LII Simpóso Brasileiro de Pesquisa Operacional
João Pessoa-PB, 3 a 5 de novembro de 2020

APLICAÇÃO DO GRASP AO PROBLEMA DE AGENDAMENTO DE CIRURGIAS ELETIVAS

Elisa de Fátima Andrade Soares
Universidade Federal de Pernambuco - UFPE
Av. Prof. Moraes Rego, 1235, Cidade Universitária, Recife, PE
efas@cin.ufpe.br

Roberval G. M. Filho, Sebastião M. Marques de Menezes, Getúlio F. da Silva Gomes, Francisco C. de Lima Júnior
Universidade do Estado do Rio Grande do Norte - UERN
Av. Prof. Antônio Campos, Pres. Costa e Silva, Mossoró, RN
robervalad@gmail.com, mateusmarques200@gmail.com, getulio.juce@gmail.com, fclimajr @gmail.com

Euller Sadraque da Costa e Lima
Universidade Federal Rural do Semi-Árido - UFERSA
Rua Francisco Mota Bairro, 572, Pres. Costa e Silva, Mossoró, RN
euller.scl@gmail.com

RESUMO

As cirurgias eletivas são procedimentos cirúrgicos que não têm urgência para serem realizadas. O agendamento deste tipo de cirurgia consiste na alocação de recursos hospitalares como salas e equipe médica e tem como objetivo atender o maior número de pessoas em menor intervalo de tempo. O problema de agendamento de cirurgias eletivas é um caso particular do problema de *Job Shop Scheduling*, então denominado de Problema de Otimização de Agendamento de Cirurgias Eletivas - POACE. Como solução computacional para o problema, realizou-se a implementação da metaheurística *Greedy Randomized Adaptive Search Procedures* (GRASP). Foram realizados experimentos com instâncias geradas a partir de dados reais obtidos em um hospital da cidade de Mossoró/RN, e com instâncias fictícias, as quais foram geradas seguindo o padrão estatístico das instâncias reais. Os resultados experimentais obtidos foram promissores e indicam que a metaheurística GRASP para o problema proposto é competitiva mesmo quando aplicado às instâncias maiores.

PALAVRAS CHAVE. GRASP, Job Shop Scheduling, Agendamento de Cirurgias Eletivas.

Tópicos (Otimização Combinatória, Metaheurísticas)

ABSTRACT

The elective surgeries are medical procedures that do not have the urgency to be done. The appointment for this surgery consists of an allocation of hospital resources, as rooms and medical staff with the goal of serving the largest number of people in the shortest time. The scheduling problem is a particular case of Job Shop Scheduling, called of Elective Surgery Scheduling Optimization Problem. As a computational solution to this problem, a Greedy Randomized Adaptive Search Procedures (GRASP) metaheuristics was implemented. Using experiments with instances created from real data got in a hospital from Mossoró City, localized in Rio Grande do Norte State, Brazil. The fictitious instances were generated following the pattern statistical from

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

the real instances. The experimental results obtained in this research were promissory and indicate that the GRASP metaheuristic is so competitive, even applied in larger instances.

KEYWORDS.GRASP, Job Shop Scheduling, Schedule of Elective Surgeries.

Paper topics (Combinatorial Optimization, Matheuristics)

1. Introdução

No Brasil, os atendimentos hospitalares na rede pública de saúde apresentam um elevado tempo de espera para os procedimentos que não são caracterizados com urgência médica. Normalmente, em tal contexto as pessoas, ao chegarem nos hospitais para serem atendidas, na maioria das vezes, encontram filas extensas, o que consome certo tempo de suas rotinas e, muitas vezes, agrava, emocionalmente, o problema de saúde existente. Desse modo, ocorrem diversas situações de atrasos no atendimento do Sistema Único de Saúde (SUS). Essas situações sucedem devido à falta de gerenciamento de recursos, de profissionais e de infraestrutura, e podem acentuar a gravidade da patologia do paciente [Oliveira et al. 2016].

O gerenciamento estratégico dos recursos hospitalares nos centros cirúrgicos é necessário para que se possa obter melhorias na organização, prestação de serviços de saúde, economia de recursos e redução de tempo de espera por uma cirurgia eletiva. Por se tratar de setores importantes para a realização de atendimentos dos casos de saúde de média e alta gravidade, no que diz respeito à população [Magerlein et al.], os centros de cirurgias consistem em unidades hospitalares que requerem atenção e dedicação por parte, gestores. Dentre os principais tipos de procedimentos que podem ser realizados nesses núcleos, existem as denominadas cirurgias eletivas. Estes procedimentos, de acordo com [Moreira et al. 2016], consistem nas intervenções que são marcadas com antecedência e possuem uma data para serem efetivadas, sendo exceções às categorias cirúrgicas consideradas de emergência ou urgência. A realização desse agendamento envolve a parte administrativa e assistencial, que são inseridas no processo de planejamento das equipes médicas.

