Revista da Estatística UFOP, Vol III(2), 2014, ISSN 2237-8111
Edição especial: 59ª Reunião Anual da Regional Brasileira da Sociedade Internacional de Biometria - RBRAS

BUSCA LOCAL PARA PÓS-PROCESSAMENTO EM RESULTADOS DE OTIMIZAÇÃO MULTI-OBJETIVO EM REDES DE FILAS GERAIS

Gabriel Lima de Souza[1], **Frederico Rodrigues Borges da Cruz**[2], **Anderson Ribeiro Duarte**[1]

Resumo: *No desafiador problema de otimização de redes de filas finitas, a capacidade total do sistema em áreas de espera deve ser a menor possível, enquanto o número de usuários atendidos por unidade de tempo deve ser o maior possível. Soluções para estes objetivos conflitantes já existem, porém podem ser melhoradas através da redistribuição de áreas de circulação entre as filas, preservando fixa a capacidade total do sistema. Um algoritmo simulated annealing foi desenvolvido, especialmente para redes de filas finitas, produzindo soluções eficientes para o problema. Um conjunto de experimentos computacionais foi conduzido, para determinar a eficiência da abordagem proposta. As conclusões apresentadas podem auxiliar aos profissionais da área no planejamento de redes de filas gerais.*

Palavras-chave: *Redes de Filas; Objetivos Confiltantes; Alocação de Áreas de Criculação; Simulated Annealing.*

Abstract: *In optimization of finite queueing networks, the buffers must be reduced, while the throughput must be maximized. Solutions can be improved by redistributing buffers, preserving the system capacity. A simulated annealing has been developed; a set of experiments was conducted. The insights obtained may helpful to practitioners analyzing queueing networks.*

Keywords: *Queueing Networks; Conflicting Objectives; Buffer Allocation; Simulated Annealing.*

1 Introdução

Sistemas de filas se encontram presentes em situações de incerteza sobre o fluxo de produtos, usuários, dentre outras. As configurações de filas em rede, considerando para cada fila uma taxa de chegada λ e uma taxa de serviço μ, determinam uma generalização natural para diversos sistemas que se busca modelar. Existem diversas discussões sobre o problema

[1]DEEST - UFOP. e-mail: *duarte.andersonr@gmail.com, gabriel.souza65@gmail.com.*
[2]DEST - UFMG. Pesquisa parcialmente financiada pelo CNPq e pela FAPEMIG.

de maximização do número de usuários atendidos por unidade de tempo (*throughput*), θ. O conhecimento de uma abordagem multi-objetivo maximizando o número de usuários atendidos, minimizando a soma áreas de circulação (*buffers*) alocadas nas filas do sistema e as soma das taxas de serviços leva a uma proposição de uma estratégia de pós-processamento para o aumento do *throughput*, preservando a soma de *buffers* alocados nas filas do sistema e as taxas de serviços através da realocação de *buffers* entre as filas do sistema.

Será tratado aqui, o problema de maximização de θ em redes de filas finitas, cada uma delas com servidor único e tempos de serviço geral. Em outras palavras, tratar-se-á de redes de filas $M/G/1/K$, na notação de Kendall [13]. Partindo das soluções propostas pela abordagem multi-objetivo, o foco central estará em redistribuir *buffers* alocados entre as filas do sistema, preservando a soma para todo o sistema, com o interesse em aumentar o valor de θ.

Uma abordagem já conhecida (veja em [8]) busca maximizar θ, simultaneamente com a minimização do espaço total de espera alocado nas filas do sistema ($\sum K_i$), e também a minimização da soma das taxas de serviço ($\sum \mu_i$), para uma topologia pré-especificada no sistema de filas. Potenciais usuários destes modelos de otimização baseados em redes de filas finitas incluem cientistas da computação e engenheiros de produção. De fato, tais modelos podem auxiliar na compreensão e melhoria de vários sistemas reais, incluindo sistemas de manufatura [1], de produção [20] e de saúde [11], sistemas de tráfego de veículos e de pedestres [9, 7, 10], sistemas de computação e de comunicação [6], aplicações baseadas na *web*, configuradas em camadas [5] e com requisitos de qualidade de serviço (QoS) definidos em termos de tempo de resposta, *throughput*, disponibilidade e segurança [17].

Foi utilizada para este trabalho, uma rede de filas com servidor único, cujos tempos entre chegadas são distribuídos exponencialmente e os tempos de serviço tem distribuição geral, configuradas em uma topologia série-paralelo acíclica arbitrária. Um exemplo desse tipo de rede pode ser visto na Figura 1.

