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

Uma Abordagem por Enxame de Partículas para o Problema de Alocação de Servidores para Redes de Filas Markovianas

João Paulo Damas de Sena

Ouro Preto-MG
2022

João Paulo Damas de Sena

Uma Abordagem por Enxame de Partículas para 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: Anderson Ribeiro Duarte

Ouro Preto

2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S474u Sena, João Paulo Damas De.
Uma abordagem por enxame de partículas para o problema de alocação de servidores para redes de filas Markovianas. [manuscrito] / João Paulo Damas De Sena. - 2022.
38 f.: il.: color., gráf..

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

1. Enxame de Particulas. 2. Rede de Filas. 3. Otimização. 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

João Paulo Damas de Sena

Uma abordagem por enxame de partículas para 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 03 de novembro de 2022

Membros da banca

Dr. Anderson Ribeiro Duarte - Orientador (Universidade Federal de Ouro Preto)
Dr. Helgem de Souza Martins - Membro (Universidade Federal de Ouro Preto)
Ms. Gabriel Lima de Souza - Membro (Universidade Federal de Ouro Preto)

Professor Dr. Anderson Ribeiro Duarte, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 03/11/2022

Documento assinado eletronicamente por **Anderson Ribeiro Duarte**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 03/11/2022, às 16:55, 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 **0421670** e o código CRC **819DB60A**.

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

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

FOLHA DE APROVAÇÃO

João Paulo Damas de Sena

Uma abordagem por enxame de partículas para 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 03 de novembro de 2022

Membros da banca

Dr. Anderson Ribeiro Duarte - Orientador (Universidade Federal de Ouro Preto)
Dr. Helgem de Souza Martins - Membro (Universidade Federal de Ouro Preto)
Ms. Gabriel Lima de Souza - Membro (Universidade Federal de Ouro Preto)

Professor Dr. Anderson Ribeiro Duarte, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 03/11/2022

Documento assinado eletronicamente por **Anderson Ribeiro Duarte**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 03/11/2022, às 16:55, 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 **0421670** e o código CRC **819DB60A**.

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

Agradecimentos

Agradeço todo mundo que um dia me ajudou levantar nos momentos em que eu me encontrava sem chão. Um agradecimento em especial a minha vó, que em momentos mais dificeis, ela nunca deixou de ajudar idenpendente da sua condição de saúde. Agradeço meus amigos, que em dias mais sofridos, sempre levantaram meu astral e foram exemplos de como seguir em frente, independente da dificuldade, afinal, ela sempre vai existir e o que determina a nossa força é como decidimos nos levantar nos momento em qua nossas pernas mal conseguem se erguer. Agredeço tambem aos meus colegas de trabalho, que em momentos de decisões mais complicadas, me deram conselhos, e me fizeram chegar onde estou hoje. Agradeço meus professores, pelo incentivo e promover o meu desejo de continuar na área de Estatística, abrindo minha mente, e mostrando como essa área pode ser cativante.

Resumo

Diversos processos produtivos demandam investigação científica. Existem altos valores financeiros envolvidos, tanto na execução do processo quanto nos resultados decorrentes do atendimento da demanda dos clientes. Em geral, uma das premissas é garantir o tempo de execução com ganho real de produtividade. O exercício projetivo destes processos sempre passa pela alocação dos recursos envolvidos. Dentre os diversos problemas de interesse, surge um problema bastante significativo, o problema de como alocar recursos (servidores) para dimensionar sistemas de filas de alta eficiência. Redes de filas multi-servidores, com topologias acíclicas arbitrárias para chegadas markovianas e serviços também markovianos são investigados neste estudo. O algoritmo por enxame de partículas (o clássico PSO - *Particle Swarm Optimization*) é utilizado como abordagem para alocação ótima de servidores em diversas topologias de redes de filas (série, fusão e divisão). A metodologia utiliza uma estratégia heurística multiobjetivo, o desempenho da utilização do servidor é considerado por meio de uma métrica de produtividade. A produtividade é maximizada simultaneamente com a minimização do tempo geral esperado dos clientes para percorrer a rede de filas. Diversos resultados de experimentos computacionais mostram a eficácia da metodologia.

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

Abstract

Several production processes demand a scientific investigation. High financial values are involved, both in the execution of the process and in the results arising from meeting customer demand. In general, one of the premises is to ensure the execution time with real productivity gains. The projective time of these processes always involves allocating the resources involved. Among the various problems of interest, a very significant problem arises: allocating resources (servers) to scale high-efficiency queueing systems. Multi-server queuing networks with arbitrary acyclic topologies for Markov arrivals and services are investigated in this study. The classical PSO (particle swarm algorithm) is used as an approach for the optimal allocation of servers in different topologies of queueing networks (series, fusion, and division). The methodology uses a multi-objective heuristic strategy. The server utilization performance is considered through a productivity metric. Productivity is maximized while minimizing the overall time expected of customers to course the queueing network. Several results of computational experiments show the effectiveness of the methodology.

Keywords: Optimization, Particle Swarm, Queueing Network, Productivity, Allocating resources.

Lista de ilustrações

Sumário

1 Introdução

A investigação acerca da teoria das filas remete para um conjunto de situações sempre presentes na vida de todos os indivíduos. De uma simples visita ao comércio, uma passagem pelo trânsito, ou até mesmo em situações de atendimento virtual, a teoria das filas sempre está presente. Particularmente diversos processos produtivos se enquandram nesta situação. Todo processo que deve seguir algum fluxo específico até chegar em seu destino final, pode ser considerado uma fila ou uma rede de filas. A compreensão do comportamente dessas filas com interesse em melhorar seu funcionamento é um desafio complexo, bastante difundido na literatura, em suma, um objeto de estudo desafiador. Diversos pesquisadores já se propuseram a solucionar problemas dessa natureza com diferentes abordagens.

Um sistema de filas é determinado pelo padrão de chegada dos usuários, o padrão de atendimento desses usuários, o volume de servidores, entre outros parâmetros. Uma rede de filas pode ser representada com um grafo direcionado, em que as arestas interconectam as filas que são representadas pelos vértices do grafo. As entidades que percorrem a rede, transitam entre as filas para receber algum tipo de serviço. A Figura 1 exemplifica uma rede de filas que possui seis nós e três possíveis rotas de acordo com o vetor de roteamentos (p_1, p_2, p_3, p_4).

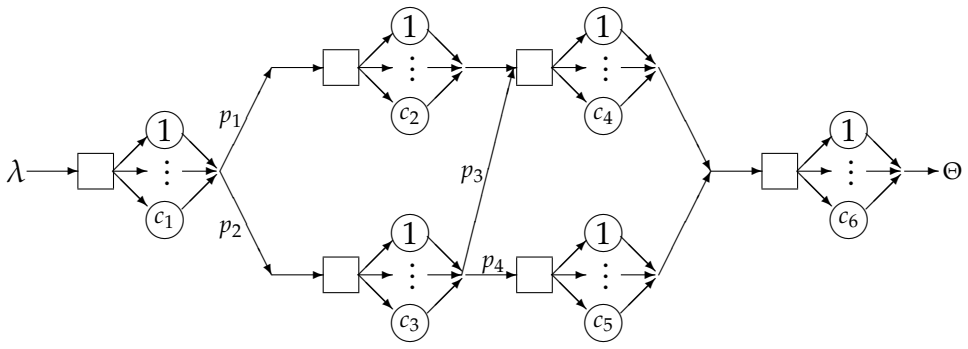

Figura 1 – Um exemplo de uma rede complexa de filas adapatado de MacGregor Smith e Cruz (2005) [1].

1.1 Motivação

Um problema de grande interesse prático é avaliar métricas que dizem respeito à eficiência do funcionamento das redes de filas. A literatura apresenta diversas métricas adequadas para aferir níveis de eficácia da operação. Particularmente, a proposta de avaliar redes de filas multi-servidoras é de fato instigante. O próposito central está na

tarefa de obter uma melhor estratégia para alocar esses servidores em cada fila da rede, de forma que a performace do sistema seja otimizada.

Uma versão específica do problema de alocação de servidores em redes de filas foi apresentada por Duarte (2022) [2]. Neste estudo, uma a abordagem que aplica o clássico algoritmo de otimização *Simulated Annealing* (SA) foi utilizada. Agora, uma investigação através de outra abordagem algorítmica será apresentada.

O problema de alocação de servidores, de uma forma resumida trata de uma relação de conflito entre objetivos. A performance do sistema tende a melhorar com o acréscimo de mais servidores, o que é vantajoso. Porém, o acréscimo de servidores acarreta em aumento de custos, o que não é desejável. Problemas de otimização com este tipo de conflito remetem a uma abordagem multi-objetivo.

O algoritmo de otimização aplicado nessa nova proposição é o *Particle Swarm Optimization* (PSO), um algoritmo que faz analogia com a natureza várias espécies que optam por viver em conjunto e tirar proveito da sociabilidade. Um exemplo clássico é um enxame de abelhas. Porém, a metáfora pode ser estendida a outros coletivos, como uma colônia de formigas, uma revoada de pássaros.

O algoritmo PSO foi proposto por Kennedy e Eberhart em 1995 [3]. É um algoritmo com motivação no movimento de várias espécies de animais. O algoritmo busca reproduzir esse movimento de enxames de vários animais na busca pelo seu objetivo. Essas atividades ocorrem sempre na busca por melhores condições para a espécie. Esse movimento é guiado pelo líder, mas também conta com a colaboração de cada um dos membros do enxame, a fim de guiar todo o enxame para uma melhor localização de bem estar da espécie.

O algoritmo PSO reproduz os movimentos de enxame por meio de equações matemáticas e faz com que um conjunto de partículas (soluções candidatas) se movimente em busca de um posicionamento ótimo. De maneira geral, as partículas são posicionadas inicialmente de maneira aleatória no espaço de soluções factíveis. A cada iteração, sucessivos movimentos são realizados para esses pontos de acordo com as informações contidas nas próprias partículas, isso de acordo com as funções objetivo. Para movimentar cada ponto, basicamente calcula-se sua direção e o tamanho de cada movimento. Essa operação é tratada como a velocidade da uma partícula desse enxame. Porsteriormente, na seção 3.3 a implementação para o PSO utilizada para este estudo específico é detalhada.

1.2 Objetivos

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

1.2.1 Objetivos Gerais

i. discutir os aspectos de utilização da meta-heurística PSO;

i. abordar formulações matemáticas para problemas de otimização.

1.2.2 Objetivos Específicos

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

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

iii. descrever a inovadora formulação do problema de otimização multiobjetivo de alocação de servidores SAP apresentada por Duarte (2022) [2], uma formulação que garante alta produtividade dos servidores, baixo tempo de clientes na rede de filas restrito à um número total fixo de servidores.

iv. 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 através das proposições presentes no estudo.

2 Fundamentação Teórica

Como mencionado no capítulo introdutório, este estudo aborda redes de filas sem limitações para a quantidade de áreas de circulação (*buffers*). A literatura apresenta diversos estudos e aplicabilidades em modelos de filas deste formato. Uma revisão de literatura com problemas desse estrutura será apresentada.

Vandaele, van Woensel e Verbruggen (2000) [4] abordam o problema de fluxo de tráfego intermitente. Procedimentos de contagem de tráfego para estimação de parâmetros em modelos analíticos de filas são utilizados. O objetivo é a investigação sobre o impacto de alguns parâmetros cruciais na estratégia de modelagem para o problema real de tráfego da autoestrada E19, que liga St-Job a Merksem na Antuérpia, Bélgica. O estudo é executado através do modelo de filas $G/G/1$.

Novamente o problema de fluxo de tráfego intermitente é abordado por van Woensel e Vandael (2006) [5]. São utilizados dados empíricos sobre velocidade e fluxo em comparação com velocidades provenientes de modelos teóricos de filas. A investigação constata que o fluxo de tráfego em uma rodovia durante períodos não congestionados é melhor descrito através de um modelo de filas $M/G/1$. Ao passo que durante o congestionamento, os modelos $GI/G/z$ de filas dependentes do estado são mais realistas.

van Woensel e Cruz (2009) [6] discutem a estocasticidade inerente ao comportamento do tráfego e seus custos atrelados aos impostos rodoviários. Os custos de tráfego atuais não refletem em sua exatidão aos custos decorrentes dos congestionamentos. Para estabelecer adequadamente esses custos de tráfego, os gestores do setor público precisam de modelos adequados. Modelos $M/M/1$ e $M/G/1$ são abordados por meio de uma análise de sensibilidade.

Gao e Liu (2013) [7] apresentam um estudo para filas $M/G/1$ com regime de férias de trabalho e possíveis interrupções de férias sob uma demanda de Bernoulli. O entrada de servidores em regime de férias acarreta em maior lentidão no atendimento. O estudo obtem a distribuição estacionária para o comprimento da fila de espera dos clientes.

Hanukov, Avinadav, Chernono, Spiegel e Yechiali (2018) [8] analisam a eficiência de sistemas de atendimento mono-servidores com pré-atendimento. A execução prévia do serviço e o armazenamento destas tarefa antes da presença física do cliente ocorre em paralelo com outros atendimentos presenciais. O objetivo é aumento do fluxo de atendimentos reduzindo a ociosidade. São utilizados modelos de filas de chegada e atendimento markoviandos sem limitação de *buffers*.

