

## **METAHEURÍSTICAS GRASP E VNS APLICADAS AO PROBLEMA DE AGENDAMENTO DE CIRURGIAS ELETIVAS COM INCERTEZA DE ALOCAÇÃO DE CIRURGIAS DE EMERGÊNCIA: Uma abordagem de minimização de *makespan***

**Giselle Paranhos de Andrade**

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)  
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte (MG), Brasil  
giselle.acao@gmail.com

**Adriano Cesar Machado Pereira**

Universidade Federal de Minas Gerais (UFMG)  
Rua Reitor Píres Albuquerque, ICEx - Pampulha, CEP 31270-901, Belo Horizonte (MG), Brasil  
adrianoc@gmail.com

**Sérgio Ricardo de Souza**

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)  
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte (MG), Brasil  
sergio@dppg.cefetmg.br

### **RESUMO**

Este artigo trata o Problema de Sequenciamento de Cirurgia Eletivas com Incerteza de Alocação de Cirurgias de Emergência (PACEE). O PACEE é tratado como um Problema de Programação em Máquinas Paralelas, no qual o objetivo é minimizar o *makespan*, ou seja o instante de término da última cirurgia. Considera-se o período de agendamento como semanal. Utiliza-se cinco tipos de movimentos para alocar, adiar ou antecipar as cirurgias no espaço de soluções. Os algoritmos desenvolvidos, baseados nas metaheurísticas GRASP e VNS, foram testados usando 75 instâncias geradas com dados reais de quatro hospitais de Minas Gerais, Brasil. Testes estatísticos comprovam a qualidade do desenvolvimento realizado.

**PALAVRAS CHAVE.** Problema de Sequenciamento de Cirurgia Eletivas. VNS. Metaheurísticas.

### **ABSTRACT**

This article deals with the Problem of Sequencing of Elective Surgery with Uncertainty of Allocation of Emergency Surgeries (PACEE). The PACEE is treated as a programming problem in parallel machines, in which the (*makespan*) is needed, that is, the end time of the last surgery. The weekly scheduling period is considered. There are five types of task management, like or not, like surgeries in the solution space. Use a rescheduling methodology to allocate as emergency surgeries that will be inserted into the agenda already in progress. The algorithms based on the GRASP and VNS metaheuristics were tested using 75 real data methods from four hospitals in Minas Gerais, Brazil. At the end date, a statistical test was performed to verify, with 95% confidence, a superiority of the VNS algorithm and its ability to solve this type of problem as tested instances.

**KEYWORDS.** Surgical Case Scheduling Problem. VNS. Metaheuristics.

## 1. Introdução

O aumento da demanda de pacientes cirúrgicos, em um ambiente no qual há recursos físicos restritos associado, além disso, a um constante aumento do custo das operações são preocupações imperativas na gestão da saúde, que exigem melhorias na forma como os serviços médicos são fornecidos ao público. Segundo Etzioni et al. [2003], o gerenciamento de salas cirúrgicas é prejudicado por sua sensibilidade a diversos fatores, como os conflitos de prioridades e as diferentes preferências das partes interessadas, mas também devido à escassez de recursos de alto valor. De acordo com Latorre-Núñez et al. [2016], as salas de cirurgias estão dentre os recursos mais caros em hospitais e clínicas. Seu agendamento é difícil, pois, além da própria sala cirúrgica, cada cirurgia requer uma combinação particular de recursos humanos, bem como diferentes equipamentos e materiais. Além disso, após cada cirurgia, um leito pós-anestésico é necessário para o paciente se recuperar. Outra questão importante é que, além das cirurgias planejadas, o agendamento deve ser feito de forma a acomodar as cirurgias de emergência que possam eventualmente chegar durante cada dia, que devem ser atendidas dentro de um prazo limitado.

Os fatores mencionados acima são tratados no Problema de Sequenciamento de Cirurgias (*Surgical Case Scheduling -SCS*), consistindo basicamente na alocação de recursos hospitalares para casos cirúrgicos, bem como na definição do melhor instante para a sua realização. Esta tarefa tem um papel decisivo na utilização de recursos hospitalares de forma eficiente Carter e Tovey [1992]. O Problema SCS é considerado, na literatura, um problema clássico de otimização combinatória, pertencente à classe NP-Difícil Carter e Tovey [1992]. Doravante, denotar-se-á o SCS como PACEE (Problema de Agendamento de Cirurgias Eletivas e de Emergência), que consiste em agendar cirurgias previamente conhecidas, e reprogramar a agenda quando do surgimento da cirurgia de emergência.

Para a resolução do PACEE, é realizado, neste trabalho, um sequenciamento semanal de cirurgias eletivas por meio de um cronograma, considerando o surgimento de até 12% de cirurgias de emergência em relação ao total de cirurgias. No cronograma, ainda é possível analisar a disponibilidade de cada recurso necessário à realização das cirurgias. A metodologia adotada tem, como objetivo, minimizar o *makespan*, ou seja o instante de término da última cirurgia realizada em horário comercial.