Segundo [Carter e Tovey 1992] o problema de agendamento de cirurgias eletivas é considerado na literatura um problema clássico de otimização combinatória, pertencente à classe NP-Difícil, e, portanto, tem sua resolução computacional inviabilizada via métodos exatos (ou exaustivos), visto que o custo envolvido no processo tornaria inviável a tomada de decisão em tempo hábil. A implicação prática da classificação de um dado problema como NP-Difícil, é a constatação teórica da não existência de algoritmos exatos que garantam resolver instâncias de grande porte, de forma ótima, em um tempo computacional aceitável [Cormen et al. 2002].

As cirurgias de emergência necessitam de salas e equipes médicas de plantão para atender os pacientes, sendo assim, é necessário sempre existir centro cirúrgico disponível para cirurgias emergenciais [Proença 2010]. Como no caso das cirurgias eletivas o atendimento é suprido por agendamento prévio, é estratégico que esse agendamento considere deixar, equipes e salas ociosas no seu planejamento. Dessa forma, este trabalho tem como objetivo reduzir o instante de término da última cirurgia que corresponde a minimizar *makespan* máximo para o problema. Para isso, propõe-se a aplicação da Metaheurística GRASP, a fim de solucionar o problema de otimização do agendamento de cirurgias eletivas.

Este artigo está organizado da seguinte maneira: a Seção 2 expõe a problematização do agendamento de cirurgias eletivas; a Seção 3 descreve a metaheurística GRASP para o problema; a Seção 4 aborda os testes computacionais e a construção das instâncias testes; a Seção 5 mostra os resultados obtidos neste estudo; e, por fim, na Seção 6, são apresentadas as conclusões.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

2. Problema de Agendamento de Cirurgias Eletivas

As cirurgias eletivas são os atendimentos prestados aos usuários que permitem o gerenciamento sem carácter de urgência e emergência. Na prática, o planejamento destas cirurgias ocorre através de etapas. Primeiramente, é checada a disponibilidade da equipe de cirurgia para os casos eletivos [Andrade et al. 2016]. Em seguida, define-se o número de salas cirúrgicas disponíveis em determinada data e os grupos que têm prioridade sobre uso das salas. Por fim, é estabelecido o agendamento através de um conhecimento diário, conhecimento empírico e experiência do gestor hospitalar. Nesse contexto, este trabalho apresenta uma proposta de otimização para solucionar esse processo de agendamento.

No problema de agendamento de cirurgias eletivas, cada sala de cirurgia (incluindo equipamentos e recursos humanos) é equiparada a uma máquina que possui tarefas a serem realizadas em um horizonte de tempo finito. Como já mencionado, de acordo com [Pham et al. 2008] o agendamento de cirurgias eletivas é um caso particular do problema de otimização Job Shop Scheduling. Dessa maneira, pode ser considerado como um problema da classe NP-Difícil, em que os resultados exatos são inviáveis quando atribuído a instâncias de grande porte [Gomes 2016].

Na literatura existem algumas abordagens para o problema de agendamento de cirurgias eletivas. O trabalho proposto por [Pham et al. 2008] propõe uma abordagem de agendamento procedimento cirúrgico utilizando uma nova extensão do problema de agendamento de Job Shop chamada Multi-Mode Blocking Job Shop (MMBJS). Nesta conjuntura, a tarefa desempenha um papel decisivo na utilização eficiente dos recursos do hospital, garantindo a qualidade do atendimento aos pacientes. Os experimentos computacionais são realizados utilizando o CPLEX com instâncias de grande porte.

Os autores [Min et al. 2010] abordaram um modelo de programação dinâmica estocástica para resolver o problema. Ademais, consideraram pacientes com prioridades diferentes para agendar intervenções cirúrgicas e com uma capacidade limitada. Desta forma, os pacientes com prioridades altas são adicionados a uma lista de espera e inseridos na agenda. Assim, a tomada de decisão é conforme o início de cada período, em que o número de pacientes a serem agendados é de acordo com os *trade-offs* entre o custo das horas extras e o custo do adiamento da cirurgia. Os resultados mostraram que a consideração da prioridade do paciente resulta em diferenças significativas nas programações de cirurgia em relação à programação que ignora a prioridade do paciente.

Já a abordagem de [Marques et al. 2014] explanaram o problema de agendamento de cirurgias eletivas com dois critérios de otimização conflitantes independendentes: maximizar a alocação das salas cirúrgicas e maximizar o número de cirurgias programadas. Assim, existe duas versões de uma heurística com uma única função objetivo aplicado a dados reais do hospital estudado. Os experimentos realizados consideram casos reais com 508–2306 cirurgias eletivas que são resolvidos com sucesso em menos de 240 segundos. Além disso, apresenta uma melhoria para qualidade dos planos cirúrgicos hospitalares e exige menos recursos para a construção dos procedimentos cirúrgicos.

