98,2039%
$(4/5, 1/5, 1/2, 1/2)$	0,04767598	0,01236458	74,0654%
$(1/2, 1/2, 1/3, 2/3)$	0,06783317	0,03477200	48,7389%
$(1/3, 2/3, 1/3, 2/3)$	0,10865310	0,01469956	86,4711%
$(1/5, 4/5, 1/3, 2/3)$	0,07025408	0,03593609	48,8484%
$(2/3, 1/3, 1/3, 2/3)$	0,17983280	0,02216912	87,6724%
$(4/5, 1/5, 1/3, 2/3)$	0,05784911	0,00594585	89,7218%
$(1/2, 1/2, 2/3, 1/3)$	0,04863190	0,02308481	52,5316%
$(1/3, 2/3, 2/3, 1/3)$	0,04068089	0,02447268	39,8423%
$(1/3, 2/3, 2/3, 1/3)$	0,04637056	0,03822689	17,5622%
$(2/3, 1/3, 2/3, 1/3)$	0,04890713	0,02097418	57,1143%
$(4/5, 1/5, 2/3, 1/3)$	0,11812960	0,03235112	57,1143%
$(1/2, 1/2, 1/5, 4/5)$	0,04229546	0,02043743	51,6794%
$(1/3, 2/3, 1/5, 4/5)$	0,02075706	0,00382738	81,5611%
$(1/5, 4/5, 1/5, 4/5)$	0,04924105	0,03132186	36,3908%
$(2/3, 1/3, 1/5, 4/5)$	0,11745670	0,03616882	69,2067%
$(4/5, 1/5, 1/5, 4/5)$	0,04974962	0,01263840	74,5960%
$(1/2, 1/2, 4/5, 1/5)$	0,05985970	0,03124937	47,7957%
$(1/3, 2/3, 4/5, 1/5)$	0,04309129	0,02300599	46,6110%
$(1/5, 4/5, 4/5, 1/5)$	0,05042639	0,01945427	61,4205%
$(2/3, 1/3, 4/5, 1/5)$	0,07047762	0,02501410	64,5077%
$(4/5, 1/5, 4/5, 1/5)$	0,08454299	0,04661423	44,8633%

Os ganhos em medida de produtividade refletem claramente na resposta do hipervolume verificado. Para todas as configurações, são apresentadas reduções de hipervolume do Pareto solução SA quando comparado ao Pareto solução inicial. Essa constatação ilustra a contribuição do algoritmo SA em vasculhar soluções eficientes para o problema de alocação de servidores em redes de filas.

Dado o volume de configurações do vetor de roteamentos investigado para a rede de filas mista. Uma representação gráfica tende a auxiliar na análise de desempenho das soluções. A Figura 89 apresenta o ganho percentual médio das soluções SA. Sendo

o eixo y, o ganho percentual médio, e o eixo x, todas as 25 configurações testadas.