Este artigo está organizado como segue. A próxima seção apresenta uma revisão bibliográfica para o problema tratado. A Seção 3 apresenta a descrição do PACEE. Já a Seção 4 descreve a metodologia de solução adotada, de modo que a Seção 5 se concentra em descrever as metaheurísticas implementadas. A Seção 6 inclui os resultados computacionais obtidos, que são submetidos à análise estatística mostrada na Seção 7. A Seção 8 conclui o artigo.

## 2. Revisão Bibliográfica

Segundo Proença [2010], o processo de planejamento de cirurgias eletivas pode ser dividido em três fases: (i) Planejamento de Casos Mistos (*Case Mix Planning*); (ii) Planejamento Mestre de Cirurgias (*Master Surgery Planning*); e (iii) Agendamento de Casos Eletivos (*Elective Case Scheduling*). A fase de Planejamento de Casos Mistos (*Case Mix Planning*) analisa a disponibilidade, em horas das salas cirúrgicas, distribuída pelos diferentes cirurgiões ou equipes cirúrgicas. Está situada em um nível estratégico de decisão e, geralmente, é realizada anualmente. A distribuição do tempo considera a capacidade operativa de cada cirurgião ou de cada grupo cirúrgico e a quantidade esperada de pacientes ao longo do correspondente horizonte temporal. Hughes e Soliman [1985] e Blake e Carter [2002] apresentam diferentes abordagens para esta fase do planejamento. A fase de Planejamento Mestre de Cirurgias (*Master Surgery Planning*) envolve o desenvolvimento de uma agenda cirúrgica. Trata-se de um documento cíclico, que define o número

Tabela 1: Classificação dos Trabalhos sobre o Planejamento de Cirurgias Eletivas e Emergências

Sistema Scheduling	1	Sistema de Sequenciamento em Bloco
	2	Sistema de Sequenciamento Aberto
Modelo Matemático	1	Programação Inteira Binária
	2	Programação Inteira
	3	Programação Linear Inteira Mista
	4	Otimização Multi-critério
	5	Otimização Não-Linear
	6	Otimização de Programação Dinâmica Estocástica
	7	Modelo de Simulação
	8	Otimização Estocástica
Solução adotada	1	Método Exato
	2	Método Heurístico
Critério	1	Número de leitos usados
	2	Custo
	3	Critério Financeiro
	4	Utilização da UTI
	5	Duração de Cirurgias
	6	Limites de Estadia
	7	<i>Makespan</i>
	8	Eficiência da sala
	9	Utilização da sala
	10	**Produtividade OT
	11	**Overtime
	12	Idade do Paciente
	13	Leitos de Recuperação Pós-Anestésica
	14	Número de Enfermeiros em Leitos de RPA
	15	Paciente Prioritário
	16	Bem-estar do Paciente
	17	Tempo de Espera do Paciente
	18	Receita
	19	Risco de Não Realização
	20	Risco de Falência Pessoal
	21	Cancelamento da Cirurgia
	22	Satisfação Pessoal
	23	Duração Total
	24	Tempo Ocioso
	25	Undertime
Recursos	1	Camas
	2	Equipamentos/materiais cirúrgicos
	3	<i>Staff</i>
Horizonte de Planejamento	1D	1 Dia
	1-2W	1-2 Semanas
	Mais	2 Semanas

e o tipo de salas de operações disponíveis, as horas em que as salas estão abertas, e definindo, ainda, cirurgiões ou grupos de cirurgias que têm prioridade sobre o tempo das salas cirúrgicas. Esta fase enquadra-se em um nível tático da gestão hospitalar. O horizonte temporal nesta fase do planejamento é mais reduzido do que na primeira fase. Blake e Carter [2002] e Beliën e Demeulemeester [2007] propõem uma série de modelos para a construção de agendamentos de cirurgias para esta fase. Na fase de Agendamento de Casos Eletivos (*Elective Case Scheduling*), é estabelecido o agendamento de cada cirurgia em uma base diária. Esta fase situa-se em um nível operacional. Trabalhos relativos a esta fase são encontrados em Magerlein e Martin [1978] e Kharraja et al. [2006], dentre outros.

Pham e Klinkert [2008] apresentam um modelo em otimização linear inteira mista, baseado em uma extensão do *Job Shop Scheduling Problem*, denominada *Multi-Mode Blocking Job Shop*. O modelo define um período de início para cada uma das três fases necessárias na realização de uma cirurgia (pré-operatório, operatório e pós-operatório) e aloca, para cada uma destas três fases, um conjunto de recursos necessários. Com técnicas de bloqueio, os autores apresentam uma solução para o problema de restrições de disponibilidade dos equipamentos. É possível, através do algoritmo apresentado, bloquear os recursos indisponíveis no momento da cirurgia. O objetivo é minimizar o período de início da última cirurgia a ser realizada. A Tabela 1 apresenta a classificação dos trabalhos sobre Planejamento de Cirurgias Eletivas e Emergências; já a Tabela 2 apresenta o resumo das referências bibliográficas considerando a classificação da Tabela 1.

Tabela 2: Resumo das referências bibliográficas sobre o Planejamento de Cirurgias Eletivas e Emergências

Referência	Sistema Scheduling	Modelo Matemático	Solução Aplicada	Critério	Recursos	Horizonte de Tempo
<b>Cirurgias Eletivas</b>						
[Balzer et al., 2017]	1	5	1	5-6-8-9-15-15-23-24	1-3	Mais
[Cardoen et al., 2009]	1	3-4	1	1-11-13-17	1-2-3	1D
[Denton et al., 2007]	1-2	3-7-8	2	11-17-25		1D
[Fei et al., 2010]	2	1-4	2	9-11-24	1-3	1-2W
[Fugener et al., 2016]	1	7	1	1-2-4-5-9-13-14-19-23-24	1-2-3	1-2W
[Guinet e Chaabane, 2003]	2	1	2	11-17	1-2-3	1-2W
[Jebali et al., 2006]	2	3	1	11-17-25	1-2-3	1D
[Ozkarahan, 2000]	1	2-4	1	11-15-17-22-25	1-2-3	1D
[Persson e Persson, 2009]	2	3-7	1	11-15-17	1-2-3	1-2W
[Pham e Klinkert, 2008]	1-2	3	1	7-11	1-2-3	1D-1-2W
[Roland et al., 2010]	1	1	2	2-11	3-2	1-2W
[Testi et al., 2007]	1	1-7	1	6-11-15-16	1-3	1-2W
<b>Cirurgias Eletivas e Emergência</b>						
[Bai, 2017]	1-2	6	1	2-4-5-9-13-14-17-19-21-24	1-2-3	1-2W
[Persson et al., 2017]	1	7	2	5-9-15-17-22-23-24	2-3	Mais
[Siar et al., 2017]	2	3	2	1-4-9-13-14-17-24	1-2	1W

### 3. Descrição do Problema

O problema abordado no presente trabalho é classificado como Problema de Programação em Máquinas Paralelas Idênticas, no qual um dos objetivos é de minimização de *makespan*. O Problema de Programação em Máquinas Paralelas Idênticas caracteriza-se por um conjunto de  $n$  tarefas, neste caso,  $N = \{1, \dots, n\}$ , a serem processadas por um conjunto de  $m$  máquinas idênticas  $M_1, \dots, M_m$ , com as seguintes características: (a) cada tarefa deve ser processada exatamente uma vez e por apenas uma máquina; (b) cada tarefa  $i$  possui um tempo de processamento  $p_i$ ; (c) existem tempos  $s_{nk}$  de preparação entre as tarefas, em que as tarefas  $n$  e  $k$  serão processadas, nesta ordem. Estes tempos de preparação são independentes da sequência, pois esta já é conhecida. Todas as tarefas estão liberadas para processamento no início do horizonte de programação, não sendo admitidas interrupções no processamento de nenhuma delas. O objetivo é encontrar um sequenciamento das  $n$  tarefas, nas  $m$  máquinas, de forma a minimizar o tempo de conclusão do sequenciamento, o chamado *makespan* ou  $C_{max}$ .

Para o estudo deste problema, foi desenvolvida uma abordagem composta por duas fases. A Fase 1, responsável pelo agendamento das cirurgias eletivas, faz uso do conceito de Módulos, proposto por Pham e Klinkert [2008], o qual propõe uma extensão do Problema de Agendamento chamada *Multi-Mode Blocking Job Shop*, porém, sob a visão de Problemas de Máquinas Paralelas. O módulo corresponde às informações contidas nas instâncias do problema, sendo, portanto, o formato da cirurgia. Cada módulo é constituído pelos recursos utilizados na realização de uma cirurgia. Assim, cada cirurgia está expressamente relacionada a um único módulo. Nesse modelo, os autores utilizam técnicas de Otimização Inteira Mista. Para Fase 2, responsável pelos agendamentos das cirurgias de emergência, utilizou-se o modelo de [Siar et al., 2017], que utiliza uma estrutura de reescalonamento de programação, baseada na chamada abordagem do horizonte de rolamento que também faz uso de Otimização Linear Inteira Mista.

### 4. Metodologia

Esta seção apresenta os procedimentos propostos para a solução do Problema de Agendamento de Cirurgias Eletivas e de Emergência (PACEE). Inicialmente, apresenta-se a estrutura de dados utilizada para a representação de uma solução. Em seguida, são apresentados os cinco tipos de movimentos utilizados na vizinhança da solução e a busca local utilizada como heurística de refinamento. Na próxima seção são apresentados as metaheurísticas implementadas, a saber, *Greedy Randomized Adaptive Search Procedure* (GRASP) e *Variable Neighborhood Search* (VNS). Os Algoritmos 1 e 3 mostram seus pseudocódigos.

salaCirurgia												
	1	2	3	4	5	6	7	8	9	10	11	12
	2	1	1	2	1	1	1	1	1	1	1	1

salas[1]										
cirurgias	7	11	3	2	8	12	5	9	10	6
slotsIniciais	0	10	18	48	56	96	99	100	105	144
slotsLivres	0	0	28	0	32	0	0	0	31	189

salas[2]		
cirurgias	4	1
slotsIniciais	0	6
slotsLivres	0	328

Figura 1: Representação da solução usando quatro vetores.

Para compreensão da metodologia adotada, considera-se horário comercial como o período de segunda à sexta-feira, das 7h às 19h; considera-se como horário não comercial os dias de sábado, domingo e feriados e os horários de segunda à sexta-feira compreendidos das 19h às 07h. O período de agendamento que a solução representa é de uma semana, sendo que cada sala cirúrgica possui 336 *slots* de tempo (7 dias por semana  $\times$  24 horas por dia  $\times$  2 *slots* por hora). Na modelagem realizada, poderá haver *slots* livres em uma sala cirúrgica entre os horários comerciais e ao final do último horário comercial. Considera-se, para o intervalo de higienização das salas e equipamentos como tempo de *setup*, associado ao uso de 1 ou 2 *slots* de tempo.

#### 4.1. Representação de uma Solução

Uma solução para o PACÉE é representada por meio de quatro vetores. O primeiro vetor, denominado *salaCirurgia*, armazena a informação referente à sequência em que as cirurgias devem ser realizadas, ou seja, o vetor armazena o número da sala em que cada cirurgia será realizada. A partir desta informação, outros três vetores são utilizados para armazenar as informações de cada sala cirúrgica. Estes vetores são: o vetor *cirurgias*, que armazena o número da cirurgia; o vetor *slotIniciais*, que armazena os números de *slots* iniciais da cirurgia, e, finalmente, o vetor *slotsLivres*, responsável por armazenar a quantidade de *slots* livres ao final no agendamento semanal. A Figura 1 apresenta a solução para o problema usando os quatro vetores.

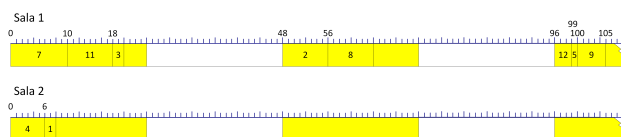


Figura 2: Representação da solução por meio de dois vetores.

A Figura 2 apresenta um exemplo de um agendamento semanal contemplando a realização de 10 cirurgias utilizando duas salas cirúrgicas. É possível notar a sequência de realização das cirurgias, os *slots* ociosos entre as cirurgias e se ocorrem em horário comercial ou não comercial. Nota-se, por exemplo, que a cirurgia 1 - com duração de 2 *slots* - e a cirurgia 4 - com duração de 6 *slots* são realizadas na sala 2; as demais cirurgias são realizadas na sala 1. Neste agendamento, o *makespan* corresponde ao *slot* de número 105; podendo ser minimizado, uma vez que há *slots* ociosos tanto na sala 1, quanto na sala 2, que podem receber as cirurgias de números 12, 5 e 9.



## 4.2. Estruturas de Vizinhanças

O espaço de busca consiste no conjunto de soluções  $s \in S$  que pertencem ao PACEE, em que se procura obter novas soluções por meio de movimentos de alocar, adiar e antecipar as cirurgias, tanto no vetor *salaCirurgia* como no vetor *cirurgias*, para a geração de vizinhos de  $s$ . Neste trabalho, foram utilizadas cinco diferentes estruturas de vizinhanças para se explorar o espaço de busca e minimizar o *makespan*. Para a exploração do espaço de soluções, vários tipos de movimentos foram testados, de forma que a escolha dos movimentos adotados reflete-se fortemente nos resultados alcançados. Na fase de construção do algoritmo GRASP foi utilizado a seguinte estrutura de vizinhança:

**AlocaCirurgiaPrimeiroHorarioComercial:** aloca a cirurgia  $i$  no final do primeiro horário comercial da sala  $j$  que comporte toda a cirurgia inteiramente em horário comercial, se houver este horário.

Na fase de Busca Local, conhecida também como fase de refinamento da construção GRASP, foram utilizados três estruturas de vizinhanças:

**AdiaCirurgiaMesmaSala:** passa a cirurgia  $i$  para o final do horário comercial em que esta está alocada, antecipando as cirurgias subsequentes a ela deste horário comercial. Caso a cirurgia  $i$  já esteja no final de um horário comercial, esta é passada para o final do horário comercial subsequente, sem antecipar o horário das demais cirurgias;

**AntecipaCirurgiaMesmaSala:** passa a cirurgia  $i$  para o final do horário comercial anterior ao que esta está alocada, antecipando as demais cirurgias subsequentes a ela do horário comercial de origem;

**RealocaCirurgiaSalaDiferente:** passa a cirurgia  $i$  para o final do mesmo horário comercial em que esta cirurgia estava alocada na sala  $j$  ou posterior, antecipando as cirurgias subsequentes do horário comercial da sala de origem.

Para a alocação das cirurgias de emergência, que surgem após a abertura do centro cirúrgico e início de realização das cirurgias, o algoritmo irá, primeiramente, percorrer o vetor *salas* e pesquisar *slots* ociosos que comportem toda a duração da cirurgia de emergência solicitada. Caso se encontrem *slots* disponíveis, a cirurgia é alocada; caso contrário, o algoritmo irá pesquisar por *slots* ociosos, considerando o período máximo de início de realização da cirurgia de emergência, a qual tem 6h após a sua solicitação para ser iniciada; por fim, o algoritmo irá alocar a cirurgia em *slots* ociosos, podendo utilizar *slots* não comercial. Foi utilizada uma estrutura de vizinhança, na forma:

**AlocaCirurgiaEmergencia:** aloca a cirurgia  $i$  no final do horário comercial que comporte toda a cirurgia, podendo utilizar os *slots* do horário não comercial subsequente, de modo o *slot* inicial da cirurgia seja o mais próximo possível após o horário em que esta foi solicitada.

## 5. Metaheurísticas Implementadas

### 5.1. Greedy Randomized Adaptive Search Procedure

A metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP), proposta por [Feo e Resende, 1995], é um procedimento de múltiplos reinícios e como tal é caracterizado pela realização de iterações composta por duas fases: (i) fase construtiva, que gera soluções factíveis

para o problema; e (ii) fase de busca local, que busca o ótimo local na vizinhança das soluções iniciais, geradas pela fase de construção.

O pseudocódigo do GRASP apresentado no Algoritmo 1 define uma *solucaoGrasp* construída de forma parcialmente gulosa com o método chamado *construcaoGRASP*, que consiste em escolher aleatoriamente a próxima cirurgia a entrar no sequenciamento, dentre uma lista de melhores candidatos. Essa lista é montada de acordo com a regra de menores tempos de execução e o parâmetro  $\alpha$  define o tamanho da lista. Na fase construtiva, uma função denominada *construcaoGRASP* é empregada, gerando uma Lista de Candidatos (LC). Tais candidatos são avaliados segundo o critério de Maior Tempo de Duração das cirurgias. Considere que o tempo necessário para a realização de uma cirurgia seja maior ou igual à diferença entre a maior duração, dada por  $T_{max}$ , e o valor  $\alpha(T_{max} - T_{min})$ , sendo  $\alpha$  um parâmetro real variando entre 0 e 1. Para  $\alpha = 1$ , tem-se uma solução totalmente aleatória; para  $\alpha = 0$ , tem-se uma solução gulosa. Os candidatos que atendem a esta condição compõem a Lista Restrita de Candidatos (LRC). Logo, o algoritmo irá selecionar aleatoriamente um dos candidatos (cirurgias) da LRC, sequenciando-o e retirando-o da LC. Este passo se repete até que todas as cirurgias sejam sequenciadas. Para a segunda fase, ou fase de busca local, foi utilizado o método de Descida Aleatória. Este método analisa parte da vizinhança da solução corrente. Os movimentos e vizinhos são escolhidos aleatoriamente. No final, efetua-se uma comparação entre a melhor solução encontrada e a solução corrente. Havendo melhora, a solução corrente é atualizada. O pseudocódigo do GRASP é mostrado no Algoritmo 1.

---

**Algoritmo 1:** Pseudocódigo GRASP

---

Entrada:  $\alpha$ , critérioParada

1. **início**
  2.   **repita**
  3.      $s \leftarrow \text{construcaoGRASP}(\alpha)$ ;
  4.      $s' \leftarrow \text{buscaLocal}(s)$ ;
  5.     **se**  $fo(s')$  melhor que  $fo(s^*)$  **então**
  6.        $s^* \leftarrow s'$ ;
  7.     **fim se**
  8.   **até** critérioParada seja satisfeito;
  9. **fim**
- 

## 5.2. Busca Local

Para a implementação da busca local, foi utilizado o Método Randômico de Descida. Esta utilização se justifica pela dimensão do espaço de busca do problema tratado e pelo número de combinações e movimentos possíveis na exploração do mesmo.

O método implementado consiste em analisar um conjunto de vizinhos gerados a partir de movimentos aleatórios, em que o melhor vizinho encontrado é comparado com a solução corrente. A solução vizinha será somente a nova solução corrente caso seja melhor ou igual à anterior. Para a geração da vizinhança, a cada iteração, todos os movimentos descritos na Seção 4.2 têm a mesma probabilidade de ocorrerem. No Algoritmo 2 é apresentado o pseudocódigo da busca local utilizada.