Outro trabalho existe para o problema abordado neste artigo é realizado por [Eun et al. 2019], que solucionaram tal problema com uma modelagem de programação inteira mista estocástica para otimizar a atribuição de cirurgias. Os autores consideraram o pior estado de saúde do paciente entre todos os pacientes que aguardam cirurgias e horas extras totais que excedem os tempos disponíveis alocados para as cirurgias. Além disso, apresentaram três abordagens de solução: o método de aproximação da média da amostra, uma pesquisa local de ascensão mais rápida e uma pesquisa tabu. Os resultados mostraram que a pesquisa tabu fornece soluções eficazes dentro de tempos de computação razoáveis.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Neste trabalho, a finalidade é minimizar o tempo do término das tarefas efetivadas por uma única máquina em um intervalo de tempo. Assim como no clássico problema de otimização Job Shop Scheduling.

Considere um conjunto de salas cirúrgicas S, sendo |S| a cardinalidade do conjunto S e o C conjunto de cirurgias a serem realizadas (com |C| sendo cardinalidade do conjunto C). Dessa forma, ao iniciar uma cirurgia nenhuma interrupção é permitida, isto é, para que se comece outra cirurgia, a anterior deve ter sido concluída.

Os procedimentos cirúrgicos só podem ser realizados em uma sala após uma preparação adequada desta. Esse tempo para os preparativos não dependem somente do tipo de cirurgia que será realizada, mas também da intervenção efetuada anteriormente, ou seja, o tempo de preparação das salas depende da sequência em que as cirurgias são realizadas.

Considere a seguinte notação para o problema:

Parâmetros:

$Q = \{c_1,\ c_2,\ ..., c_i, ...,\ c_q\}$: Conjunto de cirurgias a serem realizadas;
i : Indexação para as cirurgias a serem realizadas, $i = 1, 2, \ldots, |Q|$.
$N = \{s_1,\ s_2,\ ..., s_k, ...,\ s_n\}$: Conjunto de salas de cirurgias disponíveis.
k : Indexação para as salas de cirurgias, $k = 1, 2, \ldots, |N|$.
rt_i : Tempo de realização da cirurgia i em qualquer uma das salas.
st_i : Tempo de início da cirurgia i em qualquer uma das salas.
p_{ij} : Tempo de preparação da sala entre a cirurgia i e a cirurgia j.
T : Tempo necessário para a conclusão de todas as cirurgias.
M : Um número muito grande.

Variáveis de decisão:

X_{ik} : Variável booleana que indica se a cirurgia i é realizada na sala k.
Y_{ijk} : Variável booleana que indica se a cirurgia i precede a cirurgia j na sala k.

$Minimizar:\ T_{max}$ (*makespan máximo*)

Sujeito à:

$$T_{max} \geq st_i + rt_i \qquad \forall i \qquad (1)$$

$$\sum_{k}^{n} X_{ik} = 1 \qquad \forall i \qquad (2)$$

$$(1 - X_{ik}).M + (1 - X_{jk}).M + (1 - Y_{ijk}).M + st_j \geq st_i + rt_i + p_{ij} \qquad (3)$$

$$(1 - X_{ik}).M + (1 - X_{jk}).M + Y_{ijk}.M + st_i \geq st_j + rt_j + p_{ji} \qquad (4)$$

$$\sum_{i}^{q} X_{ik} \geq \sum_{i}^{q} X_{ik-1} \qquad \forall k > 1 \qquad (5)$$

$$X_{ik},\ Y_{ijk} \in \{0, 1\} \qquad (6)$$

No que diz respeito à função objetivo, o que se deseja é minimizar o instante de término da última tarefa que corresponde ao makespan máximo para o problema. As restrições em (1) garantem que assumirá o valor da data de finalização da última cirurgia agendada. Já as restrições em (2) garantem que cada procedimento seja agendado em uma única sala.

As restrições em (3) e (4) estabelecem o sequenciamento dos tratamentos em cada sala, de forma que, no caso das restrições em (3), se uma cirurgia precede uma outra em uma mesma sala, o tempo de início da cirurgia deve ser posterior ao término da cirurgia, mais o tempo necessário para preparar a sala para receber a cirurgia. Assim, o lado esquerdo das inequações permite que a restrição seja válida somente quando as cirurgias são executadas na mesma sala,

sendo que precede no sequenciamento. A mesma lógica é válida para o caso em que a cirurgia precede a cirurgia, tratada nas restrições em (4).

As restrições em (5) estabelecem uma espécie de corte, permitindo que, por critério de qualidade, uma sala receba maior quantidade de cirurgias do que uma outra sala. As especificações em (6) definem o conjunto universo das variáveis de decisão como sendo binário.