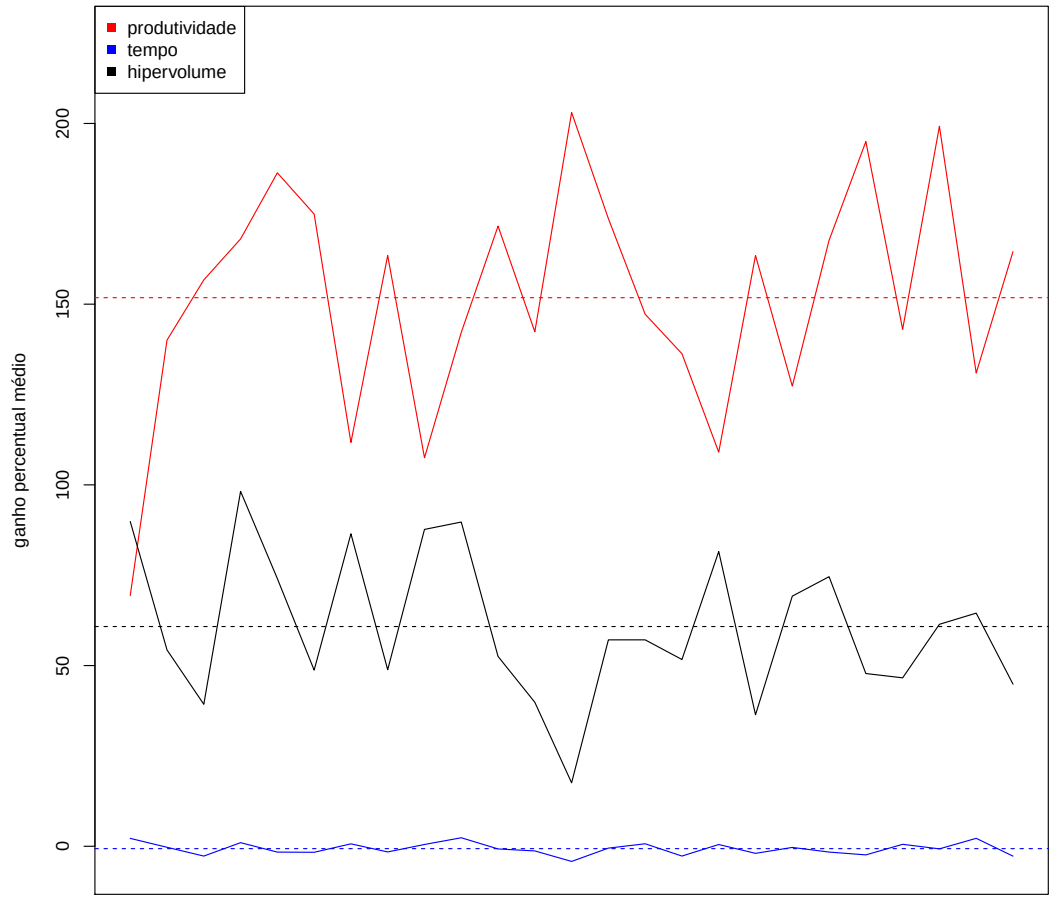

Figura 89 – Desempenho conjunto das soluções SA nas variações da rede de filas mista.

O gráfico apresenta a melhoria obtida em produtividade (em pontilhado o ganho médio entre todos os roteamentos), o ganho obtido é bastante significativo. O gráfico também mostra as alterações obtidas em tempo de percurso da rede de filas (em pontilhado o ganho médio entre todos os roteamentos), os ganhos ou perdas são apenas marginais, em outras palavras, as reconfigurações de servidores não reduziram substancialmente o tempo de percurso dos usuários da rede. Por fim, o gráfico também mostra o ganho em hipervolume na comparação entre o Pareto inicial e o Pareto SA (em pontilhado o ganho médio entre todos os roteamentos). Este resultado confirma a capacidade do algoritmo SA apresentar soluções promissoras para o problema em estudo.

5 Considerações Finais

Este estudo propôs uma investigação que trata da formulação e análise do *Problema de Alocação de Servidores* para redes de filas markovianas sem limitação de capacidade. Nesse trabalho todos os servidores possuem a mesma capacidade de atendimento. Tal abordagem, além de agregar informação para o meio acadêmico, é de grande valia para o meio industrial. Esses servidores para a indústria da manufatura podem ser considerados como máquinas idênticas, ou seja, máquinas de igual capacidade e que executam a mesma função. Suponha, por exemplo, a industria calçadista, produzindo diversos calçados em linhas produtivas distintas, com demandas distintas, mas produzidos pelo mesmo tipo de maquinário. A possibilidade de um algoritmo que aloca de forma ótima suas máquinas, que seja capaz de satisfazer a demanda dos clientes, ao invés de uma disposição completamente aleatória das máquinas, pode fornecer a companhia uma vantagem econômica para diminuir custos totais de produção.