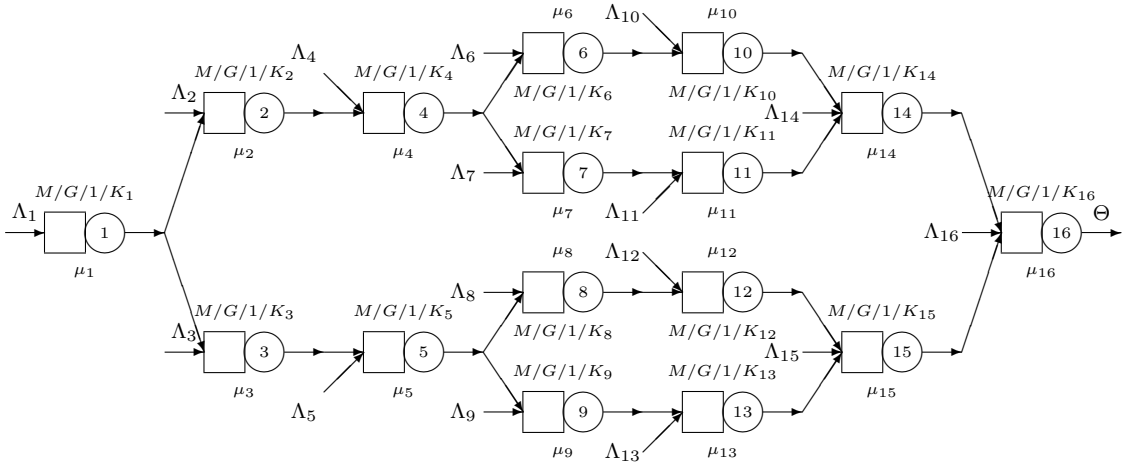

Figura 1: Um rede complexa (adaptada de Smith & Cruz [19])

Um caso mais simples de estudo pode ser obtido através da análise de filas $M/M/1/K$, em que tanto os tempos entre chegadas seguem distribuição exponencial sendo independentes entre si, bem como os tempos de serviço. Por outro lado, em muitas situações de interesse, os tempos entre chegadas preservam a distribuição exponencial, mas os tempos de serviço não, ou seja,

têm distribuição geral. Tal situação pode ser ilustrada através de sistemas que usualmente são denominados hipoexponenciais e hiperexponenciais. Em sistemas hipoexponenciais, o quadrado do coeficiente de variação dos tempos de serviços (razão entre a variância e o quadrado da média) é menor que a unidade. Já os sistemas hiperexponenciais apresentam o quadrado do coeficiente de variação dos tempos de serviços maior que a unidade. Vale ressaltar que para os sistemas markovianos (tempos de serviço independentes com distribuição exponencial) o quadrado do coeficiente de variação é igual a 1. Os casos hipoexponencial e hiperexponencial serão utilizados neste trabalho para caracterizar filas do tipo $M/G/1/K$.

O tipo de problema de interesse é o de desenvolvimento de algoritmos para otimização do *throughput* de uma rede de filas finitas gerais. O método proposto em Cruz et al. [8] executa simultaneamente a maximização de θ, a minimização de $\sum K_i$, e de $\sum \mu_i$. Contudo não garante vasculhar as diversas combinações para a distribuição de *buffers* entre as filas do sistema que preservem fixo o valor de $\sum K_i$. Em outras palavras, não se assegura que foi realizada a busca por algumas das possíveis soluções que preservam a mesma capacidade total do sistema e fornecem uma maior taxa de atendimento θ.

Alguma recombinação na distribuição de *buffers* entre as filas do sistema pode gerar melhoria no *throughput* do sistema (θ). Este fato poderá ser verificado através de uma tentativa de estratégia de pós-processamento (busca local) através do algoritmo *simulated annealing*. O objetivo central deste trabalho é buscar uma adequada distribuição de *buffers* visando desempenho ótimo do sistema de filas, mensurado através do *throughput* θ.

Formulado dessa forma, o problema em estudo, se torna uma clara analogia ao clássico problema da mochila (em inglês, *Knapsack problem*) de otimização combinatória. Trata-se da necessidade de preencher uma mochila com objetos de diferentes pesos e valores. O objetivo é preencher a mochila com o maior valor possível, não ultrapassando a capacidade de peso máximo pré-fixada pela mochila. A formulação do problema é extremamente simples, porém sua solução é reconhecidamente mais complexa.

A organização desse trabalho é a seguinte. A seção 2 discute a formulação matemática do problema, a metodologia multi-objetivo proposta em Cruz et al. [8] e introduz a abordagem de pós-processamento através do algoritmo *simulated annealing*. A seção 3 apresenta resultados de simulação para a rede de filas apresentada na Figura 1 usando diferentes valores para o quadrado do coeficiente de variação dos tempos de serviço, visando abordar sistemas exponenciais, hipoexponenciais e hiperexponenciais. A seção 4 discute conclusões alcançadas e possíveis propostas de continuidade.

2 Material e métodos

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

2.1 Otimização multi-objetivo

Em Cruz et al. [8] se encontra descrito um método de otimização multi-objetivo para determinar um conjunto de soluções eficientes para três objetivos simultâneos, $f_1(\mathbf{K}) = \sum_{\forall i \in N} K_i$, $f_2(\boldsymbol{\mu}) = \sum_{\forall i \in N} \mu_i$ e $f_3(\mathbf{K}, \boldsymbol{\mu}) = \theta(\mathbf{K}, \boldsymbol{\mu})$. Com esse método multi-objetivo proposto, o efeito da substituição de soluções pode ser avaliado por quem for tomar uma decisão. Além do mais, o tratamento multi-objetivo também permite ao usuário promover um aumento em um objetivo (por exemplo, a taxa de atendimento) enquanto reduz simultaneamente outro objetivo (por exemplo, a alocação das áreas de espera e das taxas de serviço). Os algoritmos usados em Cruz et al. [8] combinam um algoritmo evolucionário multi-objetivo (MOEA) associado ao método de expansão generalizada (GEM), sendo este um método bem sucedido para a obtenção de boas aproximações de medidas de desempenho em redes de filas finitas [14].