## 5.3. Variable Neighborhood Search

A metaheurística *Variable Neighborhood Search* (VNS), proposta por [Mladenović e Hansen, 1997], é uma metaheurística que se baseia em mudanças sistemáticas de vizinhança das soluções

---

**Algoritmo 2:** Pseudocódigo Busca Local - Método Randômico de Descida

---

**Entrada:**  $s$ ,  $\text{maxIter}$

1. **início**
  2.  $r \leftarrow$  Quantidade de movimentos (no caso,  $r = 5$ );
  3. **Para**  $i \leftarrow 1$  **até**  $\text{maxIter}$  **faça**
  4.    $s' \leftarrow$  seleciona aleatoriamente um vizinho (dentre as  $r$  vizinhanças);
  5.   **se**  $s'$  melhor que  $s$  **então**
  6.      $s \leftarrow s'$ ;
  7.   **fim**
  8. **fim**
  9. **retorna**  $s$ ;
  10. **fim**
- 

para resolver problemas de otimização. O método VNS explora vizinhanças gradativamente mais distantes da solução corrente e focaliza a busca em torno de uma nova solução se e somente se um movimento de melhora é realizado. O VNS inclui, também, um procedimento de busca local a ser aplicado sobre a solução corrente.

O pseudocódigo do VNS é apresentado no Algoritmo 3. É construída uma solução aleatória. Em seguida, a solução construída é refinada através de busca local, utilizando o método de Descida Aleatória, já explicado anteriormente. Depois um vizinho é escolhido aleatoriamente na vizinhança  $r$ . Após este procedimento, é refinada e, se houver melhora na solução, ela é atualizada e a vizinhança reinicializado.

---

**Algoritmo 3:** Pseudocódigo VNS

---

**Entrada:**  $\alpha$ ,  $\text{criterioParada}$

1. **início**
  2.  $s \leftarrow \text{construcaoAleatoria}(\alpha)$
  3.  $s^* \leftarrow \text{buscaLocal}(s)$
  4.  $r \leftarrow 1$
  5. **repita**
  6.    $s \leftarrow$  vizinho aleatório com movimento  $r$
  7.    $s' \leftarrow \text{buscaLocal}(s)$
  8.   **se**  $\text{fo}(s')$  melhor que  $\text{fo}(s^*)$  **então**
  9.      $s^* \leftarrow s'$
  10.     $r \leftarrow 1$
  11.   **senão**
  12.      $r \leftarrow r+1$
  13.   **fim**
  14.   **até**  $\text{criterioParada}$  seja satisfeito;
  15. **fim**
- 

## 6. Experimentos Computacionais

Os algoritmos GRASP e VNS foram implementados em linguagem Python. As instâncias têm os tempos de cirurgias gerados aleatoriamente entre 2 e 16 *slots* de tempo, com distribuição bi-



nomial mais um fator constante 1, se o tempo for menor que 8, e 2, caso contrário.

As instâncias testadas foram geradas a partir de dados reais de quatro hospitais localizados na região metropolitana de Belo Horizonte, Minas Gerais. Foram geradas randomicamente 71 novas instâncias. São instâncias com características diversas, determinada pela quantidade de cirurgias, quantidade de salas cirúrgicas, quantidade de salas de recuperação pós anestésica (RPA), quantidade de leitos de UTI, cirurgiões, enfermeiros e anestesistas. As instâncias variam entre 162 cirurgias e 836 cirurgias, a serem realizadas entre 7 e 53 salas cirúrgicas, respectivamente.

Os testes computacionais foram realizados em um computador com processador Intel Core i3-M330, 2.13 GHz, 3GB de RAM e sistema operacional Windows de 64 bits usando compilador G++ versão 5.4. Foram realizadas 30 execuções de cada algoritmo sobre cada instância, tendo-se, como critério de parada, o número de iterações, no caso, 100 iterações. Os resultados alcançados encontram-se na Tabela 3.

## 7. Análise Estatística e Teste *t-Student*

Para análise estatística e comprovação da existência de diferenças significativas entre os algoritmos implementados, foram utilizado vários testes *t-Student* de uma amostra unicaudal à esquerda. Considere  $\mu_0 = \{0, 8, 10, 12\}$  e  $H_1 : \mu > \mu_0$ . Seja  $f_A$  o valor de função objetivo alcançado pelo algoritmo  $A$  e  $t_A$  o tempo de execução do algoritmo  $A$ . Seguem os seguintes modelos dos testes de hipóteses:

$$\begin{cases} H_1 : t_{GRASP} < t_{VNS} \\ H_0 : t_{GRASP} \geq t_{VNS} \end{cases} \quad \begin{cases} H_1 : f_{GRASP} - f_{VNS} > \alpha \\ H_0 : f_{GRASP} - f_{VNS} \leq \alpha \\ \forall \alpha \in \{0, 8, 10, 12\} \end{cases} \quad (1)$$