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

O problema de otimização abordado é semelhante ao clássico problema de otimização combinatória da mochila, que consiste basicamente em: preencher uma mochila com o maior número de itens valiosos, porém, de modo a respeitar o espaço da mochila e o peso total que a pessoa que carrega a mochila consegue carregar. Existem diversos itens, com pesos e valores diferentes. Qual a melhor combinação que maximiza o valor dos itens, com menor peso possível? A heurística utilizada para lidar com o problema foi produzida com os conceitos do *Simulated Annealing*, focado especificamente em melhorar o desempenho do sistema por meio do aumento da produtividade e redução do tempo geral esperado na rede de filas.

Para testar o método proposto, foram experimentadas quatro topologias de rede distintas: eedes de filas em série, com fusão, com divisão e rede de filas mista, que envolveu diferentes topologias. Para cada rede, dentro de suas limitações, foram variadas as especificações pontuais para obter resultados bastante abrangentes. Para as redes com fusão, diferentes valores da taxa de entrada λ foram utilizados. Já para as redes com divisão e mista, diferentes valores para as coordenadas do vetor de probabilidades de roteamentos. Os resultados apresentados numericamente, e por meio dos gráficos, confirmam a eficácia do algoritmo SA implementado. O método proposto conseguiu fornecer soluções eficientes para o problema SAP apresentado.

Observa-se que, dos efeitos decorrentes das topologias de rede em estudo, a alocação de recursos em servidores apresenta resultados com algum padrão específico para os diversos casos investigados.

5.1 Propostas de Continuidade

Como trabalhos futuros, podem ser citados os seguintes:

- Verificação da homogeneidade das soluções com oscilações nas taxas de serviço μ_j;
- Investigação das soluções em redes de filas com atendimentos gerais e independentes, ou seja, em redes do tipo M/G/1/k, na notação de Kendall [3];
- 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. Outras investigações com filas de estruturas distintas, tais como filas markovianas multi-servidoras finitas, M/M/c/k. Estes são apenas alguns tópicos para trabalhos futuros nesta instigante linha de pesquisa.

Referências

[1] MacGregor Smith, James e Frederico Cruz: *The buffer allocation problem for general finite buffer queueing networks*. IIE Transactions, 37(4):343–365, 2005. Citado 3 vezes nas páginas 15, 1 e 18.

[2] Yang, Xin She: *Engineering optimization: An introduction with metaheuristic applications*. Wiley Publishing, 1st edição, 2010, ISBN 0470582464, 9780470582466. Citado 4 vezes nas páginas 15, 2, 8 e 9.

[3] Kendall, David G.: *Stochastic processes occurring in the theory of queues and their analysis by the method of embedded Markov chains*. Annals Mathematical Statistics, 24:338–354, 1953. Citado 4 vezes nas páginas 21, 3, 7 e 72.

[4] Yechiali, Ury e Pinhas Naor: *Queuing problems with heterogeneous arrivals and service*. Operations Research, 19(3):722–734, 1971. Citado 2 vezes nas páginas 1 e 12.

[5] Boxma, Onno Johan e I. A. Kurkova: *The $M/G/1$ queue with two service speeds*. Advances in Applied Probability, 33(2):520–540, 2001. Citado 2 vezes nas páginas 1 e 12.

[6] Baykal-Gursoy, Melike e Weihua Xiao: *Stochastic decomposition in $M/M/\infty$ queues with Markov modulated service rates*. Queueing Systems, 48(1):75–88, 2004. Citado 2 vezes nas páginas 1 e 12.

[7] Baykal-Gürsoy, Melike, Weihua Xiao e Kaan Ozbay: *Modeling traffic flow interrupted by incidents*. European Journal of Operational Research, 195(1):127–138, 2009. Citado 2 vezes nas páginas 1 e 12.

[8] Gao, Shan e Zaiming Liu: *An $M/G/1$ queue with single working vacation and vacation interruption under Bernoulli schedule*. Applied Mathematical Modelling, 37(3):1564–1579, 2013. Citado 2 vezes nas páginas 1 e 12.

