ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS APLICADO AO PROBLEMA DE PROGRAMAÇÃO DE ADMISSÃO DE PACIENTES

Leonardo dos Santos Lourenço Bastos[1]
lslbastos@tecgraf.puc-rio.br

Victor Abu-Marrul Carneiro da Cunha[1]
victor.cunha@tecgraf.puc-rio.br

Janaina Figueira Marchesi[1]
janaina.marchesi@tecgraf.puc-rio.br

Silvio Hamacher[1]
hamacher@puc-rio.br

[1] Departamento de Engenharia Industrial (DEI)
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225 – Gávea, Rio de Janeiro – RJ, 22451-900

RESUMO

Na gestão da saúde, apresentam-se diversos desafios no planejamento de recursos para garantir a satisfação da demanda no âmbito estratégico, tático e operacional, sendo que, neste último, uma das atividades relevantes é a gestão de leitos. Neste contexto, é empregada a Programação de Admissão de Pacientes (PASP), no qual pacientes devem ser alocados a leitos respeitando necessidades e preferências clínicas. Este estudo propõe uma aplicação do Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA) para o PASP. A partir dos resultados obtidos, verificou-se que o BRKGA apresentou soluções competitivas com as demais abordagens da literatura, com desvios percentuais entre 1,1-4,5% em relação às melhores soluções com baixo tempo computacional.

PALAVRAS CHAVE. Programação de Admissão de Pacientes, Algoritmo Genético de Chaves Aleatórias Viciadas, Metaheurística.

Tópicos: SA (PO na Área de Saúde), SE (PO em Serviços)

ABSTRACT

In healthcare management, there are several challenges in resource planning to ensure demand satisfaction, in strategic, tactical and operational levels, and in the latter, one of the relevant activities is bed management. In this context, the Patient Admission Scheduling Problem (PASP) is employed, in which patients must be assigned to beds, considering clinical needs and preferences. This study proposes an application of the Biased Random-Key Genetic Algorithm (BRKGA) to solve the PASP. From the results, the BRKGA presented competitive solutions compared to other methods in the literature, with deviations between 1.1-4.5% from the best known solutions with low computational time.

KEYWORDS. Patient Admission Scheduling Problem, Biased Random-Key Genetic Algorithm, Metaheuristics.

Topics: SA (PO in Healthcare), SE (PO in Services).

https://proceedings.science/p/85521?lang=pt-br

1. Introdução

Nos últimos anos, houve uma crescente demanda por serviços de saúde principalmente em decorrência do envelhecimento da população e de um aumento na conscientização sobre os cuidados preventivos [Granja et al., 2014]. Nesse contexto, busca-se alcançar o uso eficiente dos recursos e a satisfação dos pacientes pelos administradores de saúde a fim de atender adequadamente a demanda e enfrentar o aumento dos custos na prestação dos serviços [Turhan e Bilgen, 2017]. Abordar esse desafio envolve atender a questões práticas, tais como o controle de admissões, *design* de processos, planejamento agregado, alocação de capacidade e agendamento de consultas. Entre estas preocupações, o contexto da Programação de Admissão de Pacientes (*Patient Admission Scheduling Problem* - PASP) é de particular importância, pois tem implicações em vários outros problemas enfrentados nas organizações de saúde [Barz e Rajaram, 2015].

O PASP corresponde à designação de pacientes aos leitos em uma instituição de saúde (Ex: Hospital, Clínicas), considerando um horizonte de tempo determinado, de modo a maximizar a eficiência do tratamento, o conforto do paciente e a utilização hospitalar, ao mesmo tempo em que satisfaça todas as restrições médicas necessárias e que respeite as preferências dos pacientes, sempre que possível. Este problema vem sendo estudado na literatura nos níveis estratégico, tático e operacional, porém a maioria dos estudos se concentra na tomada de decisões a níveis estratégico e tático [Demeester et al., 2010].

A variante de nível operacional do PASP foi introduzida por Demeester et al. [2010] e o problema foi definido como NP-Difícil por Vancroonenburg et al. [2011]. Além disso, Demeester et al. [2010] publicaram um conjunto de 13 instâncias com base em dados hospitalares reais e um algoritmo de validação das soluções para novas pesquisas, e aplicaram um algoritmo híbrido de Busca Tabu com Busca *Token-Ring* (*Token-Ring Search*) para resolver o problema. Com isso, os autores obtiveram o primeiro conjunto de soluções para todas as instâncias, e, desde então, diversas abordagens de resolução foram utilizadas para tentar obter soluções com qualidade e em tempo computacional adequado. Bilgin et al. [2012] realizaram uma abordagem de hiper-heurísticas para solucionar o PASP e a Programação de Enfermeiras (*Nurse Rostering Problem*). Os resultados obtidos foram melhores que em Demeester et al. [2010] para as Instâncias 1 - 6. Ceschia e Schaerf [2011] aplicaram uma combinação de busca local multi-vizinhança com *Simulated Annealing*. O algoritmo adapta as relações de vizinhança para explicar diferentes pesos dos componentes da função objetivo. Os resultados obtidos foram melhores do que aqueles em Demeester et al. [2010] e Bilgin et al. [2012], para todas as instâncias. Hammouri e Alrifai [2014] relataram a primeira abordagem baseada em algoritmos evolutivos para o PASP clássico com base no Algoritmo de Otimização Baseada em Biogeografia (*Biogeography-based Optimization* - BBO), mas não conseguiram melhorar o estado da arte. Uma heurística baseada em otimização foi proposta por Range et al. [2014], no qual as técnicas de geração de colunas e de restrição dinâmica foram usadas para alcançar limites mais rígidos e gerar novas melhores soluções para um subconjunto da instâncias de referência. Os autores utilizaram heurísticas para resolução de sub-problemas (*pricing problems*) dentro do método de geração de colunas. Usando meta-heurísticas com base em um algoritmo adaptativo não-linear (*Great Deluge*), Kifah e Abdullah [2015] resolveram um subconjunto das instâncias de referência em um tempo computacional consideravelmente menor, mas não conseguiram melhorar os valores de função objetivo de Range et al. [2014]. Com foco na diminuição do tempo de execução, Turhan e Bilgen [2017] aplicaram duas heurísticas baseadas em programação inteira mista (*Fix-and-Relax* - F&R e *Fix-and-Optimize* - F&O) para sub-problemas decompostos do PASP e obtiveram soluções com lacunas de otimização da ordem de 5-15% em menos de três minutos.

Uma variante dinâmica do PASP clássico foi introduzida por Ceschia e Schaerf [2012] para lidar com tempo de internação incertos, atrasos de admissão e pacientes não eletivos. Esta variante foi definida como Programação de Pacientes sob Incerteza (*Patient Admission Scheduling Problem under Uncertainty* – PASU), resolvida usando *Simulated Annealing*. Além disso, Ceschia

https://proceedings.science/p/85521?lang=pt-br

e Schaerf [2016] incorporaram restrições de salas de cirurgia, integradas à programação de cirurgias eletivas.

Neste contexto, este trabalho considerou o estudo em relação ao PASP operacional *offline* estático, conforme definido em Demeester et al. [2010]. Para resolução do problema, propomos um algoritmo genético de chaves aleatórias viciadas (*Biased Random-Key Genetic Algorithm* - BRKGA), proposto inicialmente por Gonçalves e Resende [2011].

Algoritmos genéticos são amplamente aplicados em problemas de *scheduling* em saúde, por exemplo: planejamento de salas de cirurgia [Fei et al., 2010; Marchesi e Pacheco, 2016]; programação de equipe de enfermagem [Tsai e Li, 2009; Moz e Pato, 2007]; programação de equipe de médicos residentes [Wang et al., 2007]; programação do fluxo de pacientes em emergências [Azadeh et al., 2014]; agendamento de consultas [Qu et al., 2013]; admissão de pacientes com objetivo de estabelecer o número de pacientes por tipo de cirurgia a ser admitido em um dia a fim de diminuir o tempo de espera para admissão [Zhang et al., 2013]; entre outras aplicações. Em relação ao BRKGA, as aplicações em problemas de saúde ainda são poucas. Apenas duas aplicações foram encontradas em programação de cirurgias eletivas que usam um algoritmo genético basedo no BRKGA [Marques et al., 2014; Marques e Captivo, 2015] e uma aplicação em coleta de amostras de sangue em laboratórios clínicos [Grasas et al., 2014].

A escolha do BRKGA nesse estudo se baseia no seu sucesso em vários problemas de otimização combinatória, como por exemplo: problema de cobertura, dimensionamento de lote, layout, planejamento de transporte, entre outros [Damm et al., 2016]. Dessa forma, as principais contribuições do estudo são duas. Primeiro, propõe-se, a que se sabe, a primeira aplicação da metaheurística BRKGA para o PASP. Segundo, utiliza-se dessa proposta de solução para obter respostas para um subconjunto das instâncias de *benchmark* alcançando um balanceamento entre tempos computacionais e valor de solução comparado às demais abordagens de soluções estudadas na literatura.

O restante deste artigo está organizado da seguinte forma: além desta introdução, a Seção 2 apresenta a descrição do problema; a abordagem de solução é descrita na Seção 3; na Seção 4 apresentamos os experimentos computacionais e as análises dos resultados; na Seção 5 são apontadas as considerações finais e sugestões de trabalhos futuros.

2. Descrição do Problema

A definição, os detalhes e terminologia do problema original são fornecidos por Demeester et al. [2010], porém são relatados aqui para tornar o artigo autocontido. Nesta seção, descreve-se ainda uma visão detalhada das restrições usadas na formulação proposta destacando as diferenças em relação às usadas por Demeester et al. [2010].

2.1. Terminologia e definição do problema

Apresentamos a terminologia que será usada no restante deste artigo: Um horizonte de planejamento consiste em um conjunto de dias subsequentes de comprimento finito e fixo dentro dos quais os dias de admissão e alta são especificados. Logo, o tempo de permanência (*Length of Stay* - LoS) de cada paciente, definido em termos de dias de admissão e de alta, é conhecido antecipadamente.