O algoritmo de análise de desempenho produz uma estimativa para a taxa de atendimento da rede de filas finitas gerais, $\theta(\mathbf{K}, \boldsymbol{\mu})$, conhecida a topologia da rede e dados os vetores de alocação de áreas de espera, $\mathbf{K}$, e das taxas de serviço, $\boldsymbol{\mu}$. O algoritmo de otimização permite a maximização simultânea da taxa de atendimento, enquanto são minimizadas a alocação de áreas de espera e as taxas de serviço.

Considerando filas simples, para que seja possível a maximização da taxa de atendimento $\theta(\mathbf{K}, \boldsymbol{\mu})$ é necessário algum método para estimá-la. Em uma *única fila* $M/G/1/K$, o procedimento de estimação pode ser executado através de uma forma matemática fechada, computacionalmente eficiente, para a probabilidade p_K de ocorrência de bloqueio na fila. O método para obtenção da estimativa desta probabilidade apresentado a seguir, proposto por Smith [18], baseia-se na aproximação de dois momentos de Kimura [15] e é bastante eficaz.

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

em que $\rho = \lambda/\mu$ é a intensidade de tráfego (note que $\rho < 1$, pois, caso contrário, isto é, se $\lambda > \mu$, a fila cresce indefinidamente; CV^2 é o quadrado do coeficiente de variação da variável aleatória tempo de serviço, S, ou seja, $CV^2 = \mathrm{Var}(S)/\mathrm{E}(S)^2$. Resultados empíricos em Cruz et al. [7] indicam que esta aproximação para p_K é bastante acurada, para uma vasta gama de valores.

Para obter a taxa de atendimento para uma única fila $M/G/1/K$ é necessário o ajuste da taxa de chegada. Na verdade uma fração p_K dos recém-chegados não pode ingressar no sistema, porque eles vêm quando não há espaço deixado na área de espera. Assim, a taxa real de chegadas para ingressar no sistema deve ser ajustada convenientemente.

$$\lambda_{\text{eff}} = \lambda(1 - p_K). \tag{2}$$

Assim, a taxa de atendimento efetiva pode ser dada pela expressão seguinte:

$$\theta = \lambda_{\text{eff}} = \lambda(1 - p_K). \tag{3}$$

Já para uma *rede de filas*, a estimação da taxa de atendimento é consideravelmente mais

complicada. O método de expansão generalizada (GEM) é um algoritmo que tem sido utilizado com sucesso na estimação do desempenho de redes acíclicas de filas finitas arbitrariamente configuradas. O GEM é uma combinação de decomposição nó-a-nó e tentativas repetidas, na qual cada fila é analisada separadamente e correções são feitas para contabilizar os efeitos de inter-relacionamentos entre as filas finitas da rede. O GEM considera que os bloqueios ocorrem se, após o serviço ser concluído em uma fila, a fila seguinte estiver completamente cheia (isto é, um cliente está em serviço no único servidor da fila seguinte e todos os espaços de espera nela estão ocupados).

A estratégia de otimização usada em Cruz et al. [8] baseia-se no algoritmo NSGA-II [12]. Na aplicação dos algoritmos genéticos, os operadores de *seleção* e de *elitismo* precisam ser estruturados, para identificar corretamente as condições de otimalidade. O elitismo é baseado no conceito de dominância. Assim, um ponto $\mathbf{x}_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ é dito dominar um outro ponto $\mathbf{x}_j = (x_{j_1}, x_{j_2}, \ldots, x_{j_n})$ se $\mathbf{x}_i$ é inferior a $\mathbf{x}_j$ em pelo menos um objetivo ($f_k(\mathbf{x}_i) < f_k(\mathbf{x}_j)$, para minimização e é não superior nos demais outros objetivos ($f_\ell(\mathbf{x}_i) \not> f_\ell(\mathbf{x}_j)$, para minimização.

Os operadores de *cruzamento* e de *mutação* são um pouco independentes da natureza multi-objetivo do problema, mas são altamente dependentes na aplicação. Para o problema em questão, o mecanismo de cruzamento uniforme é selecionado [2]. O cruzamento uniforme é bem popular em codificações de variáveis múltiplas. Neste mecanismo, os cruzamentos são realizados para cada variável com uma probabilidade (`rateCr`) que está de acordo com o operador de cruzamento. O operador de cruzamento usado no algoritmo é o operador cruzamento binário simulado.

2.2 Pós-processamento

A metodologia proposta em Cruz et al. [8] fornece um conjunto Pareto ótimo, mas não garante vasculhar todas as combinações de alocação de *buffers*, $\mathbf{K} = (K_1, \ldots, K_N)$. Em outras palavras, os operadores genéticos do algoritmo não asseguram a procura por algumas das possíveis soluções que preservam a mesma capacidade total ($\sum K_i$) e fornecem um maior *throughput* (θ). Assim, alguma recombinação na distribuição de *buffers* entre as filas do sistema pode gerar melhoria no *throughput* do sistema. Este possível fato será avaliado através de alguma tentativa de estratégia de pós-processamento (busca local).

Nesse cenário, o atual problema de otimização, agora para o pós-processamento será descrito por:

$$\text{maximize}\, \theta(\mathbf{K}, \boldsymbol{\mu}), \tag{4}$$

sujeito a

$$\mathbf{K} = (K_1, \ldots, K_N); K_i \in \{1, 2, \ldots\}, \forall i \in N, \tag{5}$$

$$\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N), \forall i \in N, \tag{6}$$

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

Será utilizada uma proposta de heurística otimizadora para obter soluções para o problema

de otimização (4)–(6), partindo do conjunto inicial de soluções candidatas sendo formado pelas soluções fornecidas pelo algoritmo genético proposto em Cruz et al. [8]. A escolha por uma heurística se deve ao fato que para valores elevados de N, uma busca exaustiva se tornaria inviável do ponto de vista computacional. Uma heurística de otimização que pode se adaptar bem a natureza do problema em questão é o clássico algoritmo *simulated annealing*.

O algoritmo *simulated annealing* descrito inicialmente (veja em [16, 4]) é inspirado no processo de recozimento de sistemas físicos. Os princípios básicos têm origens em termodinâmica estatística, uma analogia com o recozimento de sólidos poderia fornecer uma estrutura para o desenvolvimento de um algoritmo genérico de otimização capaz de escapar de ótimos locais na busca pelo ótimo global. Desde a sua introdução, como um método de otimização combinatorial, o *simulated annealing* vem sendo vastamente utilizado em diversas áreas, tais como projeto de circuitos integrados auxiliado por computador, processamento de imagem, redes neurais, etc. O método não utiliza uma "estratégia" (uma lei por exemplo para convergência total), assumindo assim na maioria das vezes um mínimo ou máximo que não é o global, mais se configura como uma boa opção para solução do problema em questão, como pode ser visto em Spinellis et al. [21]

De uma forma simplista, o método depende de um funcional objetivo de otimização, neste caso $\theta(\mathbf{K})$, e de um critério de vizinhança entre as soluções candidatas. Busca-se reproduzir uma cadeia markoviana cujo espaço de estados e composto por um conjunto de possíveis soluções para o problema de otimização em estudo. O *simulated annealing* opera da seguinte forma: se o $n-$ésimo estado da cadeia de Markov é uma possível solução $\mathbf{K}_1$, então alguma solução vizinha da solução $\mathbf{K}_1$ é selecionada aleatoriamente; se o estado vizinho escolhido for $\mathbf{K}_2$, então o próximo estado da cadeia será $\mathbf{K}_2$, se este for superior a $\mathbf{K}_1$ avaliando através do funcional objetivo do problema. Caso contrário, o próximo estado da cadeia ainda poderá ser $\mathbf{K}_2$ com uma probabilidade p, ou então a cadeia se manterá em $\mathbf{K}_1$ com probabilidade $1-p$. A escolha do valor p, em geral, é dependente do número de passos já executados pela cadeia de Markov e também pelo acréscimo ou decréscimo na função objetivo, gerado pelo possível troca entre as soluções $\mathbf{K}_1$ e $\mathbf{K}_2$.

Uma escolha computacionalmente usual de p é $e^{C\log(1+n)}$ em que a constante C é dada por $C = -|\theta(\mathbf{K}_1) - \theta(\mathbf{K}_2)|$ e n é o número de passos dados pela cadeia de Markov até o instante corrente. Gerando um conjunto de m passos sucessivos da cadeia, $\Omega_K = \{\mathbf{K}_1, \ldots, \mathbf{K}_m\}$, pode-se estimar a solução ótima θ^* por $\theta(\mathbf{K}_i)$, em que $\mathbf{K}_i$ otimiza θ com respeito ao conjunto Ω_K.

A proposta em estudo é de uma busca dentre as possíveis alocações no vetor $(K_1, \ldots, K_N)$ através do algoritmo *simulated annealing*, visando obter a configuração que maximiza o *throughput* (θ) pensando em um espaço alocado total, pré-fixado $\sum K_i$, mantendo também fixo o vetor de taxas de atendimento $\boldsymbol{\mu}$. É importante a definição do conceito de vizinhança entre as possíveis alocações $(K_1, \ldots, K_N)$. Dado dois valores aleatórios i e j em $\{1, \ldots, N\}$, um possível vizinho da alocação $(K_1, \ldots, K_N)$ pode ser definido considerando a nova alocação dada por $(K_1, \ldots, K_{i-1}, K_i - 1, K_{i+1}, \ldots, K_{j-1}, K_j + 1, K_{j+1}, \ldots, K_N)$. Note que modificações na alocação individual de *buffers* entre as filas, mantendo fixo o espaço total alocado levará a alterações no valor da função objetivo $\theta(.)$.

3 Resultados e discussões

A rede complexa, apresentada na Figura 1, já foi analisada em Cruz et al. [8] através da metodologia que utiliza o algoritmo genético multi-objetivo. Neste trabalho, cada uma das soluções do conjunto Pareto solução fornecido pelo algoritmo genético foi utilizado como uma solução inicial para o algoritmo *simulated annealing*. O interesse central é verificar se a busca local pode gerar melhoria nas soluções fornecidas pelo algoritmo genético através da realocação de *buffers* entre as filas do sistema, preservando a capacidade total no sistema.

Foram analisados três diferentes valores para o quadrado dos coeficientes de variação dos tempos de serviços, $CV^2 = \{0{,}5; 1{,}0; 1{,}5\}$, visando caracterizar sistemas que fossem exponenciais, hipoexponenciais e hiperexponenciais. Em todos os casos trabalhou-se com uma entrada única no sistema através da primeira fila do sistema, sendo $\lambda = 5{,}0$. O algoritmo *simulated annealing* foi calibrado para tentar 10.000 possibilidades de trocas.