Hanukov, Avinadav, Chernonog, Yechiali (2019) [9] apresenta um sistema de filas de dois servidores ($M/M/2$) no qual os servidores ociosos podem produzir e armazenar tarefas com intuito de reduzir o tempo de atendimento e assim possibilitar que o sistema funcione bem para taxas de chegadas mais elevadas. É apresentado um estudo acerca de filas em redes de *fast-food*. Usualmente o sistema funciona com a chegada do cliente seguida do atendimento. Após este ponto, deve ser iniciada a preparação do pedido, este processo é muitas vezes bloqueado quando ocorrem taxas elevadas de chegadas de clientes. O interesse central é minimizar a probabilidade de bloqueio de filas, com a intenção de melhorar o atendimento e ampliar a satisfação dos clientes e sua fidelização. O estudo apresenta uma fila auxiliar que funciona como um pré-atendimento capaz de iniciar a etapa de preparação do pedido previamente e reduzir o intertravamento entre filas. Diversos modelos de atendimento aos clientes possuem similaridades com este estudo e podem ampliar sua aplicabilidade.

Alves, Yehia, Pedrosa, Cruz e Kerbache (2011) [10] avaliam as cotas máxima de performancem filas $M/M/c$ com servidores heterogêneos. São avaliadas três estratégias de alocação: alocar o servidor mais rápido dispovível primeiro (FSF), alocar aleatóriamente o próximo servidor disponível (RCS), e alocar o servidor mais lento dispovível primeiro (SSF).

Liu, Cao, Cao, e Zhang (2017) [11] investigam o fluxo de tráfego em praças de pedágio através de modelos de filas $M/M/c$. Um estudo de caso em dados reais é apresentado a praça de pedágio da rodovia Asbury Park Toll Plaza, em Nova Jersey/EUA.

Khodemani-Yazdi, Tavakkoli-Moghaddam, Bashiri e Rahimi (2019) [12] discutem o problema de localização estratégica de *hubs* de conexão para redistribuição de mercadorias. O objetivo e minimizar os custos maximo de instalações de *hubs* de transportes através de modelos em filas Markovianas $M/M/C$ e $M/M/1$. Uma proposta de otimização baseada em teoria dos jogos nominada *New game theory variable neighborhood fuzzy invasive weed optimization* (GVIWO) é utilizada e comparada com os clássicos algoritmos Genético (NSGA-II) e o *Simulated Annealing* híbrido (HSA).

Goodarzi, Diabat, Jabbarzadeh e Paquet, (2022) [13] apresentam uma modelagem via rede de filas $M/M/c$ para o problema de *Cross Docking*. O *Cross Docking* é um sistema utilizado em empresas de distribuição logística. Trata-se da estratégia de chegada e saída de produtos. Os pontos de carregamento e descarregamento funcionam como servidores, a ociosidade é por vezes tão maléfica quanto o atraso. Os autores desenvolvem um *software* de otimização denominado (GAMS), baseado em otimização através do algoritmo genético (GA). Exemplos com busca pela alocação ótima em problemas de larga escala são apresentados.

Particularmente, o problema foi abordado para o aumento de produtividade em

sistemas de manufatura por Duarte (2022) [2]. A alocação de servidores em redes de filas markovianas de servidor único e sem limitação de *buffers* é investigada na busca por projetar sistemas de filas mais eficientes. O estudo apresenta uma abordagem de alocação ótima por meio do clássico algoritmo Simulated Annealing. A metodologia utiliza uma estratégia multi-objetivo em que o desempenho da utilização do servidor é maximizado simultaneamente com a minimização do tempo geral esperado dos clientes na rede de filas.

Essa revisão aborda problemas em redes de filas sem limitação de *buffers*. Porém, resultados para outras estruturas de filas podem fornecer técnicas de otimização que se ajustam bem ao problema em estudo. Souza, Duarte, Moreira e Cruz (2020) [14] apresentam uma estratégia de otimização dupla, um processamento inicial é execcutado através do algoritmo genético (NSGA-II). Em seguida uma estratégia de pós-processamento através de um algoritmo de enxame de partículas (MOPSO). O estudo aborda uma otimização multi-objetivo com interesse em avaliar redes de filas do tipo $M/G/1/k$, ou seja, entrada markoviana, atendimento geral, servidor único e capacidade finita. O intuito é minimizar a soma das probabilidades de bloqueio das filas da rede simultaneamente com a minimização das somas das taxas de atendimento e dos *buffers* alocados nas filas. A abordagem multi-objetivo conduz para múltiplas soluções sub-ótimas, a tomada de decisão usualmente é conduzida de forma a se adaptar às especificidades do problema.

Diante disso, este estudo planeja associar a formulação matemática do problema apresentada por Duarte (2022) [2] com a utlização do algoritmo MOPSO apresentado por Souza, Duarte, Moreira e Cruz (2020) [14] para fornecer soluções eficientes para o clássico problema de alocação de servidores, também conhecido na literatura por *Server allocation problem* (SAP).

3 Abordagem do Problema e Aspectos Metodológicos

Este estudo apresenta uma investigação sobre algoritmos eficientes para o problema de obter a alocação ótima de servidores. Usualmente trata-se de um propósito para otimizar o funcionamento de redes de filas multi-servidor markovianas (chegadas conforme distribuição Poisson e tempos de atendimento com distribuição exponencial). A literatura apresenta diversos algoritmos propostos para resolver esse tipo de problema. A formulação do problema e o algoritmo utilizado precisam ser abordados simultaneamente dada forte dependência entre os algoritmos e a formulação de programação matemática utilizada. Este estudo será iniciado por meio da discussão de uma formulação de objetivo único do SAP.

3.1 Formulação Mono-objetivo

O problema é definido por meio 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 formulação dual da anterior, ou seja uma proposta de SAP dual pode ser também apresentada. O problema de 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 nas Equações 3.2a e 3.2b. 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.

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

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.

As formulções anteriores são bastante úteis para ajudar a desenvolver algoritmos adequados para o propósito, mas será abordada aqui uma formulação distinta e mais abrangente, um formulação multi-objetivo.

3.2 Uma Possível Formulação Matemática multi-objetivo

As formulações mono-objetivo apresentadas tem interesse em reduzir o tempo total dos clientes na rede de filas, para tanto sucessivos incrementos na alocação de servidores são utilizados. Esse procedimento é completamente alheio ao custo efetivo gerado pela inclusão desses novos servidores, principalmente devido à fração de tempo ociosa que os servidores apresentam. Diante disso, tanto o tempo geral de percurso dos clientes, quanto alguma medida associada com a produtividade dos servidores envolvidos devem ser considerados para a proposição de uma formulação mais ajustável ao problema. A fração de tempo não ocioso dos servidores em uma fila é usualmente mensurada pela razão entre o número esperado de clientes em serviço e o número efetivo de servidores existentes. Se essa relação se aproxima da unidade, a produtividade dos servidores naquela fila é mais elevada.

Esse contexto conduz para uma possível reformulação do problema de otimização de redes filas $M/M/c$ para uma versão multi-objetivo. Nessa nova formulação, a maximização da produtividade (fração de não ociosidade) dos servidores envolvidos, e a minimização do tempo total de permanência dos clientes na rede de filas são executadas simultâneamente. Essa formulação foi descrita por Duarte (2022) [2] 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 não ociosidade dos servidores e simultaneamente minimiza o tempo total esperado dos clientes na rede de filas. Decorre disso então que, $f_1(\mathbf{C}) = -W(\mathbf{C}) = -\sum_{i=1}^{m} W(c_i)$ 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 (não ociosidade) dos servidores.

O problema foi descrito como um problema apenas de maximização ao multiplicar por -1 a função objetivo associada ao tempo total de permanência dos clientes na rede de filas. Dessa forma, passa a se tratar de um objetivo de maximização.

A produtividade dos servidores na fila única com c_i servidores é descrita por $P(c_i) = [L(c_i) - L_q(c_i)]/c_i$, com $L(c_i)$ o número esperado de clientes na i-ésima estação de serviço (incluindo clientes em serviço e clientes esperando na fila), $L_q(c_i)$ o número esperado de clientes esperando na fila da i-ésima estação de serviço, e por fim c_i 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 da solução inicial. O interesse está na realocação de servidores que interesse em melhorar a performance da rede de filas. Isso conduz para uma clara analogia ao clássico problema da mochila estocástico (mais detalhes em [15]).

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

A formulação matemática apresentada nas Equações 3.3a e 3.3b são dependentes das medidas de desempenho em filas $L(c_i)$ e $L_q(c_i)$, portanto expressões para $L(c_i)$ e $L_q(c_i)$ devem ser detalhadas. Para uma única fila $M/M/c_i$, com uma taxa de chegada λ e uma taxa de atendimento μ (igual para todos os servidores), considere p_j a probabilidade de j clientes (inclusos os clientes em serviço e clientes que aguardam na fila). Para tanto, p_j é dado por:

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

Portanto,

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

É importante observar que p_0 é a probabilidade de que nenhum cliente esteja na fila e todos os servidores estejam ociosos. A existência de distribuição invariante é dependente da 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, $\sum_{j=c_i+1}^{\infty} \left(\frac{\lambda}{c_i\mu}\right)^{j-c_i} = \frac{\lambda}{c_i\mu - \lambda}$.

Diante disso,

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

Dessa forma, a expressão para p_0, e 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 (não ociosidade) dos servidores de uma única fila com c_i servidores alocados é dada por:

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

Como $L(c_i)$ é a esperança do número de clientes na i-ésima estação de serviço (inclusos os clientes em serviço e clientes que aguardam na fila). A clássica lei de Little [16] pode ser aplicada, portanto 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 *Particle Swarm Optimization*

Como mencionado no capítulo 1, o algoritmo PSO é bastante simples em sua formulação. O seu funcionamento adequado passa inicialmente pela boa definição do que representará cada partícula na formulação matemática do problema em estudo. Aqui, cada partícula deve representar uma solução possível para a alocação dos servidores que otimizam a rede de filas em estudo. Portanto, nesta formulação específica, cada partícula pode ser representada pela m-upla $(x_1, x_2, \ldots, x_m) = (c_1, c_2, \ldots, c_m)$.

Vale destacar que o problema de otimização multi-objetivo em estudo possui restrições particulares associadas às filas, a restrição de convergência da Equação 3.6 precisa sempre ser respeitada. Essas considerações garantem a viabilidade das soluções investigadas. A abordagem MOPSO proposta para o problema de otimização da rede de filas segue basicamente a execução descrita no algoritmo apresentado na Figura 2.

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

- Posição das partículas $x_i = (x_{1i}, x_{2i}, \ldots, x_{mi})$;
- Velocidade das partículas $v_i = (v_{1i}, v_{2i}, \ldots, v_{mi})$;
- Melhor posição pessoal (p_{best}) p_i;
- Melhor posição global (g_{best}) g_i.

Os parâmetros do algoritmo MOPSO foram definidos da seguinte forma: r_1 e r_2 são números aleatórios positivos com distribuição uniforme no intervalo $[0, 1]$, $w(t)$ é o componente da inércia. O componente da inércia foi definido em $w(t) = 0{,}4$ para este estudo. O MOPSO aqui descrito, é uma adaptação da implementação clássica apresentada por Coello-Coello & Lechunga [17].

Na formulação multi-objetivo, a posição da i-ésima partícula no espaço d-dimensional de busca é representada por $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. Já a velocidade da referida partícula é representada por $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. A melhor posição da i-ésima partícula durante

algoritmo
  / gera o enxame de partículas inicial */*
  $X \leftarrow$ **GeraPopulaçãoInicial**(`swarmSize`)
  $P \leftarrow X$
  / encontre fronteiras não-dominadas* $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$ **/*
  $\mathcal{F} \leftarrow$ **OrdenaçãoNãoDominante**(X)
  $g \leftarrow$ *random* $(\mathcal{F})$
  para $t = 1$ **até** `numIter` **faça**
    para $i = 0$ **até** `swarmSize` **faça**
      $v_i^{t+1} \leftarrow$ *Velocidade*(x_i^t, p_i, g)
      $x_i^{t+1} \leftarrow$ *NovaPosição*(x_i^t, v_i)
      se x_i^{t+1} *domina* p_i **então** $p_i \leftarrow x_i^{t+1}$
        senão
          se p_i *domina* x_i^{t+1} **então** $p_i \leftarrow p_i$
            senão $p_i \leftarrow$ *random* (x_i^{t+1}, p_i)
          fim se
      fim se
    fim para
    $\mathcal{F} \leftarrow$ **OrdenaçãoNãoDominante**(X)
    $g \leftarrow$ *random* $(\mathcal{F})$
  fim para
  escreva $\mathcal{F}$
fim algoritmo

Figura 2 – Algoritmo PSO multi-objetivo

as buscas é dada por $p_i = (p_{i1}, p_{i2}, \dots, p_{in})$. A velocidade e a posição das partículas são atualizadas da iteração t para a iteração $t+1$ conforme as equações:

$$v_i^{t+1} = w^t + r_1\left(p_i - x_i^t\right) + r_2\left(g_i - x_i^t\right), \tag{3.11}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}; \tag{3.12}$$

Neste estudo, as variáveis de decisão são inteiras, para as variáveis inteiras, a posição deve ser atualizada de acordo, Eq. (3.13), ou seja:

$$x_i^{t+1} = \text{int}\left(x_i^t + v_i^{t+1}\right); \tag{3.13}$$

A escolha da melhor posição da i-ésima partícula (p_i) é feita para cada iteração, da seguinte forma: se a nova posição é superior (em termos de dominância no conceito multi-objetivo) à posição p_i, a mesma é atualizada pela nova posição $x_i(t+1)$. Se a

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

4 Resultados Alcançados

O algoritmo de otimização discutido anteriormente foi implementado em R *statistical software* (R Core Team, 2021 [18]). O ambiente de execução para realização dos experimentos computacionais foi um Intel(R) Core(TM) i5-8250U, 1.60 GHz, executando, Windows 11 Home Single Language 64 bits, com 8,00 GB de memória RAM.