Após analisar 30 execuções de cada algoritmo, em cada uma das 75 instâncias, conforme resultado na Tabela 3, percebe-se que é possível afirmar, com significância de 95%, que a resposta do algoritmo VNS tem um valor de função objetivo menor que as do algoritmo GRASP para 66 das 75 instâncias testadas, lembrando que o problema é de minimização. Pode-se afirmar também que, com 95% de confiança, o algoritmo VNS reduz o *makespan* em mais de 8 *slots* de tempo em 29 das 66 instâncias do item 1. Além disso, com 95% de confiança, o algoritmos VNS reduz o *makespan* em mais de 10 *slots* de tempo em 22 das 66 instâncias e, com 95% de confiança, o algoritmos VNS reduz o *makespan* em mais de 12 *slots* de tempo em 8 das 66 instâncias.

## 8. Conclusões

Este trabalho tratou o Problema de Agendamento de Cirurgias Eletivas com Incerteza de Alocação de Cirurgias de Emergências, enquadrando-o como um Problema de Programação de Tarefas em Máquinas Paralelas. Foram implementadas dois algoritmos heurísticos baseados em GRASP e VNS. Conclui-se com este trabalho e com os resultados apresentados, que o algoritmo VNS contribui para a resolução do Problema de Agendamento de Cirurgias Eletivas e de Emergência, demonstrando, assim, a importância de explorar várias vizinhanças e manter movimentos mais robustos. Como proposta de trabalhos futuros, os autores consideram subdividir o conjunto de cirurgias por grupos de especialidades médicas e seus níveis de prioridades no agendamento.

## Agradecimentos

Os autores agradecem ao CEFET-MG, à CAPES e ao CNPq, pelo apoio ao desenvolvimento deste trabalho.

Tabela 3: Resultados

	Média (GRASP - VNS) 30 Execuções	Desvio Padrão	teste-t (H1:f(vns)) (grasp), 95%)	Quantidade
h1	5,80	8,86	0,03029	1
h2	3,70	6,02	0,00308	1
h3	6,13	10,48	0,00121	1
h4	8,70	8,98847	0,12160	0
h5	10,10	12,06085	0,00992	1
h6	10,50	10,61056	0,01345	1
h7	8,83	11,60294	0,04773	1
h8	17,07	14,10040	0,00045	1
h9	16,74	17,85648	0,02517	1
h10	17,13	14,19503	0,00018	1
h11	16,80	17,89124	0,36330	0
h12	13,57	15,95147	0,01947	1
h13	16,03	11,30217	0,08686	0
h14	16,20	20,43237	0,01994	1
h15	15,43	15,78666	0,19173	0
h16	10,17	12,33262	0,00145	1
h17	5,13	7,00201	0,00001	1
h18	11,33	11,95096	0,00855	1
h19	6,90	8,10002	0,04556	1
h20	6,47	8,41044	0,00433	1
h21	10,10	11,13077	0,00124	1
h22	7,50	11,05422	0,00147	1
h23	12,47	13,30045	0,00506	1
h24	12,33	14,65835	0,00003	1
h25	15,27	17,61244	0,21713	0
h26	11,77	12,75647	0,00017	1
h27	10,07	12,27217	0,02025	1
h28	11,53	13,12201	0,00001	1
h29	16,73	16,13366	0,02605	1
h30	17,57	18,04340	0,00079	1
h31	7,17	10,56343	0,11177	0
h32	6,87	10,13609	0,01812	1
h33	6,93	10,12132	0,00879	1
h34	7,17	9,19105	0,00006	1
h35	6,17	9,679792	0,00067	1
h36	11,30	12,99402	0,00386	1
h37	6,97	8,63227	0,00044	1
h38	10,37	13,52771	0,00046	1
h39	7,90	9,35267	0,03299	1
h40	12,13	12,74615	0,00159	1
h41	19,93	17,44337	0,00359	1
h42	16,63	15,85165	0,00416	1
h43	16,00	18,90265	0,00094	1
h44	21,00	18,13551	0,03129	1
h45	9,10	12,11539	0,00245	1
h46	4,83	7,25441	0,02660	1
h47	5,50	7,17154	0,00277	1
h48	9,17	10,51135	0,01629	1
h49	7,03	9,41196	0,00576	1
h50	9,93	10,32217	0,00462	1
h51	10,00	12,41523	0,01028	1
h52	4,73	7,93914	0,00003	1
h53	8,63	8,89976	0,00277	1
h54	13,50	13,54367	0,18564	0
h55	14,63	13,28853	0,05643	0
h56	16,63	16,78358	0,00922	1
h57	12,53	15,45345	0,00014	1
h58	11,43	13,13244	0,00062	1
h59	14,57	14,75240	0,00683	1
h60	25,30	18,37755	0,02270	1
h61	6,37	9,65074	0,00614	1
h62	6,47	10,44108	0,02758	1
h63	8,07	9,78364	0,13333	0
h64	6,70	8,65420	0,00044	1
h65	6,07	7,58371	0,00185	1
h66	11,97	12,15839	0,00301	1
h67	12,07	12,25644	0,00876	1
h68	15,60	19,01953	0,01894	1
h69	11,17	12,42823	0,00004	1
h70	13,33	13,53994	0,00478	1
h71	18,97	16,19100	0,02872	1
h72	21,27	18,34076	0,00166	1
h73	15,67	18,55993	0,00193	1
h74	15,10	14,57015	0,00282	1
h75	14,43	14,63271	0,01231	1

## Referências

Bai, M. (2017). *Optimization of Surgery Scheduling in Multiple Operating Rooms with Post Anesthesia Care Unit Capacity Constraints*. Dissertation of Doctor of Philosophy in Industrial and Systems Engineering, Lehigh University, EUA, Bethlehem - Pensilvania.

- Balzer, C., Raackow, D., Hahnenkamp, K., Flessa, S., e Meissner, K. (2017). Timeliness of operating room case planning and time utilization: Influence of first and to-follow cases. *Frontiers in Medicine*, 4(49). URL <https://doi.org/10.3389/fmed.2017.00049>.
- Beliën, J. e Demeulemeester, E. (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176(2):1185–1204.
- Blake, J. T. e Carter, M. W. (2002). A goal programming approach to strategic resource allocation in acute care hospitals. *European Journal of Operational Research*, 140(3):541–561.
- Cardoen, B., Demeulemeester, E., e Beliën, J. (2009). Optimizing a multiple objective surgical case sequencing problem. *International Journal of Production Economics*, 119(2):354–366.
- Carter, M. W. e Tovey, C. A. (1992). When is the classroom assignment problem hard? *Operations Research*, 40(Supplement 1: Optimization):S28–S39.
- Denton, B., Viapiano, J., e Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, 10(1):13–24.
- Etzioni, D. A., Liu, J. H., Maggard, M. A., e Ko, C. Y. (2003). The aging population and its impact on the surgery workforce. *Annals of Surgery*, 238(2):170–177.
- 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*, 58:221–230.
- Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133.
- Fugener, A., Guenther Michael, E., Paskal, K., Ulrike, M., Schile, J., Steiner, F., Kolisch, R., e Blobner, M. (2016). Improving intensive care unit and ward utilization by adapting master surgery schedules. *A & A Practice*, 6:172–180.
- Guinet, A. e Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics*, 85(1):69–81.
- Hughes, W. L. e Soliman, S. Y. (1985). Short-term case mix management with linear programming. *Hospital & Health Services Administration*, 30(1):52–60.
- Jebali, A., Alouane, A. B. H., e Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, 99(1):52–62.
- Kharraja, S., Albert, P., e Chaabane, S. (2006). Block scheduling: Toward a master surgical schedule. In *Proceedings of the 2006 International Conference on Service Systems and Service Management (ICSSSM 2006)*, volume 1, p. 429–435. URL <https://doi.org/10.1109/ICSSSM.2006.320501>.
- Latorre-Núñez, G., Lüer-Villagra, A., Marianov, V., Obreque, C., Ramis, F., e Neriz, L. (2016). Scheduling operating rooms with consideration of all resources, post anesthesia beds and emergency surgeries. *Computers & Industrial Engineering*, 97:248–257.
- Magerlein, J. M. e Martin, J. B. (1978). Surgical demand scheduling: A review. *Health Services Research*, 13(4):418–433.

- Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100.
- Ozkarahan, I. (2000). Allocation of surgeries to operating rooms by goal programming. *Journal of Medical Systems*, 24(6):339–378.
- Persson, M., Hvitfeldt-Forsberg, H., Unbeck, M., Sköldenberg, O. G., Stark, A., Kelly-Pettersson, P., e Mazzocato, P. (2017). Operational strategies to manage non-elective orthopaedic surgical flows: a simulation modelling study. *BMJ Open*, 7(4). URL <https://doi.org/10.1136/bmjopen-2016-013303>.
- Persson, M. e Persson, J. A. (2009). Health economic modeling to support surgery management at a swedish hospital. *Omega*, 37(4):853–863.
- Pham, D.-N. e Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, 185(3):1011 – 1025.
- Proença, I. M. (2010). *Planeamento de Cirurgias Electivas: Abordagens em Programação Inteira*. Tese de doutorado, Departamento de Estatística e Investigação Operacional, Universidade de Lisboa, Lisboa, Portugal.
- Roland, B., Martinelly, C. D., Riane, F., e Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering*, 58(2):212 – 220.
- Siar, M. M. V., Gholami, S., e Ramezani, R. (2017). Multi-period and multi-resource operating room scheduling and rescheduling using a rolling horizon approach: a case study. *Journal of Industrial and Systems Engineering*, 10(special issue on healthcare):97–115.
- Testi, A., Tanfani, E., e Torre, G. (2007). A three-phase approach for operating theatre schedules. *Health Care Management Science*, 10(2):163–172.