Em todas as situações experimentais, o conjunto de soluções iniciais era composto por 400 soluções fornecidas pelo algoritmo genético. A Tabela 1 apresenta uma avaliação dos resultados obtidos para os três valores do quadrado do coeficiente de variação.

Tabela 1: Resultados obtidos através da estratégia de pós-processamento.

CV^2	%(SIM)	%(SI3)	%(SM3)	%(S3I)
0,5	15,75	20,00	95,25	75,00
1,0	10,25	20,25	100,00	50,62
1,5	4,25	20,25	100,00	21,00

%(SIM) — Percentual de soluções iniciais que foram melhoradas;
%(SI3) — Percentual de soluções iniciais com $\theta < 3$;
%(SM3) — Percentual de soluções iniciais com $\theta < 3$ entre as soluções que foram melhoradas;
%(S3I) — Percentual de soluções iniciais com $\theta < 3$ que foram melhoradas em relação às soluções iniciais;

Os valores apresentados na Tabela 1 mostram uma avaliação para todo o conjunto de soluções inicialmente fornecidas pelo algoritmo genético. Também consideram, de forma particular, as soluções fornecidas pelo algoritmo genético cujo *throughput* era inferior ao valor 3. Inicialmente nota-se um aumento do volume de soluções iniciais que foram melhoradas pela estratégia de pós-processamento a medida que o quadrado do coeficiente de variação diminui.

Observa-se que em todos os cenários avaliados, o volume de soluções fornecidas pelo algoritmo genético com *throughput* inferior a 3, girava em torno de 20% das soluções. Dentre estas soluções, com $\theta < 3$, todas apresentavam capacidade total do sistema menor que 400, as demais soluções apresentavam valores para a capacidade total do sistema bem mais elevados (entre 400 e 1.400). Este fato leva a concluir que o algoritmo genético tende a utilizar altos valores de capacidade total do sistema para a obtenção de valores θ elevados (superiores a 3).

A grande maioria das soluções em que a técnica de pós-processamento alcançou melhoria era de casos com $\theta < 3$ e capacidade total baixa (menor que 400). Nota-se que o algoritmo genético é eficaz para fornecer soluções para os sistemas com uma alta capacidade de investimento em espaço de *buffers*. Por outro lado, a estratégia de pós-processamento através do algoritmo *simulated annealing* tende a fornecer melhorias em soluções para os sistemas com capacidade de investimento em espaço de áreas de circulação mais baixa, privilegiando assim sistemas com menor volume de recursos.

Inicialmente será feita uma análise de resultados considerando o quadrado do coeficiente de variação ($CV^2 = 1{,}0$) que servem para ilustrar a situação de filas M/M/1/K. Considerando a Figura 2, é importante salientar que as soluções apresentadas graficamente são pontos de um espaço tridimensional, mas como os valores μ_i estão fixados de acordo com a solução inicial do algoritimo genético, os pontos foram projetados no espaço bidimensional $\theta \times \sum K_i$. Esta projeção deixa uma falsa impressão de existência de pontos dominados, mas na verdade são todos pontos não-dominados no espaço tridimensional. Dessa forma, os pontos que sofreram alterações através do algoritmo *simulated annealing* eram pontos da forma $(\sum K_i, \sum \mu_i, \theta)$ e foram substituídos por pontos da forma $(\sum K_i, \sum \mu_i, \theta + \varepsilon)$ com $\varepsilon > 0$. Nos casos de alteração da solução através do pós-processamento, os pontos originais se tornaram dominados e os novos pontos (modificados pelo *simulated annealing*) são pontos não-dominados. Este mesmo raciocínio pode ser utilizado nas próximas análises para sistemas hipoexponenciais e hiperexponenciais nas avaliação das Figuras 3 e 4 que serão apresentadas posteriormente.