Diferentes topologias de redes complexas de filas foram utilizadas no experimento. 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 para testar os efeitos de série, fusão e divisão foram também utilizadas e serão detalhadas.

Em todas as execuções, as soluções iniciais foram obtidas de forma aleatória, mas sob a condição de que cada fila rede tenha no máximo 50 servidores. Essa condição não é garantida após a evolução das soluções através do MOPSO, porém a soma total de servidores é preservada para grantir factibilidade. De outra forma, o objetivo é produzir soluções mais eficientes que não acarretem em aumento de custo em servidores..

O algoritmo teve número máximo de ciclos fixado em $\texttt{numIter} = 1000$. O número de soluções iniciais `swarmSize` foi fixado em 5000, todas geradas aleatoriamente. Devido a natureza do MOPSO 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 Topologia Série

Muitos procedimentos de interesse prático são compostos por filas alocadas em série, diversos processos produtivos apresentam estas topologias. As Figuras 3, 4 e 5 apresentam topologias em série com 3, 6 e 9 filas, respectivamente.

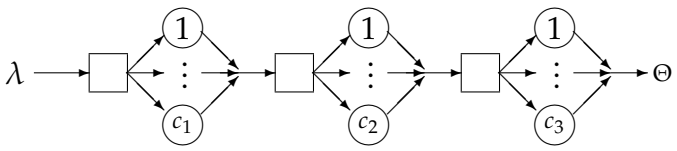

Figura 3 – Rede complexa de filas com topologia série com 3 filas.

Experimentos computacionais foram executados através da formulação matemática proposta nas Equações (3.2a) e (3.2b) utilizando o algoritmo MOPSO descrito

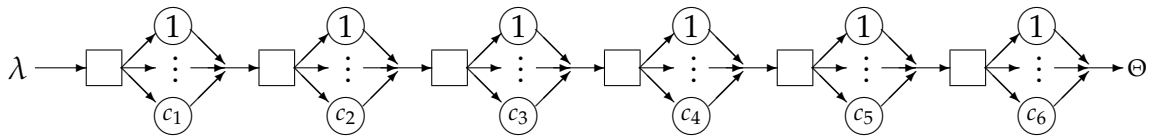

Figura 4 – Rede complexa de filas com topologia série com 6 filas.

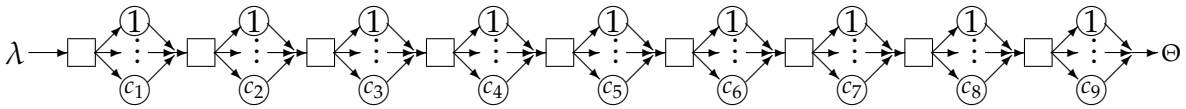

Figura 5 – Rede complexa de filas com topologia série com 9 filas.

na seção 3.3. As Figuras 6, 7, 8, 9, 10 e 11 apresentam os resultados obtidos. Para cada topologia são apresentadas duas figuras, a primeira no espaço das variáveis decisórias, para cada fila da rede. Já a segunda figura, ilustra o espaço dos objetivos do problema. Inicialmente as Figuras 6, 7 apresentam os resultados para topologia série para a rede com 3 filas.

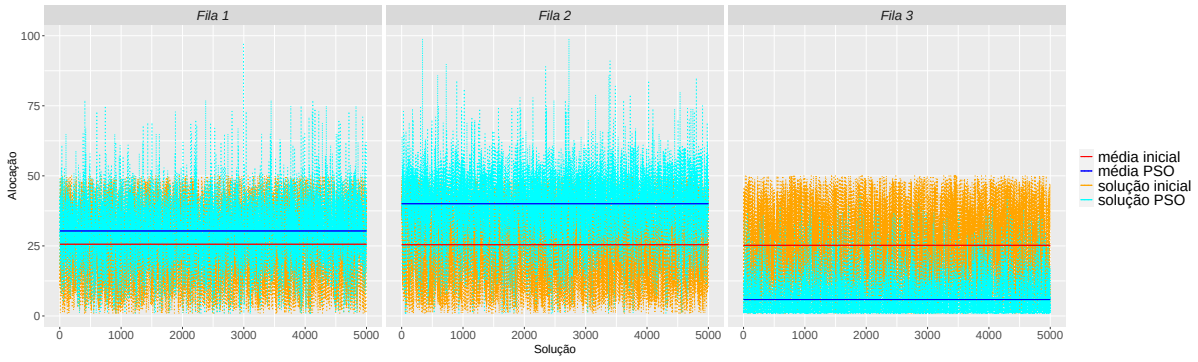

Figura 6 – Alocação de servidores nas filas da rede com 3 filas em série (de acordo com a topologia na Figura 3).

A Figura 6 mostra que o algoritmo PSO tende a reduzir sobremaneira a alocação de servidores na terceira fila da rede de filas em série. Aparentemente, um volume maior de servidores alocados nas primeiras filas da rede tende a reduzir bloqueios ao longo do percurso da rede.

A Figura 7 ilustra que algoritmo o PSO é capaz de fornecer um conjunto de soluções que preservam os tempos de percurso mas aumentam a produtividade dos servidores. É importante ressaltar que para as soluções com menor tempo de percuros, o algoritmo PSO foi capaz de fornecer uma realocação de servidores capaz de oferecer ganho em produtividade.

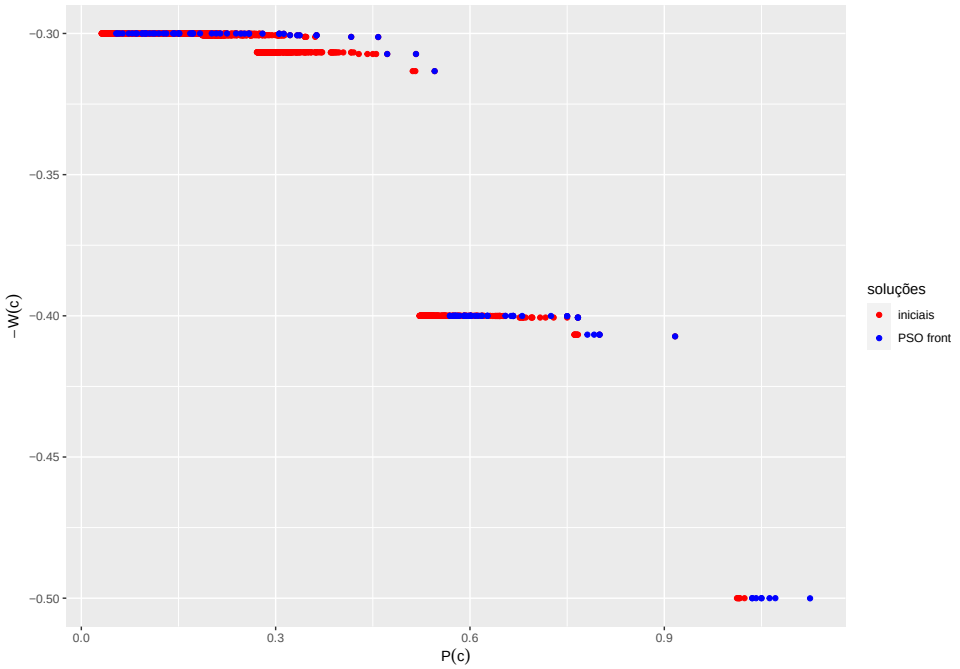

Figura 7 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 3 filas em série (de acordo com a topologia na Figura 3).

A seguir, as Figuras 8, 9 apresentam os resultados na mesma estrutura, novamente a topologia de estudo é série, porém agora numa rede mais longa, composta por 6 filas.

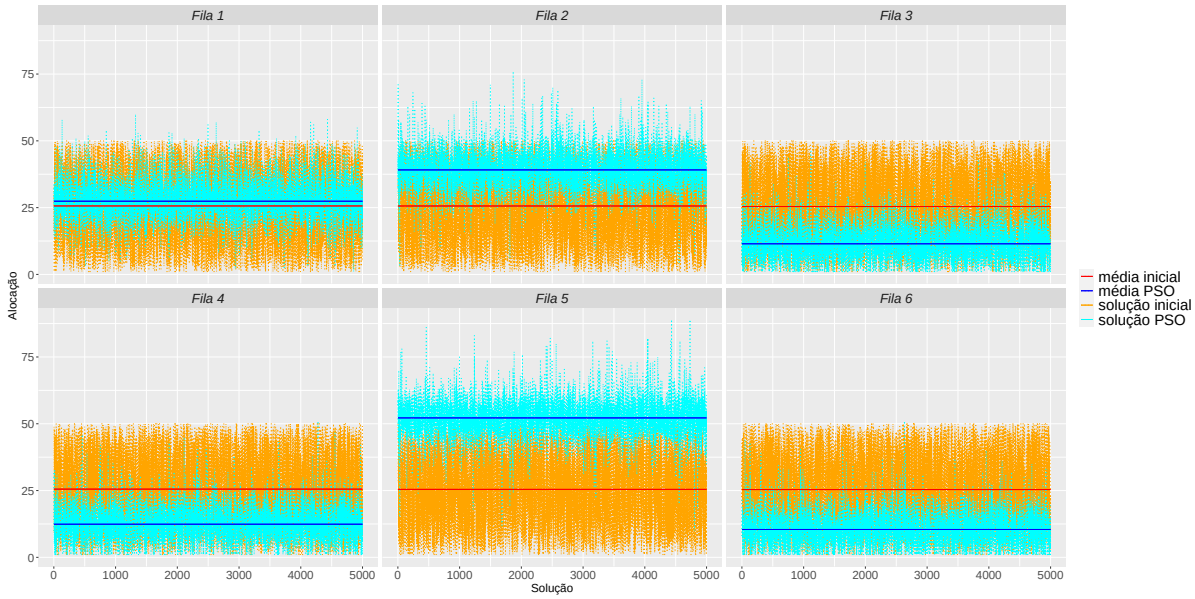

Figura 8 – Alocação de servidores nas filas da rede com 6 filas em série (de acordo com a topologia na Figura 4).

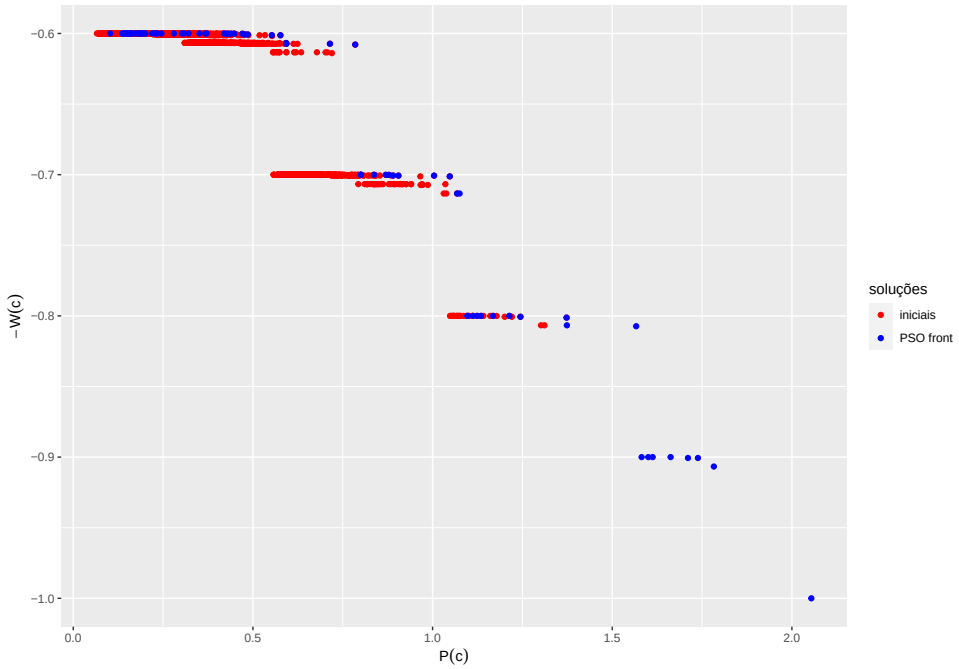

Figura 9 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 6 filas em série (de acordo com a topologia na Figura 4).