Um paciente é uma pessoa que precisa de tratamento médico, e apenas os pacientes eletivos (aqueles que ficam mais de uma noite) devem ser programados. Os pacientes eletivos podem ser classificados como pacientes planejados ou admitidos. Os primeiros são pacientes que aguardam a hospitalização, com dias de admissão e alta, e os últimos consistem em pacientes que já ocupam um dos leitos hospitalares.

Cada leito pertence a um quarto, que possui capacidade a qual define o seu tipo: individual (um leito), duplo (dois leitos) ou enfermarias (quatro leitos), e cada paciente tem preferência por um certo tipo de quarto, denominada "preferência do quarto". Cada quarto pertence a um departamento

https://proceedings.science/p/85521?lang=pt-br

especializado no tratamento de certas patologias. Portanto, cada departamento atende a um conjunto de especialidades, cujos requisitos são atendidos por certos quartos. Além disso, cada departamento e quarto tem seu próprio grau de prioridade para essas especialidades.

Os quartos possuem propriedades que são definidas em termos de diferentes equipamentos disponíveis, como oxigênio e telemetria. Os leitos no mesmo quarto se beneficiam do mesmo equipamento. Dependendo do tratamento, um paciente pode exigir ou preferir ser alocado para um quarto com determinado equipamento.

Cada quarto tem sua própria política de gênero. Os quartos que exigem que os pacientes sejam do mesmo gênero aplicam a política M (somente Masculino) ou a política F (somente Feminino). Em contraste, os quartos onde ambos os sexos são permitidos impõem a política N (gênero misto a qualquer momento) ou a política D (qualquer gênero pode estar presente, no entanto, em qualquer noite, apenas pacientes do mesmo gênero devem estar presentes). A política de idade é definida para os departamentos que impõem um limite de idade máximo ou mínimo para admissão de pacientes. As transferências correspondem à troca de pacientes entre os quartos durante todo o período de permanência e devem ser evitadas.

Nesse contexto, o objetivo do PASP é alocar pacientes aos leitos sujeitos a políticas de departamento e quarto assim como os requisitos de tratamento de cada paciente. Aumentar a satisfação do paciente está diretamente associado à capacidade de atender suas preferências evitando transferências. A solução do PASP tem como objetivo a minimização de penalidades devido à violação de restrições em relação às preferências e necessidades do paciente e às políticas de quarto. A seguir, são apresentadas as restrições definidas para o PASP e as penalidades associadas a violação dessas restrições são discutidas.

2.2. Restrições e penalidades

As restrições são categorizadas em tipo *hard* (HC), as quais devem ser estritamente satisfeitas para fornecer uma solução viável e tipo *soft* (SC), as quais podem ser violadas e estão associadas à uma penalidade considerada na função objetivo. O aumento do número de restrições tipo *soft* satisfeitas leva a melhores soluções. O conjunto inicialmente apresentado por Demeester et al. [2010] compreendem oito restrições hard e cinco restrições do tipo *soft*. Assim como feito por Turhan e Bilgen [2017], as restrições tipo *hard* (HC5)-(HC8) originalmente relatadas por Demeester et al. [2010] são consideradas como *soft* em nosso estudo, uma vez que a elas são associadas penalidades e são avaliadas na função objetivo. O conjunto de restrições inclui:

HC1: Durante o horizonte de planejamento, cada quarto deve estar disponível para receber um paciente.

HC2: A data de admissão, a data de alta e o tempo de permanência (LoS) não podem ser alterados.

HC3: O tempo de internação do paciente é contínuo e o mesmo deve ser alocado a um leito até a data de alta.

HC4: Apenas um paciente deve ser alocado para um determinado leito em uma mesma noite.

SC1: A política de gênero deve ser respeitada para cada quarto.

SC2: Um paciente deve ser alocado respeitando o limite de idade do departamento, quando aplicável.

SC3: Um paciente deve ser alocado para um quarto com propriedades necessárias para o tratamento dele.

SC4: Alguns pacientes devem ser alocados a um quarto individual quando necessário, por razões clínicas (por exemplo, quarentena).

https://proceedings.science/p/85521?lang=pt-br

SC5: A preferência de um paciente em relação ao tipo de quarto (individual, duplo ou enfermaria) deve ser considerada sempre que possível. Se o paciente é designado para um quarto em que o número de camas é maior do que a preferência, é incorrida uma violação. Caso contrário, nenhuma violação é considerada.

SC6: Um paciente deve ser designado para um departamento que atenda a sua especialidade de tratamento.

SC7: Um paciente deve ser alocado para um quarto que atenda a sua especialidade em primeiro grau de prioridade. Se o grau de prioridade é secundário ou terciário, uma violação é considerada como a diferença entre graus.

SC8: Um paciente deve ser alocado para um quarto com as propriedades preferidas. Esta restrição é uma versão complementar do SC3.

SC9: Durante a alocação, é possível que um paciente seja transferido, o que deve ser evitado. Cada transferência corresponde a uma violação.