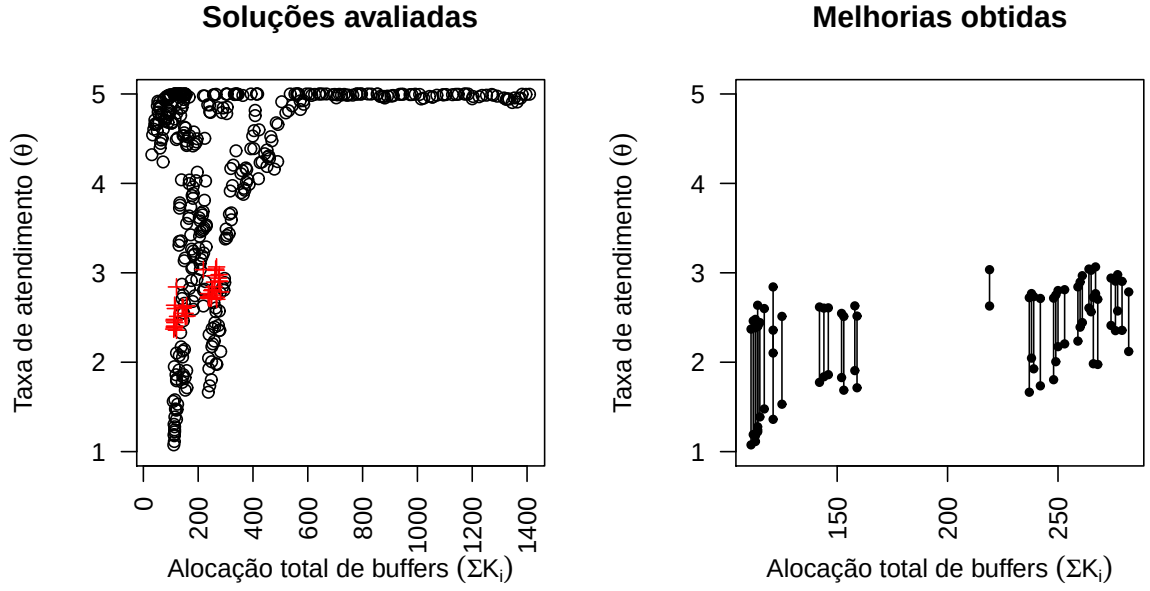

Figura 2: Sistema de filas com $CV^2 = 1$: gráfico à esqueda com as soluções inicialmente fornecidas através do algoritmo genético ($\circ$) e as soluções que foram melhoradas através do algoritmo *simulated annealing* ($+$); gráfico à direita com segmentos mostrando a melhoria alcançada em termos de θ.

A Figura 2 mostra dois gráficos, à esquerda as 400 soluções iniciais fornecidas através do algoritmo genético são apresentadas e também as soluções que foram melhoradas através da estratégia de pós-processamento através do algoritmo *simulated annealing*, à direita os segmentos ilustram a melhoria obtida para as soluções que foram modificadas.

Avaliando os gráficos que destacam os pontos que sofreram melhorias, verifica-se que a capacidade total era inferior a 300. Nota-se um maior ganho em throghput para as situações com menor alocação total de áreas de circulação. Esta observação confirma a capacidade do pós-processamento em fornecer melhorias para sistemas com menor volume de recursos.

Uma segunda análise leva em conta os resultados considerando os casos com quadrado do coeficiente de variação ($CV^2 = 0{,}5$ e $CV^2 = 1{,}5$) que servem para ilustrar a situação de filas

M/G/1/K para sistemas hipoexponenciais e hiperexponenciais, respectivamente. A Figura 3 apresenta dois gráficos análogos aos da análise anterior, agora ilustrando os resultados para sistemas hipoexponenciais.

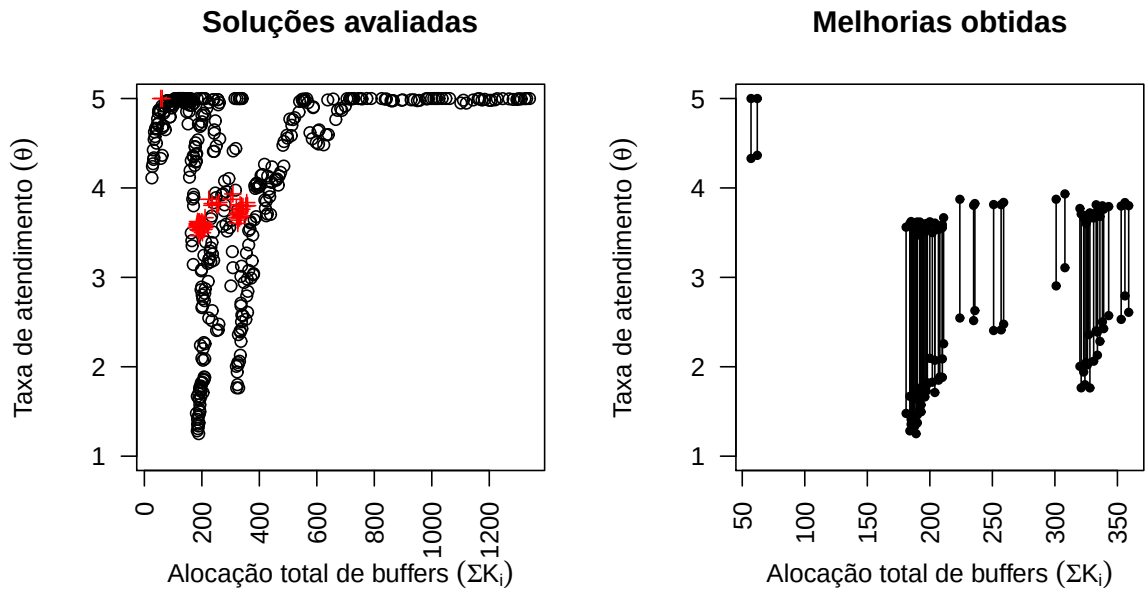

Figura 3: Sistema de filas com $CV^2 = 0{,}5$: gráfico à esqueda com as soluções inicialmente fornecidas através do algoritmo genético ($\circ$) e as soluções que foram melhoradas através do algoritmo *simulated annealing* ($+$); gráfico à direita com segmentos mostrando a melhoria alcançada em termos de θ.

Novamente as soluções que foram modificadas pelo pós-processamento apresentam valores mais baixos para a capacidade total (inferiores a 400). Considerando o aumento alcançado em termos de *throughput*, os resultados obtidos são mais significativos que os vistos na Figura 2. Trata-se de um resultado previsível. Isto se deve a um menor efeito de variabilidade nos tempos de serviço. Não existe uma clara regularidade entre o ganho em *throughput* e a capacidade total, mas é fácil observar que quanto menor o valor θ da solução inicialmente fornecida pelo algoritmo genético, maior o ganho obtido através da estretágia de pós-processamento.