Na Figura 8, é possivel observar uma estratégia diferente de alocação quanto à rede composta por 3 filas em série. O algoritmo PSO reduziu as alocações nas filas 3, 4 e 6, principalmente em detrimento da alocação na fila 5. É razoavel admitir a suposição de que a proposta foi por desafogar o fluxo da rede na fila 2 com mais servidores, posteriormente economizar recursos em servidores até que novamente possíveis retenções surgissem e por fim, alocar um alto volume de servidores na fila 5 para conter este efeito predescessor.

Através da análise da Figura 9, verifica-se a apresentação de uma novidade com relação à rede em série composta por 3 filas, analisada anteriormente. O algoritmo PSO, além de fornecer soluções com tempo médio de percurso similares aos das soluções inciais, porém com mais produtividade, o algoritmo foi capaz de fornecer algumas soluções com tempos de percurso um pouco mais elevados porém com altíssimo nível de produtividade verificado. Para alguns exemplos de atividades, a distribuição das tarefas de forma mais equânime é de extrema importância.

Por fim, as Figuras 10, 11 apresentam os resultados na mesma estrutura, mais uma vez a topologia de estudo é série, porém agora numa rede ainda mais longa, composta por 9 filas.

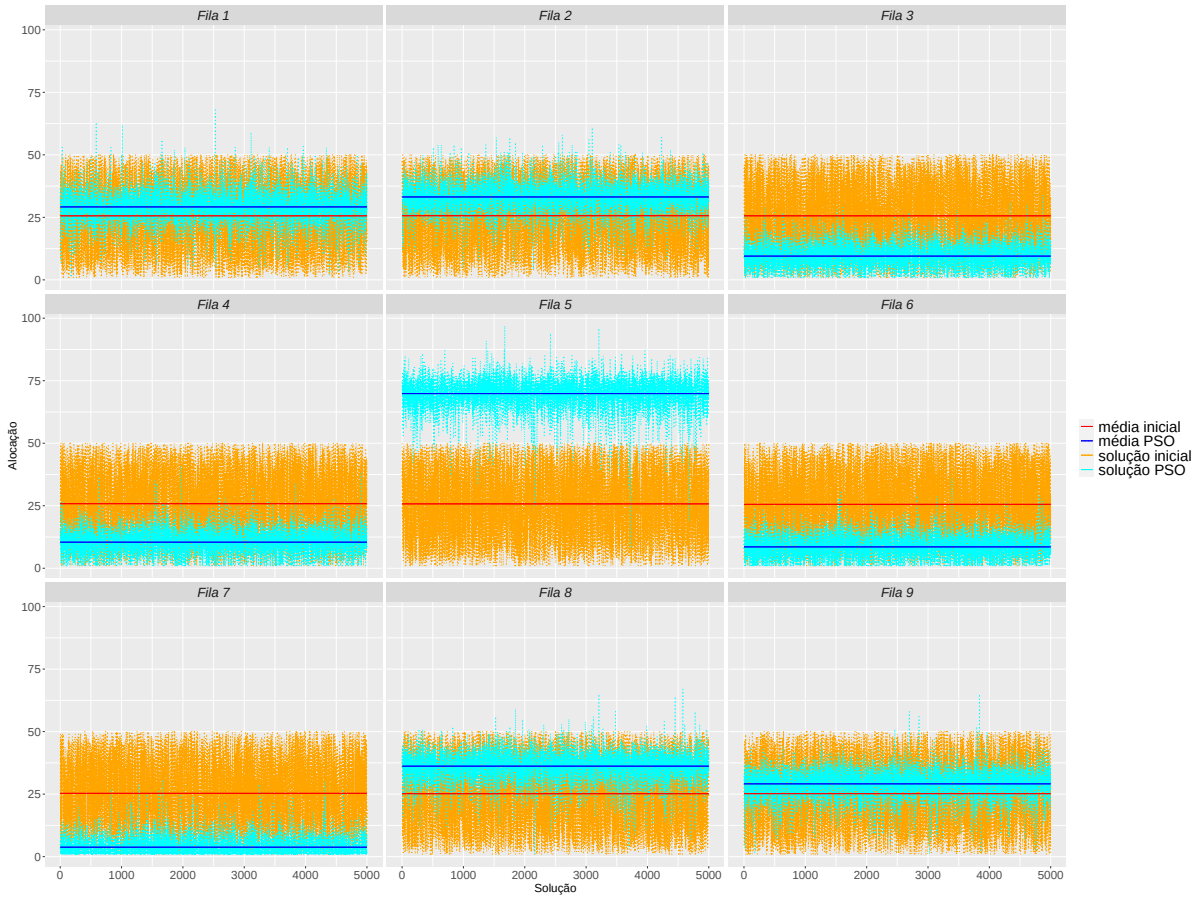

Figura 10 – Alocação de servidores nas filas da rede com 9 filas em série (de acordo com a topologia na Figura 5).

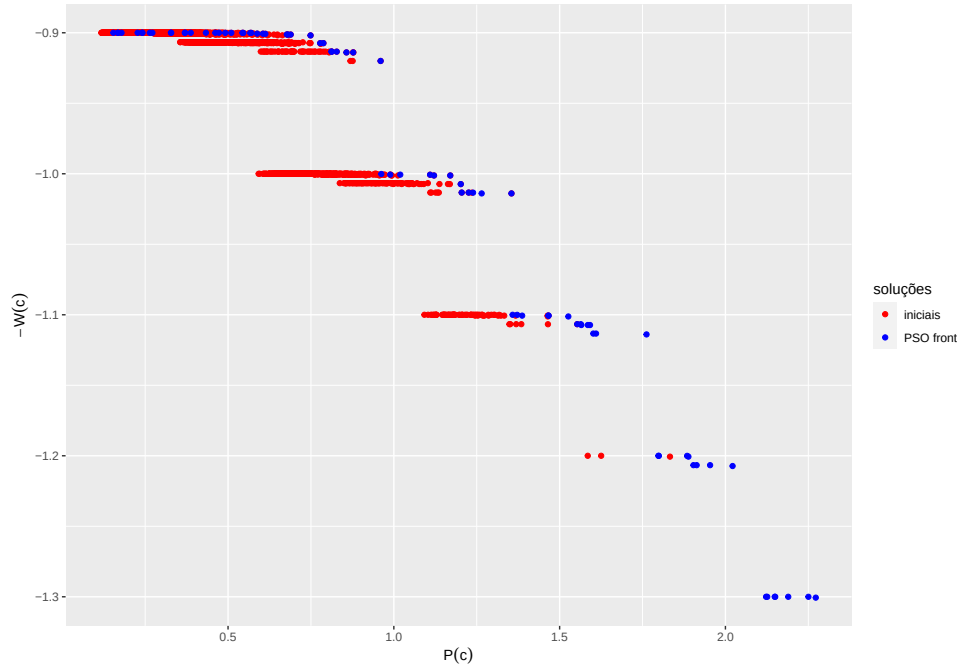

Figura 11 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 9 filas em série (de acordo com a topologia na Figura 5).

A Figura 10 apresenta maior heterogêneidade no padrão de alocação. É visível

que o algoritmo PSO alocou um volume maior de servidores na fila central da rede e dessa forma consegue apresentar alocações próximas nas filas que a alocação era maior, mas substancialmente menor nas filas 3, 4, 6 e 7.

A Figura 11 apresenta conclusões similares às verificadas para a rede em série com 6 filas. As soluções iniciais são melhoradas em termos de produtividade e algumas soluções de fluxo mais lento, porém com alta produtividade são fornecidas.

4.2 Topologia Fusão

Agora serão abordadas redes de filas com a topologia fusão. As Figuras 12, 13 e 14 apresentam topologias fusão com 3, 6 e 9 filas.

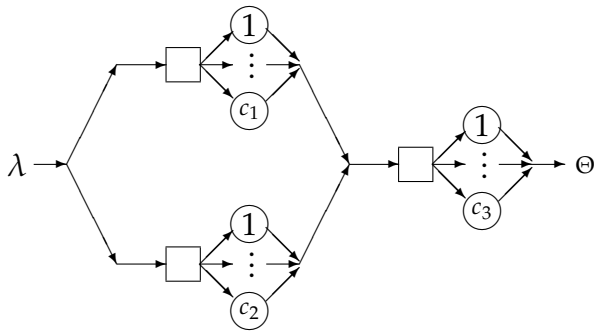

Figura 12 – Rede complexa de 3 filas com topologia fusão.

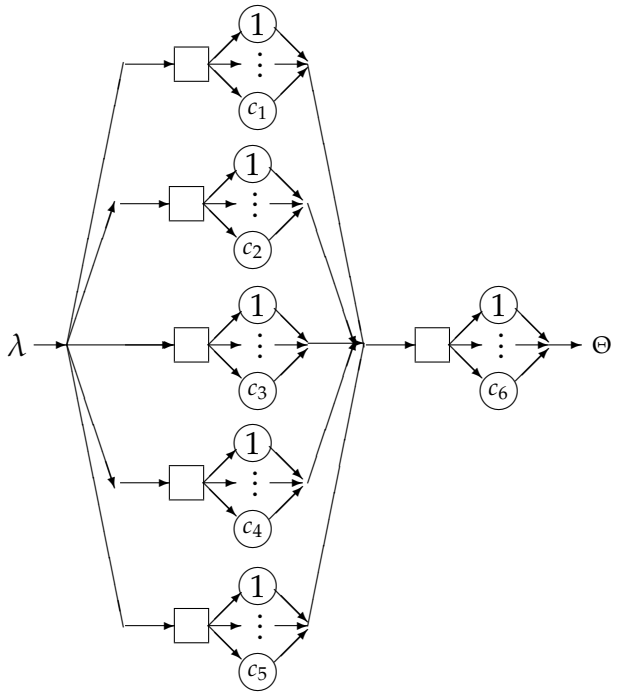

Figura 13 – Rede complexa de 6 filas com topologia fusão.

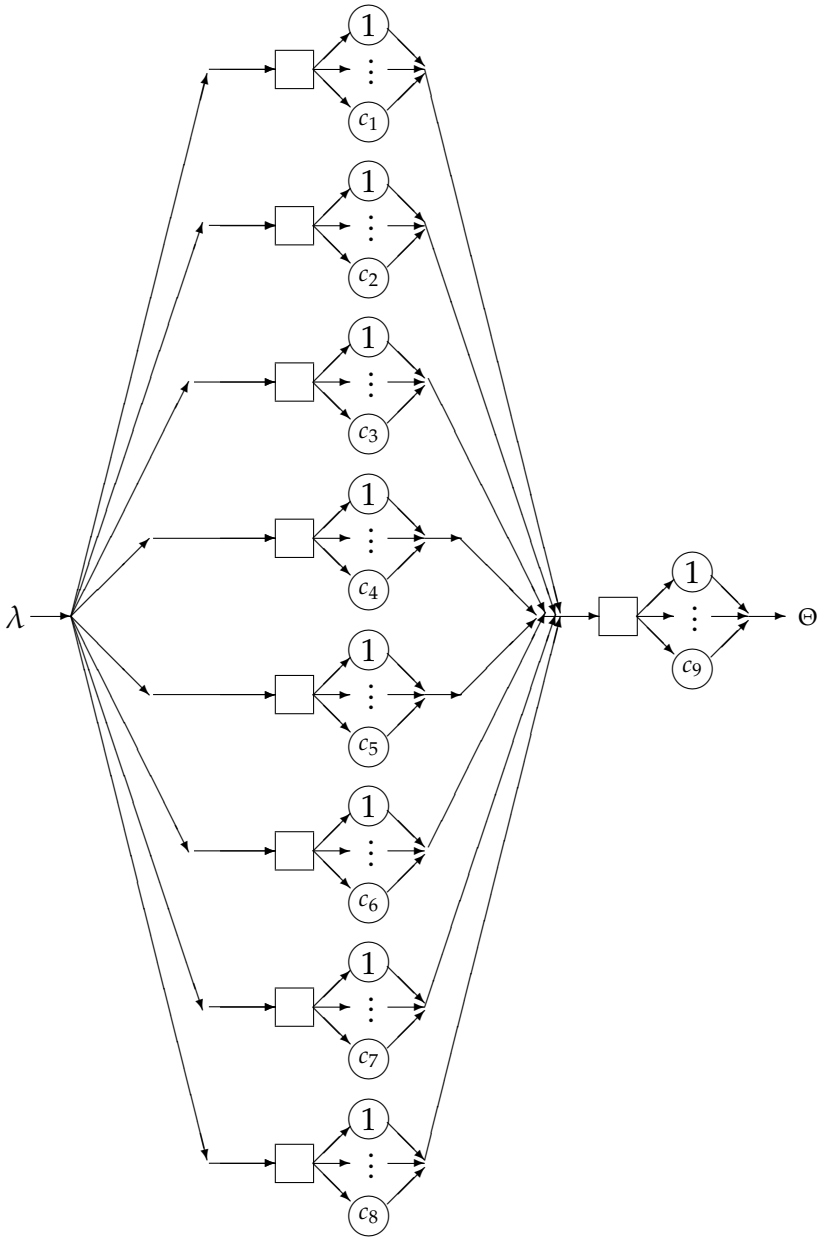

Figura 14 – Rede complexa de 9 filas com topologia fusão.