Das restrições apresentadas, tem-se que HC1-HC4 são do tipo *hard* e devem ser necessariamente respeitadas. Em relação as restrições tipo *soft*, tem-se as que representam condições necessárias e desejáveis na alocação dos pacientes aos leitos, e a distinção entre elas é dada em termos dos valores de penalidade associada a cada uma. As condições necessárias são representadas por restrições *soft* para as quais um alto valor de penalidade está associado (SC1-SC4 e SC9). Por fim, as condições desejáveis são lançadas como restrições *soft* que podem ser violadas incorrendo em uma penalidade relativamente baixa (SC5-SC8). As penalidades são calculadas por paciente, por noite, e o custo de violar essas restrições é calculado durante todo o período de permanência. Para a consistência, foram adotados os mesmos valores de penalidade que os demais trabalhos que apresentaram solução para o PASP (Tabela 1).

Tabela 1: Penalidades por violar restrições tipo *soft*

Restrição	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9
Penalidade	5	10	5	10	0,8	1	1	2	11

3. Abordagem de Solução

Nesse estudo adotamos o problema reformulado do PASP proposto por Ceschia e Schaerf [2011]. No problema reformulado a alocação do paciente acontece para quartos em vez da alocação para leitos, originalmente sugerida por Demeester et al. [2010]. Isso é possível uma vez que leitos pertencentes ao mesmo quarto possuem as mesmas características. Na reformulação também se sugere usar uma matriz de penalidades paciente-quarto a qual representa o custo de atribuir um paciente a um quarto para questões de departamento, especialidade, propriedades dos quartos, política de idade e preferências dos pacientes.

Com base na terminologia definida na Seção 2 e na reformulação sugerida por Ceschia e Schaerf [2011], o PASP pode ser resolvido usando o BRKGA cuja solução seguiu uma abordagem de três passos. O primeiro passo compreendeu a fase de pré-processamento na qual foram combinadas as informações das restrições SC2-SC8 e da restrição SC1 exceto para quartos com a política de gênero D em uma matriz única que representa o custo de atribuir um paciente a um quarto. O segundo passo consistiu na implementação do BRKGA para o PASP e a parametrização do algoritmo por meio do qual soluções com melhor relação entre tempos computacionais e valor da solução foram alcançadas. O terceiro passo incluiu o pós-processamento da solução para obter

https://proceedings.science/p/85521?lang=pt-br

as alocações do paciente ao leito, que foram posteriormente validadas usando uma aplicação disponibilizada on-line[1] por Demeester et al. [2010]. Como os leitos de um mesmo quarto são iguais, com a informação sobre as atribuições paciente-quarto, a solução final paciente-leito foi obtida distribuindo aleatoriamente os pacientes aos leitos do quarto para o qual ele foi alocado. Os detalhes em relação ao BRKGA são dados a seguir.

3.1. Algoritmo genético de chaves aleatórias viciadas

No algoritmo genético de chaves aleatórias viciadas (BRKGA), proposto por Gonçalves e Resende [2011], cada solução é codificada em um cromossomo de números reais (chaves) no intervalo [0,1]. Para que uma solução seja interpretável, a mesma deve ser decodificada, para que o seu custo, com base em uma função de avaliação determinada para o problema, possa ser avaliado. Durante a evolução, o algoritmo subdivide a população de soluções em dois subconjuntos: ELITE e NÃO-ELITE. O primeiro deles contém as melhores soluções de uma determinada geração e o outro as demais. O parâmetro p_e define a fração da população que compõe o subconjunto elite. O algoritmo prioriza as melhores soluções da população, sendo esse o aspecto viciado do método, passando suas características para as gerações seguintes.

Inicialmente os cromossomos são construídos de forma aleatória e então as etapas de seleção, *crossover* e mutação, presentes em algoritmos genéticos, são realizadas a cada nova geração.

Para a criação de uma nova geração, o subconjunto elite é copiado da geração anterior e novos indivíduos gerados aleatoriamente são inseridos nessa nova população. Esses novos indivíduos representam a mutação presente em algoritmos genéticos e a fração da população que deve ser substituída por essas soluções mutantes é definida pelo parâmetro p_m. O restante da nova população é composta por filhos, gerados a partir da combinação de soluções existentes. Para gerar um filho, são selecionados aleatoriamente um parente do subconjunto elite e outro do subconjunto não-elite, sendo estes combinados por meio do crossover uniforme parametrizado proposto por Spears e De Jong [1991]. A probabilidade de um alelo do parente do subconjunto elite passar pra o filho é definida a partir do parâmetro ρ. Os novos indivíduos são decodificados e avaliados e o algoritmo é executado até que um critério de parada seja alcançado. Dentre os critérios mais comuns estão: tempo limite de execução, custo alvo de valor da função de avaliação e número máximo de gerações.

3.1.1. Cromossomo e decodificação

O cromossomo definido para o PASP tem como tamanho o número de pacientes que precisam ser alocados em cada instância do problema e representa a sequência na qual estes pacientes serão alocados aos quartos disponíveis. Para decodificá-lo, cada alelo é inicialmente associado a um paciente de acordo com a sua posição no cromossomo. O cromossomo é então ordenado a partir do valor das chaves contidas nos alelos, resultando como consequência em um sequenciamento dos pacientes, conforme indicado na Figura 1 em um exemplo com 10 pacientes.

A partir da sequência definida após a decodificação do cromossomo, cada paciente é atribuído à um quarto por meio de uma heurística de alocação estipulada para o problema. Essa heurística consiste em alocar o paciente ao quarto, que tenha disponibilidade de leito para atendê-lo e que resulte na menor penalização total na função de avaliação do problema no momento de sua alocação. Essa penalidade total é calculada a partir da matriz de penalidades, destacada nesta seção, e da penalidade por quarto dependente, quando existente. A transferência entre quartos durante a permanência de um paciente não é permitida pela heurística e um paciente não pode trocar de quarto após ter sido alocado.

[1] https://people.cs.kuleuven.be/ wim.vancroonenburg/pas/. Acessado em 08/06/2018.

https://proceedings.science/p/85521?lang=pt-br

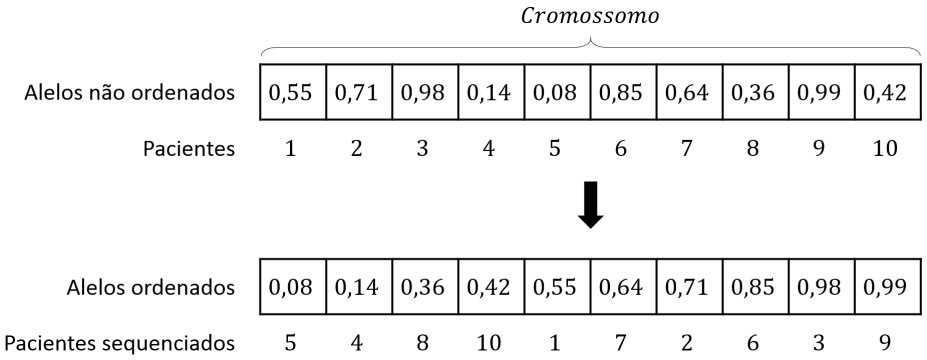


Figura 1: Decodificação do cromossomo para o PASP.

4. Experimentos Computacionais

O algoritmo proposto foi desenvolvido na linguagem *C++*, a partir do *framework* disponibilizado por Toso e Resende [2015], e executado em um processador Intel i7-3960X 3,3GHz com 64GB RAM. Os experimentos foram feitos com dez sementes diferentes a fim de verificar a estabilidade do método.

Para garantir uma melhor comparabilidade, ajustou-se o tempo de execução de cada solução da literatura com base no processador utilizado neste estudo, por meio da razão entre o processamento medido em *clocks* entre a referência e a proposta, conforme apresentado em Da Silva et al. [2012]. Para definição dos *clocks* dos processadores usados em literatura, considerou-se a maior semelhança das informações publicadas em cada referência[2].

Foram realizados dois testes para aplicação do BRKGA no PASP. No primeiro, com o objetivo de obter soluções com qualidade, considerou-se um limite de tempo de execução em cinco minutos (300s), que corresponde ao limite original usado em Demeester et al. [2010], ajustado pelo *clock*. No segundo, avaliou-se a capacidade do BRKGA em competir em tempo de execução com o método que obteve os menores tempos computacionais (Turhan e Bilgen [2017]), utilizando como critério de parada do algoritmo as soluções alcançadas por estes autores. Com isso, definem-se os testes como "Solução Proposta 1"e Solução Proposta 2", respectivamente.

A aplicação do BRKGA considerou o subconjunto das instâncias 1 à 8 do PASP. Estas instâncias foram escolhidas por serem mais homogêneas entre si, conforme discutido em Range et al. [2014], e por serem as mais abordadas na literatura. As características das instâncias (I) em termos de quantidade e leitos, quartos, pacientes (total e eletivos), departamentos (D), propriedades dos quartos (PQ), especialidades (E), o horizonte de planejamento (HP) e a taxa de ocupação por leito (TO, em %, calculada em Ceschia e Schaerf [2011]) são apresentadas na Tabela 2.

Tabela 2: Características das Instâncias 1 - 8. Fonte: Baseado em Ceschia e Schaerf [2011]

I	*Quartos*	*Leitos*	*Pacientes*	*Eletivos*	*HP*	*D*	*E*	*PQ*	*TO* (%)
1	98	286	693	652	14	4	4	2	59,69
2	151	465	778	755	14	6	6	2	59,98
3	131	395	757	708	14	5	5	2	57,07
4	155	471	782	746	14	6	6	2	54,23
5	102	325	631	587	14	4	4	2	59,69
6	104	313	726	685	14	4	4	2	64,38
7	162	472	770	519	14	6	6	4	33,52
8	148	441	895	895	21	6	6	4	43,90

[2]Obtenção dos valores de *clock* em: https://asteroidsathome.net/boinc/cpu_list.php. Acessado em 08/06/2018.

https://proceedings.science/p/85521?lang=pt-br

Range et al. [2014] comenta que as instâncias 1 - 6 possuem maior compatibilidade paciente-quarto em relação às demais, o que pode demonstrar maior facilidade em encontrar soluções viáveis para o problema. De acordo com Ceschia e Schaerf [2011], a menor taxa de ocupação dos leitos também está relacionada à obtenção de soluções mais rapidamente, visto que há mais facilidade em achar leitos para um dado paciente.