A Figura 4 mostra os gráficos para a avaliação de sistemas hiperexponenciais com $CV^2 = 1{,}5$. Assim como os casos anteriores, as soluções que foram modificadas pelo pós-processamento apresentam valores baixos para a capacidade total, neste casos todas as soluções com espaço total de áreas de circulação inferiores a 320. O volume de soluções que foram melhoradas pelo algoritmo *simulated annealing* neste caso é inferior às avaliações anteriores, trata-se de um fato decorrente do maior efeito de variabilidade nos tempos de serviço para as filas do sistema. Mesmo com um volume inferior no número de soluções melhoradas, a técnica de pós-processamento mantém a sua capacidade de fornecer melhorias para sistemas com menor volume de recursos.

Considerando o aumento alcançado em termos de *throughput*, os valores são semelhantes na maioria das situações de melhoria através do algoritmo *simulated annealing*. Analisando o ganho alcançado em θ, os resultados parecem inferiores os obtidos para $CV^2 = 0{,}5$ vistos na Figura 3, mas são ainda superiores aos alcançados para $CV^2 = 1{,}0$ observados através da Figura 2. Apesar da similaridade entre os ganhos obtidos em *throughput* para os pontos de melhoria, os

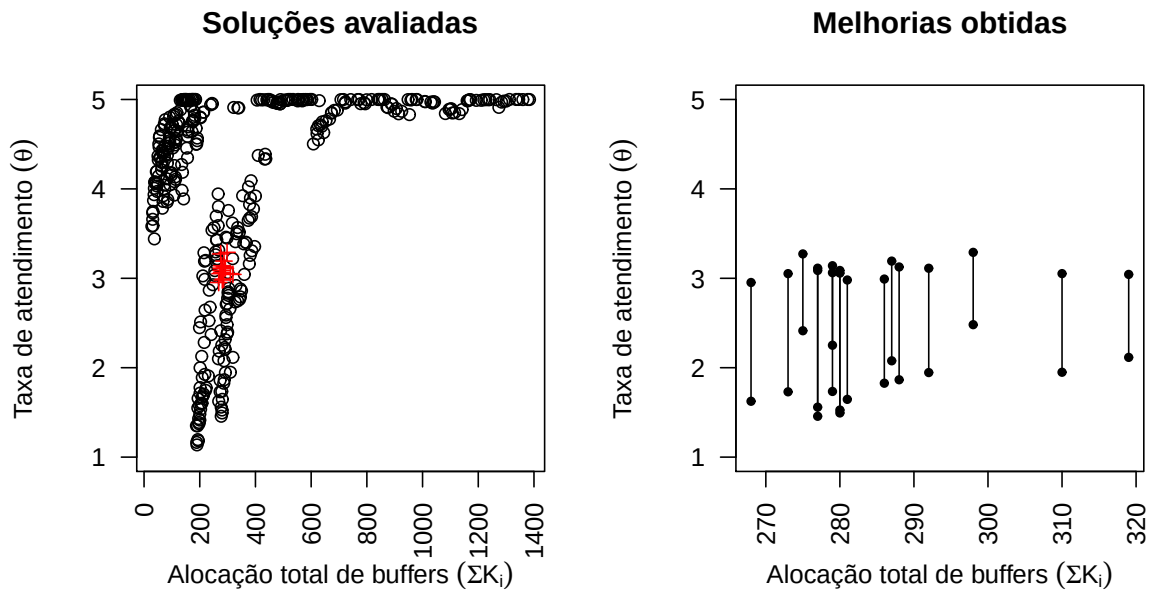

Figura 4: Sistema de filas com $CV^2 = 1{,}5$: gráfico à esqueda com as soluções inicialmente fornecidas através do algoritmo genético ($\circ$) e as soluções que foram melhoradas através do algoritmo *simulated annealing* ($+$); gráfico à direita com segmentos mostrando a melhoria alcançada em termos de θ.

casos de maior ganho são verificados para as situações com menores espaços totais de capacidade alocados, o que é bastante encorajador.

4 Conclusões

Os resultados apresentados confirmam a qualidade do algoritmo genético para produzir soluções eficiêntes para o problema em estudo em diversos cenários. Entretanto, a proposta de pós-processamento através do algoritmo *simulated annealing* se mostrou capaz de produzir melhorias em algumas das soluções fornecidas pelo algoritmo genético. Em particular, a soluções melhoradas são casos com capacidade total baixa, caracterizando assim uma capacidade do algoritmo em fornecer soluções eficientes para sistemas com um menor volume de recursos para investimentos em alocação de espaço.

Por se tratar de um algoritmo de baixo custo computacional, o acréscimo em tempo de processamento para as tentativas de pós-processamento é apenas uma fração do tempo computacional empregado para a obtenção das soluções através do algoritmo genético. Desta forma se torna bastante indicada a utilização do procedimento de pós-processamento aqui sugerido.

Considerando as soluções que foram melhoradas em sistemas com filas M/M/1/K, se a avaliação for restrita às soluções iniciais que possuiam valor $\theta < 3$ e $\sum K_i < 400$ observa-se que 50,62% das soluções foram melhoradas reforçando que a técnica de pós-processamento tende a melhorar a soluções nos sistemas com menos poder de investimento nas áreas de circulação. Dentre todas as soluções melhoradas, o aumento médio na taxa de atendimento (θ) foi da ordem de 49,69%.