[9] Heidemann, Dirk: *A queueing theory model of nonstationary traffic flow*. Transportation Science, 35(4):405–412, 2001. Citado 2 vezes nas páginas 1 e 13.

[10] Vandaele, Nico, Tom Van Woensel e Aviel Verbruggen: *A queueing based traffic flow model*. Transportation Research Part D: Transport and Environment, 5(2):121–135, 2000. Citado 2 vezes nas páginas 1 e 13.

[11] Van Woensel, Tom e Nico Vandaele: *Empirical validation of a queueing approach to uninterrupted traffic flows*. 4OR-A Quarterly Journal of Operations Research, 4(1):59–72, 2006. Citado 2 vezes nas páginas 1 e 13.

[12] Van Woensel, Tom, Bart Wuyts e Nico Vandaele: *Validating state-dependent queueing models for uninterrupted traffic flows using simulation*. 4OR-A Quarterly Journal of Operations Research, 4(2):159–174, 2006. Citado 2 vezes nas páginas 1 e 13.

[13] Van Woensel, Tom e Frederico Cruz: *A stochastic approach to traffic congestion costs*. Computers & Operations Research, 36(6):1731–1739, 2009. Citado 2 vezes nas páginas 1 e 13.

[14] Hanukov, Gabi, Tal Avinadav, Tatyana Chernonog e Uri Yechiali: *A multi-server queueing-inventory system with stock-dependent demand*. IFAC-PapersOnLine, 52(13):671–676, 2019. Citado 2 vezes nas páginas 1 e 13.

[15] Wang, Yu Bo, Cheng Qian e Jin De Cao: *Optimized M/M/c model and simulation for bank queuing system*. Em *2010 IEEE International Conference on Software Engineering and Service Sciences*, páginas 474–477. IEEE, 2010. Citado 2 vezes nas páginas 1 e 13.

[16] Liu, Y., Jian Cao, X. Cao e Y. Zhang: *Optimization of design scheme for toll plaza based on M/M/c queuing theory and cellular automata simulation algorithm*. Modern Applied Science, 11(7):1, 2017. Citado 2 vezes nas páginas 1 e 13.

[17] Khodemani-Yazdi, Melahat, Reza Tavakkoli-Moghaddam, Mahdi Bashiri e Yaser Rahimi: *Solving a new bi-objective hierarchical hub location problem with an M/M/c queuing framework*. Engineering Applications of Artificial Intelligence, 78:53–70, 2019. Citado 2 vezes nas páginas 1 e 13.

[18] Goodarzi, Asefeh Hasani, Eleen Diabat, Armin Jabbarzadeh e Marc Paquet: *An M/M/c queue model for vehicle routing problem in multi-door cross-docking environments*. Computers & Operations Research, página 105513, 2021. Citado 2 vezes nas páginas 1 e 13.

[19] Hillier, Frederick e Kut So: *On the optimal design of tandem queueing systems with finite buffers*. Queueing Systems, 21(3):245–266, 1995. Citado 2 vezes nas páginas 1 e 13.

[20] Spinellis, Diomidis, Chrissoleon Papadopoulos e James MacGregor Smith: *Large production line optimization using simulated annealing*. International journal of production research, 38(3):509–541, 2000. Citado 2 vezes nas páginas 1 e 13.

[21] Kerbache, Laoucine e James MacGregor Smith: *The generalized expansion method for open finite queueing networks*. European Journal of Operational Research, 32(3):448–461, 1987. Citado 2 vezes nas páginas 1 e 13.

[22] Hanukov, Gabi, Tal Avinadav, Tatyana Chernonog, Uriel Spiegel e Uri Yechiali: *A queueing system with decomposed service and inventoried preliminary services*. Applied Mathematical Modelling, 47:276–293, 2017. Citado 2 vezes nas páginas 1 e 13.

[23] Hanukov, Gabi, Tal Avinadav, Tatyana Chernonog, Uriel Spiegel e Uri Yechiali: *Improving efficiency in service systems by performing and storing "preliminary services"*. International Journal of Production Economics, 197:174–185, 2018. Citado 2 vezes nas páginas 1 e 13.