Dessa forma, apresentam-se a as configurações e metodologia utilizada para parametrizar o BRKGA e obter as soluções propostas. Em seguida, os resultados obtidos com o BRKGA são analisados e comparados às atuais soluções da literatura e sua qualidade é discutida.

4.1. Parametrização do BRKGA

Para a Solução Proposta 1, utilizou-se uma população de tamanho 500, e os demais parâmetros do BRKGA foram definidos com o uso da ferramenta *IRACE* (*Iterated Racing for Automatic Algorithm Configuration*) [López-Ibáñez et al., 2011], na qual a melhor configuração é determinada a partir de um procedimento iterativo automático. Foram testados os seguintes intervalos para os parâmetros: $p_e = [0,20, 0,40]$, $p_m = [0,05, 0,15]$ e $\rho = [0,55, 0,75]$ definidos em testes preliminares com o algoritmo. A melhor configuração dos parâmetros estabelecida pelo *IRACE* foi: $p_e = 0,39$; $p_m = 0,11$; e $\rho = 0,60$.

Na Solução Proposta 2, considerando os mesmos intervalos da parametrização do primeiro teste, experimentou-se variações em parâmetros que influenciam na velocidade de convergência do BRKGA. Com isso, obteve-se a configuração $p_e = 0,30$, $p_m = 0,05$, e $\rho = 0,60$ e população de tamanho 300. A redução no parâmetro p_e diminui o subconjunto elite aumentando o aspecto viciado do método, fazendo com que as características de um menor subconjunto de boas soluções prevaleçam durante as próximas gerações. Em relação ao resultado em termos de solução, a redução no parâmetro p_m e do tamanho da população diminuem a aleatoriedade do método e aceleram a sua execução, diminuindo o número de soluções mutantes que precisam ser criadas a cada geração e o número total de indivíduos que precisam ser avaliados e recombinados.

4.2. Análise e Discussão dos Resultados

Com a definição dos parâmetros, obtiveram-se os resultados para as soluções propostas. Os resultados da Solução Proposta 1 são demonstrados na Tabela 3, em termos de média das soluções, desvio-padrão (DP), valor máximo e mínimo, considerando as diferentes sementes testadas para cada instância (I), e a partir das melhores soluções encontradas na literatura (*Best Known Solutions - BKS*), foi calculado o desvio percentual do valor médio da Solução Proposta 1 em relação a essas soluções (ΔBKS).

Tabela 3: BRKGA - Resultados da Solução Proposta 1

I	*Média*	*DP*	*Mínimo*	*Máximo*	ΔBKS (%)
1	665,2	3,50	660,0	670,4	1,85
2	1153,2	3,65	1149,0	1158,8	2,30
3	782,4	4,14	776,2	792,0	2,03
4	1184,6	4,71	1174,6	1193,2	1,31
5	630,1	2,41	628,0	633,6	1,13
6	806,7	2,31	802,0	811,4	2,00
7	1243,2	10,10	1225,0	1256,0	4,48
8	4299,9	23,70	4267,0	4335,4	3,87

Observou-se que o BRKGA alcançou soluções relativamente próximas às atuais melhores, com desvio percentual máximo de 4,48%, na média, na Instância 7, e o menor desvio na Instância 5 (1,13%). Além disso, verificou-se que as Instâncias 7 e 8 apresentaram maior desvio padrão entre

https://proceedings.science/p/85521?lang=pt-br

suas diferentes sementes, o que representa uma lenta convergência dos resultados comparado às demais, possivelmente devido à distribuição das penalidades por entre os quartos de cada solução.

Os resultados da Solução Proposta 2 são demonstrados na Tabela 4 em termos de média, mínimo, máximo, desvio padrão (DP) (considerando as mesmas dez sementes), e as mesmas estatísticas para o tempo de execução. Além disso, calculou-se o desvio percentual da solução média (ΔS), e do tempo médio (Δt), da Solução Proposta 2 em relação aos resultados de Turhan e Bilgen [2017].

Tabela 4: BRKGA - Resultados da Solução Proposta 2

I	Solução					Tempo				
	Mínimo	*Média*	*Máximo*	*DP*	ΔS *(%)*	*Mínimo*	*Média*	*Máximo*	*DP*	Δt *(%)*
1	691,4	692,2	692,8	0,43	-0,11	26	16	13	3,73	-15,20
2	1283,4	1295,3	1300,0	5,47	-0,36	17	15	13	1,40	-36,78
3	886,8	896,6	900,8	4,62	-0,54	14	12	10	0,83	-23,65
4	1415,4	1426,5	1433,2	5,72	-0,58	13	11	9	1,03	-45,86
5	654,4	656,7	657,6	1,10	-0,13	9	8	7	0,87	-10,71
6	890,4	895,3	898,8	2,86	-0,41	12	10	9	0,73	-19,34
7	1415,6	1422,6	1426,4	3,03	-0,28	14	13	11	0,85	-43,51
8	4655,2	4666,5	4673,0	5,09	-0,15	37	31	27	2,82	-3,02

Observa-se que em todas as instâncias o BRKGA atingiu a solução alvo com uma média de tempo de execução menor do que em Turhan e Bilgen [2017], mostrando que o método é competitivo. Devido ao critério de parada ter sido obter uma solução menor ou igual a da referência, o BRKGA foi capaz de atingir soluções ligeiramente menores, com um ΔS de até -0,58% na Instância 4. Além disso, nas Instâncias 2 - 7, até mesmo os tempos máximos foram menores em relação aos tempos dos autores.

Analisou-se a qualidade das Soluções Propostas 1 e 2, comparando-as com os demais resultados da literatura. Utilizou-se o desvio das soluções em relação às melhores existentes (ΔBKS) como base de referência, as quais são compostas pelos resultados de Ceschia e Schaerf [2011] e Range et al. [2014]. Com isso, na Tabela 5, compararam-se as soluções propostas com estes resultados, considerando também os resultados de Turhan e Bilgen [2017] devido aos baixos tempos computacionais. As soluções de Demeester et al. [2010], Bilgin et al. [2012] e Hammouri e Alrifai [2014] foram desconsideradas da análise em todas as instâncias avaliadas pois não apresentaram resultados competitivos com as demais.

Tabela 5: Detalhamento das Soluções

I	**Ceschia e Schaerf [2011]**		**Range et al. [2014]**		**Turhan e Bilgen [2017]**		**Sol. Proposta 1**		**Sol. Proposta 2**	
	ΔBKS***(%)***	***t (s)***	ΔBKS***(%)***	***t (s)***	ΔBKS***(%)***	***t (s)***	ΔBKS***(%)***	***t (s)***	ΔBKS***(%)***	***t (s)***
1	0,18	1575	0,00	369	5,90	19	1,85	300	5,78	16
2	0,60	5355	0,00	516	15,00	23	2,30	300	14,58	15
3	0,70	2660	0,00	439	17,34	16	2,03	300	16,71	12
4	0,00	8043	0,58	3311	22,40	20	1,31	300	21,69	11
5	0,26	3917	0,00	156	5,38	9	1,13	300	5,24	8
6	0,68	1633	0,00	277	13,42	13	2,00	300	12,96	10
7	0,00	154	1,84	332	19,58	23	4,48	300	19,24	13
8	0,00	5222	6,63	1663	12,62	32	3,87	300	12,45	31
Média	**0,30**	**3570**	**1,13**	**883**	**13,96**	**19**	**2,37**	**300**	**13,58**	**15**

Apesar de Ceschia e Schaerf [2011] e Range et al. [2014] compreenderem o conjunto das melhores soluções, o tempo computacional foi bem maior em relação aos demais estudos em algumas instâncias, com tempos médios de 3570s e 883s, respectivamente. A Solução Proposta 1,

https://proceedings.science/p/85521?lang=pt-br

mesmo não tendo atingido às melhores soluções, alcançou um desvio percentual (ΔBKS) baixo em relação a elas (2,37%) em um tempo computacional consideravelmente menor. Portanto, o BRKGA se mostra um método eficiente para aplicação no PASP, com melhor balanceamento entre os objetivos de tempo e valor de solução. Destaca-se que apesar de Range et al. [2014] apresentar as melhores soluções para cinco das oito instâncias avaliadas ($\Delta BKS = 0$), demonstrou um desvio de solução máximo de 6,63%, enquanto que a Solução Proposta 1 obteve 4,88%.

A Solução Proposta 1 demonstrou valores de solução menores que os de Turhan e Bilgen [2017]. Verifica-se que as soluções desses autores não são competitivas com as demais, com desvio percentual médio de 13,96%, e um máximo de 22,40% na Instância 4. Em relação ao tempo computacional, demonstrou-se que a Solução Proposta 2, apesar de obter desvio percentual similar ao de Turhan e Bilgen [2017], o tempo médio foi ligeiramente menor (15s), conforme mencionado anteriormente.

5. Conclusão

O objetivo desse artigo foi desenvolver uma solução para Programação de Admissão de Pacientes a nível operacional considerando pacientes de internação eletivos com o uso do Algoritmo Genético de Chaves Aleatórias Viciadas. As principais contribuições do artigo foram a aplicação inédita do BRKGA para a solução do PASP, e a obtenção de soluções para o subconjunto das instâncias 1 à 8, que se apresentaram competitivas em relação à literatura.

No tempo limite de cinco minutos, o BRKGA alcançou soluções no intervalo de 1,13-4,48% de desvio em relação às melhores soluções. Porém, o método também é capaz de atingir outros objetivos a depender da parametrização adotada, conseguindo alcançar tempo computacional comparável à abordagem da literatura que detém os menores tempos.

A aplicação do BRKGA apresentou resultados satisfatórios em termos de tempo computacional e qualidade de solução, e apresenta potencial de melhoria. Dessa forma, para trabalhos futuros, sugere-se a aplicação de heurísticas de busca local para melhoria da solução ou combiná-la com modelos de otimização podendo auxiliar na obtenção da solução ótima.