Dessa forma, a situação real tem que cada sala possui uma sequência de cirurgias, determinada pelos tipos de intervenções que serão realizadas. Com isso pode-se considerar que cada cirurgia é uma tarefa e cada sala de procedimento cirúrgico é uma máquina. Isso possibilita a utilização do Job-Shop Scheduling para localizar e aplicar melhorias no agendamento das cirurgias.

De acordo com o modelo matemático apresentado anteriormente, cada procedimento cirúrgico será agendado uma única vez em uma sala, quando houver uma cirurgia posterior à que está sendo realizada, será necessário aguardar a finalização daquela para que se inicie a seguinte, tornando, assim, impossível o início de um procedimento durante o período de execução de outro.

3. Metaheurística GRASP

A metaheurística GRASP desenvolvida por [Feo e Resende 1995] funciona de forma iterativa. Assim, cada iteração é composta por duas fases, a de construção, que tem por objetivo gerar uma solução inicial; e a fase de busca local, em que a solução inicial é aprimorada. Cada uma dessas fases serão discriminadas na seção a seguir.

As iterações da metaheurística GRASP funcionam de forma independente, ou seja, elas não levam em consideração aquilo que foi criado e aprimorado na iteração anterior. O número de iterações depende de um critério de parada estabelecido por quem estiver desenvolvendo, normalmente um número máximo de iterações determinadas a priori. O Algoritmo 01 apresenta o pseudocódigo da metaheurística GRASP.

Algoritmo 01 – GRASP

```
01: procedure GRASP(D, α)
02:     S ← +∞
03:     while "critério de parada" do
04:       S1 ← Guloso_Aleatorio(D, α)
05:       S2 ← Busca_Local(S1)
06:         if S2 < S then
07:             S ← S2
08:         end if
09:       "Atualiza critério de parada"
10: end procedure
```

3.1 Fase de construção

A fase de construção gera uma solução de forma gulosa e/ou randômica. O que determina o grau de aleatoriedade é o parâmetro α, a solução é construída um elemento por vez. Esses elementos são retirados de uma lista, na qual há uma parte dos elementos da instância. O conjunto desses elementos é denominado lista restrita de candidatos (LRC), já a lista com todos os elementos da instância é denominada lista de candidatos (LC), e é desta que os elementos da LRC são retirados.

O tamanho da LRC é definido como $TLRC = 1 + \alpha\,(N - 1)$, em que consiste no N que é o número de elementos da LRC. O parâmetro α é um valor entre 0 e 1, e define a aleatoriedade

Galoá { Este trabalho foi publicado utilizando o Galoá proceedings

da fase de construção. É importante ressaltar que, se ele receber o valor 1, esta fase vai ser completamente aleatória; e que, se for 0 a solução, vai ser completamente gulosa, e o algoritmo vai retornar sempre o mesmo valor em cada iteração. Nesse caso, tal situação não é aprazível, já que o objetivo é aprimorar a resposta no decorrer das iterações.

A LRC é preenchida com os primeiros elementos da LC. Com a LRC preenchida, um de seus elementos é selecionado de maneira aleatória, e é adicionado à solução inicial, esse mesmo elemento é removido da LC. A etapa é concluída quando a LC não dispuser de elementos, assim, a solução inicial está concluída. A LC é ordenada com base em algum critério pré-estabelecido, neste caso, o critério varia em função do tipo de procedimento cirúrgico que melhor se enquadrar no critério de qualidade, dado o contexto. A avaliação de qualidade depende do tempo necessário para realização de cada procedimento e da disponibilidade de salas no momento a ser agendado. No Algoritmo 02 tem-se o pseudocódigo da fase de construção de forma gulosa aleatória.

Algoritmo 02 – Algoritmo Guloso Aleatório

```
01: procedure Guloso_Aleatorio(D, α)
02:     LC ← Ordena (D)
03:     while "solução incompleta" do
04:       LRC ← Construir_LRC(LC, α)
05:       e ← Escolhe_Elemento_Aleatorio(LRC)
06:       S ← S U {e}
07:       LC ← LC – {e}
08:     end while
09:     return S
10: end procedure
```

Fonte – Autoria própria

Onde:

D – Vetor de tarefas;

α – Parâmetro que define o grau de aleatoriedade;

LC – Lista de candidatos;

LRC – Lista restrita de candidatos;

e – Elemento selecionado para compor a solução;

S – Solução, que inicialmente está vazia.

3.2 Busca Local

A fase de busca local é um procedimento que tem por objetivo encontrar um conjunto de soluções S_{2opt} que sejam formadas a partir de pequenas modificações feitas a partir de uma solução inicial: S_i. Esse conjunto de soluções é denominado vizinhança de S_i. Essa fase tem como escopo melhorar a solução que foi criada na fase de construção.