[24] Cheng, Xueli, Linchao An e Zhenhua Zhang: *Integer Encoding Genetic Algorithm for Optimizing Redundancy Allocation of Series-parallel Systems*. Journal of Engineering Science & Technology Review, 12(1), 2019. Citado 2 vezes nas páginas 1 e 13.

[25] Liu, Xuemei, Mingliang Lei, Qingfei Zeng e Aiping Li: *Integrated optimization of mixed-model assembly line balancing and buffer allocation based on operation time complexity*. Procedia Cirp, 81:1040–1045, 2019. Citado 2 vezes nas páginas 1 e 13.

[26] Demir, Leyla e Mehmet Ulaş Koyuncuoğlu: *The impact of the optimal buffer configuration on production line efficiency: A VNS-based solution approach*. Expert Systems with Applications, 172:114631, 2021. Citado 2 vezes nas páginas 1 e 13.

[27] de Souza, Gabriel Lima, Anderson Ribeiro Duarte, Gladston Moreira e Frederico Cruz: *A novel formulation for multi-objective optimization of general finite single-server queueing networks*. Em *2020 IEEE Congress on Evolutionary Computation (CEC)*, páginas 1–8, July 2020. https://doi.org/10.1109/CEC48606.2020.9185827. Citado 2 vezes nas páginas 1 e 13.

[28] Zhang, Jianchun, Lei Li e Zhiwei Chen: *Strength–redundancy allocation problem using artificial bee colony algorithm for multi-state systems*. Reliability Engineering & System Safety, 209:107494, 2021. Citado 2 vezes nas páginas 1 e 13.

[29] Duarte, Anderson Ribeiro: *The Server Allocation Problem for markovian queueing networks*. International Journal of Services and Operations Management, (to appear), 2022. http://dx.doi.org/10.1504/IJSOM.2022.10047177. Citado 6 vezes nas páginas 2, 4, 5, 12, 13, and 18.

[30] Shanthikumar, J. George e David D. Yao: *Optimal server allocation in a system of multi-server stations*. Management Science, 33(9):1173–1180, 1987. Citado na página 4.

[31] MacGregor Smith, James, Frederico Cruz e Tom van Woensel: *Optimal server allocation in general, finite, multi-server queueing networks*. Applied Stochastic Models in Business & Industry, 26(6):705–736, 2010. Citado na página 4.

[32] Papadopoulos, Chrissoleon T, Jingshan Li e Michael EJ O'Kelly: *A classification and review of timed Markov models of manufacturing systems*. Computers & Industrial Engineering, 128:219–244, 2019. Citado na página 4.

[33] Costa, Antonio, Erica Pastore e Nicla Frigerio: *The Server Allocation Problem with non-identical machines: a meta-heuristic approach*. Computers & Industrial Engineering, página 107687, 2021. Citado na página 4.

[34] MacGregor Smith, James, Frederico Cruz e Tom van Woensel: *Topological network design of general, finite, multi-server queueing networks*. European Journal of Operational Research, 201(2):427–441, 2010. Citado 2 vezes nas páginas 4 e 8.

[35] de Souza, Gabriel Lima: *Uma Nova Formulação para Otimização Multi-objetivo em Redes de Filas Finitas Gerais e com Único Servidor*. Tese de Mestrado, Universidade Federal de Ouro Preto, 2020. Citado na página 7.

[36] Alves, F. S. Q., Hani Camille Yehia, L. A. C. Pedrosa, Frederico Cruz e Laoucine Kerbache: *Upper bounds on performance measures of heterogeneous $M/M/c$ queues*. Mathematical Problems in Engineering, 2011(Article ID 702834):18 pages, 2011. Citado na página 8.

[37] De Bruin, Arnoud M., Albert C. Van Rossum, Marieke C. Visser e Ger M. Koole: *Modeling the emergency cardiac in-patient flow: an application of queuing theory*. Health Care Management Science, 10(2):125–137, 2007. Citado na página 8.

[38] Cruz, Frederico, James MacGregor Smith e R. O. Medeiros: *An $M/G/C/C$ state dependent network simulation model*. Computers & Operations Research, 32(4):919–941, 2005. Citado na página 8.

[39] Cruz, Frederico, Anderson Ribeiro Duarte e Tom van Woensel: *Buffer allocation in general single-server queueing networks*. Computers & Operations Research, 35(11):3581–3598, 2008. Citado na página 8.

[40] Cruz, Frederico, Tom van Woensel, James MacGregor Smith e Kris Lieckens: *On the system optimum of traffic assignment in $M/G/c/c$ state-dependent queueing networks*. European Journal of Operational Research, 201(1):183–193, 2010. Citado na página 8.

[41] Ahmed, Nasir U. e Xuan Hui Ouyang: *Suboptimal RED feedback control for buffered TCP flow dynamics in computer network*. Mathematical Problems in Engineering, 2007(Article ID 54683):17 pages, 2007, ISSN 1024-123X. Citado na página 8.

[42] Chen, Jianyong, Cunying Hu e Zen Ji: *An improved ARED algorithm for congestion control of network transmission*. Mathematical Problems in Engineering, 2010(Article ID 329035):17 pages, 2010, ISSN 1024-123X. Citado na página 8.

[43] Inzillo, V., F. De Rango e A. A. Quintana: *A self clocked fair queuing MAC approach limiting deafness and round robin issues in directional MANET*. Em *2019 Wireless Days (WD)*, páginas 1–6. IEEE, 2019. Citado na página 8.

[44] Chaudhuri, K., A. Kothari, R. Pendavingh, R. Swaminathan, R. Tarjan e Y. Zhou: *Server allocation algorithms for tiered systems*. Algorithmica, 48(2):129–146, 2007. Citado na página 8.

[45] Menascé, Daniel: *QoS issues in web services*. IEEE Internet Computing, 6(6):72–75, 2002. Citado na página 8.

[46] Osorio, Carolina e Michel Bierlaire: *An analytic finite capacity queueing network model capturing the propagation of congestion and blocking*. European Journal of Operational Research, 196(3):996–1007, 2009. Citado na página 8.

[47] Dimitriou, Ioannis e Christos Langaris: *A repairable queueing model with two-phase service, start-up times and retrial customers*. Computers and Operations Research, 37(7):1181–1190, 2010. Citado na página 8.

[48] Pareto, Vilfredo: *Cours d'économie politique*, volume 1. Librairie Droz, 1896. Citado na página 10.

[49] Zitzler, E. e L. Thiele: *Multiobjective optimization using evolutionary algorithms—a comparative case study*. Em *International conference on parallel problem solving from nature*, páginas 292–301. Springer, 1998. Citado na página 11.

[50] Kirkpatrick, Scott, Charles Daniel Gelatt Jr. e Mario Vecchi: *Optimization by simulated annealing*. science, 220(4598):671–680, 1983. Citado 2 vezes nas páginas 14 e 15.

[51] Černỳ, Vladimír: *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*. Journal of optimization theory and applications, 45(1):41–51, 1985. Citado 2 vezes nas páginas 14 e 15.

[52] Metropolis, Nicholas, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller e Edward Teller: *Simulated annealing*. Journal of Chemical Physics, 21(161-162):1087–1092, 1953. Citado na página 15.

[53] Kellerer, Hans, Ulrich Pferschy e David Pisinger: *Knapsack problems*. Springer, Berlin, Heidelberg, 1ª edição, 2004, ISBN 978-3-540-40286-2. Citado na página 19.

[54] Little, John Dutton Conant: *A proof for the queuing formula: $L = \lambda W$*. Operations Research, 9(3):383–387, 1961. Citado na página 21.

[55] R Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. https://www.R-project.org/. Citado na página 25.