Na topologia fusão, os usuários, ao entrar no sistema se deparam com mais de uma fila para iniciar o procedimento e escolhem qual fila de forma aleatória e com igual probabilidade. Visto de outra forma, para k possíveis filas de entrada, uma taxa λ/k determina a taxa de entrada. Posteriormente, o usuários todos são encaminhados (em fusão) para uma única fila seguinte.

Novamente foram abordadas redes com diferentes quantidades de filas (veja as Figuras 15, 16, 17, 18, 19 e 20 com os resultados obtidos). Para cada rede de filas, novamente são apresentadas duas figuras, a primeira no espaço das variáveis decisórias, para cada fila da rede. Já a segunda figura, ilustra o espaço dos objetivos do problema. Inicialmente as Figuras 15, 16 apresentam os resultados para topologia fusão com 3 filas.

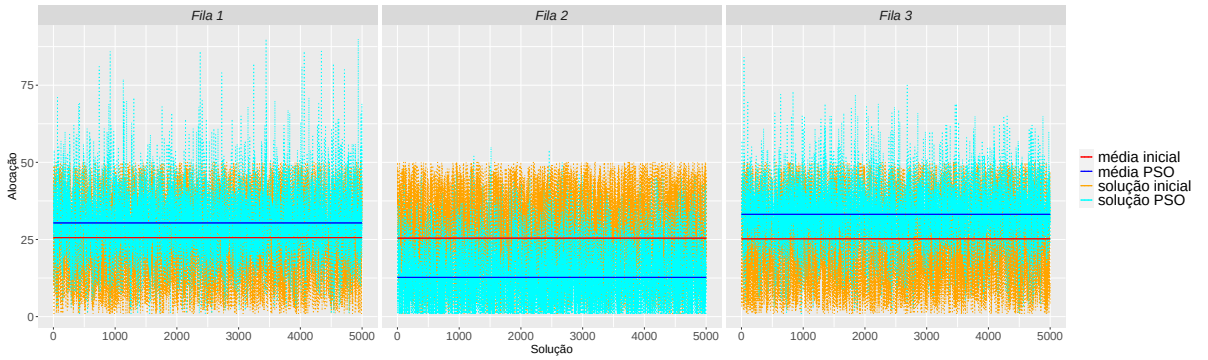

Figura 15 – Alocação de servidores nas filas da rede com 3 filas com fusão (de acordo com a topologia na Figura 12).

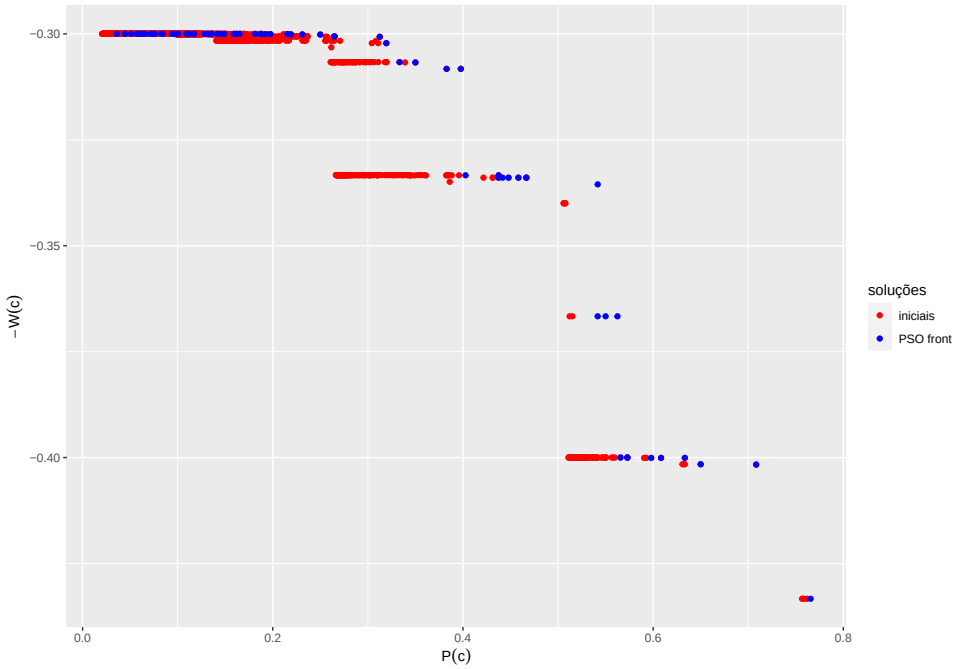

Figura 16 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 3 filas com fusão (de acordo com a topologia na Figura 12).

Na Figura 15, é possível observar que o algoritmo de PSO, reduziu as alocações na segunda fila da rede de filas que é uma das filas de entrada. Em contrapartida, as demais filas da rede tiveram maiores alocações. Já a Figura16 apresenta algum ganho em produtividade nas soluções obtidas, com pouca variação nos tempos de percurso. Fixado um tempo de percurso inicial, a menos de pequenas flutuações, o algoritmo PSO se mostrou capaz de produzir soluções com maior produtividade.

A seguir, as Figuras 17, 18 apresentam os resultados na mesma estrutura, porém para topologia fusão com 6 filas.

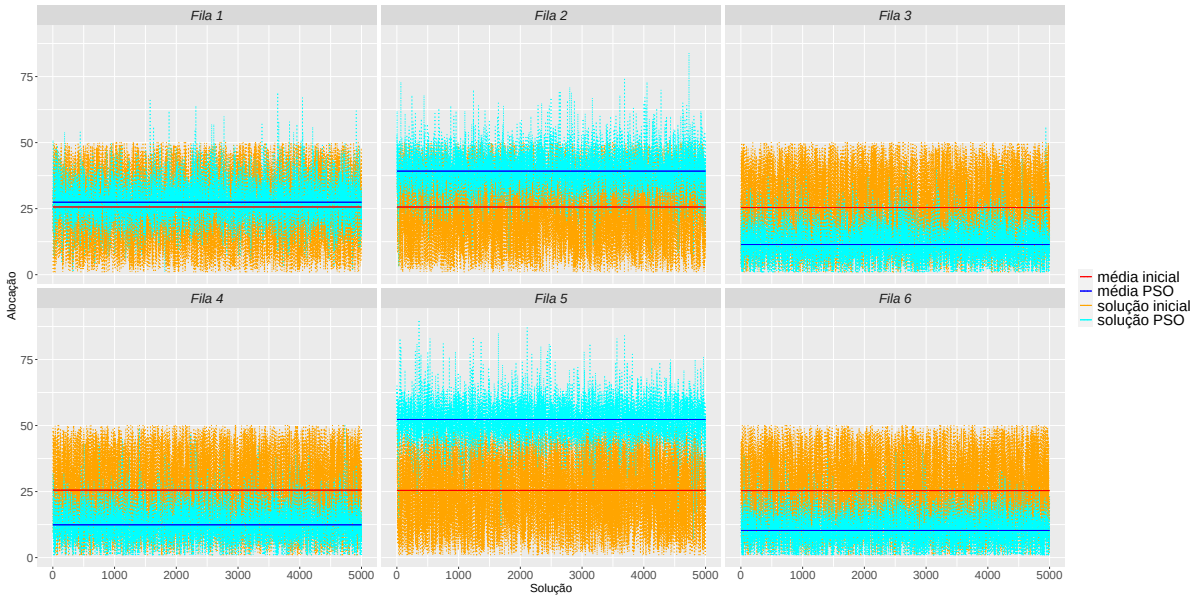

Figura 17 – Alocação de servidores nas filas da rede com 6 filas com fusão (de acordo com a topologia na Figura 13).

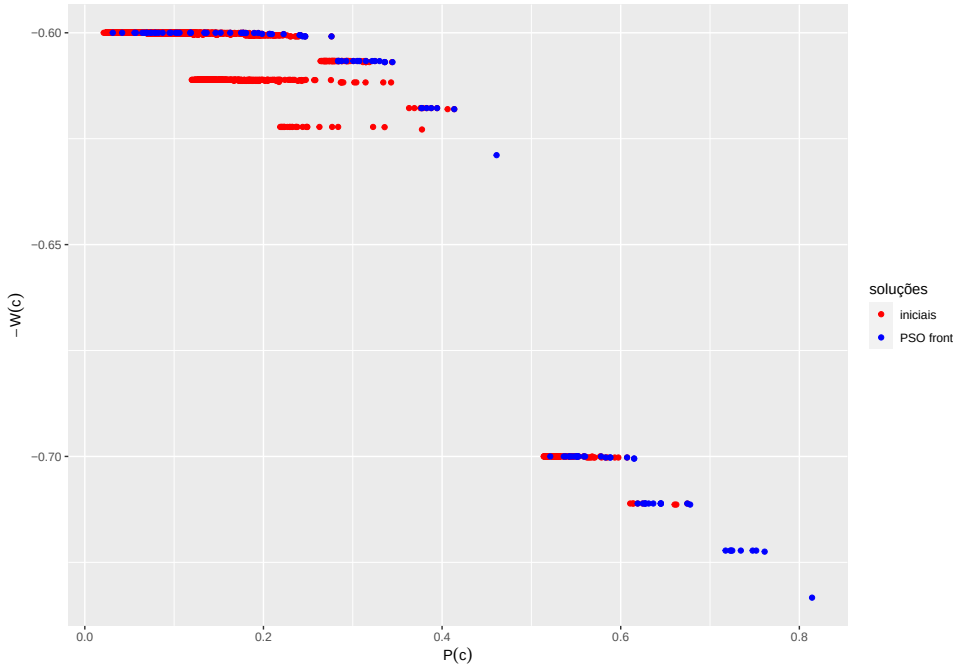

Figura 18 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 6 filas com fusão (de acordo com a topologia na Figura 13)

Ao observar a Figura 17, para a rede de filas com fusão com 6 filas, as filas 3, 4, que são filas de entrada e fila final, a fila 6 receberam alocações de servidores foram baixas. O algoritmo de PSO, privilegiou a alocação de servidores nas filas 2 e 5.

A Figura 18, ilustra algumas novidades em comparação a Figura 16 para a rede com 3 filas. O algoritmo PSO, para a topologia fusão, com 6 filas, manteve tempos de percurso semelhantes. Entretanto, a realocação de servidores acarretou em significativo

aumento da produtividade. Mesmo para soluções de tempo de percurso mais baixo, o algoritmo PSO foi capaz de fornecer soluções com elevados níveis de produtividade.

Por fim, as Figuras 19,20 apresentam os resultados na mesma estrutura, porém para topologia fusão com 9 filas.

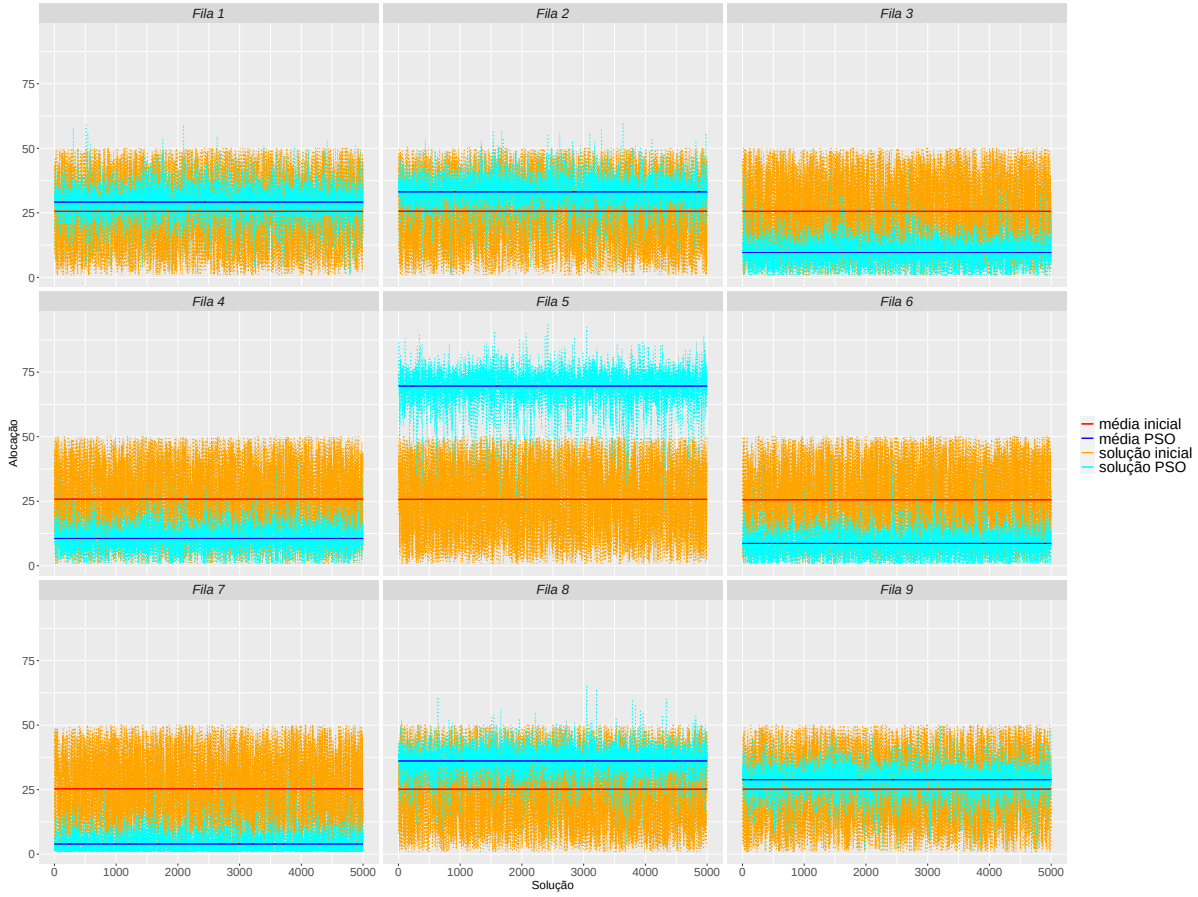

Figura 19 – Alocação de servidores nas filas da rede com 9 filas em série (de acordo com a topologia na Figura 14).

A Figura 19 apresenta as alocações de servidores na rede com nove filas para a topologia fusão. As filas de 1 até 8, são filas de entrada na rede de filas, ao passo que a fila 9 e utilizada por todos os usuários da rede de filas. O algoritmo PSO permitiu alocações elevadas de servidores na fila 5 e preservou todas as demais filas em patamares baixos de alocação, patamares menores que os verificados para as soluções iniciais ou suficientemente próximos das alocações iniciais. De alguma forma, o padrão parece similar ao verificado para as topologias de fusão, mesmo com um número menor de filas na rede de filas.

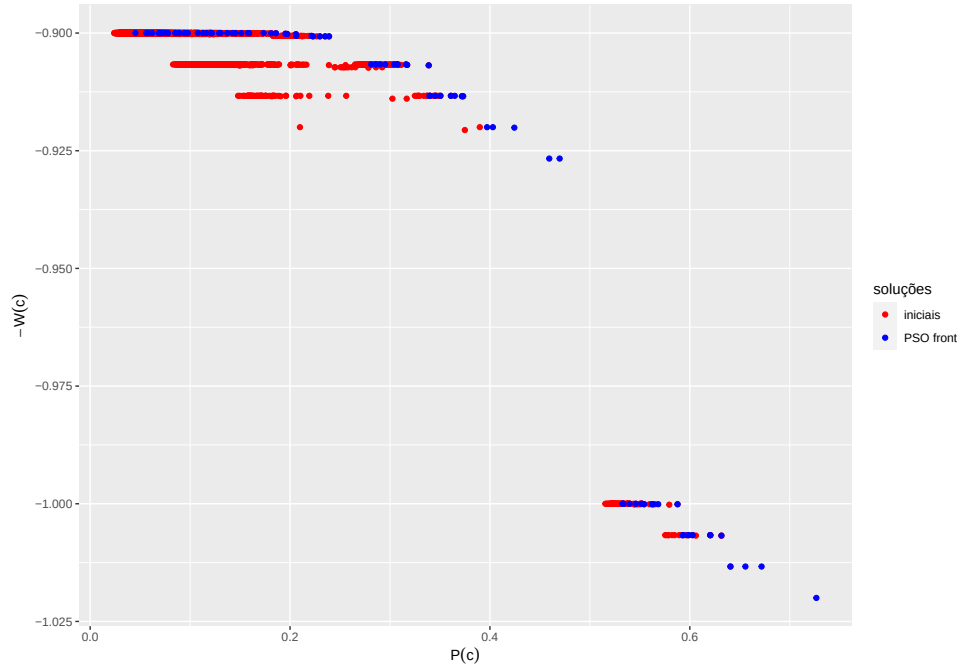

Figura 20 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 9 filas em série (de acordo com a topologia na Figura 14).

Já a Figura 20 apresenta o espaço de objetivos do problema de otimização em estudo. Dada uma maior flexibilidade, decorrente do número de filas da rede, um maior volume de soluções com tempos de percurso baixo são fornecidas. Novamente para todas estas, o algoritmo PSO foi capaz de gerar melhoria representativa na medida de produtividade, ou seja, eliminando a possibilidade de elevado número de servidores ociosos. Para as soluções de tempo de percurso maior, com níveis mais elevados de produtividade, ainda assim o algoritmo PSO foi hábil em fornecer soluções com aumento de produtividade.

4.3 Topologia Divisão

Agora serão abordadas redes de filas com a topologia divisão. As Figuras 21, 22 e 23 apresentam topologias divisão com 3, 6 e 9 filas. Na topologia divisão, os usuários sempre chegam por uma única fila do sistema, após serem atendidos nesta fila, eles se divedem de acordo com um vetor de roteamento $(p_1, p_2, \ldots, p_k)$ como pode ser observado.

Todos os experimentos (para as topologias das Figuras 21, 22 e 23) consideraram vetor de roteamento com caminhos equiprováveis. O problema de busca pelo roteamento ótimo também é um problema de interesse prático já abordado na literatura, porém não é objeto deste estudo.

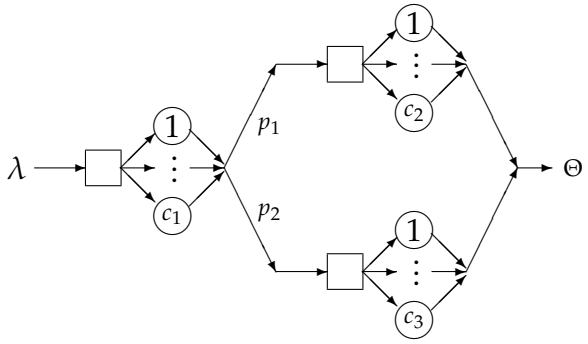

Figura 21 – Rede complexa de 3 filas com topologia divisão.

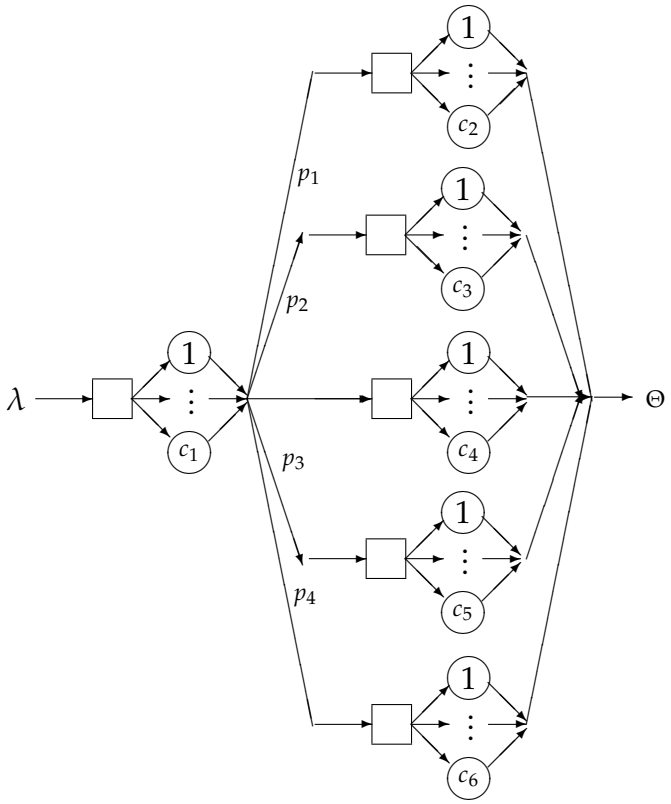

Figura 22 – Rede complexa de 6 filas com topologia divisão.

Para esta estrutura topológica, a primeira fila da rede tem papel bastante relevante, todos os clientes serão servidos por essa fila da rede para posteriormente optarem, de acordo com o vetor de roteamento, por alguma fila específica da rede de filas em estudo.

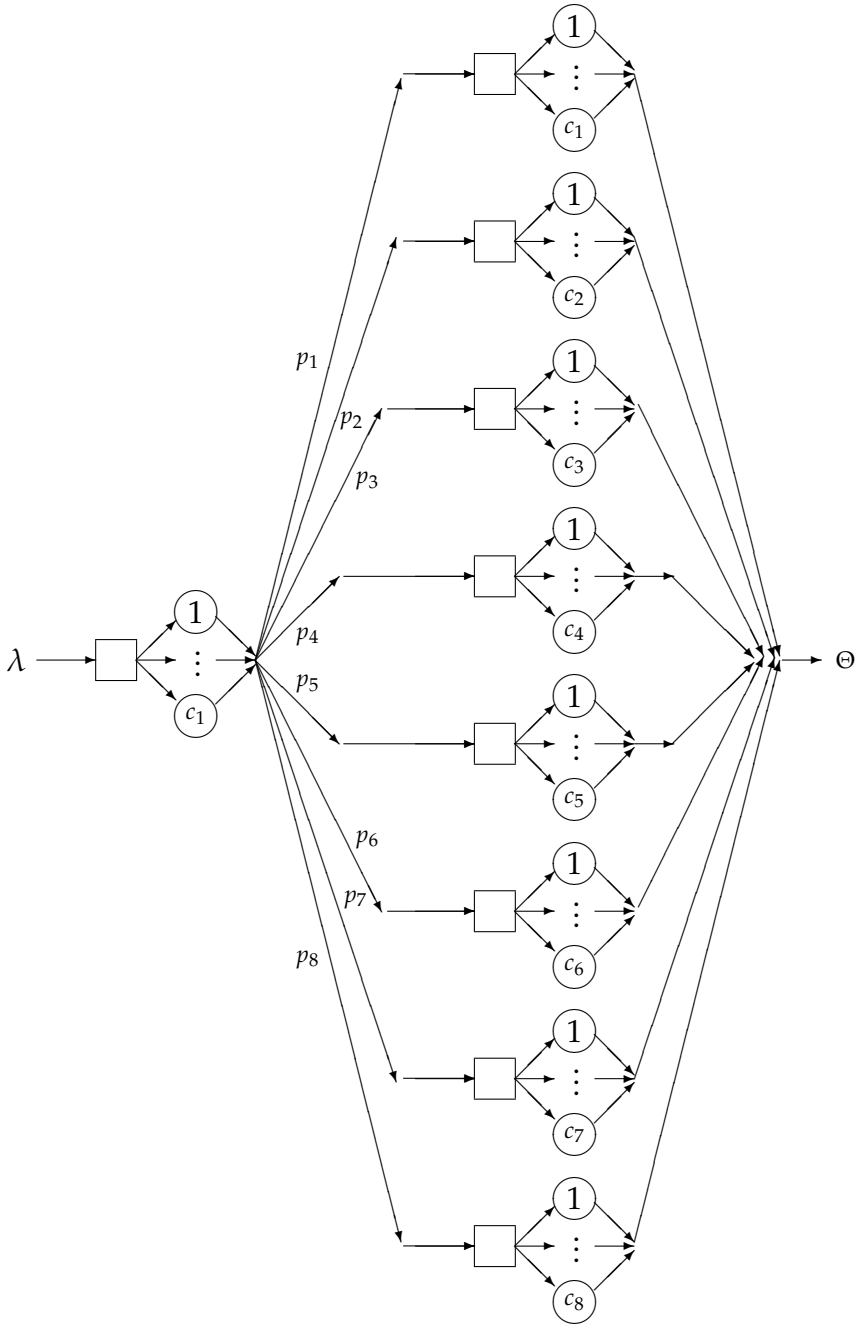

Figura 23 – Rede complexa de 9 filas com topologia divisão.

Novamente foram abordadas redes com diferentes quantidades de filas. Para cada rede de filas, novamente são apresentadas duas análises gráficas com os resultados. A primeira apresenta o espaço das variáveis decisórias, para cada fila da rede de filas. Já a segunda figura, ilustra o espaço dos objetivos do problema. Estes resultados estão apresentados nas Figuras 24, 25, 26, 27, 28 e 29.

Inicialmente são apresentados os resultados para topologia divisão com 3 filas. Na Figura 24, é possível observar que o algoritmo priorizou elevadas alocações de servidores na fila 1 de entrada e também na fila 2, na fila 3, a alocação foi menor. O algoritmo PSO propos uma redução substancial de servidores na terceira fila da rede,

aparentemente o interesse foi ampliar a vazão de clientes no percurso da rede.

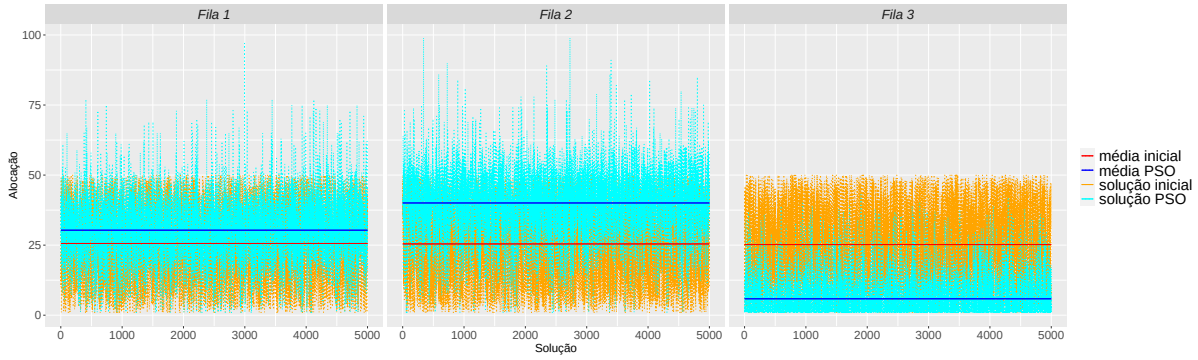

Figura 24 – Alocação de servidores nas filas da rede com 3 filas com divisão (de acordo com a topologia na Figura 21).

A análise dos resultados no espaço de objetivos confirma as verificações para a topologia anterior, a topologia com divisão. O algoritmo de otimização PSO mostra habilidade em preservar os tempos de percurso e aumentar a produtividade dos servidores empregados. A Figura 25 mostra que o algoritimo foi capaz de fornecer soluções com alta produtividade tanto nos tempos inferiores quanto para os tempos mais elevados.

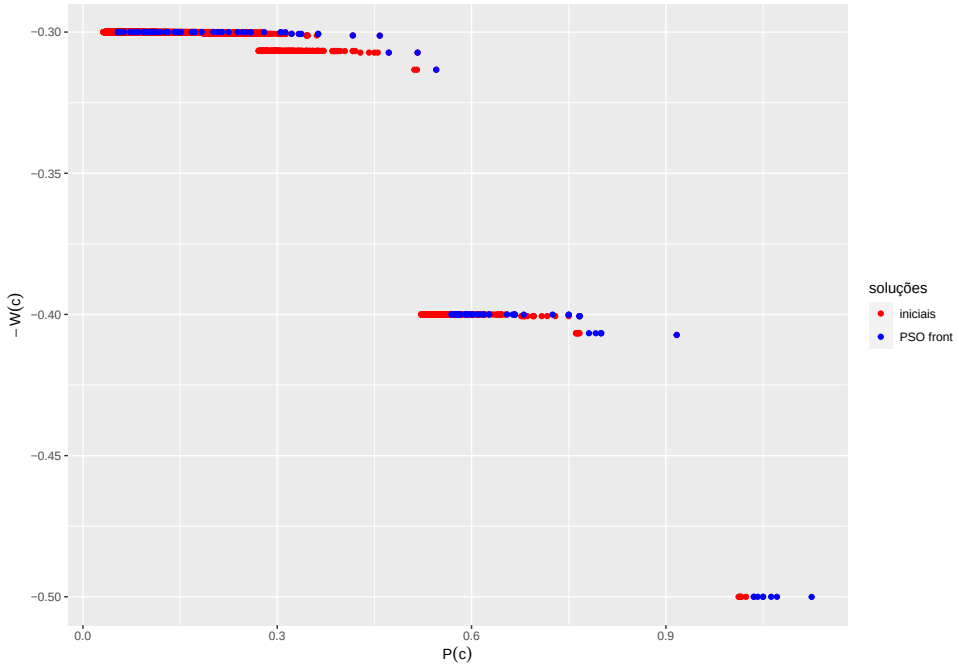

Figura 25 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 3 filas com divisão (de acordo com a topologia na Figura 21).

Em sequência, são apresentadas as Figuras 26, 27. Esta figuras contemplam os resultados na mesma estrutura topológica, porém agora em uma rede com 6 filas (veja a topologia da Figura 22).

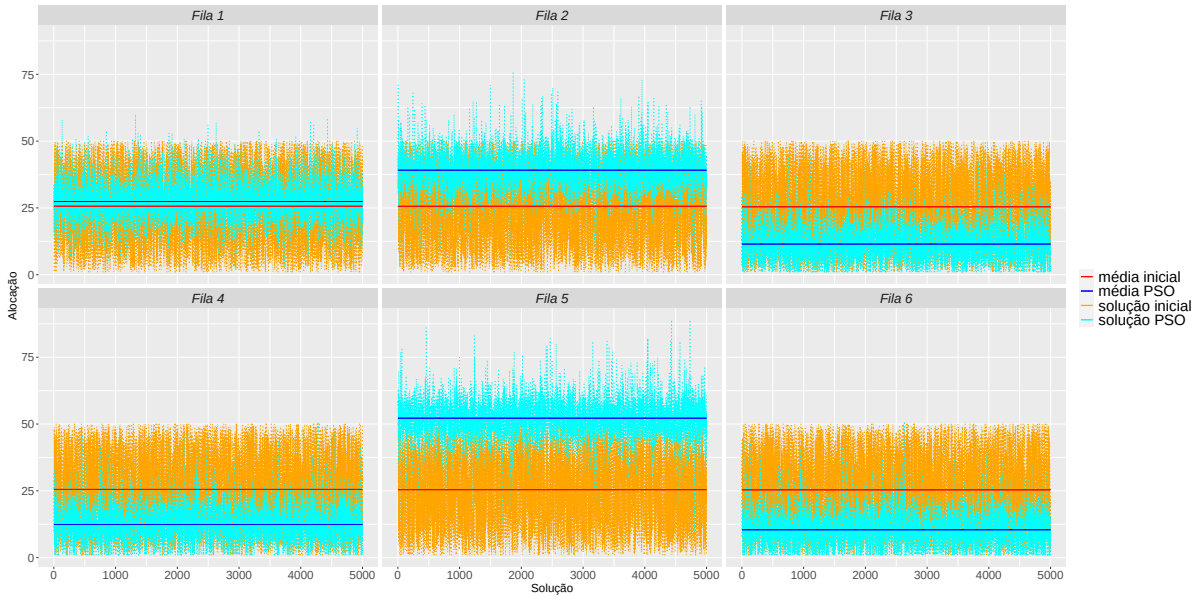

Figura 26 – Alocação de servidores nas filas da rede com 6 filas com divisão (de acordo com a topologia na Figura 22).

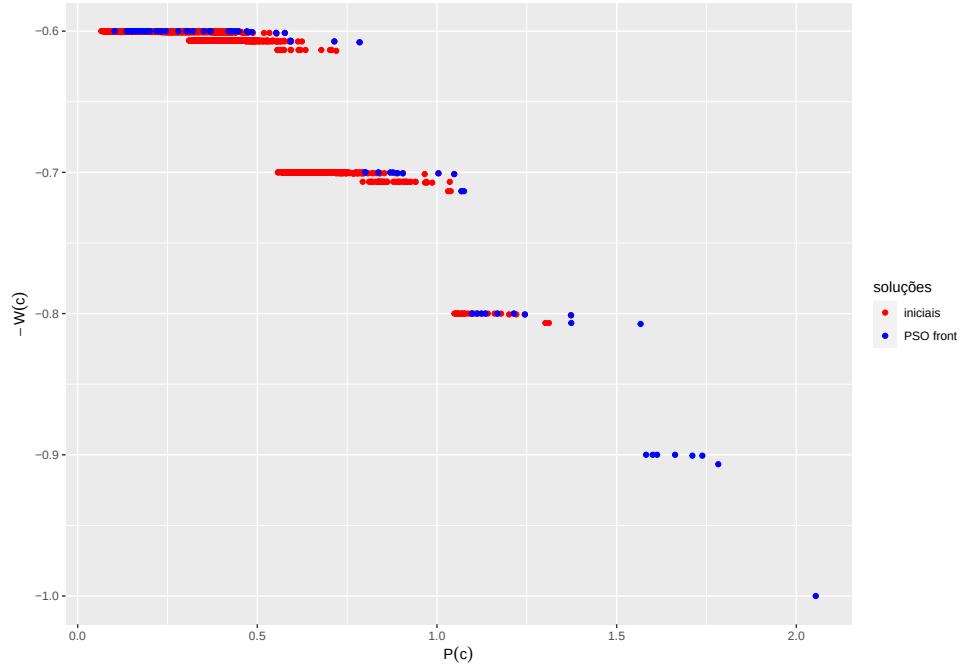

Figura 27 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 6 filas com divisão (de acordo com a topologia na Figura 22).

É possivel verificar através da Figura 26 que o algoritmo de otimização PSO manteve os níveis de alocação na fila de entrada da rede e manipulou a alocação dos demais servidores nas outras filas da rede. Nas filas 2 e 5, o algoritmo PSO priorizou altas alocações de servidores. Estas alocações mais elevadas foram compensadas nas filas 3, 4 e 6 da rede de filas.

Já a Figura 27 mostra que o PSO, foi capaz de apresentar alta produtividades nos respctivos tempos. Para a rede de 6 filas, o algoritimo apresentou soluções em tempos esperado de percurso mais altos, porém com altissima medida de produtividade verificada entre os servidores.

Por fim, as Figuras 28, 29 apresentam os resultados na mesma estrutura, porém para topologia fusão com 9 filas. A Figura 28 ilustra que o algoritmo de otimização PSO aumentou ligeiramente a alocação na fila de entrada da rede. Para as filas 3, 4, 6 e 7, o algoritmo PSO reduziu substancialmente a alocação de servidores. As alocações mais elevadas foram utilizadas nas filas 2, 5, 8 e 9.

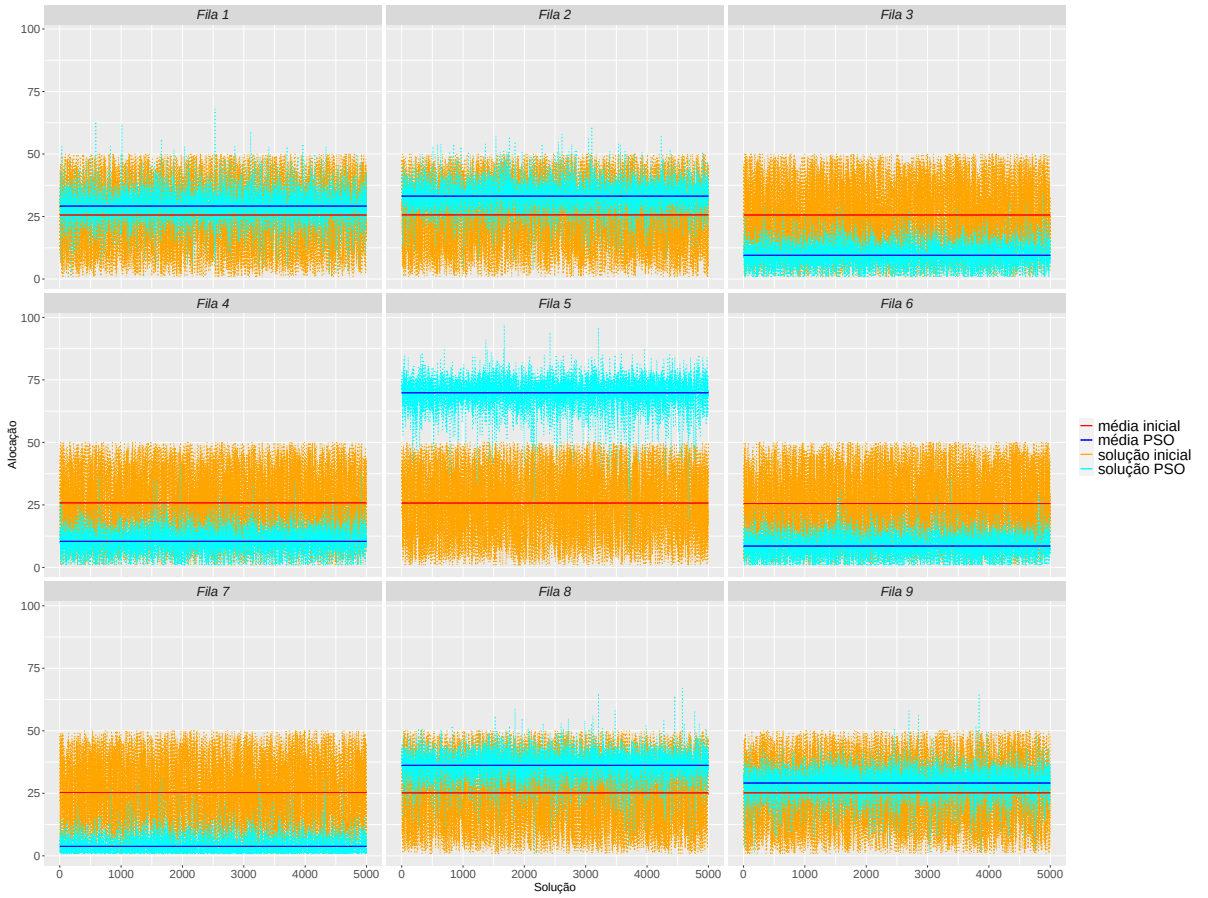

Figura 28 – Alocação de servidores nas filas da rede com 9 filas em série (de acordo com a topologia na Figura 23).

A Figura 29 mostra os resultados obtidos pelo algoritimo de otimização PSO da perspectiva do espaço de objetivos. Como verificado anteriormente o algoritmo PSO mostra capacidade em preservar os tempos de percurso e ampliar a produtividade. Além disso, na borda inferior direita, é possível verificar que o algoritmo conseguiu produzir algumas soluções com tempo de percurso superior porém com elevados níveis de produtividade.

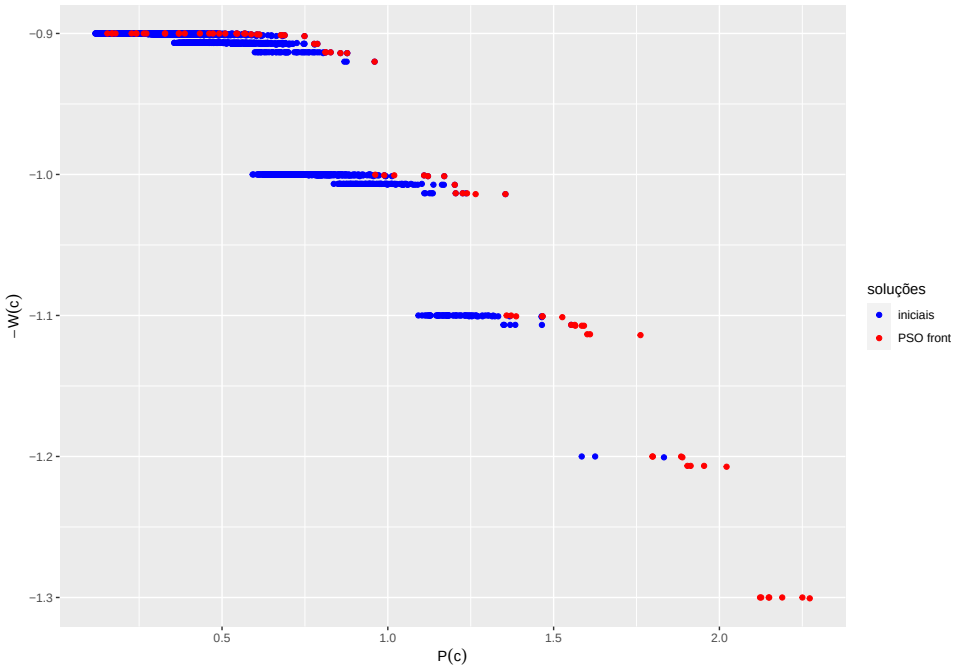

Figura 29 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 9 filas em série (de acordo com a topologia na Figura 23).

4.4 Topologia Mista

Finalmente serão abordadas redes de filas com a topologia mista, incluíndo séries, fusões e divisões. A Figura 1 apresentada na introdução foi utilizada para este estudo. Para esta investigação, apenas a rede mista de 6 filas foi utilizada. Os resultados são apresentados nas Figuras 30 e 31.

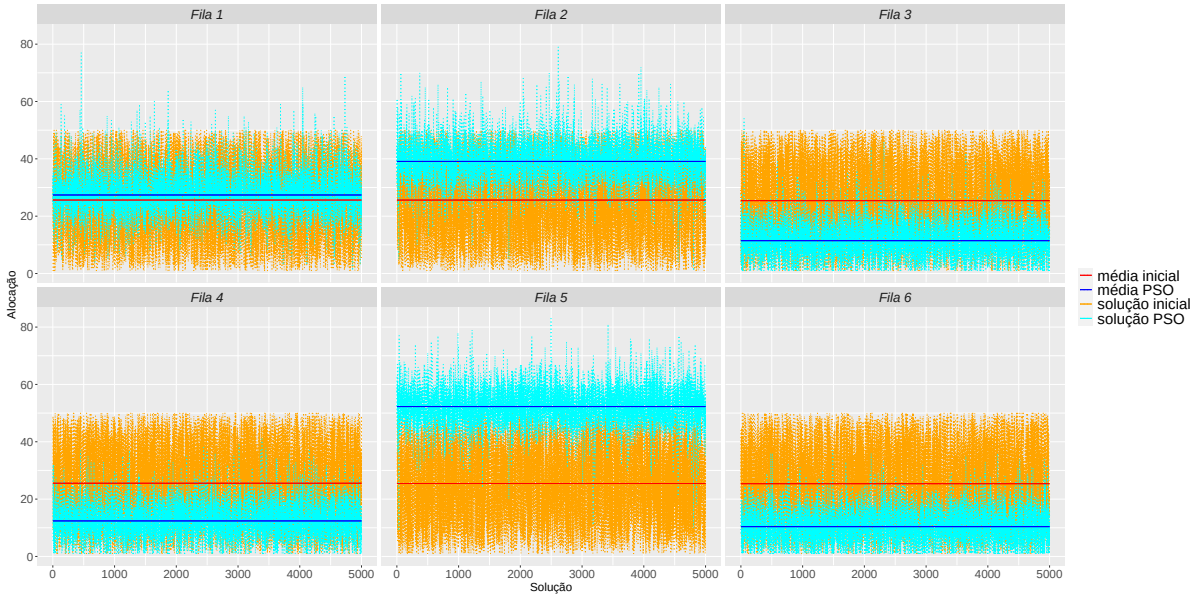

Figura 30 – Alocação de servidores nas filas da rede com 6 filas com divisão (de acordo com a topologia na Figura 1).

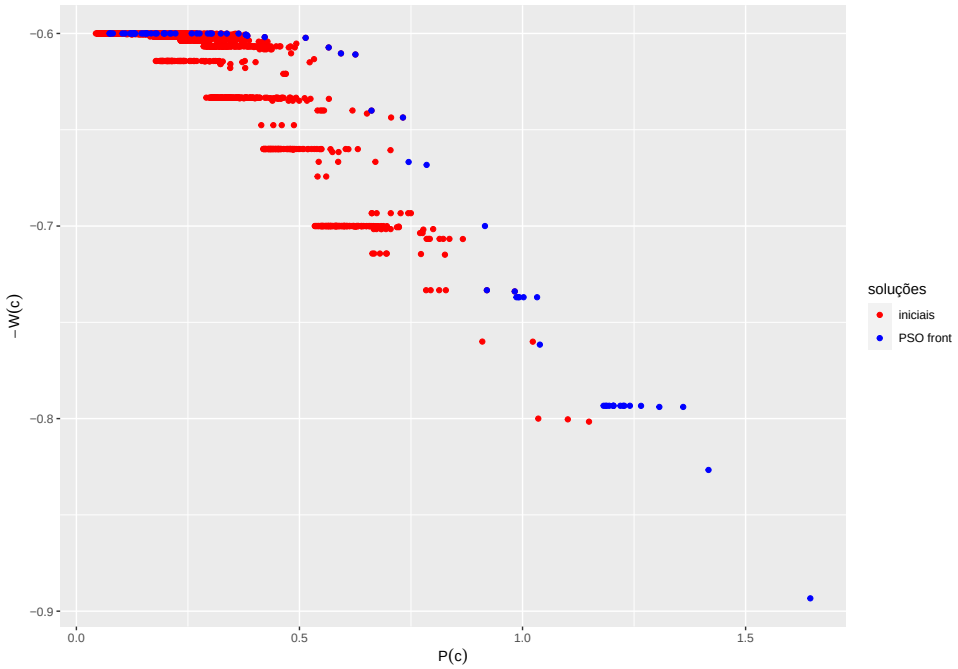

Figura 31 – Pareto-solução das soluções fornecidas pelo algoritmo PSO comparadas com o pareto das soluções iniciais nas filas da rede com 6 filas com divisão (de acordo com a topologia na Figura 1).

Como nas análises anteriores, a Figura 30 apresenta o espaço das variáveis decisórias, para cada fila da rede de filas. O algoritmo PSO priorizou baixas alocações nas 3, 4 e 6 da rede mista. Para as demais filas da rede, o algoritmo PSO utilizou alocações mais elevadas.

A Figura 31 ilustra o espaço de objetivos para a análise da rede de filas mista. É possivel verificar que o algoritmo PSO alcanço um aumento de produtividade contundente se comparado às soluções inicialmente propostas. Além disso, como critério de análise de algoritmos multi-objetivos, o Pareto-solução fornecido apresenta grande cobertura.

5 Considerações Finais

Este estudo abordou o clássico problema de alocação de servidores (SAP) em redes de filas markovianas sem limitações para a quantidade de buffers em cada fila em estudo. Uma formulação matemática específica para o problema em estudo foi apresentada. Além disso, uma adaptação do conhecido algoritmo PSO foi apresentada para o problema sob investigação. Na formulação discutida, o objetivo é obter uma alocação adequada de servidores ao longo de filas markovianda que compõem uma rede interconectada. A referida alocação adequada visa maximizar a produtividade associada aos servidores e simultaneamente minimizar o tempo médio de percurso dos clientes usuários da rede de filas.

Esse tipo de discussão demanda a análise de assuntos que se posicionam na fronteira do conhecimento científico. Dentre estes assuntos, é possível elencar, a análise de teoria de filas, o conhecimento da teoria de grafos para a representação das redes de filas, a formulação de problemas de otimização, a investigação de algoritmos de oitmização e a abordagem de classificação de soluções por meio da ótica multiobjetivo de otimização. Este conjunto de fatores estabelece o mérito inerente a este estudo enquanto um trabalho de conclusão de um curso de graduação.

O estudo apresentou um levantamento da bibliografia com pesquisas 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.

Para investigar o problema, o algoritmo PSO combinou sucessivamente, diferentes alocações em quatros topologias de rede de filas: série, fusão, divisão e mista. Cada uma delas, configuradas com diferentes quantidades de filas. Para as três primeiras topologias, foram testadas redes com 3, 6 e 9 filas. Já a rede de filas mista foi testada apenas com 6 filas. O algoritmo PSO apresentou os resultados eficientes por meio de soluções de alta produtividade com tempos de percurso competitivos.

A grande aplicabilidade de investigações desse tipo pode ser observada por meio de diversos possíveis exemplos. Entre eles, é possível mencionar o problema de produção de veículos. Os servidores são máquinas ou trabalhadores na equipe de produção. Cada fila da rede é responsavel por executar alguma etapa do processo produtivo do veículo. O algoritmo PSO se mostrou hábil para fornecer alocações eficientes dos servidores com interesse em redução do tempo produtivo e redução de ociosidade nos servidores.

Os resultados apresentados neste estudo mostram que um algoritmo de otimiza-

ção eficaz, como por exemplo o algoritmo PSO é capaz de realocar os servidores nas filas da rede e por meio disso ampliar a efetividade dos resultados produtivos. Essa informação pode ser utilizada para nortear equipes e tomadores de decisão para criarem e abordarem estratégias de alocação de recursos que permitam o aproveitamento destas informações estatísticas.

5.1 Proposta de continuidade

Este estudo, ainda apresenta propostas para propostas futuras de continuidade. Existe, por exemplo, a possibilidade de aplicação de outras formulações matemáticas do problema e também de outros métodos de otimização.

A investigação acerca da homogeneidade das soluções para oscilações nas taxas de serviço entre servidores. Abordagem para filas com atendimentos gerais (não markovianos), ou seja, filas do tipo $M/G/c$. Além disso, investigações futuras podem incluir a avaliação de qualidade no processo de estimação de medidas de desempenho em redes de filas. Os estudo abordando filas multiservidores são também abordagens instigantes para estudos futuros.

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 2 vezes nas páginas 15 e 1.

[2] 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 4 vezes nas páginas 2, 3, 7, and 10.

[3] Kennedy, James e Russel Eberhart: *Particle swarm optimization*. Em *Proceedings, IEEE International Conference on Neural Networks*, volume 4, páginas 1942–1948, 1995. Citado na página 2.

[4] 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 na página 5.

[5] van Woensel, Tom e Nico Vandaele: *Empirical validation of a queueing approach to uninterrupted traffic flows*. 4OR, 4(1):59–72, 2006. Citado na página 5.

[6] Van Woensel, Tom e Frederico Cruz: *A stochastic approach to traffic congestion costs*. Computers & Operations Research, 36(6):1731–1739, 2009. Citado na página 5.

[7] 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 na página 5.

[8] 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 na página 5.

[9] 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, ISSN 2405-8963. https://www.sciencedirect.com/science/article/pii/S2405896319310729, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019. Citado na página 6.

[10] Alves, Frederico Samartini Queiroz, Hani Camille Yehia, Luís Antônio Capanema 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 6.

[11] Liu, Yi Jian, Jian Cao, Xiao Yan Cao e Yuan Biao 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 na página 6.

[12] 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 na página 6.

[13] 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, 138:105513, 2022. Citado na página 6.

[14] 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 na página 7.

[15] Kellerer, Hans, Ulrich Pferschy e David Pisinger: *Knapsack problems*. Springer, Berlin, Heidelberg, 1ª edição, 2004, ISBN 978-3-540-40286-2. Citado na página 11.

[16] Little, John Dutton Conant: *A proof for the queuing formula: $L = \lambda W$*. Operations Research, 9(3):383–387, 1961. Citado na página 13.

[17] Coello Coello, Carlos e Maximino Salazar Lechuga: *MOPSO: A proposal for multiple objective particle swarm optimization*. Em *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, páginas 1051–1056, 2002. Citado na página 13.

[18] 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 17.