Existem vários algoritmos de busca local como melhoria iterativa, descida mais rápida ou a vizinhança *2-opt,* que consiste no algoritmo utilizado neste trabalho. A vizinhança *2-opt* foi proposto por [Croes 1958], e é um algoritmo simples de busca local. A Figura 1 apresenta um exemplo de busca local *2-opt* para o POACE. Neste exemplo da Figura 1, dois procedimentos

cirúrgicos foram trocados de lugar na perspectiva de melhorar a qualidade da solução final, ou seja, realizar o agendamento que realize o maior número de procedimentos em m enor intervalo de tempo.

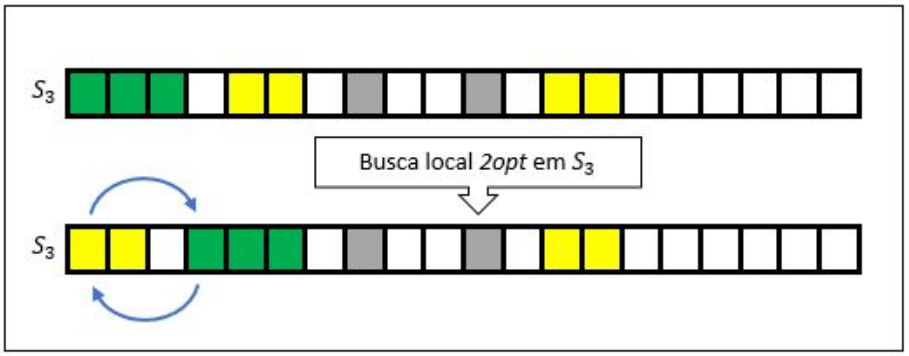

Figura 1. Exemplo de busca local *2-opt*. para o POACE. (Fonte – Autoria própria).

4. Testes Computacionais

O algoritmo foi implementado na linguagem de programação Java. Os testes foram realizados em um notebook com 6 gigabytes de memória RAM e um processador Intel core i5 com sistema operacional Windows 10 home, 64 bits.

4.1 Instâncias para o POACE

Para os experimentos, foram utilizadas sete instâncias de testes, sendo quatro geradas com dados reais e três com dados fictícios. A utilização de instâncias com dados fictícios visou a possibilidade de criar cenários que melhor exploram os limites da metaheurística para o problema. Com as instâncias fictícias é possível, por exemplo, utilizar cenários com cirurgias de maior tempo que os tempos reais, estabelecer quantidades elevadas (ou reduzidas) de determinado tipo de procedimento cirúrgico, dentre outros parâmetros que podem ser regulados para efeito experimental. Assim, foram criadas as instâncias FICT-21-S2, FICT-100-S5, FICT-200-S10, em que o número logo após a sigla "FICT" indica a quantidade de cirurgias (tarefas) e o número após o "S" indica a quantidade de salas (máquinas) disponíveis. No caso das instâncias fictícias, os tempos de processamento das tarefas foram gerados de forma aleatória, com base em critérios experimentais de definição dos cenários desejados.

Para a elaboração das instâncias reais, foram utilizados dados disponibilizados pelo Centro de Oncologia e Hematologia de Mossoró - COHM, em que cada unidade de tempo representa meia hora em tempo real. Criou-se as instâncias COHM-11-S3, COHM-50-S4, COHM-75-S5, COHM-130-S10, em que o número logo após a sigla "COHM" indica a quantidade de cirurgias (tarefas), e o número após o "S" indica a quantidade de salas (máquinas) disponíveis.

Para todas as instâncias, foram consideradas unidades de tempo em blocos de 30 minutos para composição dos tempos de execução de tarefa (tempo de realização de cada cirurgia).

4.2 Construção de Instâncias com os dados reais

Para realizar o teste com os dados cedidos, pelo Centro de Oncologia e Hematologia de Mossoró, foi utilizado os tempos de início e fim de cada cirurgia para se chegar a uma média de tempo de duração. Em seguida, analisou-se qual a cirurgia tinha maior prioridade. Após identificar o tempo médio das cirurgias, em horas (organizado em blocos de 30 minutos), e

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

construir uma lista de prioridades, elaborou-se a lista de cirurgias LC para a execução da heurística construtiva.

Com os dados obtidos no Centro de Oncologia e Hematologia de Mossoró (COHM) foram criadas instâncias de teste, dividindo as cirurgias de um determinado dia. A Figura 2 apresenta três exemplos de instâncias no formato utilizado nas heurísticas. A instância (1) denota um exemplo de agendamento realizado pelo COHM. Na sequência, é apresentado um exemplo no formato gerado pela heurística gulosa aleatória (2), em que se selecionou elementos na lista de cirurgias (LC). Posteriormente, (3) apresenta-se um exemplo de como ocorre na fase de busca local, na qual são trocadas as posições de dois elementos (cirurgias) e comparado o resultado, em que caso tal resultado seja melhor que o anterior ele assume a posição como o melhor resultado. As instâncias utilizadas neste artigo estão disponibilizadas no [GitHub 2020].