Já para os sistemas hipoexponenciais, considerando $CV^2 = 0{,}5$, o aumento médio na taxa

de atendimento (θ) foi da ordem de 92,69%, e o percentual de soluções melhoradas considerando apenas as soluções iniciais que possuiam valor $\theta < 3$ e $\sum K_i < 400$ foi de 75,00%. Avaliando os sistemas hiperexponenciais considerando $CV^2 = 1{,}5$, o aumento médio em θ foi de 69,86%. Restringindo novamente aos casos com $\theta < 3$ e $\sum K_i < 400$ nas soluções iniciais, verificou-se melhoria em 21,00% das soluções.

É importante ressaltar que as soluções melhoradas através do algoritmo *simulated annealing* utilizam o mesma capacidade total para o espaço de espera alocado, entretanto redistribuem essa área total ao longo das filas do sistema. Dessa forma, não se espera custo adicional ao sistema de filas para a obtenção de taxas de atendimento mais elevadas. Apesar dos resultados já promissores, novos estudos precisam ser realizados. A adequação de outras possíveis estruturas de vizinhança (dependentes da capacidade total alocada na solução inicial) para o algoritmo *simulated annealing* são, por exemplo, propostas de continuidade que poderiam fornecer soluções ainda mais adequadas para o problema em estudo.

Referências

[1] ALVES, F. S. Q.; YEHIA, H. C.; PEDROSA, L. A. C.; CRUZ, F. R. B.; KERBACHE, L. Upper bounds on performance measures of heterogeneous $M/M/c$ queues. **Mathematical Problems in Engineering**. (Article ID 702834): 18 pages, 2011.

[2] BÄCK, T.; FOGEL, D.; MICHALEWICZ, Z. Handbook of Evolutionary Computation. In: Bäck, Fogel e Michalewicz (eds), **Institute of Physics Publishing and Oxford University Press**. 1997.

[3] BESAG, J.; GREEN, P.; HIGDON, D.; MENGERSEN, K. Bayesian Computation and Stochastic Systems (with Discussion), **Statistical Sci.**, v. 10, p. 3-67, 1995.

[4] CERNY, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, **Journal of Optimization Theory and Applications**, v. 45, p. 41-51, 1985.

[5] CHAUDHURI, K.; KOTHARI, A.; PENDAVINGH, R.; SWAMINATHAN, R.; TARJAN, R.; ZHOU, Y. Server allocation algorithms for tiered systems. **Algorithmica**. v. 48 (2), p. 129-146, 2007.

[6] CHEN, J.; HU, C.; JI, Z. An improved ARED algorithm for congestion control of network transmission. **Mathematical Problems in Engineering**. (Article ID 329035): 14 pages, 2010.

[7] CRUZ, F. R. B.; DUARTE, A. R.; VAN WOENSEL, T. Buffer allocation in general single-server queueing network. **Computers & Operations Research** v. 35 (11), p. 3581-3598, 2008.

[8] CRUZ, F. R. B.; KENDALL, G.; WHILE, L.; DUARTE, A. R.; BRITO, N. L. C. Throughput maximization of queueing networks with simultaneous minimization of service rates and buffers. **Mathematical Problems in Engineering**. (Article ID 348262): 19 pages, 2012.

[9] CRUZ, F. R. B.; SMITH, J. M.; QUEIROZ, D. C. Service and capacity allocation in $M/G/C/C$ state dependent queueing networks. **Computers & Operations Research** v. 32 (6), p. 1545-1563, 2005.

[10] CRUZ, F. R. B.; VAN WOENSEL, T.; SMITH, J. M.; LIECKENS K. On the system optimum of traffic assignment in $M/G/c/c$ state-dependent queueing networks. **European Journal of Operational Research** v. 201 (1), p. 183-193, 2010.

[11] DE BRUIN, A. M.; VAN ROSSUM, A. C.; VISSER, M. C.; KOOLE, G. M. Modeling the emergency cardiac in-patient flow: An application of queuing theory. **Health Care Management Science** v. 10 (2), p. 125-137, 2007.

[12] DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II, **IEEE Transactions on Evolutionary Computation**, v. 6 (2), p. 182-197, 2002.

[13] KENDALL, D. G. Stochastic processes occurring in the theory of queues and their analysis by the method of imbedded markov chains, **Annals Mathematical Statistics**, v. 24, p. 338-354, 1953.

[14] KERBACHE, L.; SMITH, J. M. Multi-objective routing within large scale facilities using open finite queueing networks, **European Journal of Operational Research**, v. 121 (1), p. 105-123, 2000.

[15] KIMURA, T. A transform-free approximation for the finite capacity $M/G/s$ queue, **Operations Research**, v. 44 (6), p. 984-988, 1996.

[16] KIRKPATRICK, S.; GELATT, C.; VECCHI, M. Optimization by simulated annealing, **Science**, v. 4598, p. 671-680, 1983.

[17] MENASCE D. A. QoS issues in web services. **IEEE Internet Computing** v. 06 (6), p. 72-75, 2002.

[18] SMITH, J. M. Optimal design and performance modelling of $M/G/1/K$ queueing systems, **Mathematical and Computer Modelling**, v. 39 (9-10), p. 1049-1081, 2004.

[19] SMITH, J. M.; CRUZ, F. R. B. The buffer allocation problem for general finite buffer queueing networks. **IIE Transactions** v. 37 (4), p. 343-365, 2005.

[20] SMITH, J. M.; CRUZ, F. R. B.; VAN WOENSEL T. Topological network design of general, finite, multi-server queueing networks. **European Journal of Operational Research** v. 201 (2), p. 427-441, 2010.

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