Referências

Azadeh, A., Farahani, M. H., Torabzadeh, S., e Baghersad, M. (2014). Scheduling prioritized patients in emergency department laboratories. *Computer Methods and Programs in Biomedicine*, 117(2):61–70.

Barz, C. e Rajaram, K. (2015). Elective patient admission and scheduling under multiple resource constraints. *Production and Operations Management*, 24(12):1907–1930.

Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W., e Berghe, G. V. (2012). One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics*, 18(3): 401–434.

Ceschia, S. e Schaerf, A. (2011). Local search and lower bounds for the patient admission scheduling problem. *Computers & Operations Research*, 38(10):1452–1463.

Ceschia, S. e Schaerf, A. (2012). Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Artificial Intelligence in Medicine*, 56(3):199–205.

Ceschia, S. e Schaerf, A. (2016). Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. *Journal of Scheduling*, 19(4):377–389.

Da Silva, G. C., Bahiense, L., Ochi, L. S., e Boaventura-Netto, P. O. (2012). The dynamic space allocation problem: Applying hybrid grasp and tabu search metaheuristics. *Computers & Operations Research*, 39(3):671–677.

https://proceedings.science/p/85521?lang=pt-br

Damm, R. B., Resende, M. G., e Ronconi, D. P. (2016). A biased random key genetic algorithm for the field technician scheduling problem. *Computers & Operations Research*, 75:49–63.

Demeester, P., De Causmaecker, P., e Berghe, G. V. (2010). A tabu search algorithm for assigning patients to beds efficiently. *Artificial Intelligence Medicine*, 48(1):61–70.

Fei, H., Meskens, N., e Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, 58(2):221–230.

Gonçalves, J. F. e Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.

Granja, C., Almada-Lobo, B., Janela, F., Seabra, J., e Mendes, A. (2014). An optimization based on simulation approach to the patient admission scheduling problem: Diagnostic imaging department case study. *Journal of Digital Imaging*, 27(1):33–40.

Grasas, A., Ramalhinho, H., Pessoa, L. S., Resende, M. G., Caballé, I., e Barba, N. (2014). On the improvement of blood sample collection at clinical laboratories. *BMC Health Services Research*, 14(1):12.

Hammouri, A. I. e Alrifai, B. (2014). Investigating biogeography-based optimisation for patient admission scheduling problems. *Journal of Theoretical & Applied Information Technology*, 70 (3):413–421.

Kifah, S. e Abdullah, S. (2015). An adaptive non-linear great deluge algorithm for the patient-admission problem. *Information Sciences*, 295:573–585.

López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., e Birattari, M. (2011). The irace package, iterated race for automatic algorithm configuration. Technical report, TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium.

Marchesi, J. F. e Pacheco, M. A. C. (2016). A genetic algorithm approach for the master surgical schedule problem. In *Evolving and Adaptive Intelligent Systems (EAIS), 2016 IEEE Conference on*, p. 17–21. IEEE.

Marques, I. e Captivo, M. E. (2015). Bicriteria elective surgery scheduling using an evolutionary algorithm. *Operations Research for Health Care*, 7:14–26.

Marques, I., Captivo, M. E., e Pato, M. V. (2014). Scheduling elective surgeries in a portuguese hospital using a genetic heuristic. *Operations Research for Health Care*, 3(2):59–72.

Moz, M. e Pato, M. V. (2007). A genetic algorithm approach to a nurse rerostering problem. *Computers & Operations Research*, 34(3):667–691.

Qu, X., Peng, Y., Kong, N., e Shi, J. (2013). A two-phase approach to scheduling multi-category outpatient appointments–a case study of a women's clinic. *Health Care Management Science*, 16(3):197–216.

Range, T. M., Lusby, R. M., e Larsen, J. (2014). A column generation approach for solving the patient admission scheduling problem. *European Journal of Operational Research*, 235(1):252–264.

Spears, V. M. e De Jong, K. A. (1991). On the virtues of parameterized uniform crossover. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, p. 230–236.

https://proceedings.science/p/85521?lang=pt-br

Toso, R. F. e Resende, M. G. (2015). A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 30(1):81–93.

Tsai, C.-C. e Li, S. H. (2009). A two-stage modeling with genetic algorithms for the nurse scheduling problem. *Expert Systems with Applications*, 36(5):9506–9512.

Turhan, A. M. e Bilgen, B. (2017). Mixed integer programming based heuristics for the patient admission scheduling problem. *Computers & Operations Research*, 80:38–49.

Vancroonenburg, W., Goossens, D., e Spieksma, F. (2011). On the complexity of the patient assignment problem. Technical report, KAHO Sint-Lieven, Gebroeders De Smetstraat 1, Gent, Belgium.

Wang, C.-W., Sun, L.-M., Jin, M.-H., Fu, C.-J., Liu, L., Chan, C.-H., e Kao, C.-Y. (2007). A genetic algorithm for resident physician scheduling problem. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, p. 2203–2210. ACM.

Zhang, L. M., Chang, H. Y., e Xu, R. T. (2013). The patient admission scheduling of an ophthalmic hospital using genetic algorithm. *Advanced Materials Research*, 756–759:1423–1432.

https://proceedings.science/p/85521?lang=pt-br