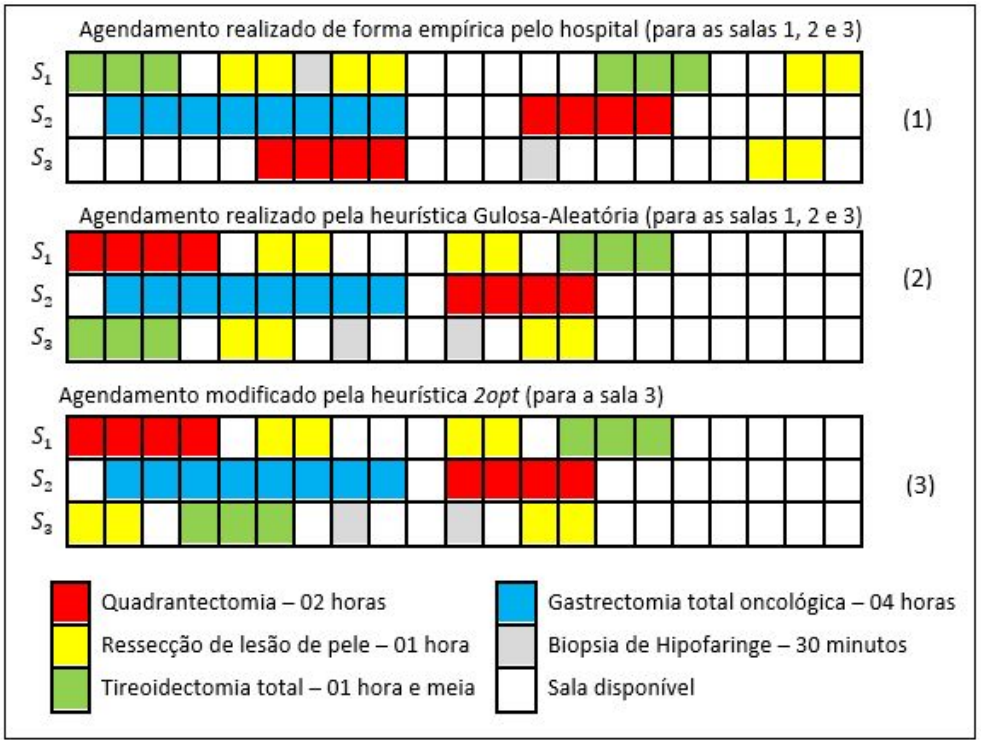

Figura 2. Exemplo do padrão de instâncias (formato de solução) para o POACE.

5. Resultados

Na implementação do GRASP para o problema de agendamento de cirurgias eletivas como um caso particular de Job Shop Scheduling, foi necessário considerar as cirurgias como tarefas e as salas como máquinas. Além disso, denota-se também que cada procedimento cirúrgico é ininterrupto, e que outras intervenções só podem ser realizadas após o término das anteriores. O algoritmo foi executado 30 vezes, para cada uma das sete instâncias e o número de iterações do GRASP foi de mil. Ademais, o critério de ordenação da LC foi a tarefa de menor tempo de processamento, e utilizou-se o parâmetro $\alpha = 0,5$. Os resultados obtidos para as sete instâncias utilizadas são apresentados na Tabela 1, em que os valores das colunas "Melhor" e "Médio" correspondem às quantidades de blocos de (unidades de tempo) de tamanho de 30 minutos.

Para efeito de comparação com a realidade vivenciada no COHM, utilizou-se o resultado do agendamento empírico realizado pelo hospital para a instância COHM-11-S3, para a qual a

metaheurística GRASP obteve resultado mais significativo (melhoria de 5 horas mesmo considerando a média) se comparado ao obtido pela equipe do COHM. Para as demais instâncias reais, por uma questão operacional, não foi possível obter os resultados do agendamento real para realizar a comparação.

Como se pode observar na Tabela 1, para todas as instâncias foram comparados os resultados da execução da fase construtiva do GRASP com a execução integral da metaheurística, apresentando, assim, o progresso entre as duas fases do algoritmo.

Para as instâncias menores (COHM-11-S3 e FICT-21-S2), há indícios de que a metaheurística tenha encontrado a solução ótima (11 e 32 unidades de tempo, respectivamente), pois o experimento com trinta execuções obteve desvio padrão nulo e N de 21 cirurgias a melhor solução foi determinada já na fase construtiva.

Tabela 1. Resultados experimentais para a metaheurística GRASP - 30 execuções.

Resultados Obtidos para 07 Instâncias - Valores de *Makespan* Máximo						
Instâncias	Empírico	Fase Construtiva - GRASP		GRASP		
	Melhor	Melhor	Médio	Melhor	Médio	Dv.Padrão
COHM-11-S3	21,00	12,00	13,86	11,00	11,00	0,00
COHM-50-S4	-	65,00	74,06	25,00	25,50	0,50
COHM-75-S5	-	32,00	34,76	29,00	30,26	0,58
COHM-130-S10	-	32,00	36,93	28,00	28,93	0,36
FICT-21-S2	-	32,00	32,00	32,00	32,00	0,00
FICT-100-S5	-	965,00	974,6	960,00	962,40	1,40
FICT-200-S10	-	211,00	215,43	208,00	210,76	1,13

No que diz respeito ao tempo de execução computacional, a metaheurística obteve resultados eficientes, mesmo para as instâncias maiores. A Figura 3 apresenta o gráfico dos resultados obtidos com a instância FICT-100-S5, considerando 30 execuções por experimento.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

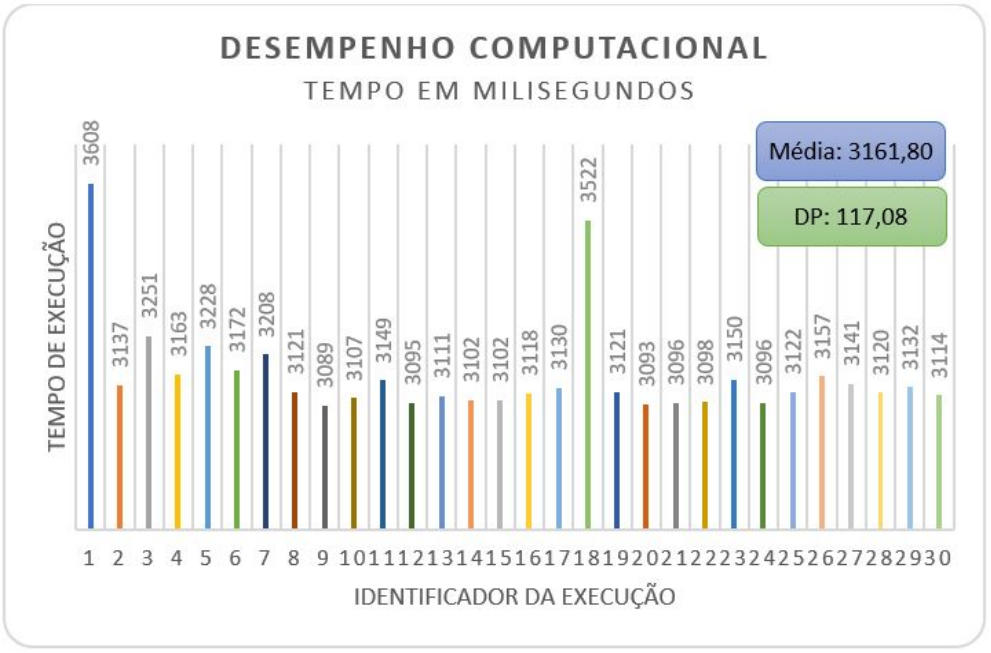

Figura 3. Desempenho computacional para instância FICT-100-S5.

Quanto ao desempenho na execução das tarefas (organização do agendamento das cirurgias), a Figura 4 apresenta o gráfico com os resultados das 30 execuções para instância FICT-100-S5, referente ao *Makespan* máximo.

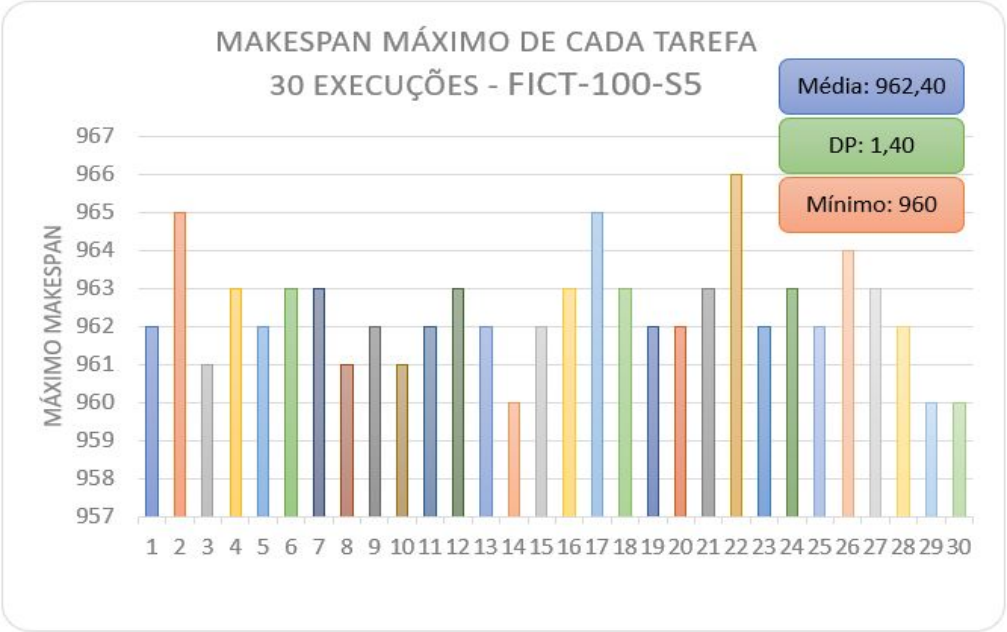

Figura 4. Resultados para instância FICT-100-S5, *Makespan* máximo.

6. Conclusões

Este artigo apresenta um estudo para o problema de agendamento de cirurgias eletivas, por conseguinte, propõe uma melhoria na agenda das salas cirúrgicas, definindo qual a sequência

de cirurgias ideal, bem como, qual o melhor momento para a realização da intervenção médico-cirúrgica. A solução do problema foi estabelecida através da proposição de uma agenda de cirurgias que pode contribuir com a tomada de decisão no contexto da gestão hospitalar, visando minimizar o horário de término de uma cirurgia e início da próxima. Este trabalho demonstrou que a metaheurística GRASP apresenta potencial quando aplicada ao problema de agendamento de cirurgias eletivas. Analisando-se os resultados, verifica-se que ocorreu uma melhora de 5 horas em relação ao gerenciamento realizado pelo COHM no contexto real do problema para uma instância de pequeno porte.

Como resultado deste trabalho, foram também geradas algumas instâncias de teste para os algoritmos implementados com dados reais obtidos no Centro de Oncologia e Hematologia de Mossoró - COHM. Todos os dados utilizados na construção das instâncias foram trabalhados de forma "blindada", ou seja, não se teve acesso ao nome ou qualquer outro dado que possibilitasse a identificação de médicos, ou pacientes envolvidos na pesquisa. Como trabalhos futuros pretendemos mostrar o comportamento da implementação da formulação proposta neste trabalho em um solver, como, por exemplo, o CPLEX para verificar os limites do método para instâncias de médio e grande porte.

Agradecimentos
Este trabalho foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq.

Referências

Andrade, G. P., Siqueira, E. C., & Souza, S. R. (2012). In XIX Encontro Nacional de Inteligência Artificial. Resolução do Problema de Agendamentos de Cirurgias Eletivas via Meta-Heurísticas GRASP, ILS e Iterated Greedy.

Carter, M. W. e Tovey, C. A. (1992). When is the classroom assignment problem hard? Operations Research, v. 40, n. 1, p. 28–30.

Croes, G. A. (1958). A method for solving traveling-salesman problems. Operations research, 6(6), 791–812.

Cormen, Thomas H. et al. (2002). Algoritmos: teoria e prática. Editora Campus, v. 2, p. 296.

Eun, Joonyup et al. (2019). Scheduling elective surgery patients considering time-dependent health urgency: Modeling and solution approaches. Omega, v. 86, p. 137–153.

Feo, T.A. e Resende, M.G.C. (1995). Greedy randomized adaptive search procedures. Journal of Global Optimization, p. 109–133.

GitHub. (2020). GitHub `https://github.com/elisandrade21/POACE-Instanc`

`ias` Acessado: 2020-05-21.

Gomes, Getúlio Ferreira da Silva. (2016). Otimização no agendamento de cirurgias eletivas modelado como job-shop scheduling, Trabalho de Conclusão de Curso, Departamento de Informática da Universidade do Estado do Rio Grande do Norte, Mossoró, RN.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Proença, Inês Marques. (2010). Planejamento de Cirurgias Eletivas - Abordagens em Programação Inteira. PhD thesis, Tese de Doutorado, Departamento de Estatística e Investigação Operacional, Lisboa.

Magerlein, J. M. e Martin, J. B. (1978). Surgical demand scheduling: A review. Health Services Research, p. 418–433.

Marques, Inês; Captivo, M. Eugénia; Pato Margarida Vaz. (2014). Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic. Operations Research for Health Care, v. 3, n. 2, p. 59–72.

Min, Daiki; Yih, Yuehwern. (2010). An elective surgery scheduling problem considering patient priority. Computers & Operations Research, v. 37, n. 6, p. 1091–1099.

Moreira, Luzimar Rangel et al. (2016). Avaliação dos motivos de cancelamento de cirurgias eletivas. Enfermagem Revista, v. 19, n. 2, p. 212–225.

Oliveira, T.G.; ALMEIDA, E.D.; ALMEIDA, J.F.F. (2016). Análise de filas em um posto de saúde da cidade de Bambuí-mg. Cadernos de Saúde Pública. São Paulo.

Pham, Dinh-Nguyen, and Andreas Klinkert. (2008). Surgical case scheduling as a generalized job shop scheduling problem. European Journal of Operational Research 185.3, p. 1011–1025.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings