

CENTRO UNIVERSITÁRIO FEI

RAPHAEL CARLOS CRUZ

O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS: uma
abordagem heurística por meio do método *Iterated Local Search*

São Bernardo do Campo

2017

RAPHAEL CARLOS CRUZ

O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS: uma
abordagem heurística por meio do método *Iterated Local Search*

Dissertação de mestrado, apresentada ao Centro Universitário FEI, como parte dos requisitos necessários para a obtenção do título de Mestre em Engenharia Mecânica com ênfase em Produção, orientada pelo Prof. Dr. João Chang Junior.

São Bernardo do Campo

2017

Carlos Cruz, Raphael.

O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS
ELETIVAS: uma abordagem heurística por meio do método Iterated
Local Search / Raphael Carlos Cruz. São Bernardo do Campo, 2017.
97 f. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. João Chang Junior.

1. Sequenciamento de Cirurgias Eletivas. 2. Métodos Heurísticos. 3.
Iterated Local Search. 4. Variable Neighborhood Descent. 5. Busca Tabu.
I. Chang Junior, João, orient. II. Título.

Aluno: Raphael Carlos Cruz

Matrícula: 215129-8

Título do Trabalho: O problema de sequenciamento de cirurgias eletivas: uma abordagem heurística por meio do método iterated local search.

Área de Concentração: Produção

Orientador: Prof. Dr. João Chang Junior

Data da realização da defesa: 19/12/2016

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

São Bernardo do Campo, 19 / 12 / 2016.

MEMBROS DA BANCA EXAMINADORA

Prof. Dr. João Chang Junior Ass.: _____

Prof. Dr. Marcone Jamilson Freitas Souza Ass.: _____

Prof. Dr. Pablo Maria Alberto Pomerantzeff Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO

REPROVADO

VERSÃO FINAL DA DISSERTAÇÃO

**APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE
FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA
EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Rodrigo Magnabosco

Aos meus pais, Elson e Vilma, por todo o apoio oferecido e pelo incentivo prestado. À minha namorada, Anita, pela paciência e por me fazer acreditar nesta grande conquista.

AGRADECIMENTOS

Ao meu orientador, professor Dr. João Chang Junior, que muito contribui para minha formação acadêmica e profissional durante todo o programa de mestrado. Todo o aprendizado e experiência transferidos foram de fundamental importância para o desenvolvimento deste trabalho.

Ao professor Dr. Marcone Jamilson Freitas Souza, por ter me ensinado tudo que sei hoje sobre otimização e pesquisa operacional. Participou ativamente do meu desenvolvimento acadêmico e, pelo qual, serei eternamente grato.

A toda equipe do Instituto do Coração (Incor), em especial ao Dr. Alfredo Fernandes, ao Dr. Pablo Maria Alberto Pomerantzeff e à Elisandra Arita, que muito contribuíram nos ensinamentos técnicos de operações hospitalares, assim como forneceram todo o suporte necessário para a modelagem do problema proposto por essa dissertação.

Aos professores do departamento de Produção do Programa de Mestrado em Engenharia Mecânica da FEI, pela transferência de conhecimento e pela prestatividade.

A todos os companheiros de turma do mestrado, pelo apoio e companheirismo durante toda essa jornada acadêmica.

A minha namorada Anita Flávia Gomes de Souza, pelo incentivo e companhia ao longo deste período.

Em especial aos meus pais, Elson Carlos Cruz e Vilma Imaculada de Andrade, pelo apoio e carinho, os quais realmente fizeram a diferença para o sucesso deste trabalho. Obrigado pelos ensinamentos e pela minha formação como pessoa.

A Deus, pela força de vontade e disciplina. Foram fatores determinantes para honrar com minhas obrigações neste presente trabalho.

Ao Centro Universitário FEI e à CAPES, pelo apoio financeiro o qual proporcionou-me a oportunidade de cursar um programa de mestrado em uma organização verdadeiramente conceituada.

Enfim, a todos que fizeram parte do desenvolvimento deste trabalho e contribuíram para o seu sucesso.

RESUMO

O presente trabalho tem seu foco no Problema de Sequenciamento de Cirurgias Eletivas. Este problema consiste em alocar um conjunto de cirurgias eletivas ao longo da agenda de programação levando em consideração os recursos hospitalares disponíveis, sendo eles: salas de cirurgias, equipamentos médicos, cirurgiões, enfermeiros, anestesistas, equipes de higienização, entre outros. O objetivo é apresentar a programação de cirurgias eletivas com o menor tempo total de processamento possível. Pelo fato de os recursos hospitalares serem na maioria das vezes escassos e de alto custo, o compartilhamento deles e a alocação de cirurgias de forma sistêmica se fazem necessários, uma vez que resolvendo este problema é possível diminuir o tempo total de execução das operações, aumentar o número de cirurgias realizadas e diminuir o tempo de espera dos pacientes. Para resolver este complexo problema combinatório, caracterizado como NP-difícil, propõe-se o algoritmo heurístico computacional denominado ESSILS (do inglês, *Elective Surgery Scheduling Iterated Local Search*), o qual combina as metaheurísticas *Iterated Local Search*, *Variable Neighborhood Descent* e Busca Tabu. Estudos bibliográficos comprovaram que a utilização de técnicas computacionais para a solução deste problema é muito pouco abordada no Brasil. À vista disso, pretende-se também que este trabalho auxilie os tomadores de decisão na tarefa de programar cirurgias eletivas nos mais diversos hospitais do país. Esta proposta foi testada em problemas-teste consagrados da literatura e também em problemas reais de um hospital de grande porte da cidade de São Paulo (Brasil), mostrando-se eficaz em relação às demais abordagens existentes.

Palavras-chave: Sequenciamento de Cirurgias Eletivas. Métodos Heurísticos. *Iterated Local Search*. *Variable Neighborhood Descent*. Busca Tabu.

ABSTRACT

The present work focuses on the Elective Surgeries Scheduling Problem. This problem consists to allocate a set of elective surgeries along the scheduling program considering the available hospital resources, namely: operating rooms, medical equipment, surgeons, nurses, anesthetists, cleaning teams, among others. The purpose is to present the elective surgeries scheduling with the lowest possible processing time. Due the hospital resources are, in most cases, scarce as well as expensive, sharing them and allocating surgeries in a systemic way are necessary, since solving this problem is possible to reduce the total execution time, increase the number of scheduled surgeries and reduce the waiting time for patients. To solve this complex combinatorial problem, characterized as NP-hard, it is proposed the computational heuristic algorithm called ESSILS (Elective Surgery Scheduling Iterated Local Search), which combines the metaheuristics Iterated Local Search, Variable Neighborhood Descent and Tabu Search. Published studies have shown that the use of computational techniques for this kind of problem is rarely addressed in Brazil. In view of this, this work also intends to contribute with decision makers during the elective surgeries scheduling activities in many hospitals of the country. This proposal was tested in benchmark problems of literature and also real problems of a large hospital in São Paulo (Brazil), being effective in relation to other existing approaches.

Keywords: Elective Surgeries Scheduling. Heuristic Methods. Iterated Local Search. Variable Neighborhood Descent. Tabu Search.

LISTA DE ILUSTRAÇÕES

Figura 1: Exemplo de um Problema de Sequenciamento de Máquinas	24
Gráfico 1: Procedimento de refinamento do ILS	29
Figura 2: Exemplo de escalonamento de tarefas em duas máquinas paralelas idênticas	33
Figura 3: Tempo médio de processamento de oito tipos diferentes de cirurgias	35
Figura 4: Tempo de preparação de oito tipos diferentes de cirurgias.....	35
Figura 5: Exemplo de um possível sequenciamento de cirurgias.....	35
Figura 6: Exemplo de sequenciamento de cirurgias com todos os recursos envolvidos.....	36
Figura 7: Exemplo de uma solução inicial para o problema de agendamento de cirurgias	38
Figura 8: Exemplo de uma solução refinada para o problema de agendamento de cirurgias ..	38
Figura 9: Parte da representação de uma solução com blocos cirúrgicos e cirurgias.....	47
Figura 10: Parte da representação de uma solução com demais recursos e cirurgias	47
Figura 11: Representação do Movimento Troca Ordem Cirurgia.....	49
Figura 12: Representação do Movimento Realoca Cirurgia	49
Figura 13: Representação do Movimento Troca Ordem Múltiplas Cirurgias	50
Figura 14: Representação do Movimento Realoca Múltiplas Cirurgias	50
Figura 15: Representação do Movimento Troca Cirurgia Entre Blocos	51
Figura 16: Representação do Movimento Realoca Cirurgia Entre Blocos.....	52
Figura 17: Representação do Movimento Troca Múltiplas Cirurgias Entre Blocos	52
Figura 18: Configuração da matriz da Lista Tabu antes do movimento	62
Figura 19: Configuração da matriz da Lista Tabu após o movimento	62
Gráfico 2: Probabilidade Empírica para o problema-teste Cirurgia Neurológica (3.7).....	81
Gráfico 3: Probabilidade Empírica para o problema-teste h79	82
Gráfico 4: Probabilidade Empírica para o problema-teste NEURO.....	83
Gráfico 5: Probabilidade Empírica para o problema-teste Congênito.....	84
Gráfico 6: <i>Boxplot</i> para o problema-teste Cirurgia Neurológica (3.7).....	87
Gráfico 7: <i>Boxplot</i> para o problema-teste Cirurgia h79	88
Gráfico 8: <i>Boxplot</i> para o problema-teste Cirurgia NEURO.....	89
Gráfico 9: <i>Boxplot</i> para o problema-teste Cirurgia Congênito.....	90

LISTA DE TABELAS

Tabela 1: Compatibilidade dos recursos de acordo com as cirurgias realizadas.....	45
Tabela 2: Conjunto de instâncias de média complexidade.....	65
Tabela 3: Conjunto de instâncias de alta complexidade.....	66
Tabela 4: Conjunto de instâncias com dados originais.....	67
Tabela 5: Conjunto de instâncias com dados retrabalhados (-10% salas de cir.).....	68
Tabela 6: Conjunto de instâncias com dados retrabalhados (-10% cirurgões).....	68
Tabela 7: Conjunto de instâncias com dados retrabalhados (+10% cirurgias).....	68
Tabela 8: Conjunto de instâncias de Daiki e Yih	69
Tabela 9: Conjunto de instâncias baseadas no problema real do hospital de grande porte de São Paulo	70
Tabela 10: Quadro de salas de cirurgias disponíveis.....	70
Tabela 11: Resultados para instâncias de média complexidade de Sperandio (2015)	74
Tabela 12: Resultados para instâncias de alta complexidade de Sperandio (2015)	75
Tabela 13: Resultados para instâncias de Andrade (2012) com dados originais.....	76
Tabela 14: Resultados para instâncias de Andrade (2012) com dados retrabalhados (-10% salas de cir.)	76
Tabela 15: Resultados para instâncias de Andrade (2012) com dados retrabalhados (-10% cirurgões).....	77
Tabela 16: Resultados para instâncias de Andrade (2012) com dados retrabalhados (+10% cirurgias).....	77
Tabela 17: Resultados para as instâncias de Daiki e Yih (2010).....	78
Tabela 18: Resultados para as instâncias baseadas no problema real do hospital de São Paulo	79
Tabela 19: Resultado para o Teste <i>t Student</i> - Algoritmo GRASP.....	85
Tabela 20: Resultado para o Teste <i>t Student</i> - Algoritmo GRASP + ILS + VND	85
Tabela 21: Resultado para o Teste <i>t Student</i> - Algoritmo ESSILS.....	86

LISTA DE ALGORITMOS

Algoritmo 1: Pseudocódigo do método GRASP	27
Algoritmo 2: Pseudocódigo do método ILS	28
Algoritmo 3: Pseudocódigo do método VND	30
Algoritmo 4: Pseudocódigo do método Busca Tabu	32
Algoritmo 5: Pseudocódigo do método de geração da solução inicial - GRASP.....	55
Algoritmo 6: Pseudocódigo do método de refinamento ILS.....	57
Algoritmo 7: Pseudocódigo do método de busca local VND.....	59
Algoritmo 8: Pseudocódigo do método de busca local BT	60
Algoritmo 9: Pseudocódigo da Lista Tabu	61
Algoritmo 10: Pseudocódigo dos Mecanismos de Perturbação	63

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	OBJETIVO GERAL.....	15
1.2	OBJETIVOS ESPECÍFICOS	15
1.3	JUSTIFICATIVA E RELEVÂNCIA	16
1.4	ESTRUTURA DO TRABALHO	16
2	REVISÃO BIBLIOGRÁFICA	18
2.1	O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS.....	18
2.2	O PROBLEMA DE SEQUENCIAMENTO DE MÁQUINAS	23
2.3	METODOLOGIAS HEURÍSTICAS.....	25
2.3.1	GRASP - Greedy Randomized Adaptive Search Procedures.....	26
2.3.2	ILS - Iterated Local Search.....	28
2.3.3	VND - Variable Neighborhood Descent.....	30
2.3.4	Busca Tabu.....	31
3	CARACTERIZAÇÃO DO PROBLEMA ABORDADO	33
3.1	DESCRIÇÃO COMPLETA DO PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS	33
3.2	FORMULAÇÃO MATEMÁTICA PARA O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS.....	39
4	METODOLOGIA.....	44
4.1	CONSIDERAÇÕES INICIAIS PARA O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS.....	44
4.2	REPRESENTAÇÃO DA SOLUÇÃO	46
4.3	ESTRUTURAS DE VIZINHANÇAS	48
4.4	FUNÇÃO DE AVALIAÇÃO	53
4.5	ALGORITMO PROPOSTO	53
4.5.1	Geração da Solução Inicial - GRASP.....	54
4.5.2	Algoritmo Proposto ESSILS - Iterated Local Search (ILS).....	56
4.5.3	Busca Local: Variable Neighborhood Descent (VND)	58
4.5.4	Busca Local: Busca Tabu.....	59
4.5.5	Mecanismos de Perturbação.....	63
5	EXPERIMENTOS COMPUTACIONAIS	64
5.1	CONJUNTO DE INSTÂNCIAS PARA O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS.....	64
5.1.1	Conjunto de Instâncias I	64

5.1.2	Conjunto de Instâncias II.....	67
5.1.3	Conjunto de Instâncias III	69
5.1.4	Conjunto de Instâncias IV	69
5.2	RESULTADOS COMPUTACIONAIS.....	71
6	ANÁLISE ESTATÍSTICA DOS RESULTADOS.....	80
6.1	TESTE DE PROBABILIDADE EMPÍRICA	80
6.2	TESTE DE DISTRIBUIÇÃO T DE STUDENT	84
6.3	GRÁFICOS BOXPLOT	86
7	CONCLUSÕES E TRABALHOS FUTUROS	91
	REFERÊNCIAS.....	94

1 INTRODUÇÃO

O problema de sequenciamento de cirurgias, do inglês *Surgical Case Scheduling* (SCS), também conhecido como o problema de agendamento de cirurgias, tem sido largamente estudado na literatura devido ao seu apelo social e sua importância prática nos hospitais, uma vez que seu foco principal está em realizar a gestão das operações médicas e reduzir os gastos com recursos hospitalares, sejam eles humanos ou de infraestrutura. Segundo Andrade (2012), hospitais querem reduzir custos e melhorar os seus ativos financeiros e administrativos, e, por outro lado, considerando a natureza da prestação do serviço, querem também elevar o nível de satisfação do paciente. Levando em consideração que o centro cirúrgico é o setor mais custoso de um estabelecimento de saúde e que, para Macario et al. (1995), é o principal gerador de receita dos hospitais, resolver este problema combinatório de alocação de recursos e agendamento de cirurgias implica em uma contribuição tanto de cunho profissional, como social e acadêmico.

É importante salientar que, segundo Jebali, Hadjalouane e Ladet (2006), a combinação do escalonamento de cirurgias com a programação do centro cirúrgico tendo como objetivo a redução das despesas hospitalares e a melhoria dos indicadores de qualidade no atendimento, passou a ser uma das prioridades mais importantes nas organizações hospitalares. É por este motivo que Min e Yih (2010) defendem que a abordagem para solução do problema de agendamento de cirurgias por meio de métodos de otimização será uma das ferramentas mais utilizadas pelos hospitais para redução de custos e melhoria dos indicadores de qualidade no atendimento de pacientes.

De acordo com Carter e Tovey (1992), o problema de sequenciamento de cirurgias consiste em definir uma agenda para a realização de cirurgias hospitalares com a intenção de minimizar a soma total dos tempos gastos nos conjuntos de operações a serem realizadas. Segundo Andrade (2012), a construção de um sequenciamento de cirurgias leva em consideração algumas variáveis já conhecidas: a duração média da cirurgia de acordo com seu porte, o tempo de higienização demandado (tempo de preparo do centro cirúrgico), o número de recursos disponíveis, sejam eles salas de cirurgias, equipamentos demandados, equipe médica, enfermeiros, anestesistas, disponibilidade da unidade de tratamento intensivo, leito pós operatório, entre muitos outros necessários à realização das operações. Sendo assim, todos estes recursos devem ser alocados, combinatoriamente, de forma que as cirurgias ocorram no

melhor momento possível respeitando o escalonamento sequencial e a prioridade de execução.

Dada a complexidade deste problema, visto que Carter e Tovey (1992) o caracterizam como sendo um problema de otimização combinatória da classe NP-Difícil (alta complexidade de solução), a abordagem de solução por métodos heurísticos tem sido fortemente explorada. De acordo com Dorigo e Stützle (2004), métodos heurísticos são algoritmos aproximados que procuram obter soluções de boa qualidade em um tempo computacional aceitável, porém não há garantia de se alcançar a solução ótima. Segundo Hertz e Widmer (2003), os problemas de otimização NP-Difíceis necessitam de um tempo computacional muito elevado quando se usa métodos exatos para a busca da solução, inviabilizando tal abordagem. Portanto, a utilização de métodos heurísticos é capaz de produzir soluções de boa qualidade e em tempo computacional viável, tornando-se a forma de abordagem mais adequada.

Baseado no trabalho de Bülbül (2011) e Silva (2012), pretende-se utilizar o método heurístico *Iterated Local Search* (ILS), combinado com dois outros métodos heurísticos: *Variable Neighborhood Descent* (VND) e Busca Tabu (BT), para resolver o Problema de Sequenciamento de Cirurgias Eletivas. Segundo Nashefa et al. (2012), cirurgias eletivas são aquelas em que o paciente é chamado para um procedimento cirúrgico com data previamente agendada. A motivação para abordar tal problema surgiu pelo fato de Bülbül (2011) aplicar tais métodos para o Problema de Sequenciamento de Máquinas tendo alcançado resultados muito bons. Já que Pham e Klinkert (2008) relacionam o problema de agendamentos de cirurgias ao problema de sequenciamento de máquinas, acredita-se que a abordagem de Bülbül (2011) será capaz de gerar resultados satisfatórios também para o contexto desta pesquisa. Portanto, questiona-se: esta metodologia é capaz de encontrar boas soluções para o atual problema estudado, uma vez que se relaciona muito à programação de cirurgias? Implementar o algoritmo proposto, assim como validá-lo em relação aos demais encontrados na literatura, será um forte indicador para verificar o quão eficiente ele será.

Com a intenção de verificar a capacidade de gerar boas soluções, ou seja, aquelas com o menor tempo de execução em relação ao melhor encontrado na literatura, a proposta em questão será testada em problemas-teste consagrados disponíveis nas bases de dados acadêmicas, assim como aplicada a um problema real de um hospital público e de grande porte da cidade de São Paulo, Brasil. Este segundo critério de avaliação consistirá em comparar os resultados obtidos pelo algoritmo proposto em relação à programação de cirurgias obtida pelos analistas de planejamento do hospital. Espera-se, portanto, alcançar

resultados ainda melhores do que aqueles encontrados pelos analistas, permitindo assim, a realização de mais cirurgias com os mesmos recursos existentes, uma vez que se objetiva reduzir o tempo total de execução do conjunto de cirurgias agendadas, permitindo a alocação de novas operações nos tempos disponíveis identificados.

1.1 OBJETIVO GERAL

Este projeto de pesquisa tem como objetivo principal desenvolver um algoritmo heurístico de otimização por meio da linguagem de programação C++ que seja eficiente para resolver o problema de sequenciamento de cirurgias eletivas. Para gerar a solução inicial será utilizado um método construtivo combinado com busca local chamado GRASP. Esta solução será então refinada pelo método heurístico *Iterated Local Search*, o qual utilizará dois outros métodos heurísticos, *Variable Neighborhood Descent* e Busca Tabu. Com isso, espera-se produzir soluções de boa qualidade, com o menor tempo possível de processamento das cirurgias, que irão contribuir para o adequado escalonamento das operações em instituições hospitalares. Espera-se também, comparar a eficiência de outros métodos heurísticos encontrados na literatura aplicados ao problema de sequenciamento de cirurgias com a atual proposta.

Além de desenvolver o algoritmo em questão, objetiva-se também validar a capacidade do algoritmo de produzir bons resultados, aplicando o método proposto tanto na solução de problemas reais como em instâncias disponíveis na literatura. E, com isso, é possível destacar a finalidade de tornar este objeto de pesquisa uma ferramenta eficaz para a programação de cirurgias eletivas, contribuindo para melhorar a qualidade no atendimento das instituições hospitalares, reduzir os custos operacionais e, sobretudo, diminuir o tempo de espera de pacientes, uma vez que será possível realizar um número maior de operações já que se espera reduzir o tempo total de execução delas.

1.2 OBJETIVOS ESPECÍFICOS

Como objetivos específicos da pesquisa, pode-se elencar os seguintes:

- a) fazer uma revisão de literatura sobre os diversos problemas relacionados a qualidade e gerenciamento do atendimento a pacientes em instituições de saúde;
- b) realizar um levantamento bibliográfico dos métodos de abordagem para solução do problema de programação de cirurgias eletivas;

- c) apresentar os conceitos sobre otimização por meio de ferramentas heurísticas, assim como seus benefícios frente aos métodos exatos de programação linear;
- d) estudar sobre métodos de solução do problema em questão: formulações estatísticas, simulação computacional e algoritmos heurísticos;
- e) comparar os diversos métodos da literatura e propor a melhor combinação deles para a solução do problema;
- f) contribuir com a divulgação de técnicas de otimização aplicadas à resolução do problema;
- g) contribuir com a formação de recursos humanos especializados nessa área do conhecimento.

1.3 JUSTIFICATIVA E RELEVÂNCIA

É um trabalho de grande importância acadêmica, já que trata de um problema de difícil solução na otimalidade. Devido à complexidade do problema abordado, sua resolução é feita normalmente por métodos heurísticos, os quais têm sido capazes de produzir soluções de alta qualidade em um curto tempo de processamento computacional, mesmo não garantindo o resultado ótimo (BELIËN; DEMEULEMEESTER, 2008).

Além disso, é um problema fortemente presente no cotidiano das instituições hospitalares. Outro fator é o centro cirúrgico, principal recurso do problema em questão, ser responsável por dois terços da receita financeira de um hospital, segundo Macario et al. (1995). É de grande importância social pelo fato de agir diretamente na satisfação do paciente e da equipe médica, já que segundo Andrade (2012) é possível diminuir a lista de espera de cirurgias, aumentar a eficiência do centro cirúrgico, melhorar a satisfação da equipe médica e dos pacientes, além de possibilitar uma gestão eficiente dos recursos da instituição, quando resolvido adequadamente.

1.4 ESTRUTURA DO TRABALHO

O restante do presente trabalho é organizado da seguinte maneira. A Seção 2 é destinada a apresentar o levantamento bibliográfico deste objeto de pesquisa, contemplando o problema de sequenciamento de máquinas, o problema de agendamentos de cirurgias eletivas, os métodos encontrados na literatura para resolvê-lo, assim como os diferentes critérios de

abordagem, desde aqueles relacionados ao padrão de qualidade de atendimento nas organizações hospitalares até aqueles relacionados à redução de custos operacionais. Na Seção 3 é apresentada a descrição do problema abordado. Na Seção 4, as metodologias heurísticas adotadas para solução do problema são então descritas, assim como as estruturas de vizinhança, a função objetivo e a representação da solução. A Seção 5 é reservada para descrever detalhadamente todas as instâncias utilizadas, desde as retiradas da literatura até aquelas criadas por meio do problema real de um hospital de grande porte da cidade de São Paulo, Brasil. Na Seção 6 são apresentados todos os resultados obtidos com o algoritmo proposto, as comparações com os demais encontrados na literatura, além das análises experimentais na ótica das soluções encontradas. Finalmente, na Seção 7, as conclusões gerais acerca do trabalho realizado, as propostas de trabalhos futuros, assim como as publicações acadêmicas alcançadas com a presente proposta são apresentadas.

2 REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica do presente trabalho é dividida em tópicos com a intenção de apresentar, com detalhes, toda a fundamentação teórica realizada para sustentar a metodologia de pesquisa adotada.

A Seção 2.1 é destinada a apresentar todos os referenciais teóricos estudados a respeito do problema de sequenciamento de cirurgias eletivas, assim como as metodologias utilizadas para resolvê-lo, desde a utilização de métodos matemáticos exatos até ferramentas de abordagem heurística. As variantes do problema em questão, as principais variáveis de entrada dos modelos de programação utilizadas, bem como os exemplos práticos de aplicação também são apresentados.

O problema de sequenciamento de máquinas, o qual é caracterizado como a base teórica para o problema de sequenciamento de cirurgias, é apresentado na Seção 2.2.

Na Seção 2.3, o conceito de métodos heurísticos é explorado, juntamente com a definição de todas as metodologias heurísticas utilizadas neste trabalho, desde a geração da solução inicial pelo método GRASP (*Greedy Randomized Adaptive Search Procedure*), até a heurística de refinamento ILS (*Iterated Local Search*), a qual é composta por dois outros métodos heurísticos, os quais atuam como métodos de busca local, sendo um deles o VND (*Variable Neighborhood Descent*) e o outro a Busca Tabu.

2.1 O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS

De acordo com Yang et al. (2015), o problema de sequenciamento de cirurgias é abordado por diversas técnicas, sejam elas ferramentas de programação matemática, métodos heurísticos ou até mesmo a união deles (algoritmos híbridos). Existem, na literatura, variados autores que estudam este tema, uma vez que trata de um problema diretamente ligado à eficiência dos hospitais, além de ser uma ferramenta muito útil para contribuir na organização e programação das tarefas que serão alocadas em todos os recursos envolvidos: cirurgião, enfermeiro, equipamentos, entre outros. Tem sido, nos dias de hoje, uma maneira de aumentar a satisfação de pacientes e funcionários dos hospitais, já que visa diminuir as filas de espera, utiliza melhor os recursos dos hospitais e contribui para o melhor escalonamento de atividades (MACARIO et al., 1995).

O agendamento de cirurgias eletivas de forma ordenada, respeitando fatores de prioridade de atendimento, assim como levando em consideração atingir objetivos

relacionados à melhora da qualidade no atendimento hospitalar e redução de custos, sejam eles minimizar o tempo total gasto na realização de todo o programa de cirurgias ou até mesmo aumentar o número de atendimentos sem a necessidade de investimentos, tem sido uma prática usual nas grandes organizações de saúde (MAY et al., 2011). De acordo com Oostrum et al. (2008), a utilização de métodos computacionais combinados com ferramentas de pesquisa operacional para alocar recursos (salas de cirurgias, equipes médicas, enfermeiros) de maneira inteligente e otimizada, é cada vez mais explorada para resolver o problema de agendamento de cirurgias eletivas nos grandes hospitais. Por se tratar de um problema combinatório complexo e quando solucionado de modo otimizado é capaz de trazer inúmeros benefícios, estudos nesta área crescem cada vez mais, tanto de cunho acadêmico, como corporativo (DENTON; VIAPIANO; VOLG, 2007).

O trabalho proposto por Cardoen, Demeulemeester e Beliën (2010) apresenta uma forte contribuição para a literatura da área de pesquisa operacional relacionada à gestão de instituições de saúde. Este artigo fornece uma revisão da pesquisa operacional recente sobre o planejamento e programação das salas de operação. Nesta mesma obra foi feita uma avaliação da literatura em vários campos que estão relacionados tanto à configuração (por exemplo, medidas de desempenho ou classes de pacientes) como às características técnicas do problema (por exemplo, a técnica de solução ou incorporação de incertezas). Reuniu-se e avaliou-se diversas metodologias aplicadas à solução do problema de agendamento de cirurgias e, além disso, foi possível identificar as tendências significativas nesta área de pesquisa sobre o planejamento das salas de operação, além de apontar as áreas carentes que precisam ser abordadas no futuro.

De acordo com Huang et al. (2012), a programação cirúrgica eficaz das salas de operações desempenha um papel significativo no rendimento e nos custos do hospital. Na grande maioria dos trabalhos encontrados na literatura, o agendamento cirúrgico considera apenas o processo de cirurgia na sala de cirurgia e ignora outras etapas que não devem ser deixadas de fora em situações reais. No trabalho de Huang et al. (2012), o problema de programação cirúrgica é considerado como um problema de programação híbrida baseado em uma variante do problema de sequenciamento de máquinas do tipo *flow-shop*. Para estes autores, leitos e salas cirúrgicas são representados como máquinas paralelas. Um modelo matemático para tal problema de programação cirúrgica é proposto e resolvido pelo *software* LINGO. Um estudo de caso com a sua solução ótima é também apresentado como forma de verificar o modelo proposto.

São diversas as abordagens encontradas na literatura para resolver o problema de agendamento de cirurgias. O trabalho de Zhong et al. (2012) utiliza de simulação computacional para resolver o problema de sequenciamento de cirurgias do Hospital de Shanghai. Assim como nos demais trabalhos, as cirurgias são consideradas como tarefas a serem realizadas e os recursos hospitalares (cirurgiões, enfermeiros, equipamentos) como as máquinas. O sistema informatizado desenvolvido aloca as cirurgias de menor tempo no início do processo e as de maior tempo no final, de modo que as primeiras sejam capazes de contribuir para o ajuste do sequenciamento, uma vez que são mais simples e de curta duração. O grau de utilização de cada recurso é utilizado como fator significativo da função objetivo, ou seja, recursos mais caros devem ter sua utilização minimizada e a quantidade de operações realizadas maximizada. Os resultados alcançados demonstraram ganhos bastante consideráveis, uma vez que aumentou o número de cirurgias em 10,33% e diminuiu o tempo de espera em 46%.

Para o sequenciamento de cirurgias, Chaabane et al. (2008) propõem comparar duas abordagens heurísticas (com restrições diferentes, mas de mesmo objetivo) baseadas nos métodos *Open Scheduling* e *Block Scheduling*. Métodos de programação linear, juntamente com as restrições de custo associadas aos recursos hospitalares são utilizadas para sustentar a função objetivo do problema. Dados reais de um hospital belga chamado Tivoli são utilizados para alimentar os dois algoritmos propostos. O objetivo é realizar a programação de cirurgias de maneira que se obtenha um cronograma cirúrgico eficaz, melhorando a coordenação dos serviços prestados pelo hospital, assim como o tempo de espera pelos pacientes. Ambas foram capazes de produzir bons resultados.

Segundo Saadouli et al. (2014), muitas variantes do problema de programação de operação são apresentadas na literatura para diferentes contextos. Na Tunísia, especialmente no Hospital Habib Bourguiba em Sfax, a política de programação aberta é considerada para agendar cirurgias eletivas. No serviço de ortopedia, a programação das salas de operações é realizada semanalmente. Para a seleção das cirurgias, a variação de duração das mesmas não é levada em consideração e, por este motivo, a média de durações das operações são consideradas para o planejamento e programação das salas. Ainda no artigo de Saadouli et al. (2014), constata-se que esta situação pode levar à geração de horas extras, o que é indesejável por cirurgiões e equipe cirúrgica, além de gerar um custo adicional para o hospital. Para lidar com este problema, foi proposto dois modelos estocásticos de programação inteira (programados no *software* CPLEX) que levam em conta a duração da cirurgia e tem como objetivo minimizar o tempo de espera dos pacientes.

A proposta de Fei, Meskens e Chu (2010) tem como objetivo maximizar a utilização dos centros cirúrgicos, minimizando o custo de horas extras e o tempo de ociosidade dos recursos físicos (equipamentos, unidade cirúrgica). Os autores dividem o problema em duas fases, sendo uma responsável por definir o escalonamento de cirurgias e a outra por definir a sequência de cirurgias em cada dia, levando em consideração a disponibilidade de leitos de recuperação. Para a primeira fase os autores propõem um modelo de programação inteira resolvido por um procedimento heurístico baseado em geração de colunas. Já para a segunda fase, é proposto um algoritmo genético híbrido baseado no problema de sequenciamento de máquinas *flow-shop*.

A proposta de Guo et al. (2014) também divide o problema de sequenciamento de cirurgias em duas partes, uma relacionada ao agendamento da cirurgia e a outra responsável pela alocação de recursos cirúrgicos. De início, os autores utilizam da programação inteira para definir o processo de agendamento de cirurgias e, em seguida, utiliza-se da heurística baseada em algoritmos genéticos para resolver o problema de sequenciamento de recursos. Devido à complexidade da segunda etapa, a abordagem heurística é muito indicada, uma vez que é capaz de produzir bons resultados para o problema em questão.

Segundo Kim e Horowitz (2002), o problema de sequenciamento de cirurgias também pode ser abordado por meio de simulação computacional. Os autores propõem utilizar um agendamento antecipado para cirurgias eletivas explorando o uso de um sistema de quotas diárias com agendamento de uma ou duas semanas. O objetivo é melhorar o desempenho de uma unidade de terapia intensiva (UTI) que atende pacientes provenientes de diferentes departamentos do hospital. A simulação computacional proposta realiza uma busca local analisando as possibilidades de agendamento de cirurgias de acordo com a demanda. Efeitos benéficos são observados para todos os problemas-teste disponíveis na literatura.

Mancilla e Storer (2001) abordam o problema de sequenciamento de cirurgias de uma forma diferente em relação aos demais trabalhos encontrados na literatura. Os autores alocam a mesma equipe cirúrgica em duas salas de operação paralelas, assim como no problema de sequenciamento de máquinas. Enquanto se realiza a operação em uma delas pelos cirurgiões, a equipe de higienização realiza o preparo da outra. O objetivo é reduzir o tempo de espera da equipe cirúrgica, assim como o custo com horas extras e o tempo de ociosidade. Para resolver este problema de otimização foi proposto um algoritmo inteiro estocástico que utiliza a técnica de aproximação média da amostra para realizar o sequenciamento lógico das cirurgias. Testes de cálculo com base em dados reais mostraram que os métodos propostos foram eficientes.

O trabalho de Khanna et al. (2012) apresenta a aplicação da metodologia de sistemas multiagentes para a modelagem de problemas com compartilhamento de recursos, como é o caso do agendamento de cirurgias eletivas. Neste artigo discute-se os desafios da otimização em problemas complexos como este, além de propor técnicas inteligentes capazes de conduzir algoritmos heurísticos para o melhor planejamento e gestão da programação de cirurgias.

A proposta de Andrade (2012) utiliza exclusivamente métodos heurísticos para resolver o problema de agendamento de cirurgias eletivas. As heurísticas GRASP (*Greedy Randomized Adaptive Search Procedure*), ILS (*Iterated Local Search*), VNS (*Variable Neighborhood Search*) e IGS (*Iterated Greedy Search*) são aplicadas com o objetivo de minimizar o *makespan*, ou seja, minimizar o instante de término da última cirurgia. O algoritmo proposto foi testado em problemas reais de hospitais de grande porte da cidade de Belo Horizonte (Brasil) e se mostrou capaz de produzir bons resultados.

O autor Sperandio (2015) resolve o problema de escalonamento de cirurgias por meio da combinação de dois métodos, otimização e mineração de dados. Sua intenção é focar na redução de custos do sistema hospitalar Português, o qual vem enfrentando graves crises nos últimos anos. As técnicas de programação inteira mista, algoritmos genéticos e programação multiobjetivo são aplicadas para resolver o problema de agendamento de cirurgias eletivas. O objetivo é desenvolver uma ferramenta eficiente para auxiliar a tomada de decisão das equipes de planejamento cirúrgico. A combinação destes métodos com simulação computacional também é aplicada afim de diminuir as incertezas quanto ao agendamento, ou seja, a finalidade principal é alcançar resultados realmente fidedignos, onde evita-se alterações do plano durante a execução do mesmo.

Os autores Daiki e Yih (2010) propõem resolver o problema de cirurgias eletivas por meio de procedimentos de otimização estocástica combinados com métodos de aproximação média da amostra. O objetivo principal proposto é a obtenção de um cronograma de cirurgias direcionado à minimização do custo total gerado pelos pacientes e também do custo de horas extras gerados pelos recursos humanos do hospital.

O trabalho proposto por Marchesi (2015) utiliza modelagem de processos e simulação de Monte Carlo com o intuito de estruturar a tomada de decisão dos hospitais quanto ao planejamento e programação das cirurgias. O objetivo é reduzir o índice de suspensão das operações por falta de vagas na unidade hospitalar. Ainda segundo a autora Marchesi (2015), nos dias de hoje a maioria das decisões em relação à gestão de leitos e salas de cirurgias são tomadas a partir de conhecimento tácito e observação empírica e, por isso, propõe-se a

aplicação de modelos sistêmicos para melhorar os indicadores de rendimento do atendimento hospitalar.

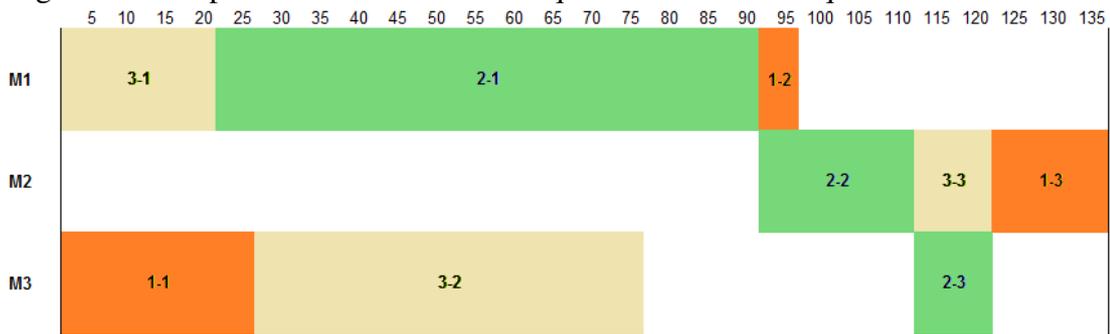
É importante salientar que os problemas-teste tanto do trabalho de Andrade (2012), de Sperandio (2015) e de Daiki e Yih (2010), assim como o problema real de um hospital de grande porte da cidade de São Paulo, Brasil, serão utilizados como laboratório de pesquisa para a validação do algoritmo proposto por esta dissertação.

2.2 O PROBLEMA DE SEQUENCIAMENTO DE MÁQUINAS

O Problema de Sequenciamento de Máquinas, conhecido do inglês como *Job-Shop Scheduling Problem*, é definido por Brucker, Jurisch e Sievers (1994) da seguinte forma: considera-se n tarefas (*jobs*) T_1 até T_n e m máquinas diferentes M_1 até M_m , onde cada tarefa T_i consiste de um número determinado de operações O_{i1} até O_{in} e cada uma destas operações contemplam a sequência de máquinas (recursos) em que cada uma das tarefas devem ocupar para que o trabalho, como um todo, seja concluído. Sabe-se também que cada operação O_{in} pode ser processada em somente uma máquina M_{in} em um determinado tempo de processamento e que uma tarefa T_{in} não pode ser processada por duas ou mais máquinas ao mesmo tempo. O principal objetivo da solução deste problema, fortemente estudado na literatura, é minimizar o tempo total de processamento das tarefas (conhecido do inglês como *makespan*), aferido no início da execução da primeira tarefa no tempo inicial t_0 até o término do processamento da última tarefa na última máquina t_n (PINEDO, 2005).

Um exemplo prático e ilustrado do problema de sequenciamento de máquinas pode ser observado na Figura 1. Este exemplo é composto por três diferentes máquinas (M1, M2 e M3) e por três diferentes tarefas, cada uma delas com três operações. No topo da Figura 1 pode-se notar uma escala de unidades de tempo que varia de 5 a 135. A tarefa de índice 3 (representada pela cor bege) realiza a primeira operação de índice 1 na máquina M1 durante 20 unidades de tempo, partindo em seguida para a máquina M3, onde a operação de índice 2 da tarefa 3 é processada durante 50 unidades de tempo e, por último, passa-se pela máquina M2 para processar a terceira operação desta mesma tarefa durante 10 unidades de tempo. Neste exemplo, quando as três operações são concluídas, pode-se dizer que a tarefa em questão foi encerrada com êxito. Portanto, onde se lê $i-j$, tem-se tarefa i e operação j , ou seja, onde se lê 1-3, por exemplo, tem-se tarefa de índice 1 e terceira operação de índice 3. O sequenciamento obtido para o problema representado pela Figura 1 possui tempo total de processamento 135 unidades de tempo, como pode ser observado.

Figura 1: Exemplo de um Problema de Sequenciamento de Máquinas.



Fonte: Autor.

Ainda na Figura 1 é possível notar que todas as restrições do problema são atendidas, isto é, a operação subsequente só é iniciada quando a anterior a esta é então encerrada, cada tarefa não é processada ao mesmo tempo por mais de uma máquina e que cada máquina processa uma operação por vez. Observa-se também que para a tarefa ser totalmente concluída, esta deve ter todas as operações finalizadas por cada uma das máquinas disponíveis. O processo em si é contínuo, não havendo interrupção durante a execução de nenhuma das tarefas e cada uma das operações possui um determinado tempo de processamento em cada uma das máquinas.

De acordo com Leung (2004), o sequenciamento é uma atividade baseada na administração de recursos que, na maioria das vezes, são escassos e de alto custo. Sua responsabilidade é designar qual atividade será alocada no determinado recurso necessário em um intervalo de tempo específico. Dessa forma, portanto, o sequenciamento estabelece o quanto de recurso é alocado para a execução de um conjunto de tarefas, tendo como principal objetivo a minimização do tempo total de processamento, assim como a redução do custo operacional. O objetivo, portanto, é encontrar a melhor maneira possível de alocar as operações, de cada uma das tarefas, nos recursos disponíveis, de tal forma que seja possível diminuir o tempo total de processamento (PINEDO, 2005).

De acordo com Adams, Balas e Zawack (1988), o sequenciamento de máquinas por meio de ferramenta computacional é uma maneira inteligente de obter os resultados de um problema tão complexo como este, uma vez que quanto maior é o número de tarefas e recursos, mais difícil é o encontro de uma solução factível. É por este motivo que abordagens por métodos heurísticos na solução de problemas como este são cada vez mais estudadas na literatura, já que realizar a programação de máquinas manualmente se torna algo completamente inviável, custoso e demorado. Desta forma, o algoritmo computacional tem a

capacidade de explorar as possibilidades de combinações entre tarefas e recursos de uma maneira extremamente rápida e eficaz (CHENG; GEN; TSUJIMURA, 1996).

Segundo Jain e Meeran (1999), o problema de sequenciamento de máquinas é um tema largamente encontrado no cotidiano das indústrias do setor de manufatura por parte do planejamento da produção, onde os produtos a serem fabricados são considerados como as tarefas a serem processadas e, os recursos disponíveis, são representados pelos maquinários da fábrica ou até mesmo pela mão de obra, dependendo do tipo de processo produtivo. Sabe-se, então, que para se obter um produto completo e acabado, é necessário a passagem deste por diversas etapas da produção. Cada uma destas etapas (operações) é realizada por um determinado tipo de recurso (máquina, mão de obra ou a combinação dos dois) em um tempo específico e, a etapa consecutiva só pode ser iniciada quando a anterior for concluída, uma vez que existe uma ordem de processos a ser seguida. É de conhecimento ainda que a grande maioria do setor manufatureiro produz uma gama variada de produtos, onde o compartilhamento de recurso se faz necessário, ou seja, diversos produtos são produzidos em linhas de montagem comuns, como é o caso da fabricação de automóveis, por exemplo. Este é um dos motivos principais para a realização de um sequenciamento de máquinas inteligente por meio computacional, já que alocar as operações nos recursos disponíveis, seguindo uma ordem conhecida, levando em consideração uma gama diversificada de produtos e a utilização de recursos compartilhados, é uma função muito complexa dentro da engenharia de operações (HADDAD, 2012).

2.3 METODOLOGIAS HEURÍSTICAS

Os problemas de otimização combinatória da classe NP-difícil possuem alto nível de complexidade e, na grande maioria das vezes, não é possível obter a solução ótima por meio de métodos matemáticos em tempo polinomial, ou seja, obter o melhor resultado possível em um tempo de processamento computacional aceitável. Ao contrário dos métodos matemáticos exatos, que garantem a solução ótima do problema, as metodologias heurísticas são capazes de encontrar soluções de boa qualidade, porém em um tempo computacional factível. Mesmo não garantindo a obtenção da solução ótima, é importante salientar que o tempo gasto por um método exato para encontrar a melhor solução é muito maior em ordem de magnitude do que aquele gasto pelas abordagens heurísticas (MARTÍ; REINELT, 2011).

De acordo com Mine (2009), metodologias heurísticas, no âmbito da otimização combinatória, são ferramentas de busca exploratória baseadas na experiência e na inteligência

artificial que são utilizadas com a intenção de encontrar soluções para os mais variados problemas de pesquisa operacional. Segundo o autor Bueno (2009), os métodos heurísticos são classificados como uma busca contínua e empírica, com vários ótimos locais, cujo resultado é o melhor que se pode encontrar sob determinadas condições, sejam elas pelo número de iterações ou pelo tempo de execução do algoritmo.

O trabalho de Mine (2009) classifica as metodologias heurísticas em dois grupos, sendo um de heurísticas construtivas e o outro de refinamento. De acordo com Souza (2009), uma heurística construtiva tem por objetivo construir uma solução, elemento por elemento. Estes elementos são constituídos pelo conjunto de soluções disponíveis no espaço de busca e são escolhidos a cada passo de acordo com a função de avaliação adotada, a qual, dependendo do problema abordado, pode ser de maximização ou minimização. Estabelece-se, então, uma regra para alocação dos elementos com o intuito de formar uma possível solução inicial.

Já as heurísticas de refinamento consistem em promover modificações em uma solução de forma a tentar melhorá-la. Elas são baseadas no conceito de vizinhança, ou seja, caminham pelo espaço de soluções possíveis por meio de alterações em seus elementos, encontrando assim, os vizinhos (novas soluções) obtidos a partir da solução inicial. Essa metodologia, portanto, inicia-se de uma determinada solução e percorre, a cada iteração, seus possíveis vizinhos de acordo com a regra de movimento estabelecida, obtendo assim, novas soluções. Na maioria dos casos, o algoritmo aceita apenas as soluções de melhora em relação a anterior gerada (SOUZA, 2009).

As próximas subseções do item 2.2 são utilizadas para apresentar todas as metodologias heurísticas utilizadas no presente trabalho.

2.3.1 GRASP - Greedy Randomized Adaptive Search Procedures

Segundo Souza (2009), a metaheurística GRASP, do inglês *Greedy Randomized Adaptive Search Procedures* e proposta por Feo e Resende (1995), consiste de um método adaptativo e iterativo formado de duas fases. Uma delas é a etapa de construção, na qual gera-se uma solução, elemento a elemento, seguindo um determinado padrão previamente estabelecido, sendo este determinístico, aleatório ou, até mesmo, com a combinação destes dois. A segunda fase é caracterizada pelo processo de busca local, ou seja, a partir de uma solução inicial gerada, aplica-se um método de busca local na vizinhança para garantir que o ótimo local desta solução construída seja encontrado.

O Algoritmo 1 ilustra o pseudocódigo de um procedimento GRASP para um problema de minimização. Por se tratar de um exemplo para um problema de minimização, a função objetivo f^* recebe um valor verdadeiramente grande representado por ∞ . Para um determinado número de iterações, definido previamente, constrói-se uma solução inicial (linha 3) de acordo com o padrão estabelecido e de acordo com o fator de aleatoriedade α adotado (varia de 0 a 1, sendo $\alpha = 0$ gera soluções puramente determinísticas e $\alpha = 1$ faz produzir soluções totalmente aleatórias). Na linha 4, pode ser observado a aplicação da busca local que garante o encontro de um ótimo local por meio da pesquisa na vizinhança gerada através da solução inicial. Se a função objetivo da solução encontrada for menor que a da solução corrente, esta então passa a ser a de melhor resultado encontrado (linha 5 a 8). Terminada todas as iterações, o algoritmo retorna a melhor solução encontrada (linha 11).

Algoritmo 1: Pseudocódigo do método GRASP.

Procedimento <i>GRASP</i>	
1	$f^* \leftarrow \infty$
2	para ($Iter = 1, 2, \dots, GRASPmax$) faça
3	<i>Construção</i> (α, s);
4	<i>BuscaLocal</i> (f, s);
5	se ($f(s) < f^*$) então
6	$s^* \leftarrow s$;
7	$f^* \leftarrow f(s)$;
8	fim se;
9	fim para;
10	$s \leftarrow s^*$;
11	Retorne s ;
fim <i>GRASP</i>	

Fonte: Souza (2009).

De acordo com Souza (2009), a fase de construção caracteriza-se pela formação da solução propriamente dita, elemento a elemento. A cada iteração desta fase, os próximos elementos candidatos a serem inseridos na solução são armazenados em uma lista de candidatos, seguindo um critério de alocação previamente determinado. A heurística é dita adaptativa porque os benefícios associados com a escolha de cada elemento são atualizados a cada iteração da fase de construção para refletir as mudanças oriundas da seleção do elemento anterior. A componente probabilística (representada por α) do procedimento reside no fato de que cada elemento é selecionado de forma aleatória a partir de um subconjunto restrito formado pelos melhores elementos que compõem a lista de candidatos.

2.3.2 ILS - Iterated Local Search

O método ILS, sigla do inglês para *Iterated Local Search*, foi proposto por Stützle e Hoos (1999). Segundo Lourenço, Martin e Stützle (2003), a metaheurística ILS é constituída por quatro componentes fundamentais, sendo eles: geração da solução inicial, mecanismo de perturbação, método de busca local e, por fim, critério de aceitação. A geração da solução, como o próprio nome diz, consiste em produzir uma solução inicial, seja por métodos determinísticos, heurísticos ou até mesmo híbridos, para então ser refinada pelo ILS. Esse refinamento é caracterizado pela busca no espaço de soluções possíveis com o objetivo de melhorar a solução já existente. O que se chama de mecanismo de perturbação é a operação realizada para modificar a solução corrente por meio de movimentos nos elementos que formam a solução. Esses movimentos são utilizados para guiar a solução corrente para outras intermediárias que, possivelmente, podem ser melhores. A busca local é responsável por percorrer a vizinhança originada a partir da solução corrente à procura de uma melhor ainda. O critério de aceitação é responsável por decidir a partir de qual solução a perturbação seguinte será efetuada.

Algoritmo 2: Pseudocódigo do método ILS.

Procedimento <i>ILS</i>	
1	Considere s_0 uma solução inicial;
2	$s \leftarrow BuscaLocal(s_0)$;
3	enquanto (<i>critério de parada não satisfeito</i>) faça
4	$s' \leftarrow Perturbação(s)$;
5	$s'' \leftarrow BuscaLocal(s')$;
6	$s \leftarrow CritérioAceitação(s, s'')$;
7	fim enquanto;
8	Retorne s ;
	fim <i>ILS</i>

Fonte: Souza (2009).

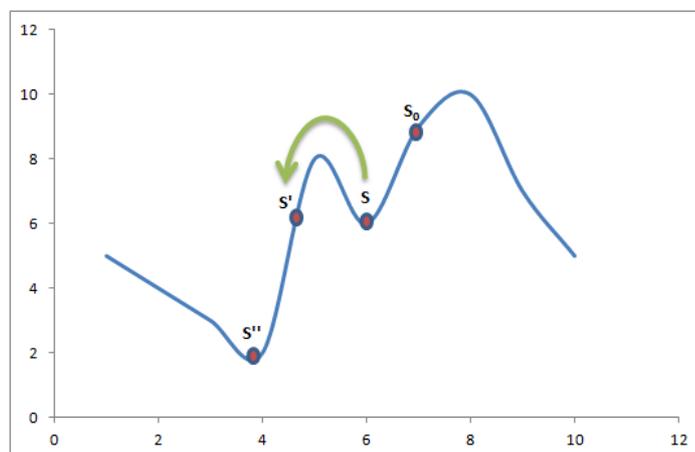
O Algoritmo 2 ilustra o pseudocódigo do método ILS onde os quatro elementos citados são observados. Assim que a solução inicial é gerada (linha 1), o método de busca local (linha 2) é aplicado com a intenção de se encontrar uma solução derivada da corrente que seja ainda melhor, ou seja, um ótimo local. Em seguida inicia-se o processo de refinamento, o qual é repetido até que um determinado critério de parada previamente determinado não seja atingido. Durante esta etapa (linha 4, 5 e 6), são realizados movimentos com os elementos da solução corrente com a intenção de produzir novas soluções por meio da

perturbação. A ferramenta de busca local também é aplicada nesta fase do algoritmo seguida do critério de aceitação, responsável por decidir de qual solução se continuará a exploração, bem como qual será a perturbação a ser aplicada.

Segundo Souza (2009), o sucesso do ILS é centrado no conjunto de amostragem de ótimos locais, juntamente com a escolha do método de busca local, das perturbações e do critério de aceitação. Em princípio, qualquer método de busca local pode ser usado, mas o desempenho do ILS com respeito à qualidade da solução final e a velocidade de convergência depende fortemente do método escolhido. A intensidade da perturbação deve ser forte o suficiente para permitir escapar do ótimo local corrente e permitir explorar diferentes regiões. Ao mesmo tempo, ela precisa ser fraca o suficiente para guardar características do ótimo local corrente.

O Gráfico 1 representa um bom exemplo de como a metaheurística ILS funciona para um problema de minimização. A partir de uma solução inicial qualquer S_0 aplica-se um método de busca local com o objetivo de melhorar a solução corrente, ou seja, encontrar um ótimo local. Neste caso a solução s é vizinha de s_0 e também melhor, uma vez que o valor da função objetivo de s é menor que s_0 e o exemplo trata de um problema de minimização. O processo de refinamento é iniciado juntamente com as perturbações, que neste caso são representadas pela seta em verde. A partir das perturbações gera-se, então, uma nova solução denominada de s' . Aplica-se então a busca local e um novo ótimo local s'' é encontrado. É possível notar que s'' é menor que s' , sendo aquela a nova solução corrente e a melhor encontrada até então. É por este motivo que o ILS é conhecido como um método eficiente, uma vez que é capaz de percorrer o espaço de soluções de maneira inteligente à procura de melhores resultados.

Gráfico 1: Procedimento de refinamento do ILS.



Fonte: Autor.

2.3.3 VND - Variable Neighborhood Descent

O método VND, do inglês *Variable Neighborhood Descent*, é um método de refinamento proposto por Mladenovic e Hansen (1997). A metaheurística VND resume-se em explorar o espaço de soluções através de mudanças sistemáticas na estrutura de vizinhanças, ou seja, por meio da realização de movimentos com os elementos da solução, buscando novas possibilidades para serem avaliadas.

O pseudocódigo deste método, o qual para o exemplo em questão considera-se um problema de minimização, onde uma determinada solução s é guiada no espaço de busca por um função objetivo f e é representada por um conjunto r de possíveis vizinhanças (soluções derivadas de S) denominadas de $N = \{N^{(1)}, N^{(2)}, \dots, N^{(r)}\}$, é representado pelo Algoritmo 3.

Algoritmo 3: Pseudocódigo do método VND.

Procedimento VND	
1	Seja r o número de estruturas diferentes de vizinhança;
2	$k \leftarrow 1$;
3	enquanto ($k \leq r$) faça
4	Encontre o melhor vizinho $s' \in N^{(k)}(s)$;
5	se ($f(s') < f(s)$)
6	então
7	$s \leftarrow s'$;
8	$k \leftarrow 1$;
9	senão
10	$k \leftarrow k + 1$;
11	fim se;
12	fim enquanto;
13	Retorne s ;
	fim VND

Fonte: Souza (2009).

De acordo com o Algoritmo 3, é possível observar que o método VND inicia-se verificando a primeira estrutura de vizinhança $N^{(1)}$, ou seja, analisa a próxima solução (vizinho) derivada da corrente que pertence à mesma bacia de atração. Segundo Souza (2012), dizer mesma bacia de atração significa que os vizinhos gerados a partir da solução inicial estão localizados na mesma região do espaço de busca que dará origem a um único ótimo local. A cada iteração do método, a função "Encontre o melhor vizinho" (linha 4) armazena o melhor vizinho s' derivado da solução corrente s e encontrado por meio da vizinhança $N^{(k)}$. Em caso de s' ser melhor que s , aquela solução passa a ser a corrente e a vizinhança retorna à

posição inicial com identificador I . Caso contrário, o algoritmo continua o refinamento pela próxima vizinhança $N^{(k+1)}$ até o encontro de uma solução melhor que a corrente. O método é encerrado quando não é possível encontrar uma solução $s' \in N^{(r)}$ melhor que a solução corrente.

Segundo Souza (2009), o método de refinamento do VND indica que um ótimo local frequentemente fornece algum tipo de informação sobre o ótimo global. Este é o caso em que os ótimos local e global compartilham muitas variáveis com o mesmo valor, o que sugere uma investigação sistemática da vizinhança de um ótimo local até a obtenção de uma nova solução de melhor valor.

2.3.4 Busca Tabu

A metaheurística Busca Tabu, proposta por Glover (1986) e descrita com mais detalhes por Glover e Laguna (1997), tem como objetivo percorrer o espaço de soluções, movimentando-se de um vizinho para o outro, procurando sempre o melhor vizinho da solução corrente. Segundo Silva (2012), esse processo de busca é auxiliado por uma estrutura de memória, chamada de Lista Tabu, responsável por armazenar movimentos proibidos, fazendo com que o algoritmo percorra outras posições do espaço de busca sem retornar a uma solução já gerada anteriormente.

A Lista Tabu funciona como uma fila de tamanho determinado onde, toda vez que um novo movimento tabu é adicionado à lista, aquele que é o mais antigo é retirado da lista. A Lista Tabu impede o retorno de uma solução já gerada anteriormente, mas pode proibir movimentos para soluções que ainda não foram visitadas. Para minimizar os efeitos desse problema, existe uma função de aspiração, que é um mecanismo que retira, sob certas circunstâncias, o status tabu de um movimento (SILVA, 2012).

Apresenta-se, no Algoritmo 4, o pseudocódigo do método Busca Tabu para um problema de minimização. O termo s^* armazena a melhor solução encontrada até então pelo algoritmo, enquanto $iter$ representa o número total de iterações da Busca Tabu. A variável $melhorIter$ guarda o número de iterações sem melhora a partir da solução inicial e T a Lista Tabu propriamente dita. Segundo Souza (2009), o algoritmo de Busca Tabu parte de uma solução inicial e , a cada iteração, explora um subconjunto V da vizinhança $N(s)$ da solução corrente. O membro s' de V com melhor valor nesta região segundo a função objetivo torna-se a nova solução corrente mesmo que s' seja pior que s , isto é, que $f(s') > f(s)$ para um problema de minimização, como é o caso do exemplo.

Duas regras são normalmente utilizadas de forma a interromper o procedimento. A primeira delas é quando se atinge um certo número máximo de iterações sem melhora no valor da melhor solução. Já a segunda, é quando o valor da melhor solução chega a um limite inferior conhecido. Esse segundo critério evita a execução desnecessária do algoritmo quando uma solução ótima é encontrada ou quando uma solução é julgada suficientemente boa (SOUZA, 2009).

De acordo com Souza (2012), os parâmetros principais de controle do método de Busca Tabu são a cardinalidade $|T|$ da Lista Tabu, a função de aspiração A , a cardinalidade do conjunto V de soluções vizinhas testadas em cada iteração e o número máximo de iterações $IterMaxBT$ sem melhora no valor da melhor solução. Todos estes parâmetros representados no Algoritmo 4 são responsáveis por guiar o procedimento de Busca Tabu pelo espaço de soluções, evitando que soluções já avaliadas sejam novamente consideradas.

Algoritmo 4: Pseudocódigo do método Busca Tabu.

```

Procedimento BuscaTabu
1    $s^* \leftarrow s$ ;
2    $iter \leftarrow 0$ ;
3    $melhorIter \leftarrow 0$ ;
4    $|T| \leftarrow 0$ ;
5   enquanto ( $f(s) \leq f_{min}$  &  $iter - melhorIter \leq iterMaxBT$ ) faça
6        $iter \leftarrow iter + 1$ ;
7       Seja  $s'$  o melhor elemento  $V \subset N(s)$  tal que o movimento  $m$  não
          seja tabu ou  $s'$  atenda a condição de aspiração ( $f(s') \leq A(f(s))$ );
8       Atualiza a Lista Tabu  $T$ ;
9        $s \leftarrow s'$ ;
10      se ( $f(s) < f(s^*)$ ) então
11           $s^* \leftarrow s$ ;
12           $melhorIter \leftarrow iter$ ;
13      fim se;
14      Atualize a função de aspiração  $A$ ;
15  fim enquanto;
16  Retorne  $s^*$ ;
fim BuscaTabu

```

Fonte: Souza (2012).

3 CARACTERIZAÇÃO DO PROBLEMA ABORDADO

O Problema de Sequenciamento de Cirurgias Eletivas abordado pelo trabalho em questão é, então, descrito na presente Seção. A Seção 3.1 é destinada a descrever o problema de sequenciamento de cirurgias, assim como suas restrições e principais premissas. Logo em seguida, na Seção 3.2, a formulação matemática para o problema de sequenciamento de cirurgias eletivas é apresentada.

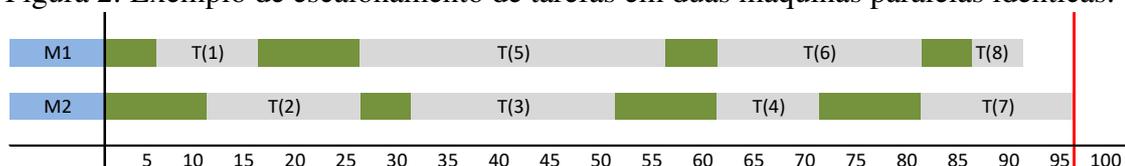
3.1 DESCRIÇÃO COMPLETA DO PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS

O sequenciamento de cirurgias eletivas abordado por esta dissertação é baseado no problema de programação em máquinas paralelas, onde o objetivo principal é a minimização do somatório dos tempos de processamento. De acordo com Pinedo (2008), o problema de programação em máquinas paralelas é representado por um conjunto de n tarefas, onde $N = \{1, \dots, n\}$, a serem processadas por um conjunto de m máquinas idênticas M_1, \dots, M_m , estruturado sob as seguintes restrições:

- cada tarefa deve ser processada uma única vez e por apenas uma máquina;
- cada tarefa i possui um tempo de processamento p_i ;
- existe um tempo de preparação da máquina denominado de t_i , onde a cada execução de uma tarefa i a máquina em questão deve ser preparada para o recebimento da próxima tarefa $i+1$;
- o processo é contínuo, não havendo interrupção durante a execução de nenhuma das tarefas.

A Figura 2 representa um exemplo para a programação de tarefas em máquinas paralelas idênticas. É possível observar todas as tarefas $T(n)$ sequenciadas em cada uma das duas máquinas idênticas disponíveis ($M1$ e $M2$), assim como o tempo de preparação anterior à execução das tarefas representado em verde. A escala de tempo é representada pelo eixo X.

Figura 2: Exemplo de escalonamento de tarefas em duas máquinas paralelas idênticas.



Fonte: Autor.

Os autores Pham e Klinkert (2008) propõem utilizar as técnicas de Programação Inteira Mista para a solução do Problema de Agendamento chamado de *Multi-Mode Blocking Job-Shop*. Relacionando o trabalho destes autores com o problema de máquinas paralelas, define-se, para a atual proposta, a equivalência entre máquina e bloco cirúrgico, assim como tarefa e cirurgia. Essa mesma abordagem é também encontrada no trabalho de Andrade (2012) que utiliza técnicas heurísticas para solucionar o problema de agendamento de cirurgias eletivas.

Desta forma, segundo Pham e Klinkert (2008), o problema de sequenciamento de cirurgias eletivas é representado por um conjunto C de n cirurgias, onde $C = \{1, \dots, n\}$, a serem realizadas em um conjunto B de b blocos cirúrgicos B_1, \dots, B_b . As restrições deste problema são assim apresentadas:

- a) cada cirurgia n deve ser processada uma única vez e por apenas um bloco cirúrgico b ;
- b) cada cirurgia n possui um tempo de processamento t_n ;
- c) um determinado bloco cirúrgico b não pode realizar mais de uma cirurgia no mesmo período de tempo. Essa restrição também é válida para os demais recursos envolvidos, sejam eles médicos cirurgiões, anestesistas, equipamentos, entre outros. Esta restrição garante que determinado recurso não seja alocado em mais de uma cirurgia no mesmo instante de tempo;
- d) existe um tempo de higienização do bloco cirúrgico denominado de p_n , onde a cada realização de uma cirurgia n o bloco cirúrgico utilizado deve ser higienizado para o recebimento da próxima cirurgia $n + 1$;
- e) o processo em questão é contínuo, ou seja, não há interrupção de nenhuma cirurgia. Uma vez iniciada, a mesma deve ser concluída.

A solução do problema de sequenciamento de cirurgias só é possível quando se tem um mapeamento completo dos recursos hospitalares, ou seja, quantidade de enfermeiros, de médicos cirurgiões, de anestesistas, de blocos cirúrgicos, de unidades de tratamento intensivo, de equipamentos em geral, entre outros itens necessários para a realização das operações. É importante também ter conhecimento do tempo médio de duração de cada uma das cirurgias realizadas, além do tempo necessário para a preparação (higienização, configuração e alocação de equipamentos, separação de instrumentos, etc.) que cada cirurgia necessita. Tendo todas essas informações, o algoritmo de otimização é capaz de combinar a alocação de cirurgias de acordo com os recursos disponíveis, buscando resultados onde o tempo total de processamento seja cada vez menor.

Na Figura 3 é possível observar um exemplo com os tempos médios de processamento (t_n) de oito tipos diferentes de cirurgias (n), enumeradas de 1 a 8. Em seguida, na Figura 4, nota-se o tempo de higienização (p_n) necessário para cada uma destas oito cirurgias (n).

Figura 3: Tempo médio de processamento de oito tipos diferentes de cirurgias.

n	1	2	3	4	5	6	7	8
t_n (min)	86	64	122	49	136	36	71	53

Fonte: Autor.

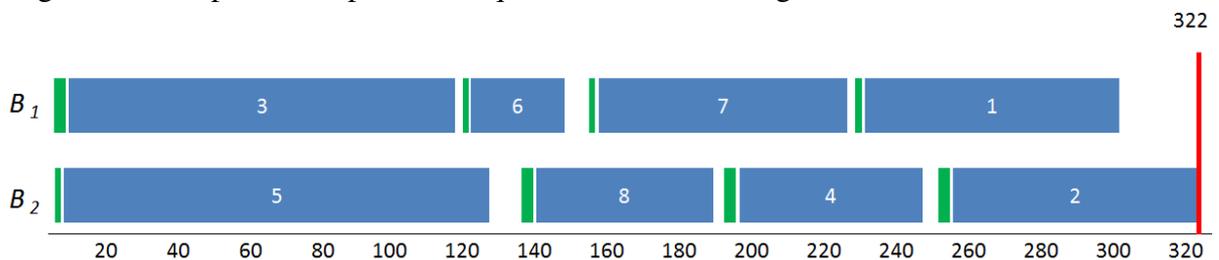
Figura 4: Tempos de preparação de oito tipos diferentes de cirurgias.

n	1	2	3	4	5	6	7	8
p_n (min)	5	10	10	10	5	5	5	10

Fonte: Autor.

Um exemplo prático de programação de cirurgias pode ser observado na Figura 5, onde as oito cirurgias citadas no parágrafo anterior são alocadas de forma sequencial em dois blocos cirúrgicos (recursos hospitalares), denominados de B_1 e B_2 . Os retângulos pintados em verde representam o tempo de preparação e higienização demandado por cada uma das cirurgias. A cirurgia de índice 6 ($n=6$), por exemplo, é realizada no bloco cirúrgico B_1 durante 36 unidades de tempo ($t_n=36$), tendo a cirurgia 3 ($n=3$) como predecessora e a cirurgia 7 ($n=7$) como sucessora. O instante de término da última cirurgia, ou seja, o tempo total (*makespan*) gasto para a realização de todas as cirurgias deste sequenciamento é equivalente a 322 unidades de tempo.

Figura 5: Exemplo de um possível sequenciamento de cirurgias.



Fonte: Autor.

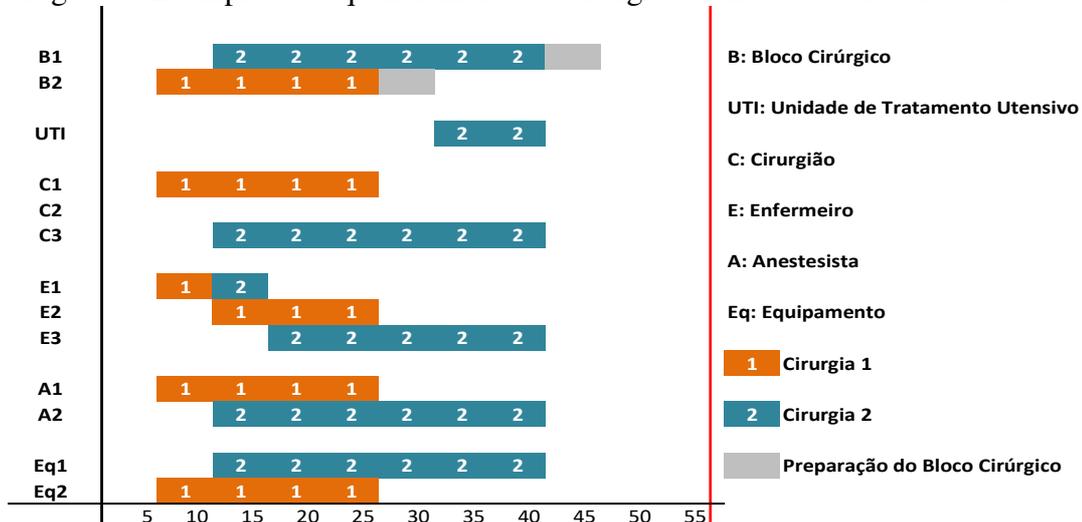
O problema de agendamento de cirurgias apresentado até então está intimamente relacionado à alocação de cirurgias em salas de operação, assim como se caracteriza o sequenciamento em máquinas paralelas. Porém, conforme já mencionado anteriormente, o problema real do sequenciamento de cirurgias envolve diferentes tipos e quantidades de recursos, podendo ser compartilhados ou não, além de diversos tipos de cirurgias, cada uma

com suas particularidades: tempo médio de duração, tempo de preparação e grau de risco (MARQUES; CAPTIVO; PATO, 2012).

De acordo com todas essas definições apresentadas, assim como o real entendimento da complexidade do problema em questão, percebe-se a necessidade de modelar o sequenciamento de cirurgias eletivas numa ótica mais detalhada em relação aos recursos hospitalares. Recursos estes que são de alto custo, escassos e, por isso, devem ser utilizados de forma planejada (MACARIO et al., 1995). É por este motivo que o problema de sequenciamento de cirurgias eletivas, abordado por esta dissertação, adota um planejamento completo de cirurgias levando em consideração a utilização de diversos recursos (blocos cirúrgicos, equipamentos médicos, enfermeiros, médicos cirurgiões, anestesistas) e o compartilhamento deles.

A Figura 6 ilustra um exemplo hipotético para planejamento de cirurgias, assim como todos os recursos necessários envolvidos no processo. Nota-se que a Cirurgia 1 é realizada no bloco cirúrgico 2 (B2) pelo cirurgião 1 (C1), além de ser iniciada pelo enfermeiro 1 (E1) e concluída pelo 2 (E2). Esta utilizou-se do anestesista 1 (A1) e do conjunto de equipamentos 2 (Eq2). Já a cirurgia 2 foi realizada no bloco cirúrgico 1 (B1) pelo cirurgião 3 (C3). A unidade de tratamento intensivo foi utilizada durante o processo por algum motivo específico. Os enfermeiros 1 (E1) e 3 (E3) foram envolvidos nesta segunda cirurgia, além do anestesista 2 (A2) e conjunto de equipamentos 1 (Eq1). É importante salientar a utilização comutativa dos enfermeiros e também o tempo de preparo (representado pelo retângulo cinza) do bloco cirúrgico logo após cada cirurgia para garantir o adequado recebimento das próximas operações. O eixo X representa a escala em unidades de tempo.

Figura 6: Exemplo de sequenciamento de cirurgias com todos os recursos envolvidos.

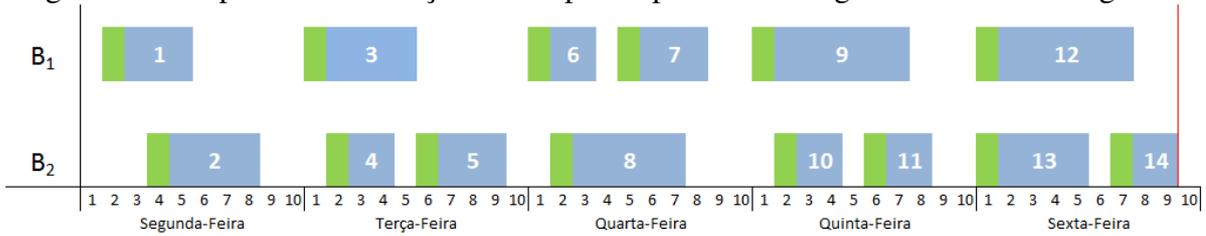


Fonte: Autor.

Até então foi possível entender a origem do problema de sequenciamento de cirurgias eletivas, como representá-lo e, sobretudo, tomar conhecimento sobre o compartilhamento de recursos hospitalares necessários para a realização de cada operação. A partir de então, o objetivo será compreender que, por meio da realocação dos recursos hospitalares juntamente com as cirurgias, é possível obter resultados da programação de cirurgias cada vez melhores, ou seja, realizar as mesmas cirurgias, porém em um período de tempo menor. Isto significa que o algoritmo consegue alcançar resultados melhores que aqueles obtidos pela programação manual, simplesmente realizando intensas modificações na alocação de recursos, os quais ora são usados em uma cirurgia, ora em outra. A velocidade de processamento do método computacional permite a avaliação de inúmeras possibilidades de realocação de recursos em um curto período de tempo, tendo como objetivo a minimização do tempo total de execução. Essa busca rápida e constante por novas combinações de recursos é que possibilita o encontro de soluções ainda melhores.

A Figura 7 ilustra um exemplo de agendamento de quatorze cirurgias a serem realizadas em um período de uma semana, iniciando na segunda-feira e terminando na sexta-feira. É possível observar que cada dia possui dez horas para a utilização dos recursos hospitalares na execução das cirurgias e que dois blocos cirúrgicos B_1 e B_2 estão sendo utilizados. Cada dia foi dividido em um período de tempo equivalente a uma hora. Os retângulos em verde representam o tempo de preparação necessário para cada cirurgia e em azul o tempo médio de realização de cada uma das cirurgias propriamente ditas. Além disso, nota-se que o tempo total (*makespan*) de execução de todas as quatorze cirurgias foi de 49 horas corridas, ou seja, ao final da nona hora de sexta-feira quando a última cirurgia foi concluída. O exemplo da Figura 7 não representa diretamente a utilização de todos os recursos, apenas do bloco cirúrgico. Porém, os demais recursos estão implicitamente sendo utilizados, uma vez que a programação apresentada como resultado leva em consideração a utilização de cada recurso para cada cirurgia em um determinado período de tempo, já que o mesmo recurso não pode ser usado simultaneamente por mais de uma operação. Por exemplo, os recursos utilizados na cirurgia 3 de terça-feira não podem ser os mesmos utilizados na cirurgia 4, já que essas duas cirurgias estão ocorrendo simultaneamente. Isto prova, portanto, que a solução apresentada na Figura 7 contempla a utilização de todos os recursos hospitalares necessários para a realização das cirurgias, mesmo que não representado diretamente no esquema.

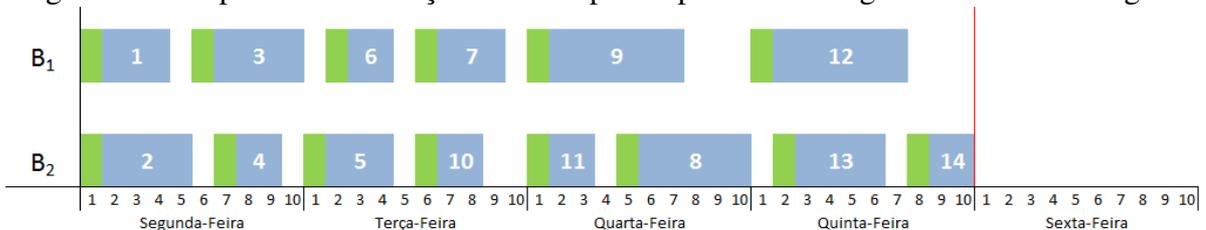
Figura 7: Exemplo de uma solução inicial para o problema de agendamento de cirurgias.



Fonte: Autor.

Já na Figura 8 é possível observar uma solução refinada a partir daquela apresentada na Figura 7. Por meio de processos de realocação de recursos e cirurgias, realizados pelo algoritmo heurístico computacional, foi possível obter uma solução melhor que a inicial, ou seja, com um tempo total de execução de 40 horas (última cirurgia concluída na décima hora de quinta-feira). Toda essa realocação de recursos fez a cirurgia 8, por exemplo, ser realizada depois que a cirurgia 10 e 11 que, inicialmente, estavam programadas para acontecer antes. Essas realocações são caracterizadas por movimentar as cirurgias e recursos disponíveis de acordo com as combinações possíveis do esquema de agendamentos, como por exemplo, mudar a ordem de execução das cirurgias, alocar as cirurgias em diferentes recursos, mas que sejam suficientes para realizá-las, entre outras possibilidades. Este exemplo mostra a capacidade que algoritmos de otimização tem de obter soluções ainda melhores que as encontradas manualmente pelos analistas. Com este resultado, mostra-se que é possível reduzir a fila de espera das cirurgias, uma vez que as mesmas 14 cirurgias foram realizadas num período de tempo menor que o anterior. O espaço ocioso obtido após o refinamento da solução inicial possibilita que mais cirurgias sejam agendadas, contribuindo para a melhoria dos indicadores de desempenho do hospital, uma vez que se nota a possibilidade de realizar mais com a mesma quantidade de recursos.

Figura 8: Exemplo de uma solução refinada para o problema de agendamento de cirurgias.



Fonte: Autor.

3.2 FORMULAÇÃO MATEMÁTICA PARA O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS

Existem na literatura inúmeros modelos matemáticos que representam o problema de sequenciamento de cirurgias. Nesta presente Seção será apresentado a formulação matemática proposta por Proença (2010), a qual aborda o agendamento de cirurgias com foco na redução da fila de espera dos pacientes e na melhoria da eficiência hospitalar.

Conforme adotado por Proença (2010), sejam os seguintes conjuntos e parâmetros de entrada do problema em questão:

- a) C : conjunto de cirurgias eletivas a serem realizadas;
- b) E : conjunto das especialidades cirúrgicas;
- c) P : conjunto dos níveis de prioridade, sendo $P = \{1$ (urgência identificada), 2 (prioridade alta), 3 (prioridade média), 4 (prioridade normal) $\}$;
- d) C^e_j : conjunto de cirurgias da especialidade $j \in E$ a serem realizadas;
- e) C^p_i : conjunto de cirurgias com nível de prioridade $i \in P$ a serem realizadas;
- f) C^n : conjunto de cirurgias a serem realizadas que não são classificadas como “urgência identificada”;
- g) H : conjunto de cirurgiões disponíveis;
- h) S : conjunto de salas de cirurgia disponíveis;
- i) D : conjunto de dias disponíveis para a marcação de cirurgias;
- j) T : conjunto de períodos de tempo diários disponíveis para a marcação de cirurgias;
- k) e_c : especialidade da cirurgia $c \in C$;
- l) h_c : cirurgião que realiza a cirurgia $c \in C$;
- m) t_c : tempo de duração previsto para a cirurgia $c \in C$, em número de períodos de tempo;
- n) T_c : subconjuntos de períodos de tempo em que a cirurgia $c \in C$ pode ser iniciada de modo a ser concluída previsivelmente dentro do tempo regular do bloco operatório;
- o) i_{ctd} : impossibilidade da cirurgia $c \in C$ ser iniciada no período de tempo $t \in T_c$ no dia $d \in D$;
- p) T^{MAXD}_{hd} : número máximo de períodos de tempo em operação para o cirurgião $h \in H$ no dia $d \in D$;
- q) T^{MAXS}_h : número máximo semanal de períodos de tempo em operação para o cirurgião $h \in H$;

As variáveis de decisão utilizadas no modelo para compor as restrições e função objetivo, são assim descritas de acordo com Proença (2010):

- a) x_{cst1} : variável do tipo inteira e binária, de tal modo que $x_{cst1} = 1$ se o início da cirurgia c , classificada como “1 - urgência identificada”, é agendada para o período de tempo t do dia 1 na sala s e, para $x_{cst1} = 0$, o contrário;
- b) x_{cstd} : variável do tipo inteira e binária, de tal modo que $x_{cstd} = 1$ se o início da cirurgia c , não classificada como “1 - urgência identificada”, é agendada para o período de tempo t do dia d na sala s e, para $x_{cstd} = 0$, o contrário;
- c) y_{jsd} : variável do tipo inteira e binária, de tal modo que $y_{jsd} = 1$ se pelo menos uma cirurgia da especialidade j é iniciada na sala s no dia d e, para $y_{jsd} = 0$, o contrário;

O modelo matemático adotado por Proença (2010) trata as cirurgias com maior prioridade de urgência identificada (fator de prioridade $P = 1$) como um caso a parte, uma vez que essas cirurgias devem ser alocadas no primeiro dia do horizonte de planejamento. No modelo apresentado a seguir essa tratativa será notada em algumas restrições, onde a variável $d \in D$ recebe o valor 1 quando se trata de cirurgias de urgência identificada.

$$\text{Minimizar } z = \sum_{c \in C_1^p} \sum_{s \in S} \sum_{t \in T_c} t_c x_{cst1} + \sum_{c \in C^n} \sum_{s \in S} \sum_{t \in T_c} \sum_{d \in D} t_c x_{cstd} \quad (1)$$

Sujeito a:

$$\sum_{s \in S} \sum_{t \in T_c} x_{cst1} = 1 \quad \forall c \in C_1^p \quad (2)$$

$$\sum_{s \in S} \sum_{t \in T_c} \sum_{d \in D} x_{cstd} = 1 \quad \forall c \in C_2^p \quad (3)$$

$$\sum_{s \in S} \sum_{t \in T_c} \sum_{d \in D} x_{cstd} \leq 1 \quad \forall c \in (C_3^p \cup C_4^p) \quad (4)$$

$$\sum_{c \in C} \sum_{[t'=t-t_c-1 | t' \in T_c]}^t x_{cst'1} \leq 1 \quad \forall s \in S, t \in T \quad (5)$$

$$\sum_{c \in C^n} \sum_{[t'=t-t_c-1 | t' \in T_c]}^t x_{cst'd} \leq 1 \quad \forall s \in S, t \in T, d \in D \setminus \{1\} \quad (6)$$

$$\sum_{s \in S} x_{cst1} \leq i_{ct1} \quad \forall c \in C_1^p, t \in T_c \quad (7)$$

$$\sum_{s \in S} x_{cstd} \leq i_{ctd} \quad \forall c \in C^n, t \in T_c, d \in D \quad (8)$$

$$\sum_{j \in E} y_{jsd} \leq 1 \quad \forall s \in S, d \in D \quad (9)$$

$$\sum_{c \in C_j^e} \sum_{t \in T_c} x_{cst1} \leq y_{js1} |T| \quad \forall j \in E, s \in S \quad (10)$$

$$\sum_{c \in (C_j^e \cap C^n)} \sum_{t \in T_c} x_{cstd} \leq y_{jsd} |T| \quad \forall j \in E, s \in S, d \in D \setminus \{1\} \quad (11)$$

$$\sum_{[c \in C: h_c = h]} \sum_{s \in S} \sum_{[t' = t - t_c + 1 | t' \in T_c]} x_{cst'1} \leq 1 \quad \forall h \in H, t \in T \quad (12)$$

$$\sum_{[c \in C^n: h_c = h]} \sum_{s \in S} \sum_{[t' = t - t_c + 1 | t' \in T_c]} x_{cst'd} \leq 1 \quad \forall h \in H, d \in D \setminus \{1\}, t \in T \quad (13)$$

$$\sum_{[c \in C: h_c = h]} \sum_{s \in S} \sum_{t \in T_c} t_c x_{cst1} \leq T_{h1}^{MAXD} \quad \forall h \in H \quad (14)$$

$$\sum_{[c \in C^n: h_c = h]} \sum_{s \in S} \sum_{t \in T_c} t_c x_{cstd} \leq T_{hd}^{MAXD} \quad \forall d \in D \setminus \{1\}, h \in H \quad (15)$$

$$\sum_{[c \in C_1^p: h_c = h]} \sum_{s \in S} \sum_{t \in T_c} t_c x_{cst1} + \sum_{[c \in C^n: h_c = h]} \sum_{d \in D} \sum_{s \in S} \sum_{t \in T_c} t_c x_{cstd} \leq T_h^{MAXS} \quad \forall h \in H \quad (16)$$

$$x_{cst1} \in \{0,1\} \quad \forall c \in C_1^p, s \in S, t \in T_c \quad (17)$$

$$x_{cstd} \in \{0,1\} \quad \forall c \in C^n, s \in S, t \in T_c, d \in D \quad (18)$$

$$y_{jsd} \in \{0,1\} \quad \forall j \in E, s \in S, d \in D \quad (19)$$

As expressões matemáticas do modelo apresentado serão descritas a seguir com o intuito de apresentar o papel que elas desempenham. A expressão (1) representa a função objetivo do problema de sequenciamento de cirurgias eletivas modelado, a qual é responsável por calcular o tempo total de execuções das cirurgias durante a programação realizada. A primeira parcela da expressão (1) calcula o tempo total gasto na realização das cirurgias com prioridade de urgência identificada e a segunda parcela calcula o tempo total para as demais cirurgias. Como já mencionado, o foco em questão é minimizar o tempo total de execução das cirurgias, conhecido como *makespan*.

A expressão (2) garante que as cirurgias eletivas de maior prioridade de urgência identificadas sejam marcadas no primeiro dia da agenda de planejamento cirúrgico, uma vez que estas devem ser executadas antes das demais. A expressão (3) faz com que as cirurgias eletivas de alta prioridade ($P = 2$) sejam marcadas durante a agenda de planejamentos prioritários, ou seja, dentro de um período previamente determinado que considera a prioridade do procedimento cirúrgico. A expressão (4) determina que as demais cirurgias de prioridade média e normal sejam ou não marcadas durante o período da agenda de planejamentos prioritários.

É importante garantir que não haja sobreposição das cirurgias dentro da mesma sala de operação, isto é, que decorra, a cada momento, apenas uma cirurgia por bloco cirúrgico. Isto é garantido por meio da expressão (5) que se refere ao primeiro dia do planejamento e pela expressão (6) que se refere aos demais dias da agenda. Estas restrições também reservam dois períodos em vazio, após a realização de cada cirurgia, para o procedimento de higienização da sala de cirurgia.

As restrições (7) e (8) consideram a possibilidade de não haver disponibilidade por parte do cirurgião e do paciente para o agendamento da cirurgia, sendo a (7) relativa às cirurgias de urgência identificadas e a (8) para as demais.

A expressão (9) trata de uma restrição específica para o hospital em questão, onde não é permitida a troca de especialidade cirúrgica em uma sala de operação no mesmo dia. Sendo assim, para um determinado dia, tal sala cirúrgica irá receber apenas uma especialidade de cirurgia. Desta forma, a marcação das cirurgias numa determinada sala deve respeitar a especialidade cirúrgica determinada a ela. As restrições (10) e (11) fazem uma relação entre as variáveis x_{cstd} e y_{jsd} , ou seja, garantem a marcação de cirurgias de acordo com a especialidade que as salas de operação estão determinadas a realizar. A expressão (10) se refere ao primeiro dia da agenda e a (11) aos demais dias da agenda.

Deve-se garantir também que não haja sobreposição dos cirurgiões entre as várias salas de cirurgias disponíveis, ou seja, cada cirurgião só pode realizar uma cirurgia por vez. As expressões (12) e (13) tratam de representar tal restrição, sendo aquela para o primeiro dia da agenda e esta última para os demais. É importante ressaltar que restrições como essa podem ser criadas para outros tipos de recursos (anestesiastas, enfermeiros, equipamentos, entre outros), os quais se deseja considerar para o planejamento da agenda de cirurgias eletivas.

O limite máximo de tempo em operação disponível para cada cirurgião é considerado nas restrições (14), (15) e (16). A expressão (14) limita o tempo gasto diariamente em

procedimento cirúrgico por cada cirurgião no primeiro dia da agenda de planejamento e, a expressão (15), limita o tempo gasto diariamente em procedimento cirúrgico por cada cirurgião nos demais dias da agenda de planejamento. Já a expressão (16) limita o tempo semanal utilizado em procedimento cirúrgico por cada cirurgião.

Finalmente, as restrições (17), (18) e (19) representam o domínio das variáveis consideradas no modelo em questão.

4 METODOLOGIA

Nesta Seção é apresentada a metodologia utilizada para resolver o problema de Sequenciamento de Cirurgias Eletivas. Como mencionado anteriormente, o problema combinatório de agendamento de cirurgias, abordado nesta dissertação, será resolvido por meio de métodos heurísticos implementados com base em programação de computadores.

O algoritmo proposto, nomeado de ESSILS (iniciais em inglês para *Elective Surgery Scheduling Iterated Local Search*), utiliza o método *Greedy Randomized Adaptive Search Procedure* (GRASP) para a geração da solução inicial e a metaheurística *Iterated Local Search* (ILS) para a fase de refinamento. O ILS, por sua vez, utiliza dois métodos de busca local, *Variable Neighborhood Descent* (VND) e Busca Tabu. Todos estes procedimentos, assim como as variáveis e parâmetros adotados, serão descritos detalhadamente nesta Seção.

Ainda nesta Seção, serão detalhados a forma com que a solução do problema é representada, a estrutura de vizinhanças utilizada para explorar o espaço de soluções, assim como a função de avaliação necessária para conduzir a busca do algoritmo proposto e os mecanismos de perturbação da busca local.

4.1 CONSIDERAÇÕES INICIAIS PARA O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS ELETIVAS

A estrutura dos dados de entrada adotada neste trabalho se baseia na proposta de Pham e Klinkert (2008). Essa estrutura contempla todos os dados necessários para resolver o problema de sequenciamento de cirurgias eletivas e está disponível nos problemas-teste encontrados na literatura, assim como no mapa de processos dos problemas reais. Os elementos que compõem os dados de entrada são justamente os recursos hospitalares disponíveis para a realização das cirurgias, sendo eles humanos e materiais, os tempos de utilização e disponibilidade de cada um destes recursos, além das janelas de tempo disponíveis para alocação das cirurgias. Desta forma, portanto, é possível alocar as cirurgias a serem realizadas de acordo com a disponibilidade dos recursos existentes.

Uma matriz de recursos com suas respectivas compatibilidades e tempos médios de utilização é então criada para servir como dado de entrada para o algoritmo proposto. A Tabela 1, portanto, ilustra a matriz de compatibilidades mencionada anteriormente. Como pode ser observado, a primeira coluna é representada pelos recursos disponíveis, tanto humanos como materiais. A partir da segunda coluna até a nona, pode-se observar os oito

tipos de cirurgias realizadas, enumeradas de 1 a 8 no topo da Tabela 1. Nas células, onde se lê um número diferente de zero, significa que aquele determinado recurso é capaz de realizar aquela determinada cirurgia durante um tempo médio previamente estabelecido. Por consequência, onde se lê zero, significa que aquele recurso não é capaz de realizar aquela determinada cirurgia. Portanto, a primeira célula com o número 78 informa que o Bloco Cirúrgico 1 tem capacidade para realizar a Cirurgia do tipo 1 no tempo médio de 78 unidades de tempo. Já a cirurgia 2 não pode ser realizada neste mesmo bloco por algum motivo específico, já que a célula possui o número zero. Vale salientar que os números apresentados na Tabela 1 são exemplos fictícios e, portanto, não são representativos de um problema real.

Tabela 1: Compatibilidade dos recursos de acordo com as cirurgias realizadas.

Recurso / Tipo Cirurgia	1	2	3	4	5	6	7	8
Bloco Cirúrgico 1	78	0	89	245	218	163	197	97
Bloco Cirúrgico 2	0	125	89	0	218	163	0	97
Bloco Cirúrgico 3	78	125	0	245	0	0	197	0
Médico Cirurgião 1	78	125	89	245	218	163	197	97
Médico Cirurgião 2	78	0	89	0	218	163	197	97
Médico Cirurgião 3	78	125	0	0	218	0	197	97
Médico Cirurgião 4	0	125	89	245	218	0	0	97
Anestesista 1	12	17	10	45	55	34	31	33
Anestesista 2	12	17	0	0	0	0	31	33
Enfermeiro 1	78	0	89	245	218	163	197	97
Enfermeiro 2	78	125	89	245	218	0	0	97
Enfermeiro 3	0	125	0	0	0	0	0	0
UTI 1	0	57	76	76	66	0	89	66
UTI 2	33	57	76	76	66	79	89	54
Equipamento 1	78	0	89	245	218	163	197	97
Equipamento 2	0	125	0	0	218	0	197	0
Equipamento 3	0	125	89	0	0	163	0	0

Fonte: Autor.

Desta forma, o algoritmo é capaz de se orientar e alocar cada um dos recursos necessários para cada cirurgia, de acordo com sua disponibilidade e compatibilidade. É importante considerar também que cada cirurgia demanda um determinado tipo de recurso. Esta informação também é disponibilizada como dado de entrada do algoritmo e é tratada pelas restrições do problema. Isto significa que, se uma determinada cirurgia necessita de um bloco cirúrgico, de um médico cirurgião, de um anestesista e de um enfermeiro, por exemplo, as restrições do problema (Seção 3.2) irão garantir a alocação destes recursos para viabilizar a realização da operação, levando em consideração a não utilização destes recursos em mais de uma cirurgia ao mesmo tempo, conforme explicado anteriormente.

Para o trabalho em questão, considera-se a elaboração de um planejamento semanal gerado como resultado do algoritmo proposto. Este agendamento, portanto, é realizado de

segunda-feira à sexta-feira onde, para cada dia, a disponibilidade dos recursos é de dez horas. É com base em todas essas informações de entrada que o algoritmo aloca os recursos de acordo com a demanda de cirurgias existente, tendo como objetivo a minimização do tempo total de duração das operações. Essa alocação é feita passo a passo, onde para cada cirurgia escolhida pelo algoritmo, seguindo regras previamente estabelecidas, cada um dos recursos necessários e disponíveis são direcionados para a execução de cada operação de acordo com o tempo médio determinado.

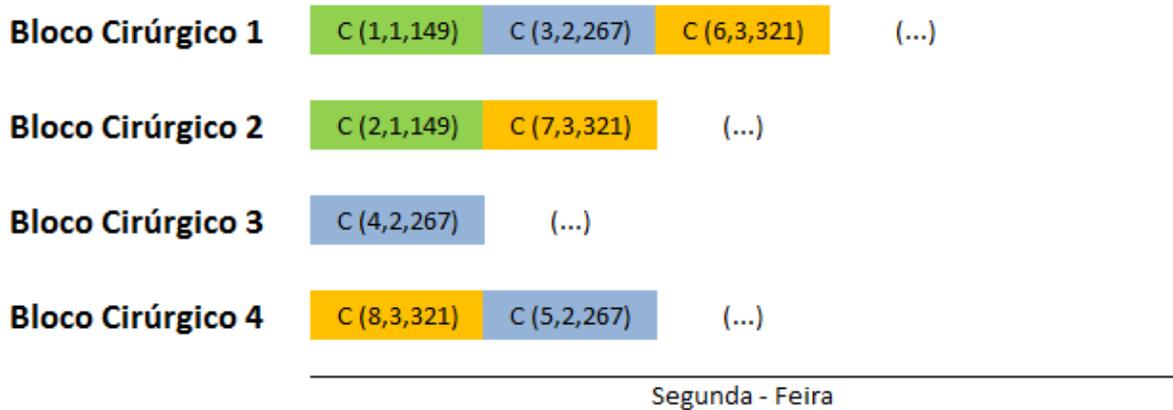
4.2 REPRESENTAÇÃO DA SOLUÇÃO

Uma solução para o Problema de Sequenciamento de Cirurgias Eletivas é representada por meio de uma matriz tridimensional, onde temos os blocos cirúrgicos representados em um plano, os demais recursos em um segundo plano e, por último, as cirurgias eletivas reproduzidas em um terceiro plano. A adoção desta representação de solução se deve ao fato da possibilidade de utilização de recursos compartilhados para a execução das cirurgias e também pelos diferentes tempos de ocupação destes mesmos recursos em uma determinada cirurgia (SPERANDIO, 2015). Vale salientar, portanto, que esta maneira de representar a solução permite o compartilhamento de recursos, caso este seja possível. Em caso de problemas reais, onde determinados recursos não podem ser compartilhados, basta restringi-los no algoritmo.

As Figuras 9 e 10 são utilizadas para ilustrar a forma com que a solução é representada. Como pode ser observado na Figura 9, os blocos cirúrgicos enumerados de 1 a 4 estão representados pelos vértices da matriz e as cirurgias eletivas estão alocadas de forma a preencher as arestas da matriz. Cada célula $C(x,y,z)$ desta matriz contém informações fundamentais para a alocação da cirurgia e também para a avaliação da solução (Seção 4.4), onde o primeiro fator (x) é o identificador da cirurgia, o segundo fator (y) é o tipo de cirurgia e o terceiro fator (z) é o tempo médio de duração daquela determinada cirurgia. Logo, a cirurgia $C(1,1,149)$ possui identificador 1, é do tipo 1 e tem tempo médio de duração 149 unidades de tempo. É importante salientar que cada cirurgia possui um identificador, mesmo que esta seja do mesmo tipo. Essa característica pode ser percebida na cirurgia $C(6,3,321)$ e $C(7,3,321)$, por exemplo, onde aquela (realizada no bloco cirúrgico 1) tem identificador 6 e esta (realizada no bloco cirúrgico 2) tem identificador 7, mas são do mesmo tipo 3. Desta forma, pode-se perceber que cada tipo de cirurgia é representado por uma cor diferente, como por exemplo a cirurgia do tipo 1 em verde, a do tipo 2 em azul e a do tipo 3 em amarelo.

Nota-se também que a representação da solução em questão é parcial, uma vez que apenas parte do planejamento de segunda-feira é apresentado.

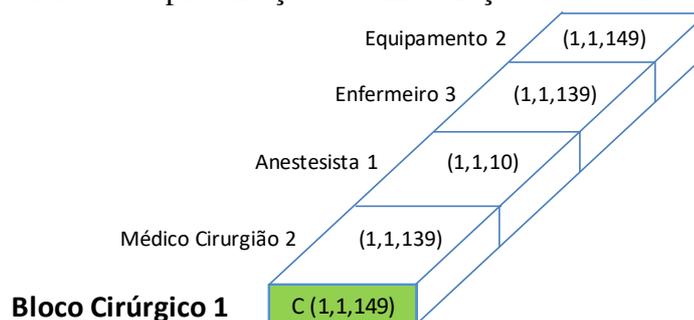
Figura 9: Parte da representação de uma solução com blocos cirúrgicos e cirurgias.



Fonte: Autor.

A Figura 10 é uma complementação da Figura 9, uma vez que representa a utilização dos demais recursos necessários à realização das cirurgias eletivas. Como é possível observar, a cirurgia C(1,1,149), a primeira a ser realizada pelo bloco cirúrgico 1 de acordo com a Figura 10, necessita do médico cirurgião 2 por 139 unidades de tempo, do anestesista 1 por 10 unidades de tempo, do enfermeiro 3 por 139 unidades de tempo e do equipamento 2 em sua total duração, ou seja, 149 unidades de tempo. É desta forma, portanto, que a solução para o agendamento de cirurgias será apresentada, exibindo o recurso que será alocado para cada cirurgia em um determinado período de tempo, respeitando a utilização dos recursos individualmente para cada operação até que esta seja encerrada. É possível perceber que os tipos de recursos e cirurgias são facilmente alterados, uma vez que são representados de forma matricial e são utilizados como dados de entrada. Sendo assim, cada problema-teste tem seus dados de entrada configurados de acordo com a necessidade do problema real, que pode variar de um hospital para outro.

Figura 10: Parte da representação de uma solução com demais recursos e cirurgias.



Fonte: Autor.

4.3 ESTRUTURAS DE VIZINHANÇAS

O espaço de busca, ou seja, aquele que consiste no conjunto de soluções possíveis para o Problema de Sequenciamento de Cirurgias Eletivas, será explorado por meio de sete tipos diferentes de movimentos. Estes movimentos são responsáveis por obter novas soluções (vizinhos) para o problema em questão. Para melhor compreensão, é importante salientar que vizinho é considerado como uma solução gerada a partir de um movimento aplicado na solução corrente, modificando-a (ARMENTANO; BRANCHINI, 2013).

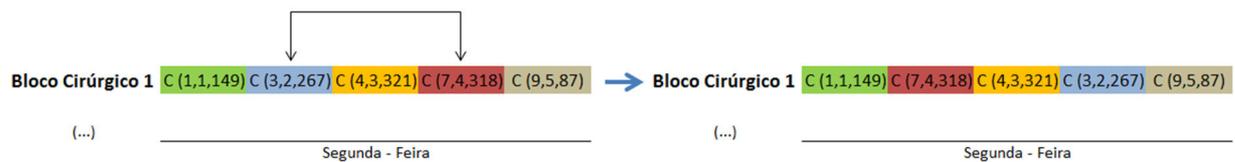
É importante destacar que os movimentos adotados pelo algoritmo proposto são baseados na troca de ordem e realocação de posição das cirurgias, sendo todos eles realizados a partir do bloco cirúrgico. Isto significa que os demais recursos alocados para uma determinada cirurgia, de acordo com a disponibilidade e compatibilidade, acompanharão a movimentação da cirurgia sempre que possível. Em caso da não possibilidade de utilização dos demais recursos devido a problemas de sobreposição (onde o mesmo recurso passa a ser utilizado por mais de uma cirurgia) oriundos do processo de movimentação, o algoritmo pode tomar duas decisões. Uma delas é não aceitar o movimento e a outra é aceitá-lo, porém realizando uma nova realocação dos demais recursos de acordo com a nova disponibilidade e compatibilidade (se determinado recurso tem capacidade para realizar a cirurgia que está sendo alocada a ele) dos mesmos após a movimentação realizada. A proposta deste trabalho se baseia no artigo de Bam et al. (2015), o qual propõe uma função para auxiliar o algoritmo na tomada de decisão entre estas duas possibilidades. Esta função escolhe pela realização de uma nova alocação dos demais recursos com probabilidade de 40% em relação a outra escolha. Isto significa que a não aceitação do movimento quando ocorre sobreposição dos demais recursos tem maior probabilidade de acontecer, uma vez que uma nova realocação torna o problema ainda mais complexo e com alto custo de avaliação, ou seja, o algoritmo leva mais tempo para avaliar a nova solução, já que esta foi fortemente modificada em relação à anterior. Porém, segundo Bam et al. (2015) é importante aceitar parcialmente a opção de realocação dos demais recursos, pois é uma maneira de explorar o espaço de soluções viáveis caminhando-se também no espaço de soluções inviáveis, o que permite encontrar uma solução ótima mesmo se o conjunto de soluções viáveis for desconexo. Contudo, para que o algoritmo não apresente um longo tempo de execução, Bam et al. (2015) adotaram essa função para aceitar, com 40% de chance, a realocação dos demais recursos.

Em seguida são apresentados todos os movimentos de troca e realocação de cirurgias a partir dos blocos cirúrgicos adotados neste trabalho. As características e resultados dos

movimentos também são descritos para melhor compreensão. Os movimentos são ilustrados com o dia de segunda-feira, mas são realizados linearmente até a sexta-feira, ou seja, uma cirurgia realizada na quinta-feira pode ser trocada com uma executada na terça-feira, por exemplo:

- a) Movimento Troca Ordem Cirurgia: este movimento tem como objetivo trocar a posição de duas cirurgias entre si de um mesmo bloco cirúrgico. Como pode ser observado na Figura 11, a cirurgia C(3,2,267) foi trocada com a C(7,4,318), onde aquela que era a segunda a ser realizada passa a ser a quarta. O resultado deste movimento pode ser observado no diagrama localizado no canto direito da Figura 11. É importante ressaltar que este movimento é aplicado em todos os blocos cirúrgicos disponíveis;

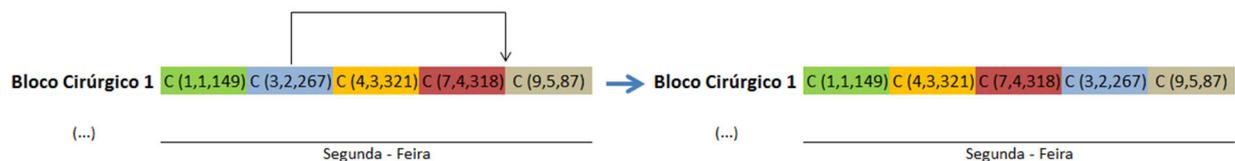
Figura 11: Representação do Movimento Troca Ordem Cirurgia.



Fonte: Autor.

- b) Movimento Realoca Cirurgia: este movimento consiste em escolher uma determinada cirurgia e realocá-la para uma outra posição de realização no mesmo bloco cirúrgico. Este movimento, então, muda a ordem de realização da cirurgia escolhida. A nova posição deve ser diferente da original, garantindo assim a movimentação. Como se pode notar na Figura 12 a cirurgia C(3,2,267) foi realocada para o final da cirurgia C(7,4,318), onde aquela passa a ser realizada ao término desta. No canto direito observa-se o resultado do movimento realizado;

Figura 12: Representação do Movimento Realoca Cirurgia.



Fonte: Autor.

- c) Movimento Troca Ordem Múltiplas Cirurgias: este movimento consiste em trocar a posição de um conjunto de duas ou mais cirurgias adjacentes por outro conjunto de duas ou mais cirurgias adjacentes entre si. A quantidade k de cirurgias que

formam este conjunto é determinada aleatoriamente de acordo com o intervalo $[2,4]$, ou seja, o número de elementos deste conjunto varia de duas a quatro cirurgias. Para um valor de $k = 2$, a Figura 13 ilustra o conjunto de cirurgias adjacentes $C(1,1,149)$ e $C(3,2,267)$ sendo trocado com outro conjunto de cirurgias também adjacentes $C(7,4,318)$ e $C(9,5,87)$. O resultado deste movimento pode ser observado no canto direito da Figura 13, onde a cirurgia $C(3,2,267)$ passa a ser realizada na quinta posição e não mais na segunda, por exemplo;

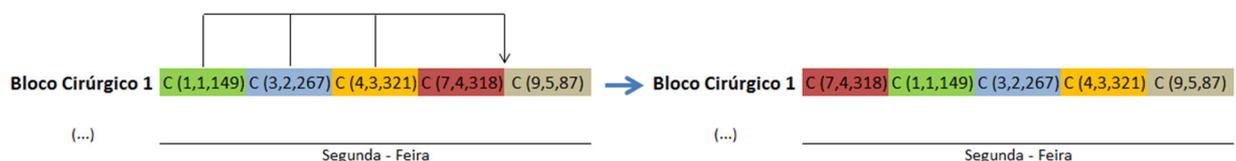
Figura 13: Representação do Movimento Troca Ordem Múltiplas Cirurgias.



Fonte: Autor.

- d) Movimento Realoca Múltiplas Cirurgias: este movimento consiste em escolher um determinado conjunto de cirurgias e realocá-lo para uma outra posição de realização no mesmo bloco cirúrgico. A quantidade k de cirurgias que formam este conjunto é determinada randomicamente no intervalo $[2,4]$. Este movimento, então, muda a ordem de realização deste conjunto de cirurgias. A nova posição deve ser diferente da original, garantindo assim a movimentação. A Figura 14 mostra a realização deste movimento para um conjunto de três cirurgias ($k = 3$). Por esta figura nota-se que as cirurgias $C(1,1,149)$, $C(3,2,267)$ e $C(4,3,321)$ foram realocadas para uma posição logo após o término da cirurgia $C(7,4,318)$;

Figura 14: Representação do Movimento Realoca Múltiplas Cirurgias.



Fonte: Autor.

- e) Movimento Troca Cirurgia Entre Blocos: este movimento tem como objetivo trocar duas cirurgias de diferentes blocos cirúrgicos entre si. Primeiramente, escolhe-se dois blocos cirúrgicos que terão uma de suas cirurgias trocadas entre si em cada um deles. Em seguida, selecionam-se as cirurgias de cada um dos blocos cirúrgicos escolhidos que serão trocadas. A Figura 15 ilustra a realização deste

movimento. É possível notar que a cirurgia $C(7,4,318)$, antes realizada no bloco cirúrgico 1 na quarta posição, passa a ser realizada no bloco cirúrgico 2 na segunda posição. O mesmo acontece com a cirurgia $C(5,3,321)$, antes realizada pelo bloco cirúrgico 2 na segunda posição, passa a ser realizada no bloco cirúrgico 1 na quarta posição. É importante ressaltar que o movimento só é realizado se os blocos cirúrgicos que estão recebendo a nova cirurgia têm compatibilidade para sua execução, conforme exemplificado na Tabela 1. A compatibilidade do bloco cirúrgico, explicada na Seção 4.1, está associada à capacidade deste recurso de realizar determinado tipo de cirurgia;

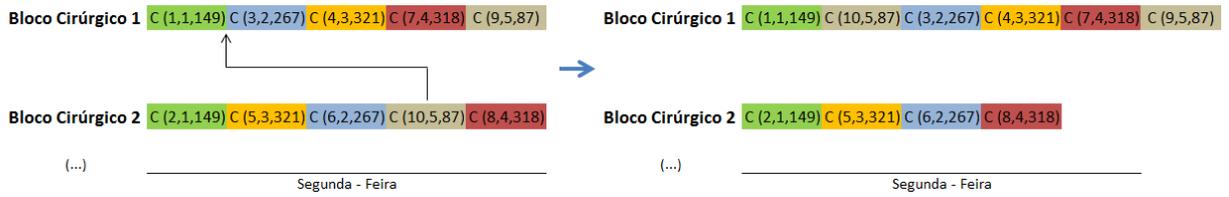
Figura 15: Representação do Movimento Troca Cirurgia Entre Blocos.



Fonte: Autor.

- f) Movimento Realoca Cirurgia Entre Blocos: este movimento consiste em escolher um bloco cirúrgico e uma determinada cirurgia realizada neste mesmo. Em seguida um novo bloco cirúrgico é escolhido juntamente com uma determinada posição, a qual irá receber a cirurgia escolhida inicialmente oriunda do outro bloco. A Figura 16 ilustra um exemplo deste movimento, onde a cirurgia $C(10,5,87)$, antes alocada no bloco cirúrgico 2 na quarta posição, passa a ser realizada no bloco cirúrgico 1 na segunda posição. Vale ressaltar que este movimento entre blocos cirúrgicos é avaliado pelas restrições do problema, ou seja, só é realizado se existe compatibilidade do recurso em questão, conforme exemplificado na Tabela 1. A compatibilidade do bloco cirúrgico, explicada na Seção 4.1, está associada à capacidade deste recurso de realizar determinado tipo de cirurgia;

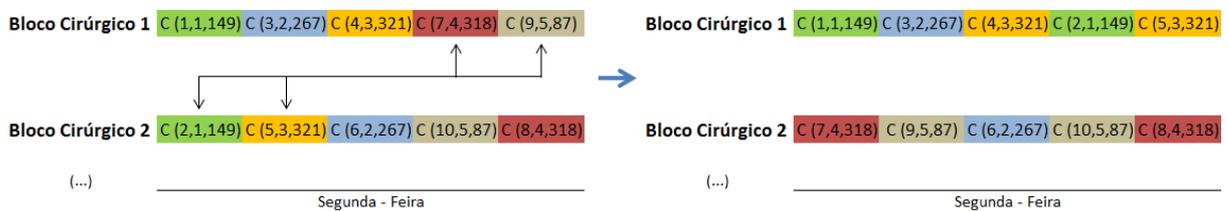
Figura 16: Representação do Movimento Realoca Cirurgia Entre Blocos.



Fonte: Autor.

- g) Movimento Troca Múltiplas Cirurgias Entre Blocos: este movimento tem como objetivo trocar um conjunto de duas ou mais cirurgias adjacentes de um determinado bloco cirúrgico com outras duas ou mais cirurgias adjacentes de um outro bloco cirúrgico. A quantidade k de cirurgias envolvidas neste movimento é definida de maneira aleatória de acordo com o intervalo $[2,4]$. A Figura 17 ilustra um exemplo deste movimento envolvendo um conjunto de duas ($k = 2$) cirurgias adjacentes de dois blocos cirúrgicos, 1 e 2. Como pode ser observado, as cirurgias $C(7,4,318)$ e $C(9,5,87)$ pertencentes ao bloco cirúrgico 1 são trocadas com outras duas cirurgias $C(2,1,149)$ e $C(5,3,321)$ pertencentes ao bloco cirúrgico 2. O resultado deste movimento pode ser observado no canto direito da Figura 17. É possível observar também que a ordem das cirurgias adjacentes é permanecida de acordo com a original. Por ser um movimento em que a cirurgia é trocada de um bloco cirúrgico para o outro, existe a necessidade de certificar se o bloco cirúrgico, o qual está recebendo a cirurgia em questão, tem capacidade de recebê-la. Este assunto fora explicado na Seção 4.1, a qual também trata da compatibilidade dos recursos.

Figura 17: Representação do Movimento Troca Múltiplas Cirurgias Entre Blocos.



Fonte: Autor.

4.4 FUNÇÃO DE AVALIAÇÃO

Para uma determinada solução x , pertencente ao conjunto de soluções possíveis X , tem-se a seguinte função de avaliação utilizada para quantificar (avaliar) o custo de cada solução produzida pelo algoritmo:

$$\emptyset(s) = \sum_{c \in C} \sum_{d \in D} \sum_{s \in S} t_c x_{csd} \quad (1)$$

A expressão (1) calcula o tempo total (*makespan*) utilizado para a realização de todas as cirurgias eletivas agendadas na programação de operações. A variável t_c armazena o tempo previsto da cirurgia c e a variável binária x_{csd} indica que a cirurgia c é agendada na sala cirúrgica s no dia d quando recebe o valor 1 e, quando recebe 0, indica o contrário. Os conjuntos C , D e S representam, respectivamente, todas as cirurgias a serem programadas na agenda de operações, todos os dias livres para a marcação das cirurgias em questão e todas as salas cirúrgicas disponíveis para a realização das cirurgias. Sendo assim, a partir de uma determinada cirurgia, percorre-se em todos os dias da agenda de programação todas as salas cirúrgicas disponíveis com o intuito de identificar onde a mesma fora alocada. Uma vez identificado, a variável x_{csd} recebe o valor 1 que é multiplicado pelo tempo de duração da cirurgia em questão. O processo se repete até que todas as cirurgias a serem realizadas sejam alocadas e seus respectivos tempos armazenados. No final deste processo tem-se o tempo total da programação de cirurgias eletivas, o qual se objetiva minimizar.

4.5 ALGORITMO PROPOSTO

O algoritmo proposto nesta dissertação, nomeado ESSILS (das iniciais em inglês de *Elective Surgery Scheduling Iterated Local Search*), tem como proposta a utilização do método GRASP (*Greedy Randomized Adaptive Search Procedure*) para gerar a solução inicial e, como método de refinamento, a metaheurística ILS (*Iterated Local Search*), a qual por sua vez utiliza as metaheurísticas: VND (*Variable Neighborhood Descent*) e Busca Tabu como métodos de busca local.

Na Subseção 4.5.1 o método GRASP, utilizado para gerar a solução inicial, é explicado com detalhes. A Subseção 4.5.2 é destinada ao algoritmo de refinamento proposto,

o qual utiliza a metaheurística ILS. Os métodos de busca local, VND e Busca Tabu, são apresentados separadamente nas subseções 4.5.3 e 4.5.4, respectivamente. Em seguida, na Subseção 4.5.5, são apresentados os mecanismos de perturbação utilizados para modificar a solução corrente em busca de novas ainda melhores, localizadas em outras regiões do espaço de busca.

4.5.1 Geração da Solução Inicial – GRASP

A solução inicial do algoritmo ESSILS proposto é gerada por meio da metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*), proposta por Feo e Resende (1995). Este procedimento iterativo, caracterizado por possuir diversos ciclos de reinício, é estruturado em duas fases. A primeira delas é a fase construtiva, responsável pela geração de soluções factíveis para o problema em questão. A segunda fase é conhecida como a de busca local ou refinamento, sendo esta responsável pela busca do ótimo local a partir das soluções iniciais geradas durante a fase de construção.

Na fase de construção, a função denominada *Constroi_Solução_GRASP* é aplicada, onde é gerada uma Lista de Candidatos (LC). Os possíveis candidatos para compor a LC são justamente as cirurgias eletivas a serem realizadas e são alocadas na solução inicial de acordo com a seguinte expressão: $c \in LC : t(c) \geq c_{min} + \gamma (c_{max} - c_{min})$. Para tal, seja $t(c)$ o tempo médio de duração da cirurgia c , c_{min} a cirurgia com o menor tempo de duração e c_{max} a cirurgia com o maior tempo de duração. Considera-se também o fator γ como sendo um número real que varia entre 0 e 1. A expressão, portanto, é interpretada como um critério para a escolha dos candidatos da lista, onde o tempo necessário para a realização de uma cirurgia seja maior ou igual à soma da cirurgia de menor duração com o produto de γ pela diferença da cirurgia de maior tempo pela de menor tempo. É importante salientar que para valores de $\gamma = 0$, obtém-se uma solução puramente gulosa, ou seja, aquela em que se espera um ótimo local ou global por meio de um critério determinístico de busca. Já, para valores de $\gamma = 1$, obtém-se uma solução exclusivamente aleatória. Sendo assim, os candidatos que atendem ao critério apresentado farão parte da Lista Restrita de Candidatos (LRC). Isto significa que, para a proposta deste algoritmo, serão consideradas as cirurgias de maior tempo de duração para a fase de construção. Finalizando, este algoritmo irá elencar aleatoriamente um dos candidatos da LRC, alocando-o na solução inicial e retirando-o da LC. Esta etapa é repetida até que todas as cirurgias eletivas a serem realizadas sejam alocadas.

O método de busca local utilizado na segunda fase foi o VND (*Variable Neighborhood Descent*). Este procedimento também será utilizado na fase de refinamento do algoritmo proposto e será descrito em detalhes na Subseção 4.5.3. Em resumo, o VND explora o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança, aceitando somente soluções de melhora em relação à solução corrente. Havendo melhora da solução encontrada, a solução corrente é então atualizada.

Algoritmo 5: Pseudocódigo do método de geração da solução inicial - GRASP.

```

Procedimento Geração da Solução Inicial - GRASP
1    $f^* \leftarrow \infty$ 
2   enquanto (critério de parada não satisfeito) faça
3        $s_0 \leftarrow \text{Constroi\_Solução\_GRASP}(T(c), \gamma)$ ;
4        $s \leftarrow \text{VND}(s_0)$ ;
5       se ( $f(s) < f^*$ ) então
6            $s^* \leftarrow s$ ;
7            $f^* \leftarrow f(s)$ ;
8       fim se;
9   fim enquanto;
10   $s \leftarrow s^*$ ;
11  Retorne  $s$ ;
fim

```

Fonte: Souza (2009).

O Algoritmo 5 exemplifica o procedimento GRASP utilizado para a geração da solução inicial para o problema proposto. Na linha 3, uma solução é gerada por meio de um critério parcialmente guloso implementado na função *Constroi_Solução_GRASP*, a qual é responsável por elencar, aleatoriamente, a próxima cirurgia a entrar no sequenciamento, dentre uma lista de melhores candidatos. Esta lista com os melhores candidatos é elaborada por meio da regra de cirurgias com os maiores tempos de execução, juntamente com o fator γ que define o tamanho da lista. Como pode ser observado na linha 4, o método VND é aplicado com a intenção de se realizar um refinamento prévio da solução encontrada. Uma vez encontrada uma solução melhor que a corrente, esta é então atualizada (linha, 5, 6, e 7). O método é aplicado até que o critério de parada seja atendido. No caso deste trabalho, o procedimento é repetido durante um período de tempo *time*, o qual é definido com base no tempo em que o algoritmo não melhora mais o resultado encontrado para cada problema-teste.

4.5.2 Algoritmo Proposto ESSILS - Iterated Local Search (ILS)

Esta Subseção é destinada a apresentar o Algoritmo de Refinamento ESSILS proposto por esta dissertação, o qual utiliza a metaheurística *Iterated Local Search* combinada com os métodos *Variable Neighborhood Descent* e Busca Tabu, os quais atuam como métodos de busca local do ILS, sendo apresentados nas subseções 4.5.3 e 4.5.4, respectivamente.

De acordo com Lourenço, Martin e Stützle (2003), a metaheurística *Iterated Local Search* é um procedimento de busca iterativa utilizado na diversificação das soluções, uma vez que atua com o objetivo de desviar e encontrar diferentes ótimos locais na procura de soluções ainda melhores. Essa busca iterativa é caracterizada pelo uso constante de modificações na solução corrente. Esta metodologia é também conhecida como perturbação da solução. O método ILS, mais especificamente, consiste na aplicação das quatro etapas apresentadas a seguir: geração da solução inicial, busca local, perturbação da solução corrente e, por último, critério de aceitação.

Em síntese, de acordo com Subramanian et al. (2013), a metaheurística ILS fundamenta-se na ideia de explorar o espaço de soluções por meio de perturbações em ótimos locais gerados durante o processo de busca. Os mecanismos de perturbação, apresentados na Subseção 4.5.5, consistem em um conjunto de modificações feitas no ótimo local com a intenção de gerar soluções progressivamente mais distantes desse ótimo local e, possivelmente, melhores. No caso da busca local não for bem-sucedida, ou seja, não estiver gerando ótimos locais melhores, a intensidade da perturbação é aumentada. Contudo, segundo Souza (2009), a intensidade da perturbação deve ser forte o suficiente para permitir escapar do ótimo local corrente e permitir explorar diferentes regiões, ao passo que, ela precisa também ser fraca o suficiente para guardar características do ótimo local corrente.

O pseudocódigo do método ILS é apresentado no Algoritmo 6. Como pode ser observado na linha 2, uma solução inicial é gerada por meio do método GRASP, apresentado na Subseção 4.5.1. Na linha 1, o parâmetro γ , o qual determina o tamanho da lista de candidatos, influenciando na qualidade e diversidade da solução gerada na fase de construção do GRASP, é gerado por meio de uma função que obtém um número real aleatório entre 0 e 0,6. O parâmetro *Time* (linha 2) também é um dado de entrada para o GRASP, o qual é responsável por executar este procedimento durante um determinado período de tempo. A partir da linha 5 a solução gerada é então refinada pela metaheurística ILS. Para escapar das armadilhas de ótimos locais e se dirigir para outras regiões do espaço de busca, mecanismos de perturbação são aplicados na solução s gerada (linha 6). Estes mecanismos são

apresentados na Subseção 4.5.5 e o nível de perturbação β vai sendo aumentado a cada iteração sem melhora do algoritmo. O valor de β é caracterizado pelo número de vezes que um dos movimentos descritos na Seção 4.3 é aplicado e é atualizado de acordo com o número de iterações, com melhora da solução (retorna ao valor 1) ou sem melhora (é incrementado). Para aperfeiçoar a solução, os métodos de busca local VND (Subseção 4.5.3) e Busca Tabu (Subseção 4.5.4) são aplicados nas linhas 8 e 10, respectivamente. A variável *iterMaxSemMelhoraVND* controla o número de iterações do método VND, ou seja, a Busca Tabu é acionada somente após certo número de iterações sem melhora do procedimento *Variable Neighborhood Descent*. Após cada aplicação de um dos métodos de busca local, a solução é então avaliada (linha 12). Se a solução obtida for melhor que a corrente, aquela passa a ser a solução em vigor. O método ILS é interrompido quando o número máximo de iterações sem melhora na solução corrente (*maxIter*) for atingido.

Algoritmo 6: Pseudocódigo do método de refinamento ILS.

```

Procedimento ESSILS - Iterated Local Search
1    $\gamma \leftarrow$  número real aleatório no intervalo  $[0,0 ; 0,6]$ ;
2    $s \leftarrow$  GRASP ( $\gamma$ , Time);
3    $iter \leftarrow 1$ ;
4    $\beta \leftarrow iter$ ;
5   enquanto (  $iter \leq maxIter$  ) faça
6        $s' \leftarrow$  Perturbação ( $s$ ,  $\beta$ );
7       se (  $iter \leq iterMaxSemMelhoraVND$  ) então
8            $s'' \leftarrow$  VND ( $s'$ );
9       senão
10           $s'' \leftarrow$  BuscaTabu (  $s'$ , tamanhoListaTabu, iterMaxBT );
11      fim se;
12      se (  $f(s'') \leq f(s)$  ) então
13           $s \leftarrow s''$ ;
14           $iter \leftarrow 1$ ;
15           $\beta \leftarrow iter$ ;
16      senão
17           $iter \leftarrow iter + 1$ ;
18           $\beta \leftarrow iter$ ;
19      fim se;
20  fim enquanto;
21  Retorne  $s$ ;
fim

```

Fonte: Silva (2012).

4.5.3 Busca Local: Variable Neighborhood Descent (VND)

Um dos métodos de busca local utilizado no algoritmo ESSILS (apresentado na Subseção 4.5.2) é a metaheurística VND (*Variable Neighborhood Descent*) proposta por Mladenovic e Hansen (1997). O método consiste em explorar o espaço de soluções por meio de trocas sistemáticas das estruturas de vizinhanças (apresentadas na Seção 4.3) disponíveis. Salienta-se ainda que o VND é caracterizado por um procedimento de avaliação que aceita somente soluções de melhora em relação a solução corrente e, quando isso acontece, o processo de busca retorna para a primeira estrutura de vizinhança da lista.

O Algoritmo 7 apresenta o pseudocódigo do método VND aplicado ao problema onde se considera o refinamento de uma solução s . Ele é dividido em duas estratégias, assim como adotado no trabalho de Mine (2009). A primeira delas é caracterizada por definir, aleatoriamente, uma ordem das vizinhanças a serem exploradas. O conjunto V de vizinhanças (já em ordem aleatória) exploradas pelo VND é formado por todas aquelas descritas na Seção 4.3. A outra estratégia utilizada é a de intensificação da busca nas salas cirúrgicas modificadas em cada iteração do método. Esse processo de intensificação é executado a partir dos procedimentos de busca local estruturados com base nos movimentos Troca Ordem Cirurgia, Realoca Cirurgia, Troca Ordem Múltiplas Cirurgias, Realoca Múltiplas Cirurgias, Troca Cirurgia Entre Blocos, Realoca Cirurgia Entre Blocos, Troca Múltiplas Cirurgias Entre Blocos. Salienta-se que os movimentos que envolvem a modificação de um conjunto de cirurgias ao mesmo tempo, ou seja, os quais são caracterizados pela movimentação de múltiplas cirurgias, são conduzidos pela variável k , a qual recebe um valor aleatório dentro do intervalo $[2,4]$ (linha 10). Ainda no Algoritmo 7, seja s a solução corrente e n o número de estruturas de vizinhanças disponíveis no conjunto V , ou seja, $V = V^{(1)} \cup V^{(2)} \cup V^{(3)} \cup \dots \cup V^{(n)}$. O VND, portanto, inicia o procedimento analisando a primeira estrutura de vizinhança $V^{(1)}$ e, para cada iteração, produz o melhor vizinho s' a partir da solução corrente s na vizinhança $V^{(contador)}$, como pode ser observado na linha 5. Caso s' encontrado seja melhor que s , s' passa a ser a nova solução corrente e o método é retornado à vizinhança $V^{(1)}$ (linhas 6, 7 e 8). Se s' não for melhor que s , o procedimento caminha para a próxima estrutura de vizinhança $V^{(contador + 1)}$, como pode ser observado na linha 18 quando a variável *contador* é incrementada. O método só é concluído quando não se encontra uma solução s' pertencente a $V^{(n)}$ melhor que a melhor solução encontrada até então.

Algoritmo 7: Pseudocódigo do método de busca local VND.

```

Procedimento Busca Local - VND
1   Considere  $n$  o número de estruturas de vizinhanças distintas;
2    $V \leftarrow$  conjunto das vizinhanças (descritas na seção 4.3) ordenadas de forma aleatória;
3    $contador \leftarrow 1$ ;
4   enquanto (  $contador \leq n$  ) faça
5       Encontre o melhor vizinho  $s' \in V^{(contador)}(s)$ ;
6       se (  $f(s') \leq f(s)$  ) então
7            $s' \leftarrow s$ ;
8            $contador \leftarrow 1$ ;
9           {Intensificação nas salas cirúrgicas alteradas}
10           $k \leftarrow$  número aleatório [2,4];
11           $s \leftarrow$  Movimento_Troca_Ordem_Cirurgia( $s$ );
12           $s \leftarrow$  Movimento_Realoca_Cirurgia( $s$ );
13           $s \leftarrow$  Movimento_Troca_Ordem_Múltiplas_Cirurgias( $s,k$ );
14           $s \leftarrow$  Movimento_Realoca_Múltiplas_Cirurgias( $s,k$ );
15           $s \leftarrow$  Movimento_Troca_Cirurgia_Entre_Blocos( $s$ );
16           $s \leftarrow$  Movimento_Realoca_Cirurgia_Entre_Blocos( $s$ );
17           $s \leftarrow$  Movimento_Troca_Múltiplas_Cirurgias_Entre_Blocos( $s,k$ );
18          senão
19               $contador \leftarrow contador + 1$ ;
20          fim se
21      fim enquanto
22  Retorne  $s$ ;
fim

```

Fonte: Mine (2009).

4.5.4 Busca Local: Busca Tabu

O segundo método de busca local utilizado pelo algoritmo ESSILS foi a metaheurística Busca Tabu (GLOVER, 1986; GLOVER; LAGUNA, 1997). Segundo Souza (2012), a Busca Tabu (BT) é um procedimento adaptativo que utiliza uma estrutura de memória para explorar o espaço de soluções do problema em questão. A partir de uma determinada solução, o algoritmo BT explora, a cada iteração, um conjunto de vizinhos. Sendo assim, o procedimento BT percorre o espaço de busca caminhando de uma solução para outra que seja seu melhor vizinho, mesmo que este vizinho seja obtido por meio de um movimento de piora (SOUZA, 2012).

De acordo com Armentano e Branchini (2013), a estratégia de elencar o melhor vizinho como a nova solução é utilizada com o intuito de escapar de ótimos locais, contudo pode fazer com que o algoritmo forme ciclos, ou seja, retorne a uma solução já encontrada anteriormente percorrendo o mesmo caminho de busca. Portanto, para não ocorrer ciclagem

do método BT, ou seja, retornar a uma solução já gerada anteriormente, utiliza-se a Lista Tabu (LT) para armazenar informações inerentes às soluções já obtidas. O tamanho da LT, responsável por determinar o tempo em que o movimento vai permanecer proibido (tabu), é definido como um parâmetro do algoritmo de Busca Tabu. O funcionamento da Lista Tabu é descrito ainda nesta Subseção.

Algoritmo 8: Pseudocódigo do método de busca local BT.

```

Procedimento Busca Local - BT ( tamanhoListaTabu, iterMaxBT, iterAux, s )
1    $s^* \leftarrow s$ ;           [melhor solução encontrada até então]
2    $tamanhoOriginal \leftarrow tamanhoListaTabu$ ;           [tamanho da Lista Tabu]
3    $iter \leftarrow 0$ ;           [número de iterações do algoritmo BT]
4    $melhorIter \leftarrow 0$ ;           [número de iterações sem melhora da solução]
5   enquanto (  $iter - melhorIter \leq iterMaxBT$  ) faça
6       se (  $iter - melhorIter \leq iterAux$  ) então
7            $tamanhoListaTabu \leftarrow tamanhoListaTabu + 1$ ;
8       fim se
9        $s' \leftarrow Movimento\_Troca\_Ordem\_Cirurgia(s)$ ;
10       $s'' \leftarrow Movimento\_Realoca\_Cirurgia(s)$ ;
11       $s''' \leftarrow Movimento\_Troca\_Cirurgia\_Entre\_Blocos(s)$ ;
12       $s'''' \leftarrow Movimento\_Realoca\_Cirurgia\_Entre\_Blocos(s)$ ;
13       $s \leftarrow \text{mínimo} ( f(s'), f(s''), f(s'''), f(s''') )$ ;
14       $atualizaListaTabu ( tamanhoListaTabu, Lista, Anterior, Proximo, iter, \Delta$ 
);
15      se (  $f(s) \leq f(s^*)$  ) então
16           $s^* \leftarrow s$ ;
17           $melhorIter \leftarrow iter$ ;
18           $tamanhoListaTabu \leftarrow tamanhoOriginal$ ;
19      fim se
20       $iter \leftarrow iter + 1$ ;
21  fim enquanto

```

Fonte: Silva (2012).

Descreve-se, a seguir, o pseudocódigo do procedimento de Busca Tabu baseado na abordagem adotada por Silva (2012) e apresentado pelo Algoritmo 8. Como pode ser observado, a Busca Tabu implementada nesta dissertação possui quatro parâmetros de entrada, sendo eles *tamanhoListaTabu*, *IterMaxBT*, *iterAux* e *s*. O primeiro parâmetro representa o tamanho da Lista Tabu, o segundo o número máximo de iterações do procedimento em questão, o terceiro um número de iterações consideravelmente grande e o último a solução corrente. Entre as linhas 9 e 12 do Algoritmo 8 são geradas quatro soluções a cada iteração utilizando os movimentos Troca Ordem Cirurgia, Realoca Cirurgia, Troca Cirurgia Entre Blocos e Realoca Cirurgia Entre Blocos descritos na Seção 4.3. A partir de

então, seleciona-se a melhor solução gerada entre esses quatro movimentos que, para o problema de cirurgias eletivas, considera-se a que possuir o menor tempo total de processamento (linha 13). Salienta-se que a aplicação destes movimentos é feita de forma sequencial: um após o outro. A cada vez que se move para uma nova solução, a Lista Tabu é atualizada (linha 14). Toda vez que o número de iterações sem melhora alcança um limiar, o tamanho da Lista Tabu é aumentado em uma unidade (linha 7). Quando uma nova solução encontrada é melhor que a corrente, o tamanho da lista tabu retorna ao original e a variável *iter* recebe o valor correspondente a esta iteração (linha 16 até 19). O algoritmo BT só é interrompido quando o número máximo de iterações sem melhora é atingido, representado pela variável *IterMaxBT*.

O processo de atualização da Lista Tabu é descrito a seguir e seu pseudocódigo é apresentado no Algoritmo 9. Como mencionado anteriormente, a Busca Tabu adota uma técnica para impedir que o movimento tabu se repita, evitando que o algoritmo retorne para uma solução já gerada. O movimento tabu adotado para o método em questão caracteriza-se por proibir que a cirurgia afetada pelo movimento gerado seja sucessora da cirurgia adjacente a ela antes do movimento. Assim como Silva (2012), adotou-se como critério de aspiração do algoritmo desenvolvido a permissão de realizar um movimento tabu que gere uma solução melhor que a melhor solução encontrada até então. Este é um mecanismo que retira, sob certas circunstâncias, o status tabu de um movimento, evitando proibir movimentos para soluções que ainda não foram visitadas.

Algoritmo 9: Pseudocódigo da Lista Tabu.

Procedimento <i>atualizaListaTabu</i> (<i>tamanhoListaTabu</i> , <i>Lista</i> , <i>Anterior</i> , <i>Proximo</i> , <i>iter</i> , Δ) 1 <i>mínimo</i> \leftarrow <i>tamanhoListaTabu</i> + <i>iter</i> - Δ ; 2 <i>máximo</i> \leftarrow <i>tamanhoListaTabu</i> + <i>iter</i> + Δ ; 3 <i>duraçãoTabu</i> \leftarrow valor aleatório entre [<i>mínimo</i> , <i>máximo</i>]; 4 <i>Lista</i> (<i>Anterior</i> , <i>Proximo</i>); 5 Retorne <i>Lista</i> ; fim

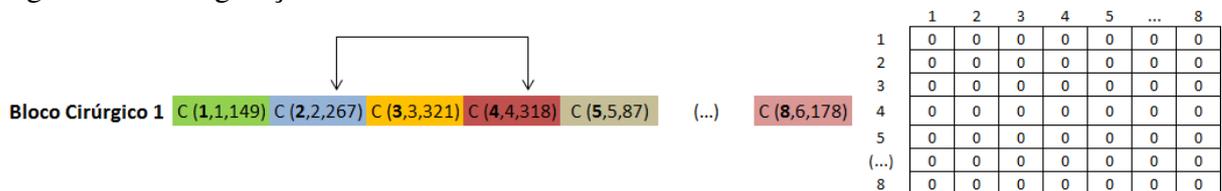
Fonte: Silva (2012).

Para o presente trabalho foi utilizada uma Lista Tabu que assumiu o tamanho 10 e um valor de Δ igual a 2. Sendo assim, a lista assume tamanhos diferentes ao longo da execução do algoritmo, uma vez que existe um fator de aleatoriedade presente no método (linha 3). Sendo assim, é justamente a mudança do tamanho da Lista Tabu que permite produzir caminhos diferentes na exploração do espaço de busca, diminuindo a probabilidade de

ciclagem. Quanto maior é o tamanho da Lista Tabu, mais restrita ela é, ou seja, impede o retorno a uma solução já gerada anteriormente e também a outras soluções ainda não geradas.

A Lista Tabu é representada por uma matriz quadrada $O(1)$ com baixa complexidade de consulta, onde em cada célula (i, j) é gravado o período em iterações pela qual a cirurgia j está proibida de ser sucessora da cirurgia i . A Figura 18 e a Figura 19 ilustram o procedimento da Lista Tabu adotado pelo presente trabalho, exemplificando o modelo para um conjunto de oito cirurgias alocadas em um determinado bloco cirúrgico. Em ambas as figuras, a Lista Tabu é representada do lado direito e, do lado esquerdo, a ilustração do movimento adotado que, para este exemplo, adotou-se o Movimento Troca Ordem Cirurgia.

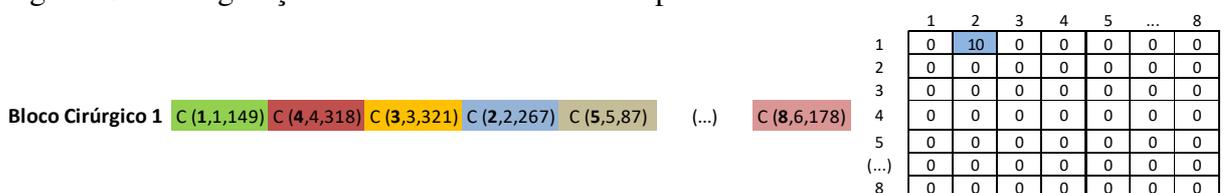
Figura 18: Configuração da matriz da Lista Tabu antes do movimento.



Fonte: Autor.

A Figura 18 ilustra o início do procedimento, uma vez que não existe nenhum movimento tabu ainda, já que a matriz do lado direito está totalmente preenchida com o valor 0. Ainda na Figura 18, pode-se observar que o Bloco Cirúrgico 1 está realizando uma sequência de cirurgias, iniciando em $C(1,1,149)$ e terminando em $C(8,6,178)$. Na Figura 19 observa-se o resultado do Movimento Troca Ordem Cirurgia, inicialmente representado pelo conjunto de setas na Figura 18. Neste movimento, a cirurgia $C(2,2,267)$ é trocada de posição com a cirurgia $C(4,4,318)$. A nova ordem está claramente representada pela Figura 19. A matriz da Lista Tabu foi então atualizada na célula $(1, 2)$ de acordo com o identificador (em negrito) da cirurgia, já que antes do movimento a cirurgia $C(2,2,267)$ era sucessora da cirurgia $C(1,1,149)$. Para este exemplo, qualquer movimento em que a cirurgia $C(1,1,149)$ seja sucedida pela cirurgia $C(2,2,267)$ estará impedido de ser realizado até a iteração de número 10 (valor gravado na célula em azul).

Figura 19: Configuração da matriz da Lista Tabu após o movimento.



Fonte: Autor.

4.5.5 Mecanismos de Perturbação

Os mecanismos de perturbação são responsáveis por diversificar a busca, fazendo com que o algoritmo não permaneça em uma mesma região do espaço de soluções. Este procedimento é caracterizado por realizar modificações na solução, como por exemplo a aplicação de uma série de movimentos como aqueles descritos na Seção 4.3.

A estrutura de perturbação utilizada neste trabalho foi baseada em uma função adotada por Andrade (2012). O Algoritmo 10 representa o princípio de funcionamento do mecanismo de perturbação implementado no algoritmo ESSILS (linha 6 do Algoritmo 6). Como pode ser observado no Algoritmo 10, o procedimento denominado de *Mecanismos_Perturbação* possui dois parâmetros de entrada, s e β , onde o primeiro representa a solução corrente que será modificada pela perturbação e o segundo representa o nível de perturbação, no caso, o número de vezes que um dos movimentos, escolhido de forma aleatória, será aplicado. Os movimentos utilizados são todos aqueles descritos na Seção 4.3. O valor que β assume é variável e depende do andamento do Algoritmo 6 (linha 15 e 18). Como mencionado, as linhas 2 e 3 do Algoritmo 10 são executadas β vezes, sendo a variável n responsável por armazenar o identificador, gerado aleatoriamente, de um dos sete movimentos da Seção 4.3 (linha 2). Em seguida, o movimento escolhido é aplicado na solução s (linha 3).

Algoritmo 10: Pseudocódigo dos Mecanismos de Perturbação.

```

Procedimento Mecanismos_Perturbação (  $s, \beta$  )
1   para  $i \leftarrow 1$  até  $\beta$  faça
2        $n \leftarrow$  Escolha aleatoriamente 1 dos 7 movimentos descritos na Seção 4.3;
3        $s \leftarrow$  realizeMovimento (  $n$  );
4   fim para
5   retorne  $s$ ;
fim

```

Fonte: Andrade (2012).

5 EXPERIMENTOS COMPUTACIONAIS

Nesta Seção são apresentados os experimentos computacionais que foram realizados com todos os algoritmos desenvolvidos por esta dissertação e que foram descritos, individualmente, na Seção 4. A Seção 5.1 é destinada a apresentar o conjunto de instâncias utilizadas para testar e validar os algoritmos propostos pelo presente trabalho.

A Seção 5.2, por sua vez, é responsável por apresentar os resultados computacionais obtidos em cada uma das instâncias testadas pelos algoritmos desenvolvidos. A estratégia adotada por esta dissertação será apresentar os resultados obtidos por cada um dos algoritmos desenvolvidos. Sendo assim, serão apresentados os resultados gerados pelo método GRASP (solução inicial), pelo método ILS combinado com a busca local VND e com solução inicial gerada pelo GRASP e, por fim, pelo método ILS combinado com dois métodos de busca local: VND e Busca Tabu, além da solução inicial obtida pela GRASP (nomeado de ESSILS).

5.1 CONJUNTO DE INSTÂNCIAS PARA O PROBLEMA DE SEQUENCIAMENTO DE CIRURGIAS

Esta Seção é dividida em quatro Subseções, cada uma representada pelas instâncias oriundas tanto de trabalhos encontrados na literatura, como de problemas reais. A Subseção 5.1.1 descreve as instâncias retiradas da tese de doutorado de Sperandio (2015), a Subseção 5.1.2 apresenta as instâncias utilizadas no trabalho de Andrade (2012), a Subseção 5.1.3 mostra as instâncias oriundas do trabalho de Daiki e Yih (2010) e, por último, a Subseção 5.1.4 descreve as instâncias desenvolvidas a partir de um problema real de um hospital de grande porte da cidade de São Paulo, Brasil.

5.1.1 Conjunto de Instâncias I

Esta Subseção é responsável por apresentar o conjunto de instâncias extraídas do trabalho de Sperandio (2015), o qual se baseou em um problema real de um hospital português de grande porte. As características dos problemas-teste em questão são apresentadas nas Tabelas 2 e 3. A Tabela 2 possui um conjunto de instâncias de tamanho médio e a Tabela 3 possui um conjunto de instâncias de maior complexidade, uma vez que aumenta o número de cirurgias a serem realizadas e mantém a quantidade de recursos disponíveis.

Tabela 2: Conjunto de instâncias de média complexidade.

Especialidade	Instância	Número Pacientes	Número Sala de Operação	Número Cirurgiões	Tempo Médio Duração (min)
1. Cirurgia Vascular	1.1	91	2	15	53,74
	1.2	214	4	15	43,52
	1.3	346	6	15	39,22
	1.4	214	2	15	43,52
	1.5	473	4	15	37,79
	1.6	721	6	17	36,95
2. Cirurgia Oral e Maxilofacial	2.1	32	2	11	47,69
	2.2	75	4	12	38,17
	2.3	115	6	13	36,78
	2.4	75	2	12	38,17
	2.5	165	4	14	33,19
	2.6	179	6	14	33,53
3. Cirurgia Neurológica	3.1	96	2	10	118,91
	3.2	179	4	15	124,34
	3.3	268	6	16	123,52
	3.4	179	2	15	124,34
	3.5	286	4	16	122,67
4. Cirurgia Oftalmológica	4.1	299	2	34	26,59
	4.2	299	2	34	26,59
5. Cirurgia Ortopédica	5.1	143	2	27	90,76
	5.2	282	4	29	92,34
	5.3	416	6	33	94,19
	5.4	282	2	29	92,34
	5.5	558	4	33	93,45
	5.6	861	6	35	90,38
6. Cirurgia Urológica	6.1	93	2	20	73,37
	6.2	206	4	20	63,73
	6.3	287	6	21	62,86
	6.4	206	2	20	63,73
	6.5	287	4	21	62,86
7. Cirurgia de Rinoplastia	7.1	87	2	14	56,22
	7.2	170	4	16	57,89
	7.3	253	6	16	58,36
	7.4	170	2	16	57,89
	7.5	335	4	16	58,80
	7.6	448	6	16	58,20
8. Cirurgia Geral I	8.1	59	2	9	91,17
	8.2	140	4	11	77,19
	8.3	204	6	13	77,10
	8.4	140	2	11	77,19
	8.5	275	4	15	75,91
	8.6	329	6	16	77,34
9. Cirurgia Geral II	9.1	65	2	8	70,51
	9.2	129	4	8	69,69
	9.3	192	6	8	70,33
	9.4	129	2	8	69,69
	9.5	214	4	8	69,12
10. Cirurgia Geral III	10.1	64	2	11	66,31
	10.2	123	4	11	63,10
	10.3	163	6	12	63,17
	10.4	123	2	11	63,10
	10.5	163	4	12	63,17

Fonte: Sperandio (2015).

Em ambas as tabelas a primeira coluna mostra a especialidade cirúrgica e a segunda coluna o identificador da instância. A partir da terceira coluna em diante, é apresentado o número de pacientes a serem operados, a quantidade de salas de operação que estão disponíveis a cada dia para o determinado tipo de cirurgia em questão, o número de cirurgiões e, por fim, o tempo médio de duração do procedimento cirúrgico em minutos.

Tabela 3: Conjunto de instâncias de alta complexidade.

Especialidade	Instância	Número Pacientes	Número Sala de Operação	Número Cirurgiões	Tempo Médio Duração (min)
1. Cirurgia Vascular	1.7	205	2	15	44,03
	1.8	457	4	15	37,67
	1.9	691	6	15	37,18
	1.10	457	2	15	37,67
	1.11	767	4	15	36,84
2. Cirurgia Oral e Maxilofacial	2.7	71	2	12	39,00
	2.8	156	4	14	33,81
	2.9	179	6	14	33,83
	2.10	156	2	14	33,81
	2.11	179	4	14	33,83
3. Cirurgia Neurológica	3.6	178	2	15	121,92
	3.7	286	4	16	122,67
	3.8	286	2	16	122,67
4. Cirurgia Oftalmológica	4.3	299	2	34	26,59
	4.4	299	2	34	26,59
5. Cirurgia Ortopédica	5.7	277	2	29	92,35
	5.8	551	4	33	92,81
	5.9	847	6	35	90,17
	5.10	551	2	33	92,81
	5.11	1153	4	37	87,95
	5.12	1281	6	37	86,59
6. Cirurgia Urológica	6.6	202	2	20	63,69
	6.7	287	4	21	62,86
	6.8	287	2	21	62,86
7. Cirurgia de Rinoplastia	7.7	165	2	16	58,13
	7.8	327	4	16	58,73
	7.9	448	6	16	58,20
	7.10	327	2	16	58,73
	7.11	448	4	16	58,20
8. Cirurgia Geral I	8.6	138	2	11	73,86
	8.7	268	4	15	76,41
	8.8	329	6	16	77,34
	8.9	268	2	15	76,41
	8.10	329	4	16	77,34
9. Cirurgia Geral II	9.6	126	2	8	70,01
	9.7	214	4	8	69,12
	9.8	214	2	8	69,12
9. Cirurgia Geral III	10.6	120	2	11	63,32
	10.7	163	4	12	63,17
	10.8	163	2	12	63,17

Fonte: Sperandio (2015).

5.1.2 Conjunto de Instâncias II

Nesta Subseção é apresentado o conjunto de instâncias retirado do trabalho de Andrade (2012) e que também será utilizado para avaliar os resultados produzidos pelos algoritmos propostos por essa dissertação. Essas instâncias foram produzidas com base em dados reais de quatro hospitais de grande porte da cidade de Belo Horizonte, Brasil. A Tabela 4 é composta pelos dados reais obtidos em 2011 destes 4 hospitais e as demais Tabelas, sendo elas 5, 6 e 7 representam a composição da instância real, porém com redução de 10% no número das salas de cirurgia, redução de 10% no número de cirurgias e aumento de 10% no número de cirurgias a serem realizadas, respectivamente.

As Tabelas 4, 5, 6 e 7 são assim estruturadas: a primeira coluna representa a instância, iniciada pelo prefixo *h*, a segunda coluna a quantidade de cirurgias previamente conhecida para o agendamento semanal, a terceira coluna a quantidade de salas cirúrgicas disponíveis, a quarta coluna a quantidade de salas de recuperação pós-anestésica disponíveis, a quinta coluna a quantidade de leitos de UTI (Unidade de Tratamento Intensivo) disponíveis, a sexta coluna a quantidade de cirurgias disponíveis, a sétima coluna a quantidade de enfermeiros disponíveis, a oitava coluna a quantidade de anestesistas disponíveis e, por último, o tempo do procedimento cirúrgico de cada instância em horas.

Tabela 4: Conjunto de instâncias com dados originais.

Instância	Cirurgias	Salas Cirúrgicas	Salas de RPA	Leitos de UTI	Cirurgias	Enfermeiros	Anestesistas	Duração (horas)
<i>h</i> 1	162	7	6	24	74	18	20	3,5
<i>h</i> 2	192	12	15	29	112	21	25	4,5
<i>h</i> 3	216	18	20	40	126	27	31	4,0
<i>h</i> 4	266	16	55	18	157	34	37	5,6
<i>h</i> 5	354	19	21	53	186	39	45	4,2
<i>h</i> 6	378	25	26	64	200	45	51	2,6
<i>h</i> 7	428	23	61	42	231	52	57	4,6
<i>h</i> 8	408	30	35	69	238	48	56	4,5
<i>h</i> 9	458	28	70	47	269	55	62	3,5
<i>h</i> 10	482	34	75	58	283	61	68	4,8
<i>h</i> 11	570	37	41	93	312	66	76	5,3
<i>h</i> 12	620	35	76	71	343	73	82	5,1
<i>h</i> 13	644	41	81	82	357	79	88	3,7
<i>h</i> 14	674	46	90	87	395	82	93	3,1
<i>h</i> 15	836	53	96	111	469	100	113	4,2

Fonte: Andrade (2012).

Tabela 5: Conjunto de instâncias com dados retrabalhados (-10% salas de cir.).

Intância	Cirurgias	Salas Cirúrgicas	Salas de RPA	Leitos de UTI	Cirurgiões	Enfermeiros	Anestesiastas	Duração (horas)
h 16	162	6	6	24	74	18	20	3,5
h 17	192	11	15	29	112	21	25	4,5
h 18	216	16	20	40	126	27	31	4,0
h 19	266	14	55	18	157	34	37	5,6
h 20	354	17	21	53	186	39	45	4,2
h 21	378	22	26	64	200	45	51	2,6
h 22	428	20	61	42	231	52	57	4,6
h 23	408	27	35	69	238	48	56	4,5
h 24	458	25	70	47	269	55	62	3,5
h 25	482	30	75	58	283	61	68	4,8
h 26	570	33	41	93	312	66	76	5,3
h 27	620	31	76	71	343	73	82	5,1
h 28	644	36	81	82	357	79	88	3,7
h 29	674	41	90	87	395	82	93	3,1
h 30	836	47	96	111	469	100	113	4,2

Fonte: Andrade (2012).

Tabela 6: Conjunto de instâncias com dados retrabalhados (-10% cirurgiões).

Intância	Cirurgias	Salas Cirúrgicas	Salas de RPA	Leitos de UTI	Cirurgiões	Enfermeiros	Anestesiastas	Duração (horas)
h 31	162	7	6	24	67	18	20	3,5
h 32	192	12	15	29	100	21	25	4,5
h 33	216	18	20	40	114	27	31	4,0
h 34	266	16	55	18	140	34	37	5,6
h 35	354	19	21	53	167	39	45	4,2
h 36	378	25	26	64	181	45	51	2,6
h 37	428	23	61	42	207	52	57	4,6
h 38	408	30	35	69	214	48	56	4,5
h 39	458	28	70	47	240	55	62	3,5
h 40	482	34	75	58	254	61	68	4,8
h 41	570	37	41	93	281	66	76	5,3
h 42	620	35	76	71	307	73	82	5,1
h 43	644	41	81	82	321	79	88	3,7
h 44	674	46	90	87	354	82	93	3,1
h 45	836	53	96	111	421	100	113	4,2

Fonte: Andrade (2012).

Tabela 7: Conjunto de instâncias com dados retrabalhados (+10% cirurgias).

Intância	Cirurgias	Salas Cirúrgicas	Salas de RPA	Leitos de UTI	Cirurgiões	Enfermeiros	Anestesiastas	Duração (horas)
h 76	180	7	6	24	74	18	20	3,5
h 77	210	12	15	29	112	21	25	4,5
h 78	237	18	20	40	126	27	31	4,0
h 79	293	16	55	18	157	34	37	5,6
h 80	390	19	21	53	186	39	45	4,2
h 81	417	25	26	64	200	45	51	2,6
h 82	473	23	61	42	231	52	57	4,6
h 83	447	30	35	69	238	48	56	4,5
h 84	503	28	70	47	269	55	62	3,5
h 85	530	34	75	58	283	61	68	4,8
h 86	627	37	41	93	312	66	76	5,3
h 87	683	35	76	71	343	73	82	5,1
h 88	710	41	81	82	357	79	88	3,7
h 89	740	46	90	87	395	82	93	3,1
h 90	920	53	96	111	469	100	113	4,2

Fonte: Andrade (2012).

5.1.3 Conjunto de Instâncias III

Nesta Subseção será apresentado outro conjunto de instâncias também utilizado como critério de comparação e validação dos algoritmos propostos por essa dissertação. A Tabela 8 apresenta o conjunto de problemas-teste proposto por Daiki e Yih (2010) e é assim caracterizada: a primeira coluna representa o grupo cirúrgico descrito em inglês, a segunda coluna o tempo médio de duração do procedimento cirúrgico em minutos, a terceira coluna a quantidade de cirurgias a serem realizadas, a quarta coluna a quantidade de salas de cirurgia disponíveis e, por último, a quantidade de cirurgiões disponíveis para cada grupo cirúrgico.

Tabela 8: Conjunto de instâncias de Daiki e Yih.

Grupo Cirúrgico	Duração (min)	Quantidade de Cirurgias	Salas de Cirurgia	Quantidade de Cirurgiões
ENT	74	788	10	4
OBGYN	86	342	10	3
ORTHO	107	859	10	5
NEURO	160	186	10	2
GEN	93	817	10	3
OPHTH	38	110	10	1
VASCULAR	120	303	10	2
CARDIAC	240	90	10	6
UROLOGY	64	198	10	1

Fonte: Daiki e Yih (2010).

5.1.4 Conjunto de Instâncias IV

Nesta Subseção são apresentadas as instâncias produzidas a partir de dados reais obtidos de um hospital de grande porte da cidade de São Paulo, Brasil. Essas instâncias, geradas a partir dos dados de 2015 do hospital em questão, também serão utilizadas para validar os resultados obtidos pelos algoritmos propostos por esta dissertação. A Tabela 9 é assim caracterizada: a primeira coluna armazena o identificador da instância, a segunda coluna a especialidade cirúrgica, a terceira coluna a duração média de cada especialidade cirúrgica, a quarta coluna o tempo de preparação da sala de operação, a quinta coluna a quantidade de salas de cirurgia disponíveis, a sexta coluna a quantidade de cirurgiões disponíveis, a sétima coluna a quantidade de anestesistas disponíveis e, por fim, a quantidade de cirurgias a serem realizadas (março de 2015).

Tabela 9: Conjunto de instâncias baseadas no problema real do hospital de grande porte de São Paulo.

Id Instância	Especialidade Cirúrgica	Duração Média (horas)	Tempo Preparação (horas)	Quantidade de Salas de Cirurgia	Quantidade de Cirurgiões	Quantidade de Anestesiastas	Quantidade de Cirurgias
1	Válvula	5	0,67	7	3	7	46
2	Coronária	5	0,67	7	7	7	25
3	Congênito	5	0,67	7	5	7	33
4	Aorta	5	0,67	7	1	7	10
5	Marca-passo	2	0,33	3	4	3	68
6	Tórax	2	0,67	7	10	7	7
7	Plástica	2	0,67	7	1	7	2
8	Geral	3	0,67	6	1	6	7
9	Ortopédica	5	0,67	6	1	6	4

Fonte: Autor.

Como pode ser observado na Tabela 9, o conjunto de instâncias apresentado é formado por nove tipos diferentes de especialidades cirúrgicas. A duração média e os respectivos tempos de preparação foram fornecidos pela Gerência Operacional do hospital de grande porte de São Paulo que apoiou esta pesquisa. De acordo com o cenário estrutural do hospital, existem nove salas de operação disponíveis para realizar as cirurgias em questão. Destas nove salas, sete (incluindo a híbrida) possuem condições estruturais para realizar as cirurgias de válvula, coronária, congênito, aorta, tórax e plástica. Das nove salas existentes, três delas (incluindo a híbrida) são disponibilizadas para a realização de cirurgias de marca-passo. A sala de operação híbrida, dentre as nove disponíveis, está disponível exclusivamente para o conjunto de instâncias de 1 a 7 e, portanto, as cirurgias geral e ortopédica devem ser realizadas dentre as seis salas do conjunto de nove. A Tabela 10 representa o quadro de salas de cirurgia disponíveis de acordo com a especialidade cirúrgica. Quando a célula está preenchida com o numeral 1 significa que a sala cirúrgica em questão tem capacidade para receber a cirurgia da linha corresponde e 0, caso contrário.

Tabela 10: Quadro de salas de cirurgias disponíveis.

Especialidade Cirúrgica	Sala_1	Sala_2	Sala_3 (Híbrida)	Sala_4	Sala_5	Sala_6	Sala_7	Sala_8	Sala_9
Válvula	0	0	1	1	1	1	1	1	1
Coronária	0	0	1	1	1	1	1	1	1
Congênito	0	0	1	1	1	1	1	1	1
Aorta	0	0	1	1	1	1	1	1	1
Marca-passo	1	1	1	0	0	0	0	0	0
Tórax	0	0	1	1	1	1	1	1	1
Plástica	0	0	1	1	1	1	1	1	1
Geral	0	0	0	1	1	1	1	1	1
Ortopédica	0	0	0	1	1	1	1	1	1

Fonte: Autor.

Segundo a agenda de trabalho dos cirurgiões do hospital em questão, dos sete cirurgiões disponíveis para a cirurgia de coronária, um deles não opera às tardes, o outro não opera às tardes e nem nas quintas-feiras, um terceiro não opera nas terças-feiras e quintas-feiras durante a tarde, dois deles não possuem restrições (operam manhã e tarde durante todos os dias úteis da semana) e, o último, atende apenas convênio (manhã e tarde todos os dias úteis da semana). Os cirurgiões de válvula não possuem restrições, ou seja, operam manhã e tarde durante todos os dias úteis da semana. Dos cinco cirurgiões de congênito, apenas um opera somente de manhã durante a semana e, os demais, operam todos os dias úteis durante a manhã e tarde. O cirurgião de aorta não opera nas segundas-feiras e nas quintas-feiras durante a tarde. Dos quatro cirurgiões de marca-passo, um não opera nas segundas-feiras, dois deles não operam nas segundas-feiras, nas quintas-feiras e nas sextas-feiras e, o último, não opera nas terças-feiras. Para as cirurgias plástica, geral e ortopédica são alocadas apenas um cirurgião por equipe de acordo com a demanda, sendo as duas primeiras realizadas pela tarde e a última pela manhã. Os dez cirurgiões de tórax são assim organizados: dois cirurgiões por dia útil, sendo um na manhã e outro na tarde.

Existe um anestesista para cada sala de cirurgia e, para o cenário em questão, considerou-se um horizonte de planejamento para o mês de março de 2015, o qual houve um histórico de realização de cirurgias apresentado na última coluna da Tabela 9.

5.2 RESULTADOS COMPUTACIONAIS

O algoritmo ESSILS foi codificado na linguagem de programação C++ utilizando a plataforma *Eclipse IDE for C/C++ Developers* e o compilador GCC (*GNU Compiler Collection*). Para executar o algoritmo foi utilizado um microcomputador com processador *Intel Core I5 1,70 GHz*, 8 GB de memória RAM e sistema operacional Windows 10. Apesar do microcomputador utilizado possuir dois núcleos, o algoritmo proposto não explora multiprocessamento.

Com o propósito de avaliar individualmente o ganho apresentado por cada método utilizado no algoritmo ESSILS, sendo eles o GRASP (algoritmo responsável pela geração da solução inicial) e o ILS (algoritmo de refinamento), o qual utiliza os métodos VND e Busca Tabu (que atuam como métodos de busca local do ILS), decidiu-se apresentar os resultados obtidos por cada um deles. Sendo assim, apresenta-se nesta Seção os resultados obtidos exclusivamente pelo algoritmo GRASP, assim como os resultados apresentados pelo método

ILS com busca local VND a partir da solução inicial com GRASP e, por último, os resultados obtidos pelo ILS com VND e Busca Tabu a partir dos resultados do GRASP (ESSILS).

Para validar os algoritmos em questão foram utilizadas 92 instâncias de Sperandio (2015), 60 instâncias de Andrade (2012), 9 instâncias de Daiki e Yih (2010) e 9 instâncias produzidas com dados reais de 2015 fornecidos por um hospital de grande porte de São Paulo, Brasil. Os parâmetros adotados, obtidos com base no trabalho de Silva (2012) e também por meio de uma bateria preliminar de testes, foram os seguintes: $maxIter = 10000$, $iterMaxSemMelhoraVND = 600$, $iterMaxBT = 300$, $tamanhoListaTabu = 10$ e $\Delta = 2$. Cada um dos algoritmos foi executado 30 vezes. O parâmetro *Time* do GRASP foi estabelecido experimentalmente para cada problema-teste de acordo com o que fora observado durante as execuções. O melhor tempo, a partir do momento que a solução inicial não é mais melhorada, será então o valor do parâmetro *Time*.

As tabelas a seguir apresentam os resultados obtidos tanto pelos algoritmos propostos por esta dissertação, assim como pelos algoritmos encontrados na literatura utilizados como critério de comparação. A coluna nomeada de “Instância” denota o problema-teste em questão. Já a coluna “Melhor Lit.” representa o melhor resultado encontrado na literatura para determinado problema-teste, ou seja, o valor do *makespan* encontrado para a instância. A coluna “Resultado” apresenta o melhor valor encontrado por aquele determinado trabalho a ser comparado com a presente dissertação. As colunas “GRASP”, “GRASP+ILS+VND” e “ESSILS” representam, respectivamente, o melhor valor encontrado pelo algoritmo GRASP, pelo algoritmo ILS combinado com VND e solução inicial GRASP e pelo algoritmo ILS combinado com VND, Busca Tabu e solução inicial GRASP (ESSILS), todos eles desenvolvidos na presente dissertação. O desvio do valor médio das soluções em relação ao melhor resultado da literatura, denotado por $Desv^{Avg}$, é calculado pela expressão (1). Já o desvio do melhor resultado encontrado pelo algoritmo em relação ao melhor conhecido na literatura, denotado por $Desv^{Best}$, é calculado pela expressão 2. Os valores de desvio médio das soluções e desvio do melhor resultado encontrado são apresentados com base nos resultados do algoritmo ESSILS, uma vez que este obteve os melhores resultados frente aos demais algoritmos propostos por esta dissertação. Ao final, as colunas Tempo representam, respectivamente, a duração (em segundos) dos algoritmos GRASP, ILS com VND e solução inicial GRASP e ILS combinado com VND, Busca Tabu e solução inicial GRASP (ESSILS).

$$Desv^{Avg} = \frac{Média - MelhorLiteratura}{MelhorLiteratura} \quad (1)$$

$$Desv^{Best} = \frac{Melhor - MelhorLiteratura}{MelhorLiteratura} \quad (2)$$

Nas Tabelas 11 e 12, o trabalho de Sperandio 2015, o qual utiliza programação inteira mista e algoritmos genéticos, é comparado com os algoritmos propostos por esta dissertação. Nota-se, na Tabela 11, que Sperandio (2015) encontra 30 melhores resultados dentre os 52 problemas-teste de média complexidade e alcança desvio médio de 0,02. Já o algoritmo ESSILS obtém 39 melhores soluções e desvio médio 0,01, mostrando-se mais eficiente que a proposta de Sperandio (2015). Já na Tabela 12, a qual apresenta os resultados para instâncias de alta complexidade, Sperandio (2015) tem 28 dos melhores resultados e desvio médio de 0,01 e o algoritmo ESSILS encontra 26 melhores resultados dentre os 40 problemas-teste e desvio médio de 0,01. Observa-se que Sperandio (2015) se sobressaiu em relação à quantidade de melhores resultados para as instâncias de maior complexidade, mas ESSILS ainda se mostrou capaz de apresentar desvio médio baixo. Isso prova que as soluções apresentadas pelo algoritmo ESSILS estão próximos dos valores ótimos da literatura e, portanto, forneceu soluções de qualidade assim como Sperandio (2015).

Tabela 11: Resultados para instâncias de média complexidade de Sperandio (2015).

Especialidade	Instância	Melhor Lit.	Sperandio (2015)				Cruz (2017)						
			Resultado	Desv ^{Best}	GRASP	GRASP+ILS+VND	ESSILS	Desv ^{Avg(ESSILS)}	Desv ^{Best(ESSILS)}	Tempo ^{GRASP}	Tempo ^{GRASP+ILS+VND}	Tempo ^{ESSILS}	
1. Cirurgia Vascular	1.1	163	171	0,05	247	183	163	0,01	0,00	9,24	50,79	78,50	
	1.2	155	155	0,00	233	175	155	0,01	0,00	6,93	55,41	101,59	
	1.3	151	153	0,01	231	165	151	0,01	0,00	5,77	41,56	83,12	
	1.4	310	310	0,00	388	329	310	0,00	0,00	6,93	42,71	85,43	
	1.5	298	298	0,00	352	314	298	0,00	0,00	5,77	66,96	79,65	
	1.6	261	267	0,02	323	279	261	0,00	0,00	6,93	42,71	87,73	
2. Cirurgia Oral e Maxilofacial	2.1	69	78	0,12	154	88	69	0,01	0,00	6,93	68,11	80,81	
	2.2	60	60	0,01	154	74	60	0,02	0,01	8,08	58,87	87,73	
	2.3	54	54	0,00	153	68	54	0,18	0,16	5,77	41,56	76,19	
	2.4	119	119	0,00	217	142	127	0,07	0,06	8,08	57,72	81,96	
	2.5	98	98	0,00	152	112	98	0,01	0,00	9,24	40,40	96,97	
	2.6	71	73	0,02	155	86	71	0,01	0,00	8,08	40,40	75,04	
3. Cirurgia Neurológica	3.1	571	571	0,00	656	584	577	0,01	0,01	5,77	62,34	70,42	
	3.2	371	373	0,01	431	381	371	0,00	0,00	6,93	51,95	102,74	
	3.3	345	345	0,00	425	368	357	0,04	0,04	6,93	48,48	83,12	
	3.4	742	751	0,01	841	753	742	0,00	0,00	8,08	45,02	81,96	
	3.5	548	548	0,00	639	563	548	0,00	0,00	9,24	48,48	79,65	
4. Cirurgia Oftalmológica	4.1	117	117	0,00	179	134	119	0,03	0,02	6,93	43,87	96,97	
	4.2	117	117	0,00	206	124	119	0,03	0,02	9,24	50,79	84,27	
5. Cirurgia Ortopédica	5.1	240	240	0,00	316	253	240	0,00	0,00	8,08	45,02	76,19	
	5.2	224	224	0,00	285	237	224	0,00	0,00	8,08	68,11	88,89	
	5.3	198	205	0,04	250	214	198	0,01	0,00	6,93	43,87	70,42	
	5.4	449	465	0,04	528	460	449	0,00	0,00	5,77	57,72	84,27	
	5.5	395	395	0,00	489	415	403	0,02	0,02	8,08	40,40	91,20	
	5.6	371	371	0,00	466	387	371	0,00	0,00	10,39	60,03	77,34	
6. Cirurgia Urológica	6.1	171	171	0,00	263	196	177	0,04	0,04	9,24	42,71	102,74	
	6.2	164	164	0,00	257	170	164	0,01	0,00	10,39	34,63	77,34	
	6.3	143	149	0,04	201	161	143	0,01	0,00	10,39	48,48	72,73	
	6.4	328	337	0,03	426	342	328	0,00	0,00	5,77	35,79	93,51	
	6.5	215	215	0,00	270	233	215	0,01	0,00	10,39	50,79	88,89	
	6.6	175	175	0,00	251	193	175	0,01	0,00	8,08	34,63	70,42	
7. Cirurgia de Rinoplastia	7.1	154	168	0,09	251	160	154	0,01	0,00	5,77	63,49	83,12	
	7.2	154	154	0,00	248	171	154	0,01	0,00	6,93	35,79	100,43	
	7.3	308	308	0,00	384	336	319	0,04	0,04	6,93	65,80	83,12	
	7.4	308	308	0,00	389	335	319	0,04	0,04	10,39	34,63	95,82	
	7.5	272	272	0,00	355	284	272	0,01	0,00	5,77	42,71	95,82	
	7.6	272	272	0,00	355	284	272	0,01	0,00	5,77	42,71	95,82	
8. Cirurgia Geral I	8.1	299	304	0,02	355	319	299	0,00	0,00	5,77	56,57	70,42	
	8.2	246	249	0,01	306	262	246	0,01	0,00	5,77	47,33	101,59	
	8.3	202	205	0,02	298	219	202	0,01	0,00	5,77	49,64	72,73	
	8.4	491	491	0,00	603	523	517	0,05	0,05	10,39	40,40	78,50	
	8.5	348	348	0,00	448	365	348	0,00	0,00	10,39	60,03	100,43	
	8.6	265	265	0,00	346	277	265	0,00	0,00	9,24	51,95	70,42	
9. Cirurgia Geral II	9.1	286	293	0,02	370	300	286	0,00	0,00	6,93	66,96	73,88	
	9.2	281	288	0,03	371	300	291	0,04	0,04	9,24	45,02	98,12	
	9.3	281	288	0,02	354	296	291	0,04	0,03	6,93	69,26	80,81	
	9.4	562	562	0,00	623	579	562	0,00	0,00	9,24	39,25	71,57	
	9.5	462	462	0,00	520	470	462	0,00	0,00	10,39	42,71	71,57	
10. Cirurgia Geral III	10.1	193	193	0,00	283	202	193	0,01	0,00	8,08	58,87	78,50	
	10.2	176	179	0,01	252	181	176	0,00	0,00	6,93	53,10	101,59	
	10.3	143	166	0,16	218	148	143	0,01	0,00	10,39	48,48	72,73	
	10.4	353	356	0,01	447	372	353	0,00	0,00	9,24	48,48	103,90	
	10.5	215	223	0,04	269	235	215	0,01	0,00	8,08	49,64	90,04	
Média	-	-	0,02	-	-	-	-	0,02	0,01	7,90	49,82	84,67	

Fonte: Autor.

Tabela 12: Resultados para instâncias de alta complexidade de Sperandio (2015).

Especialidade	Instância	Melhor Lit.	Sperandio (2015)		Cruz (2017)							
			Resultado	Desv ^{Best}	GRASP	GRASP+HLS+VND	ESSILS	Desv ^{Avg(ESSILS)}	Desv ^{Best(ESSILS)}	Tempo ^{GRASP}	Tempo ^{GRASP+HLS+VND}	Tempo ^{ESSILS}
1. Cirurgia Vascular	1.7	301	301	0,00	365	312	301	0,00	0,00	12,70	64,65	94,66
	1.8	287	287	0,00	381	303	287	0,00	0,00	11,54	61,18	111,98
	1.9	285	285	0,00	370	295	285	0,00	0,00	12,70	84,27	87,73
	1.10	574	574	0,00	670	591	574	0,00	0,00	15,01	54,26	103,90
	1.11	471	471	0,00	521	483	471	0,00	0,00	15,01	80,81	92,35
2. Cirurgia Oral e Maxilofacial	2.7	115	115	0,00	200	134	115	0,01	0,00	15,01	47,33	106,20
	2.8	94	105	0,11	152	113	94	0,01	0,00	16,16	81,96	90,04
	2.9	72	84	0,17	170	88	72	0,01	0,00	16,16	61,18	92,35
	2.10	188	188	0,00	267	209	188	0,00	0,00	15,01	81,96	87,73
	2.11	108	108	0,00	213	125	117	0,09	0,08	17,32	79,65	113,13
3. Cirurgia Neurológica	3.6	723	739	0,02	814	728	723	0,00	0,00	11,54	73,88	107,36
	3.7	548	567	0,03	614	569	548	0,00	0,00	12,70	68,11	107,36
	3.8	1096	1104	0,01	1204	1124	1119	0,02	0,02	11,54	51,95	88,89
4. Cirurgia Oftalmológica	4.3	117	117	0,00	214	131	117	0,01	0,00	17,32	58,87	87,73
	4.4	117	117	0,00	206	130	117	0,01	0,00	12,70	63,49	99,28
5. Cirurgia Ortopédica	5.7	441	441	0,00	552	466	453	0,03	0,03	16,16	76,19	109,67
	5.8	387	387	0,00	456	404	387	0,00	0,00	11,54	62,34	108,51
	5.9	364	364	0,00	453	387	372	0,03	0,02	11,54	65,80	102,74
	5.10	775	775	0,00	883	796	791	0,02	0,02	11,54	65,80	96,97
	5.11	685	689	0,01	744	699	685	0,00	0,00	16,16	50,79	115,44
	5.12	500	500	0,00	585	512	507	0,02	0,01	15,01	55,41	115,44
6. Cirurgia Urológica	6.6	322	322	0,00	384	332	322	0,00	0,00	11,54	48,48	101,59
	6.7	215	219	0,02	291	232	215	0,01	0,00	16,16	58,87	109,67
	6.8	430	439	0,02	525	448	430	0,00	0,00	11,54	69,26	110,82
7. Cirurgia de Rinoplastia	7.7	300	319	0,06	351	310	300	0,00	0,00	15,01	85,43	96,97
	7.8	300	319	0,06	362	318	300	0,00	0,00	16,16	50,79	95,82
	7.9	272	272	0,00	332	285	272	0,01	0,00	12,70	61,18	100,43
	7.10	600	600	0,00	714	622	614	0,02	0,02	17,32	58,87	88,89
	7.11	407	407	0,00	523	447	428	0,05	0,05	15,01	72,73	100,43
8. Cirurgia Geral I	8.6	463	463	0,00	571	490	479	0,04	0,03	13,85	47,33	95,82
	8.7	341	341	0,00	431	356	341	0,00	0,00	16,16	50,79	110,82
	8.8	265	265	0,00	336	277	265	0,00	0,00	11,54	57,72	99,28
	8.9	683	683	0,00	738	702	683	0,00	0,00	13,85	53,10	109,67
	8.10	398	398	0,00	488	413	398	0,00	0,00	16,16	46,18	107,36
9. Cirurgia Geral II	9.6	551	551	0,00	679	599	581	0,06	0,05	16,16	48,48	100,43
	9.7	462	469	0,01	556	478	462	0,00	0,00	15,01	56,57	100,43
	9.8	924	962	0,04	1022	956	941	0,02	0,02	15,01	85,43	107,36
9. Cirurgia Geral III	10.6	345	345	0,00	461	388	367	0,07	0,06	13,85	50,79	91,20
	10.7	215	215	0,00	274	232	215	0,01	0,00	13,85	48,48	91,20
	10.8	429	429	0,00	504	440	429	0,00	0,00	12,70	46,18	108,51
Média	-	-	0,01	-	-	-	-	0,01	0,01	14,20	62,16	101,15

Fonte: Autor.

Nas Tabelas 13, 14, 15 e 16 são apresentados os resultados obtidos pelos algoritmos propostos por esta dissertação em relação ao conjunto de instâncias elaborado por Andrade (2012). Para resolver o problema de sequenciamento de cirurgias eletivas, Andrade (2012) propõe uma série de algoritmos, sendo eles *GARSP*, *Iterated Local Search*, *Variable Neighborhood Search* e *Iterated Greedy Search*. O resultado apresentado como critério de comparação é o melhor valor encontrado por Andrade (2012) dentre os quatro algoritmos implementados por ela.

Na Tabela 13 observa-se que o algoritmo *ESSILS* melhora dois resultados frente aos resultados obtidos por Andrade (2012). Isto significa que além de alcançar todos os melhores resultados encontrados por Andrade (2012) neste primeiro conjunto de instâncias, o algoritmo proposto foi capaz de melhorar dois deles nas instâncias *h5* e *h13*. O desvio médio em relação ao melhor resultado de Andrade (2012) foi de 0,01 frente a 0,00 do algoritmo proposto, uma vez que este encontrou os melhores resultados na literatura para este conjunto de instâncias.

Tabela 13: Resultados para instâncias de Andrade (2012) com dados originais.

Instância	Melhor Lit.	Andrade (2012)		Cruz (2017)							
		Resultado	Desv ^{Best}	GRASP	GRASP+ILS+VND	ESSILS	Desv ^{Avg(ESSILS)}	Desv ^{Best(ESSILS)}	Tempo ^{GRASP}	Tempo ^{GRASP+ILS+VND}	Tempo ^{ESSILS}
<i>h1</i>	125	125	0,00	146	134	125	0,07	0,00	5,77	28,86	60,03
<i>h2</i>	92	92	0,00	121	98	92	0,09	0,00	5,77	26,55	55,41
<i>h3</i>	69	69	0,00	92	77	69	0,13	0,00	5,77	36,94	55,41
<i>h4</i>	172	172	0,00	198	186	172	0,05	0,00	6,93	43,87	53,10
<i>h5</i>	104	108	0,04	132	119	104	0,08	0,00	5,77	46,18	57,72
<i>h6</i>	93	93	0,00	115	107	93	0,09	0,00	6,93	20,78	53,10
<i>h7</i>	110	110	0,00	131	125	110	0,08	0,00	5,77	41,56	64,65
<i>h8</i>	86	86	0,00	111	93	86	0,10	0,00	8,08	33,48	63,49
<i>h9</i>	107	107	0,00	133	113	107	0,08	0,00	6,93	45,02	51,95
<i>h10</i>	95	95	0,00	124	108	95	0,09	0,00	6,93	31,17	51,95
<i>h11</i>	100	100	0,00	126	110	100	0,09	0,00	5,77	33,48	54,26
<i>h12</i>	110	110	0,00	138	125	110	0,08	0,00	8,08	19,62	62,34
<i>h13</i>	101	105	0,04	127	113	101	0,09	0,00	8,08	23,09	69,26
<i>h14</i>	89	89	0,00	109	95	89	0,10	0,00	6,93	35,79	69,26
<i>h15</i>	107	107	0,00	133	122	107	0,08	0,00	5,77	23,09	61,18
Média	-	-	0,01	-	-	-	0,09	0,00	6,62	32,63	58,87

Fonte: Autor.

Tabela 14: Resultados para instâncias de Andrade (2012) com dados retrabalhados (-10% salas de cir.).

Instância	Melhor Lit.	Andrade (2012)		Cruz (2017)							
		Resultado	Desv ^{Best}	GRASP	GRASP+ILS+VND	ESSILS	Desv ^{Avg(ESSILS)}	Desv ^{Best(ESSILS)}	Tempo ^{GRASP}	Tempo ^{GRASP+ILS+VND}	Tempo ^{ESSILS}
<i>h16</i>	141	141	0,00	163	149	141	0,06	0,00	9,24	32,32	64,65
<i>h17</i>	103	103	0,00	131	109	103	0,08	0,00	9,24	23,09	77,34
<i>h18</i>	80	80	0,00	102	90	80	0,11	0,00	9,24	41,56	83,12
<i>h19</i>	122	122	0,00	157	141	129	0,13	0,06	10,39	12,70	81,96
<i>h20</i>	121	121	0,00	145	129	121	0,07	0,00	13,85	16,16	61,18
<i>h21</i>	100	100	0,00	126	106	100	0,09	0,00	12,70	18,47	55,41
<i>h22</i>	130	130	0,00	154	145	130	0,07	0,00	9,24	38,10	83,12
<i>h23</i>	98	98	0,00	123	113	98	0,09	0,00	13,85	24,24	61,18
<i>h24</i>	125	125	0,00	152	144	131	0,12	0,05	9,24	27,71	64,65
<i>h25</i>	104	104	0,00	128	113	104	0,08	0,00	10,39	43,87	68,11
<i>h26</i>	110	110	0,00	134	119	110	0,08	0,00	13,85	40,40	70,42
<i>h27</i>	123	123	0,00	143	128	123	0,07	0,00	9,24	42,71	57,72
<i>h28</i>	116	116	0,00	143	129	123	0,14	0,06	13,85	39,25	70,42
<i>h29</i>	121	121	0,00	141	128	121	0,07	0,00	10,39	33,48	61,18
<i>h30</i>	122	122	0,00	151	133	122	0,07	0,00	9,24	36,94	83,12
Média	-	-	0,00	-	-	-	0,09	0,01	10,93	31,40	69,57

Fonte: Autor.

A Tabela 14 apresenta os resultados para o segundo conjunto de instâncias de Andrade (2012), aquele em que a quantidade de salas de cirurgia é reduzida em 10%. Nota-se que Andrade (2012) alcançou os 15 melhores resultados, enquanto o algoritmo ESSILS encontrou 12. O desvio médio em relação ao melhor resultado encontrado se manteve em 0,01, comprovando que ESSILS é um algoritmo de qualidade, uma vez que os resultados obtidos para as instâncias *h19*, *h24* e *h28* foram muito próximos ao melhor da literatura.

A Tabela 15 apresenta os resultados para o terceiro conjunto de instâncias de Andrade (2012), aquele em que a quantidade de cirurgias é reduzida em 10%. Observa-se que tanto os algoritmos de Andrade (2012) como o algoritmo ESSILS encontram os mesmos resultados, ou seja, ambos obtêm os 15 melhores resultados da literatura para o conjunto de instâncias em questão.

Tabela 15: Resultados para instâncias de Andrade (2012) com dados retrabalhados (-10% cirurgões).

Instância	Melhor Lit.	Andrade (2012)		Cruz (2017)							
		Resultado	Desv ^{Best}	GRASP	GRASP+ILS+VND	ESSILS	Desv ^{Avg(ESSILS)}	Desv ^{Best(ESSILS)}	Tempo ^{GRASP}	Tempo ^{GRASP+ILS+VND}	Tempo ^{ESSILS}
<i>h31</i>	114	114	0,00	135	121	114	0,08	0,00	12,70	46,18	86,58
<i>h32</i>	106	106	0,00	134	117	106	0,08	0,00	16,16	30,01	79,65
<i>h33</i>	73	73	0,00	101	80	73	0,12	0,00	15,01	21,93	80,81
<i>h34</i>	181	181	0,00	202	195	181	0,05	0,00	16,16	27,71	76,19
<i>h35</i>	113	113	0,00	136	123	113	0,08	0,00	12,70	38,10	72,73
<i>h36</i>	91	91	0,00	121	100	91	0,10	0,00	15,01	25,40	70,42
<i>h37</i>	129	129	0,00	157	142	129	0,07	0,00	15,01	51,95	79,65
<i>h38</i>	88	88	0,00	117	103	88	0,10	0,00	16,16	34,63	77,34
<i>h39</i>	108	108	0,00	128	117	108	0,08	0,00	15,01	25,40	83,12
<i>h40</i>	102	102	0,00	132	115	102	0,09	0,00	13,85	49,64	63,49
<i>h41</i>	102	102	0,00	129	112	102	0,09	0,00	16,16	21,93	86,58
<i>h42</i>	121	121	0,00	141	134	121	0,07	0,00	13,85	23,09	63,49
<i>h43</i>	102	102	0,00	127	116	102	0,09	0,00	12,70	48,48	71,57
<i>h44</i>	108	108	0,00	130	120	108	0,08	0,00	15,01	32,32	75,04
<i>h45</i>	125	125	0,00	152	133	125	0,07	0,00	13,85	23,09	64,65
Média	-	-	0,00	-	-	-	0,08	0,00	14,62	33,32	75,42

Fonte: Autor.

Tabela 16: Resultados para instâncias de Andrade (2012) com dados retrabalhados (+10% cirurgias).

Instância	Melhor Lit.	Andrade (2012)		Cruz (2017)							
		Resultado	Desv ^{Best}	GRASP	GRASP+ILS+VND	ESSILS	Desv ^{Avg(ESSILS)}	Desv ^{Best(ESSILS)}	Tempo ^{GRASP}	Tempo ^{GRASP+ILS+VND}	Tempo ^{ESSILS}
<i>h76</i>	143	143	0,00	171	152	143	0,06	0,00	17,32	31,17	84,27
<i>h77</i>	99	99	0,00	119	110	99	0,09	0,00	17,32	38,10	81,96
<i>h78</i>	88	88	0,00	114	101	88	0,10	0,00	19,62	43,87	83,12
<i>h79</i>	215	222	0,03	243	226	215	0,04	0,00	19,62	55,41	83,12
<i>h80</i>	124	124	0,00	148	134	124	0,07	0,00	17,32	34,63	86,58
<i>h81</i>	103	103	0,00	133	116	103	0,08	0,00	21,93	45,02	75,04
<i>h82</i>	131	131	0,00	157	140	131	0,07	0,00	20,78	28,86	68,11
<i>h83</i>	102	102	0,00	129	110	102	0,09	0,00	17,32	46,18	86,58
<i>h84</i>	136	144	0,06	162	142	136	0,06	0,00	18,47	34,63	83,12
<i>h85</i>	108	108	0,00	134	117	108	0,08	0,00	20,78	54,26	81,96
<i>h86</i>	117	117	0,00	140	131	117	0,07	0,00	18,47	48,48	78,50
<i>h87</i>	134	134	0,00	161	142	134	0,06	0,00	21,93	45,02	88,89
<i>h88</i>	117	117	0,00	145	128	117	0,07	0,00	19,62	45,02	76,19
<i>h89</i>	119	119	0,00	148	131	119	0,07	0,00	16,16	36,94	75,04
<i>h90</i>	123	123	0,00	147	131	123	0,07	0,00	16,16	35,79	81,96
Média	-	-	0,01	-	-	-	0,07	0,00	18,86	41,56	80,96

Fonte: Autor.

A Tabela 16 apresenta os resultados para o quarto conjunto de instâncias de Andrade (2012), aquele em que a quantidade de cirurgias é aumentada em 10%. É possível notar que o algoritmo ESSILS melhora dois resultados em relação ao trabalho proposto por essa autora. As instâncias *h79* e *h84* tiveram seus resultados melhorados pelo algoritmo ESSILS frente aqueles obtidos por Andrade (2012). O ESSILS alcança os 15 melhores resultados da literatura para este conjunto de instâncias, enquanto Andrade (2012) encontra 13 e desvio padrão médio em relação ao melhor resultado de 0,01.

Observa-se, na Tabela 17, os resultados obtidos pelos algoritmos propostos por esta dissertação para o conjunto de instâncias de Daiki e Yih (2010), os quais aplicam ferramentas de otimização estocástica e métodos de aproximação média da amostra para resolver o

problema de sequenciamento de cirurgias eletivas. Como pode ser observado, Daiki e Yih (2010) encontram os melhores resultados para todas as 9 instâncias disponíveis, enquanto ESSILS obtém 8 melhores resultados e desvio padrão médio de 0,02. Mesmo não superando Daiki e Yih (2010), o algoritmo ESSILS se mostrou capaz de produzir soluções de qualidade como pode ser notado na Tabela 17.

Tabela 17: Resultados para as instâncias de Daiki e Yih (2010).

Instância	Melhor Lit.	Daiki e Yih (2010)		Cruz (2017)							
		Resultado	$Desv^{Best}$	GRASP	GRASP+ILS+VND	ESSILS	$Desv^{Avg(ESSILS)}$	$Desv^{Best(ESSILS)}$	$Tempo^{GRASP}$	$Tempo^{GRASP+ILS+VND}$	$Tempo^{ESSILS}$
ENT	200	200	0,00	220	213	200	0,01	0,00	11,54	31,17	62,34
OBGYN	212	212	0,00	238	219	212	0,01	0,00	11,54	32,32	49,64
ORTHO	215	215	0,00	236	222	215	0,01	0,00	11,54	30,01	54,26
NEURO	217	217	0,00	240	228	217	0,01	0,00	12,70	28,86	63,49
GEN	218	218	0,00	240	227	218	0,01	0,00	11,54	31,17	60,03
OPHTH	219	219	0,00	259	249	235	0,08	0,07	11,54	34,63	54,26
VASCULAR	219	219	0,00	240	230	219	0,01	0,00	11,54	36,94	58,87
CARDIAC	220	220	0,00	241	235	220	0,01	0,00	12,70	30,01	54,26
UROLOGY	221	221	0,00	251	234	221	0,01	0,00	12,70	34,63	61,18
Média	-	-	0,00	-	-	-	0,02	0,01	11,93	32,19	57,59

Fonte: Autor.

A Tabela 18 é responsável por apresentar os resultados gerados pelos algoritmos desenvolvidos nesta dissertação aplicados ao problema real de um hospital de grande porte da cidade de São Paulo, Brasil. Como critério de comparação, os algoritmos desenvolvidos pelo presente trabalho serão executados com dados históricos do mês de março de 2015 e os resultados obtidos serão comparados com o planejamento que fora realizado. Isto significa que a agenda de cirurgias eletivas realizadas no mês de março de 2015, proposta pelo departamento de planejamento cirúrgico, é utilizada para verificar a eficiência dos algoritmos propostos. As colunas dos resultados, as quais representam o somatório do tempo total de processamento das cirurgias (*makespan*), foram baseadas em um horizonte de planejamento mensal contínuo com quatro semanas e considerando oito horas diárias de trabalho durante cinco dias úteis.

Analisando a Tabela 18, portanto, é possível notar que o algoritmo ESSILS foi capaz de melhorar todos os resultados obtidos pelo departamento de planejamento cirúrgico. Mesmo com restrições em relação à disponibilidade de salas de cirurgias e escala de cirurgiões, assim como fora descrito na Subseção 5.1.4, o algoritmo ESSILS obteve melhores resultados para todas as nove especialidades de cirurgia em questão quanto ao planejamento de realização delas. Comprova-se, desta forma, a eficiência do algoritmo proposto por esta dissertação em produzir resultados de qualidade.

Em geral, com base nos resultados obtidos pelos três algoritmos propostos, nota-se a qualidade superior dos resultados produzidos pelo algoritmo ESSILS. Isso evidencia que a combinação da geração da solução inicial pelo método GRASP e refinamento desta por meio do ILS com VND e Busca Tabu obteve sucesso. Observa-se que o algoritmo que utilizou GRASP como método de geração da solução inicial e como método de refinamento o ILS com VND também foi capaz de produzir resultados de qualidade, porém, com a inserção da Busca Tabu os resultados encontrados foram ainda melhores para todas as instâncias testadas. Não há dúvida que houve um acréscimo em relação ao tempo computacional, uma vez que o procedimento de busca local a mais foi inserido, mas houve compensação devido a melhoria da qualidade das soluções produzidas.

Tabela 18: Resultado para as instâncias baseadas no problema real do hospital de São Paulo.

Instância	Melhor Lit.	Dados Históricos (mar/2015)		Cruz (2017)							
		Resultado	Desv ^{Best}	GRASP	GRASP+ILS+VND	ESSILS	Desv ^{Avg(ESSILS)}	Desv ^{Best(ESSILS)}	Tempo ^{GRASP}	Tempo ^{GRASP+ILS+VND}	Tempo ^{ESSILS}
Válvula	135	155	0,15	148	137	135	0,02	0,00	9,24	28,86	40,40
Coronária	95	105	0,11	108	97	95	0,02	0,00	6,93	25,40	39,25
Congênito	115	135	0,17	128	117	115	0,02	0,00	8,08	23,09	48,48
Aorta	60	65	0,08	72	64	60	0,04	0,00	9,24	25,40	38,10
Marca-passo	152	160	0,05	163	155	152	0,01	0,00	8,08	28,86	43,87
Tárax	36	40	0,11	48	39	36	0,06	0,00	6,93	31,17	46,18
Plástica	22	26	0,18	37	24	22	0,10	0,00	5,77	30,01	39,25
Geral	42	48	0,14	57	45	42	0,05	0,00	5,77	30,01	45,02
Ortopédica	55	60	0,09	66	57	55	0,04	0,00	9,24	26,55	43,87
Média	-	-	0,12	-	-	-	0,04	0,00	7,70	27,71	42,71

Fonte: Autor.

6 ANÁLISE ESTATÍSTICA DOS RESULTADOS

Esta Seção é responsável por avaliar a eficiência dos algoritmos propostos em relação aos resultados obtidos por cada um deles por meio de métodos estatísticos de análise. Por se tratar de algoritmos heurísticos, a eficiência deles está relacionada à capacidade de gerar soluções de qualidade com a menor variabilidade possível (VONDER; DEMEULEMEESTER; HERROELEN, 2008).

A Seção 6.1 apresenta uma análise do desempenho computacional dos procedimentos heurísticos utilizados por meio do Teste de Probabilidade Empírica. Já a Seção 6.2 mostra uma análise estatística dos resultados com base nos valores de desvio padrão encontrados juntamente com a distribuição t de *Student*. Por fim, a Seção 6.3 apresenta os gráficos *Boxplot* para avaliar a distribuição empírica dos resultados.

Afim de viabilizar a apresentação dos procedimentos estatísticos utilizados para validação dos algoritmos e seus respectivos resultados, será utilizado um problema-teste de cada conjunto de instâncias daquelas apresentadas na Seção 5.1. Assim, para as análises estatísticas, foi escolhido um problema-teste denominado de Cirurgia Neurológica (3.7) de Sperandio (2015), um segundo denominado de h79 de Andrade (2012), outro denominado de NEURO de Daiki e Yih (2010) e, por último, um denominado de Congênito, do problema real do hospital de grande porte de São Paulo.

6.1 TESTE DE PROBABILIDADE EMPÍRICA

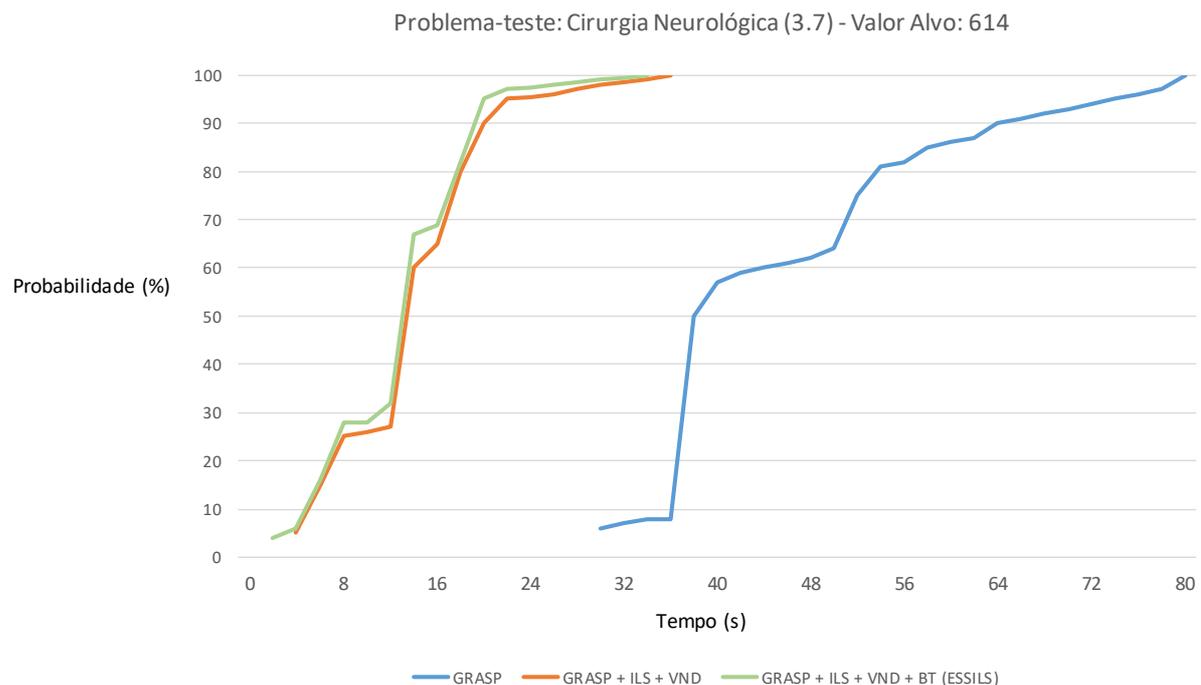
Nesta Seção é apresentado o Teste de Probabilidade Empírica para comparar o desempenho dos algoritmos propostos por esta dissertação. Segundo Aiex, Resende e Ribeiro (2002), a eficiência dos algoritmos heurísticos para encontrar uma determinada solução pode ser analisada por meio de gráficos TTTPlot (*Time-To-Target Plots*). Para a construção destes gráficos são considerados o tempo gasto e o número de vezes que os algoritmos foram capazes de chegar ao resultado estabelecido. Para este método em questão realizou-se 100 execuções e ordenou-se os tempos $T = \{t_1, t_2, \dots, 100\}$ que os algoritmos encontraram o valor alvo. Em seguida foi calculada a probabilidade acumulada $p_i = (i - \frac{1}{2}) / 100$ associada ao i -ésimo tempo de execução. Ao final, os 100 pontos $z_i = (t_i, p_i)$ foram plotados.

Pelo fato de o algoritmo GRASP também ter sido analisado estatisticamente e por este fazer parte da geração da solução inicial dos demais algoritmos, o parâmetro *Time* teve que ser alterado. Para os demais algoritmos, sendo eles GRASP combinado com ILS e VND, além

do ESSILS, o tempo (parâmetro *Time*) de execução do GRASP para geração da solução inicial foi limitado em 8 segundos, afim de avaliar a atuação dos métodos de refinamento quanto ao desempenho. Desta maneira, quando o GRASP é executado isoladamente o parâmetro *Time* assume um valor suficientemente alto a ponto de alcançar o valor alvo, que nada mais é que o melhor resultado encontrado pelo GRASP, conforme apresentado na Seção 5.2. Já para os outros algoritmos, que usam o GRASP para gerar a solução inicial, o parâmetro *Time* foi fixado com o intuito de focar a análise no desempenho obtido pelo método ILS, ora combinado somente com a busca local VND e ora combinado com as buscas locais VND e Busca Tabu.

Para o problema-teste Cirurgia Neurológica (3.7), o valor alvo definido foi 614 unidades de tempo, ou seja, o resultado do menor valor encontrado para o *makespan* obtido por meio do método GRASP. O Gráfico 2 apresenta o resultado do teste de probabilidade empírica para o problema-teste em questão, onde os três algoritmos propostos são comparados entre si. Nota-se que todos os algoritmos heurísticos propostos por esta dissertação são eficazes para a solução do problema, uma vez que todos eles alcançaram o valor alvo em todas as execuções. Porém, em relação ao desempenho computacional, percebe-se que o algoritmo ESSILS é o mais eficiente, já que alcança o valor alvo mais rapidamente (34 segundos) que todos os demais.

Gráfico 2: Probabilidade Empírica para o problema-teste Cirurgia Neurológica (3.7).

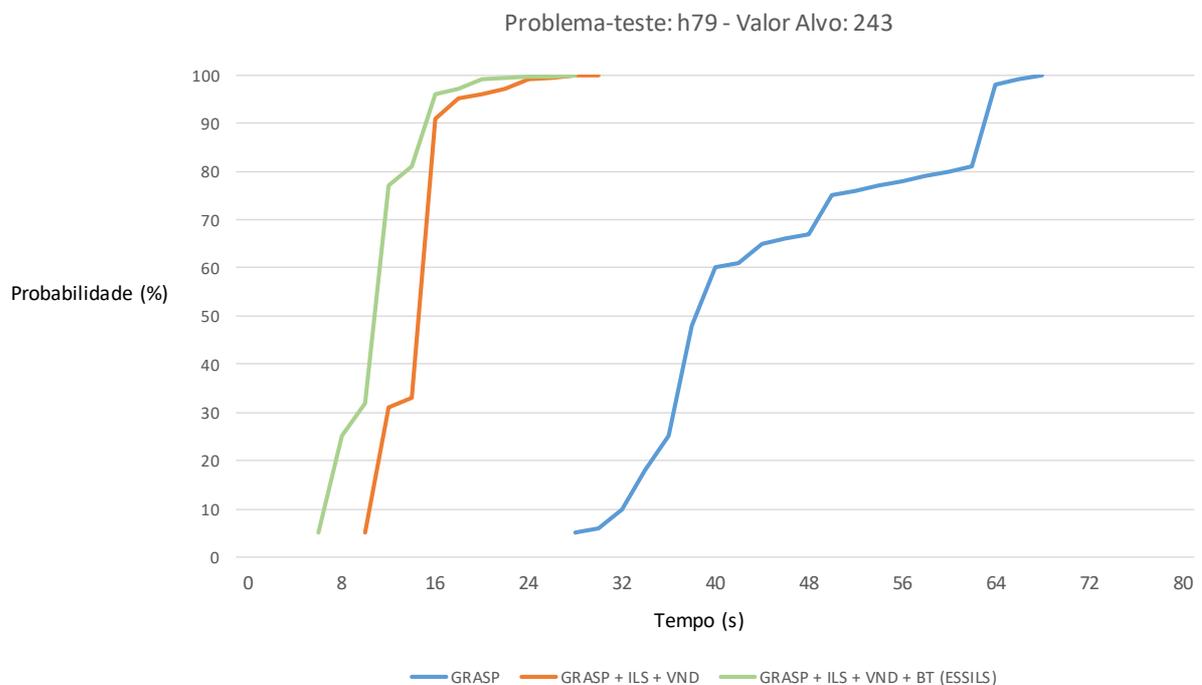


Fonte: Autor.

Não há dúvida que o algoritmo GRASP + ILS + VND tem eficiência muito próxima ao ESSILS, contudo o módulo Busca Tabu deste último contribuiu para que o valor alvo fosse encontrado em menor tempo. É possível observar também que a aplicação do método de refinamento ILS, após o método de geração da solução inicial, fez com que o valor alvo fosse encontrado de maneira bem mais rápida quando comparado ao uso exclusivo do GRASP.

O valor alvo definido para o problema-teste h79 de Andrade (2012) foi de 243 unidades de tempo, ou seja, o menor valor obtido pelo GRASP para o instante final da agenda de cirurgias. O Gráfico 3 apresenta o resultado do teste de probabilidade empírica dos algoritmos propostos executados neste problema-teste em questão. Como pode ser observado, o comportamento do gráfico muito se assemelha ao que fora representado pelo Gráfico 2. Porém, por se tratar de um problema-teste de menor complexidade, todos os algoritmos propostos por este trabalho levaram menos tempo para atingir o valor alvo quando comparados com a instância de Sperandio (2015), por exemplo. Ainda assim, todos encontraram o valor alvo e o ESSILS alcançou o melhor tempo (28 segundos), como já era de se esperar. Novamente se comprova o benefício da inserção da Busca Tabu como método de busca local do ILS.

Gráfico 3: Probabilidade Empírica para o problema-teste h79.

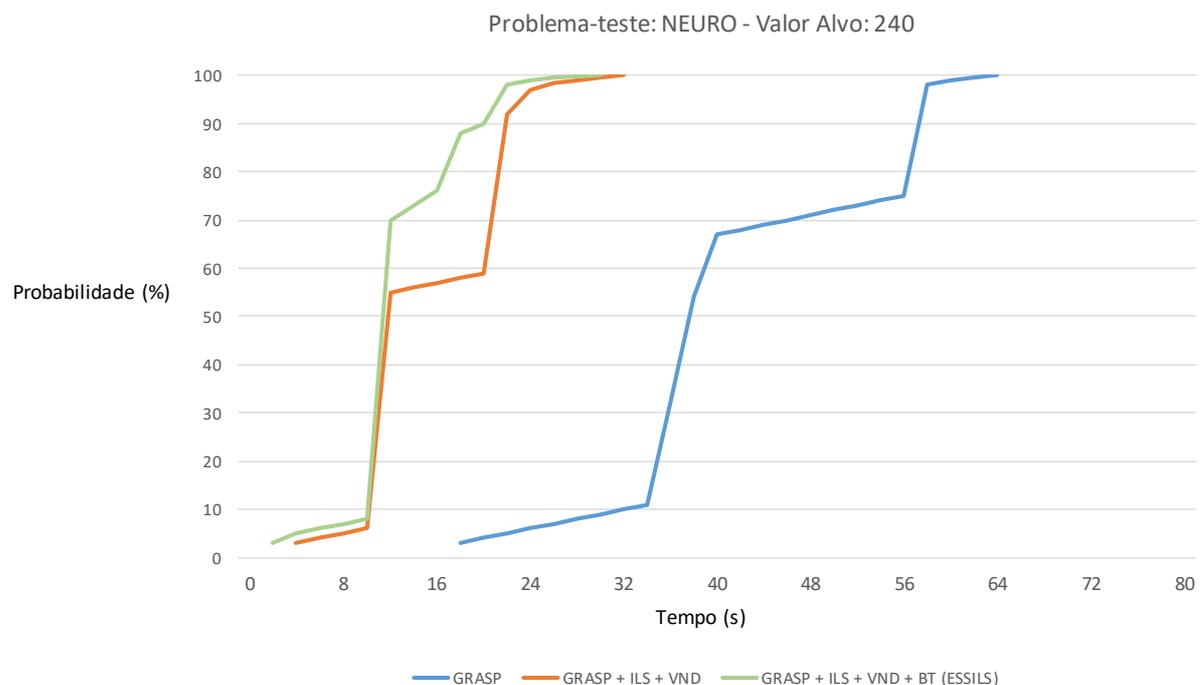


Fonte: Autor.

Para o problema-teste NEURO definiu-se o valor alvo de 240 unidades de tempo obtido como sendo o melhor resultado encontrado pelo método GRASP. O Gráfico 4 apresenta o resultado do teste de probabilidade empírica dos algoritmos propostos aplicados ao problema-teste em questão com comportamento bem semelhante ao observado nos outros dois anteriores. O valor alvo também é encontrado por todos os algoritmos, mas o ESSILS o alcança mais rapidamente, no caso, algo em torno de 30 segundos. Entre os intervalos de tempo 10s e 22s, o algoritmo ESSILS apresenta uma convergência mais rápida rumo ao valor alvo, mas, após certo tempo volta a se comportar muito próximo ao algoritmo GRASP + ILS + VND, sobressaindo-se levemente.

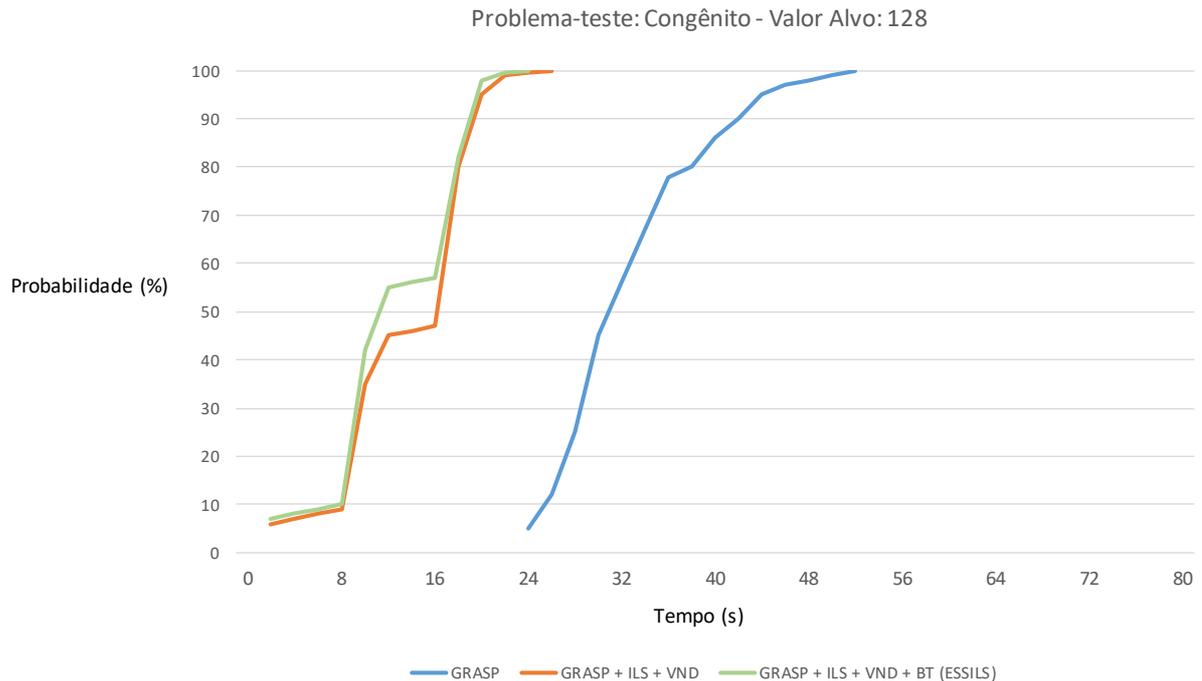
Foi definido o valor alvo de 128 unidades de tempo para o problema-teste Congênito, ou seja, o menor valor do *makespan* encontrado pelo método GRASP. O Gráfico 5 apresenta o resultado do teste de probabilidade empírica com base no problema-teste em questão. Mesmo tendo o comportamento parecido com os demais gráficos, nota-se que todos os algoritmos propostos por esta dissertação encontraram o valor alvo no menor tempo quando comparado com os demais problemas-teste utilizados para análise nesta Seção. O algoritmo ESSILS continua sendo o mais eficiente de todos os demais, uma vez que leva cerca de 24 segundos para alcançar o valor alvo.

Gráfico 4: Probabilidade Empírica para o problema-teste NEURO.



Fonte: Autor.

Gráfico 5: Probabilidade Empírica para o problema-teste Congênito.



Fonte: Autor.

6.2 TESTE DE DISTRIBUIÇÃO T DE STUDENT

Com a finalidade de comparar os resultados obtidos pelos algoritmos propostos por essa dissertação em relação à variabilidade utilizou-se o Teste de Distribuição t de *Student*, assim como valores de média e desvio padrão calculados sobre as amostras de resultados. Para tal, cada um dos algoritmos fora executado 30 vezes em cada um dos problemas-teste descritos na Seção 5.1 e os 30 resultados obtidos em cada um deles foram utilizados como critério de comparação entre si quanto à variabilidade.

De acordo com Doane e Seward (2014), o Teste t *Student* é um dos métodos mais utilizados para se avaliar a diferença entre as médias de dois grupos, comparar a média de uma amostra com a de uma população, comparar duas amostras independentes e, até mesmo, comparar duas amostras pareadas. Nesta Seção, portanto, a variação dos resultados encontrados por cada algoritmo proposto por este trabalho é avaliada em torno do valor médio da amostra. Sendo assim, é possível identificar se os algoritmos em questão produzem resultados que converjam para o valor da média.

As Tabelas 19, 20 e 21 apresentam, respectivamente, os resultados estatísticos obtidos por cada um dos algoritmos propostos: GRASP, GRASP com ILS e busca local VND e, por último, GRASP com ILS e busca local VND e Busca Tabu (ESSILS). Os quatro problemas-

teste utilizados para validação dos algoritmos no Teste *t* de *Student* são os mesmos descritos no início da Seção 6. Objetiva-se, portanto, analisar a hipótese nula (H_0) $\bar{x} \leq \mu_0$, onde o primeiro fator representa a média da amostra e o segundo fator o valor da amostra para comparação. Desta forma, a hipótese nula visa comprovar que os valores da amostra são de fato próximos ao valor da média. Sendo assim, quanto menor o valor do resultado do Teste *t* de *Student*, mais certeza tem para se afirmar que os valores da amostra são próximos ao valor da média que está se comparando, ou seja, concluir que a hipótese nula não deve ser rejeitada (DOANE; SEWARD, 2014).

De acordo com a Tabela 19, pode-se observar que os resultados do Teste *t* de *Student* são menores que 2 e, desta forma, conclui-se que a hipótese nula de igualdade de valores entre as 30 rodadas de cada problema não deve ser rejeitada. Portanto, em um nível de significância menor que 5%, conclui-se que a variação é bem pequena em torno do valor médio e que o algoritmo GRASP convergiu para o valor da média.

Tabela 19: Resultado para o Teste *t Student* - Algoritmo GRASP.

	GRASP			
	Cirurgia Neurológica (3.7)	h79	NEURO	Congênito
Média	615,60	244,27	240,97	129,80
Desvio Padrão	11,9788	3,8773	4,7866	2,5774
<i>t Student</i>	0,0195	0,0159	0,0199	0,0199

Fonte: Autor.

De acordo com a Tabela 20, pode-se observar que os resultados do Teste *t* de *Student* são menores que 2 e, desta forma, conclui-se que a hipótese nula de igualdade de valores entre as 30 rodadas de cada problema não deve ser rejeitada. Portanto, em um nível de significância menor que 5%, conclui-se que a variação é bem pequena em torno do valor médio e que o algoritmo GRASP combinado com o método de refinamento ILS e busca local VND convergiu para o valor da média.

Tabela 20: Resultado para o Teste *t Student* - Algoritmo GRASP + ILS + VND.

	GRASP + ILS + VND			
	Cirurgia Neurológica (3.7)	h79	NEURO	Congênito
Média	562,60	226,50	231,40	119,53
Desvio Padrão	1,8868	1,2551	0,9789	0,8604
<i>t Student</i>	0,0034	0,0055	0,0042	0,0072

Fonte: Autor.

De acordo com a Tabela 21, pode-se observar que os resultados do Teste *t* de *Student* são menores que 2 e, desta forma, conclui-se que a hipótese nula de igualdade de valores entre as 30 rodadas de cada problema não deve ser rejeitada. Portanto, em um nível de significância

menor que 5%, conclui-se que a variação é bem pequena em torno do valor médio e que o algoritmo ESSILS convergiu para o valor da média.

Tabela 21: Resultado para o Teste *t Student* - Algoritmo ESSILS.

	ESSILS (GRASP + ILS + VND + Busca Tabu)			
	Cirurgia Neurológica (3.7)	h79	NEURO	Congênito
Média	548,67	215,47	217,47	115,47
Desvio Padrão	1,0613	0,7303	0,8193	0,7761
<i>t Student</i>	0,0019	0,0034	0,0038	0,0067

Fonte: Autor.

Com essas análises é possível afirmar que os algoritmos propostos, mesmo sendo métodos heurísticos, são capazes de produzir resultados de qualidade e com baixa variabilidade. Comprova-se, então, que a parametrização dos algoritmos e a combinação dos métodos estão equilibrados. Salienta-se, ainda, que o algoritmo GRASP foi o que obteve maior variabilidade quando se compara os valores do Teste *t* de *Student* com os demais algoritmos propostos, mas ainda assim está dentro do parâmetro para a não rejeição da hipótese nula.

6.3 GRÁFICOS BOXPLOT

Nesta Seção é apresentada uma análise estatística acerca da dispersão dos resultados obtidos pelos algoritmos propostos por esta dissertação por meio de gráficos *Boxplot*. De acordo com Benjamini (2012), os gráficos *Boxplot* fornecem uma série de informações sobre as características do conjunto de dados que está sendo analisado, sendo elas locação, dispersão, assimetria, comprimento da cauda e *outliers* (medidas discrepantes). Todas essas informações são importantes para avaliar os resultados obtidos pelos algoritmos em relação à variabilidade das soluções finais produzidas.

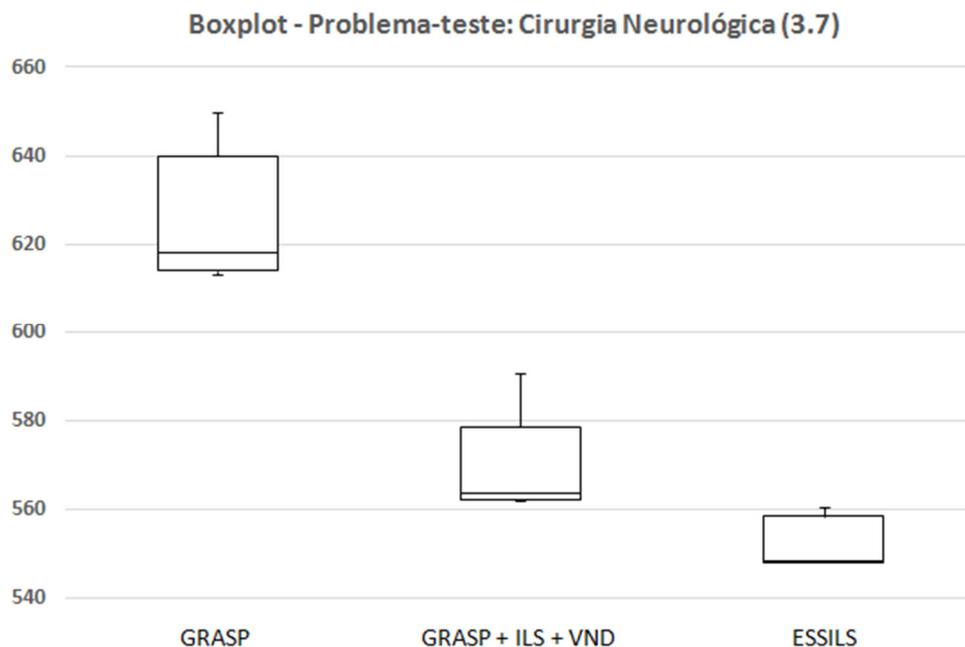
Os gráficos *Boxplot* necessitam de alguns dados para construí-los: Primeiro Quartil, Mediana (ou Segundo Quartil), Terceiro Quartil e o Intervalo Interquartil. Os quartis são valores que dividem uma amostra de dados em quatro partes iguais. A mediana é o valor que divide os dados no centro da amostra. O primeiro quartil, Q1, é o número que divide a amostra em 25% das observações abaixo e 75% acima, enquanto que o terceiro quartil, Q3, é o número que divide a amostra em 75% das observações abaixo e 25% acima. O Intervalo Interquartil (IIQ) é a diferença entre Q3 e Q1 (BENJAMINI, 2012).

O gráfico *Boxplot* por si só consiste de uma caixa que contém 50% dos dados amostrais. O limite superior da caixa indica o percentual de 75% dos dados e o limite inferior

da caixa indica o percentual de 25% dos dados. Quanto maior a distância entre a aresta inferior e a aresta superior da caixa, maior é a dispersão dos dados. A linha localizada no interior da caixa indica o valor de mediana dos dados. Por fim, as duas linhas externas à caixa representam duas possibilidades: caso não existam *outliers*, as duas linhas demarcam os valores máximo e mínimo do conjunto de dados e, caso existam *outliers*, estas duas linhas demarcam o maior valor não *outlier* e o menor valor não *outlier*.

Os Gráficos 6, 7, 8 e 9 apresentam a análise por meio de *Boxplot* relativa à variabilidade dos resultados de cada um dos algoritmos propostos acerca dos quatro problemas-teste relatados no início do Capítulo 6.

Gráfico 6: *Boxplot* para o problema-teste Cirurgia Neurológica (3.7).

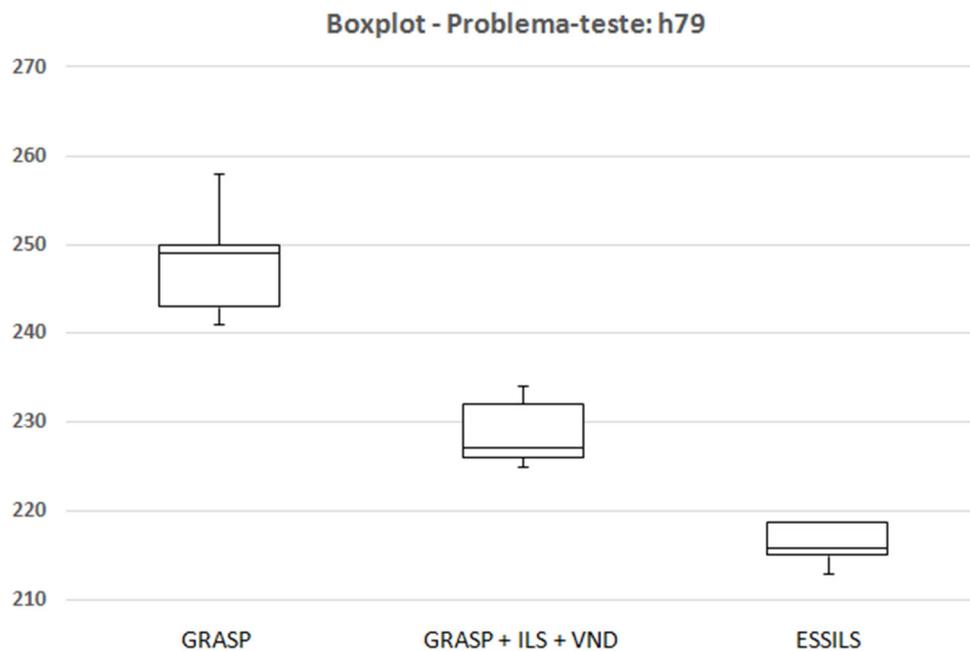


Fonte: Autor.

Como pode ser observado no Gráfico 6, o qual apresenta a análise acerca do problema-teste Cirurgia Neurológica (3.7), o gráfico *Boxplot* que indicou maior dispersão dos resultados foi o do algoritmo GRASP e, por outro lado, o que apresentou menor dispersão foi o do algoritmo ESSILS. Assim como constatado em outras análises, observa-se nesta também que o algoritmo ESSILS é o que apresenta de fato os melhores resultados, uma vez que alcança o menor valor de média, o menor valor de mediana e também o menor valor absoluto (*makespan*). Em se tratando de um problema de minimização, os melhores resultados se encontram na parte inferior do gráfico.

O Gráfico 7 apresenta o *Boxplot* em relação ao problema-teste h79. Observa-se que a dispersão dos resultados foi menor em todos os algoritmos propostos quando se compara com o Gráfico 6. O algoritmo ESSILS apresenta ponto de mínimo externo à caixa, mas a mediana está próxima do menor valor (*makespan*). Constata-se, portanto, que a maioria dos resultados obtidos por ESSILS se aproximam do menor valor encontrado e possuem baixa variabilidade entre si. O segundo melhor desempenho é o do algoritmo GRASP combinado com ILS e VND. Mesmo tendo um ponto de máximo fora da caixa, a maioria dos resultados obtidos se aproximam do menor valor encontrado por esse algoritmo. Esse resultado é diferente do que acontece exclusivamente com o algoritmo GRASP neste problema-teste h79, onde a mediana está próxima do limite superior, ou seja, próxima dos maiores valores encontrados. Significa, então, que a maioria dos resultados encontrados não se aproximam do melhor valor encontrado por esse algoritmo para o *makespan* de 243 unidades de tempo. Destaca-se, desta forma, que ESSILS continua sendo o melhor algoritmo entre os demais propostos por esta dissertação. Verifica-se também que nenhum *outlier* foi encontrado.

Gráfico 7: *Boxplot* para o problema-teste Cirurgia h79.

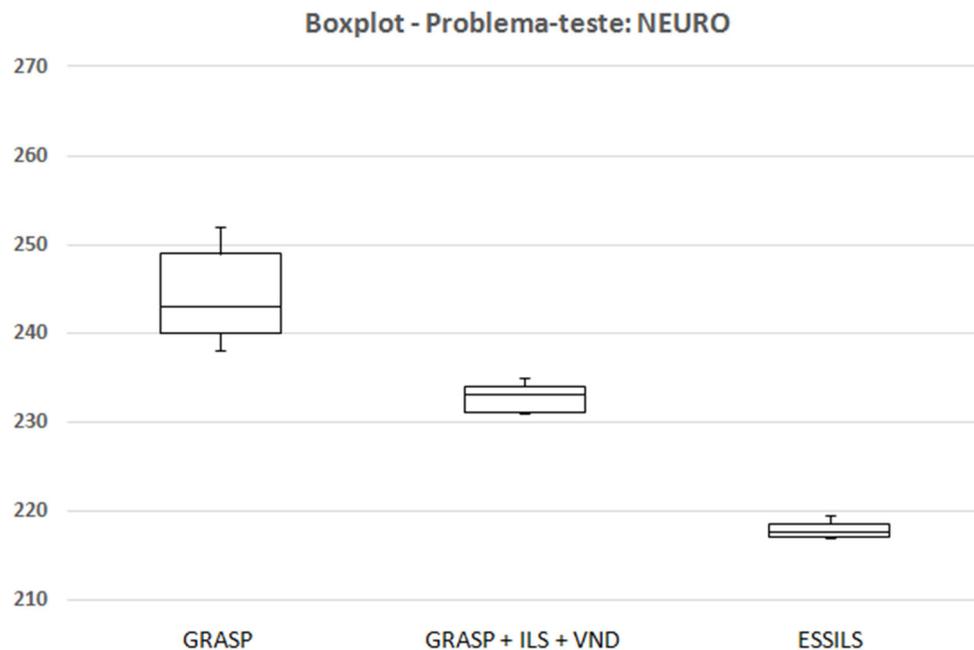


Fonte: Autor.

O Gráfico 8 apresenta a análise por meio do *Boxplot* relativo ao problema-teste NEURO para cada um dos algoritmos propostos. Claramente, o algoritmo ESSILS é o de melhor desempenho. É possível notar, também, que o algoritmo ESSILS e o algoritmo GRASP combinado com ILS e VND obtiveram resultados com baixa variabilidade, ao

contrário do algoritmo GRASP, quando é feita uma comparação usando o Gráfico 8. Por outro lado, a mediana do algoritmo GRASP está localizada mais próxima dos melhores resultados. Tanto o algoritmo ESSILS como o algoritmo GRASP + ILS + VND obtiveram pontos extremos fora da caixa, mas é possível notar que devido à baixa variabilidade dos resultados obtidos ainda assim são próximos dos melhores resultados encontrados. Salienta-se a melhor eficiência de ESSILS, assim como sua dispersão simétrica para este problema-teste, uma vez que a mediana está localizada no centro da caixa. Salienta-se que nenhum *outlier* foi encontrado.

Gráfico 8: *Boxplot* para o problema-teste Cirurgia NEURO.

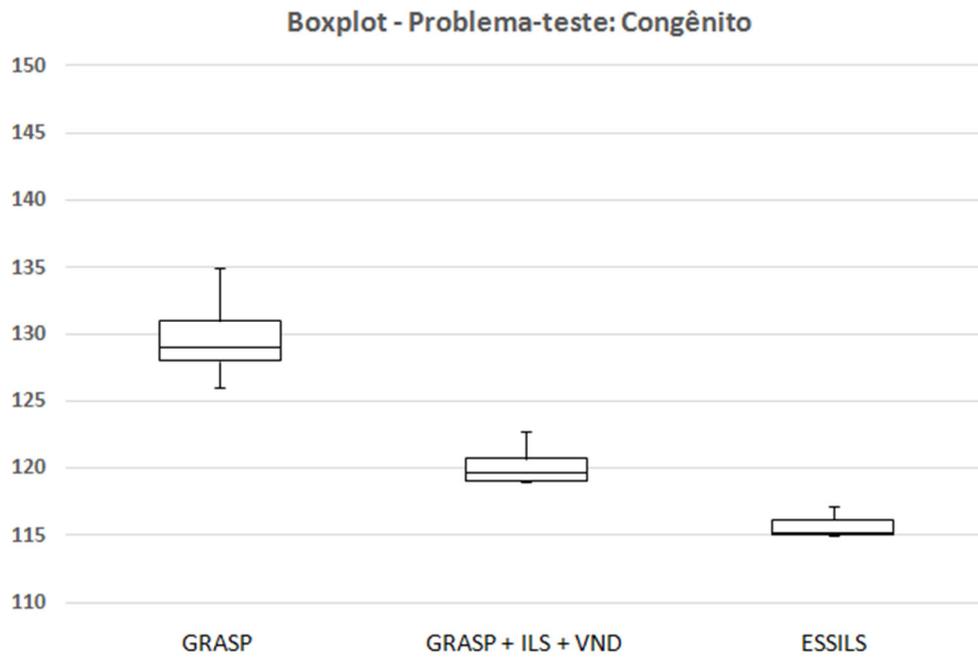


Fonte: Autor.

O Gráfico 9 apresenta o *Boxplot* relativo ao problema-teste Congênito. É possível observar que a dispersão dos resultados obtidos pelos algoritmos propostos é bem menor se comparado com os demais gráficos dos diferentes problemas-teste desta Seção. Constatou-se pontos de máximo e mínimo fora da caixa no algoritmo GRASP, mas não identificados como *outlier*. O comportamento do *Boxplot* para o GRASP foi quase simétrico, embora a mediana se posicionou levemente para os resultados de menor valor. O Algoritmo GRASP combinado com ILS e VND apresentou um gráfico simétrico e ponto de máximo fora da caixa, mas ainda assim com baixa variabilidade dos resultados da amostra. O algoritmo ESSILS apresentou linha de mediana quase que coincidente com o limite inferior (maioria dos resultados próximo ao melhor *makespan*) e um ponto de máximo fora da caixa, mas se tratando da baixa

variabilidade detectada, pode-se garantir que os resultados encontrados são muito próximos entre si. Para esta instância nenhum *outlier* foi identificado.

Gráfico 9: *Boxplot* para o problema-teste Cirurgia Congênito.



Fonte: Autor.

7 CONCLUSÕES E TRABALHOS FUTUROS

A presente dissertação abordou a solução do Problema de Sequenciamento de Cirurgias Eletivas por meio de métodos computacionais heurísticos. Assim como no Problema de Sequenciamento de Máquinas Paralelas, porém na ótica do planejamento cirúrgico, o Problema de Sequenciamento de Cirurgias Eletivas objetiva realizar o agendamento das cirurgias previamente conhecidas (por isso o fato de serem nomeadas de eletivas) de acordo com os recursos humanos e físicos do hospital com o intuito de minimizar o *makespan*, ou seja, o instante de término da última cirurgia. Considerou-se para esta dissertação recursos hospitalares tais estes como salas de cirurgia, salas de UTI, equipamentos cirúrgicos, cirurgiões, anestesistas, enfermeiros, entre outros.

Para resolver o problema combinatório em questão foi proposto o algoritmo denominado de ESSILS (*Elective Surgery Scheduling Iterated Local Search*), o qual combinou uma série de métodos heurísticos. O método GRASP foi utilizado na fase de geração da solução inicial e, na etapa de refinamento, utilizou-se o método ILS combinado com dois métodos de busca local: VND e Busca Tabu. Por meio da análise dos resultados experimentais e validação estatística, foi possível identificar a contribuição individual de cada algoritmo implementado. Conclui-se, portanto, que a combinação dos métodos proposta no algoritmo ESSILS contribuiu para a produção de melhores resultados em relação à utilização exclusiva do algoritmo GRASP e do algoritmo que combinou GRASP com ILS tendo somente o VND como busca local. Esses resultados mostram que a inclusão do método Busca Tabu como busca local do ILS foi totalmente justificável, uma vez que trouxe melhoria dos resultados obtidos.

Por todas as análises estatísticas realizadas, verificou-se que ESSILS se mostrou o melhor algoritmo, tanto na capacidade de gerar soluções de qualidade como na variabilidade dos resultados obtidos. O teste de desempenho por meio dos gráficos de probabilidade empírica mostrou que o algoritmo ESSILS alcançou o valor alvo no menor tempo. Por outro lado, o teste de t de *Student*, assim como os gráficos *Boxplot*, comprovaram a baixa variabilidade entre os resultados gerados.

Os algoritmos propostos foram testados em diversos problemas-teste da literatura, sendo eles 92 de Sperandio (2015), 60 de Andrade (2012), 9 de Daiki e Yih (2010) e 9 a partir do problema real de um hospital de grande porte de São Paulo. De acordo com os resultados computacionais obtidos, o algoritmo ESSILS se mostrou muito eficiente. Ele superou resultados nas instâncias de Sperandio (2015) e Andrade (2012) quando comparado com os

métodos propostos por esses dois autores. Quando não superado, ESSILS foi capaz de encontrar os mesmos resultados de Sperandio e Andrade (2015) na maioria dos problemas-teste. Apenas em 3 problemas-teste de Andrade (2012) ESSILS não foi capaz de encontrar o melhor resultado da literatura. Já para as instâncias de Sperandio (2015), o algoritmo ESSILS não superou o melhor da literatura em 25 problemas-teste dentre os 92 testados. Mesmo não alcançando o melhor da literatura, ESSILS chega muito próximo, uma vez que o desvio em relação ao melhor encontrado é muito baixo. Quanto aos problemas-teste de Daiki e Yih (2010), o algoritmo ESSILS alcançou a maioria dos resultados encontrados por esses autores e, em apenas um, obteve resultado inferior ao obtido pelo método deles. Quando comparado com os dados históricos da programação cirúrgica do hospital de grande porte de São Paulo, o algoritmo ESSILS foi capaz de superar todos os resultados, melhorando o *makespan* em todos os problemas-teste. Justifica-se, portanto, sua utilização para apoiar o planejamento cirúrgico, já que obtém resultados melhores que os alcançados manualmente pela equipe de agendamentos de cirurgia.

É importante ressaltar que a proposta deste trabalho se direcionou também para o desenvolvimento de uma ferramenta prática e inteligente para a realização da programação de cirurgias eletivas. A intenção, portanto, foi apresentar um método que fosse capaz de auxiliar os tomadores de decisão responsáveis pela programação das cirurgias eletivas. Por se tratar de um método computacional, é possível garantir que o problema combinatório de alocação de cirurgias nos recursos disponíveis seja explorado de tal forma que consiga obter resultados melhores do que aqueles obtidos manualmente pela equipe de agendamentos, como pode ser observado pelos resultados obtidos no problema real do hospital de grande porte de São Paulo adotado nesta dissertação. Sabe-se também que imprevistos nos procedimentos cirúrgicos podem acontecer, assim como acontecem nas linhas de produção industriais, e uma nova reprogramação das cirurgias pode eventualmente ser necessária. É por este motivo que o resultado do algoritmo é uma sugestão de agendamentos a ser seguida e, se necessário for, uma nova programação de operações pode rapidamente ser realizada com o intuito de ajustar o agendamento face aos imprevistos dos procedimentos cirúrgicos previamente agendados. Esta é a grande vantagem do algoritmo computacional, uma vez que tem compilação rápida e flexível, gerando novos resultados em um curto período de tempo.

Conclui-se, portanto, que o modelo heurístico utilizado no algoritmo ESSILS se mostrou adequado, já que foi capaz de alcançar resultados de qualidade perante as demais abordagens da literatura, as quais foram utilizadas como critério de comparação. Os objetivos da presente dissertação também foram atingidos, uma vez que o algoritmo heurístico

implementado foi capaz de diminuir o *makespan* do planejamento cirúrgicos de diversos problemas-teste o qual este fora testado. ESSILS também se mostrou eficiente e eficaz em relação aos resultados alcançados e às validações estatísticas realizadas.

Este trabalho proporcionou, até então, a publicação de um artigo intitulado *Improvement of Performance Indicators in Hospitals: An Innovative Approach Through Computational Optimization* no evento científico internacional denominado de *POMS Annual Conference 2016 of Production and Operations Management*.

É importante ressaltar que aperfeiçoamentos ainda podem ser feitos no algoritmo proposto com o intuito de melhorar, ainda mais, a qualidade de seus resultados. Como trabalho futuro pretende-se aplicar outros métodos heurísticos (como é o exemplo de algoritmos genéticos e colônias de formigas) afim de comparar a performance deles entre si e a presente proposta e, por fim, aplicar também métodos matemáticos para a solução do problema de Sequenciamento de Cirurgias Eletivas.

REFERÊNCIAS

- ADAMS, J; BALAS, E.; ZAWACK, D. The shifting bottleneck procedure for job shop scheduling. **Management Science**, v. 34, p. 391-401, 1988.
- AIEX, R. M.; RESENDE, M. G. C.; RIBEIRO, C. C. Probability distribution of solution time in GRASP: an experimental investigation. **Journal of Heuristics**, v. 8, p. 343-373, 2002.
- ANDRADE, G. P. **Estudo de Metaheurísticas Aplicadas ao Problema de Agendamento de Cirurgias Eletivas**. Dissertação de Mestrado - Programa de Pós-graduação em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2012.
- ARMENTANO, V. A.; BRANCHINI, R. M. Uma introdução à Busca Tabu. In: **Meta-Heurísticas em Pesquisa Operacional**. [s.L]: Editora Omnipax, 2013, v. 1, c. 3, p. 33-52.
- BAM, M. et al. Surgery scheduling with recovery resources. **INFORMS Healthcare Conference**, v.3, p. 231-240, 2015.
- BELIËN, J.; DEMEULEMEESTER, E. A branch-and-price approach for integrating nurse and surgery scheduling. **European Journal of Operational Research**, v. 189, p. 652-668, 2008.
- BENJAMINI, Y. Opening the box of a boxplot. **The American Statistician**, v. 42, p. 257-262, 2012.
- BRUCKER, P.; JURISCH, B.; SIEVERS, B. A branch and bound algorithm for the job-shop scheduling problem. **Discrete Applied Mathematics**, v. 49, p. 107-127, 1994.
- BUENO, F. **Métodos heurísticos: teoria e implementações**. Instituto Federal de Santa Catarina, Florianópolis, 2009.
- BÜLBÜL, K. A hybrid shifting bottleneck-tabu search heuristic for the job shop total weighted tardiness problem. **Computers & Operations Research**, v. 38, p. 967-983, 2011.
- CARDOEN, B.; DEMEULEMEESTER, E. E BELIËN, J. Operating room planning and scheduling: a literature review. **European Journal of Operational Research**, v. 3, p. 333-333, 2010.
- CARTER, M. W.; TOVEY, C. A. When is the classroom assignment problem hard? **Operations Research**, v. 40, n. 1, p.28-30, 1992.
- CHAABANE, S. et al. Comparison of two methods of operating theatre planning: application in Belgian Hospital. **Systems Engineering Society of China & Springer-Verlag**, v. 17, p. 171-186, 2008.
- CHENG, R.; GEN, M.; TSUJIMURA, Y. A tutorial survey of job-shop scheduling problems using genetic algorithms. **Computers & Industrial Engineering**, v. 30, p. 983-997, 1996.

- DAIKI, M.; YIH, Y. Scheduling elective surgery under uncertainty and downstream capacity constraints. **European Journal of Operational Research**, v. 206, p. 642-652, 2010.
- DENTON, B.; VIAPIANO, J.; VOLG, A. Optimization of surgery sequence and scheduling decision under uncertainty. **Health Care Management Science**, v. 10, p. 13-24, 2007.
- DEXTER, F.; TRAUB, R. D. How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. **Economics and Health Systems Research**, v. 94, p. 933-942, 2002.
- DOANE, D. P.; SEWARD, L. E. Testes de hipóteses baseados em uma amostra. In: **Estatística Aplicada à Administração e Economia**. São Paulo: McGraw-Hill, 4. ed., 2014.
- DORIGO, M.; STÜTZLE, T. Ant colony optimization. **MIT Press Journals**, Cambridge, MA, EUA, 2004.
- FEI, H.; MESKENS, N.; CHU, C. A planning and scheduling problem for an operating theatre using an open scheduling strategy. **Computers & Industrial Engineering**, v. 58, p. 221-230, 2010.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v. 6, p. 109-133, 1995.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, v. 13, p. 533-549, 1986.
- GLOVER, F.; LAGUNA, M. **Tabu Search**. Boston: Kluwer Academic Publisher, 1997.
- GUO, M. et al. Integrated scheduling of elective surgeries and surgical nurses for operating room suites. **Springer Science Business**, New York, v.28, p. 166-181, 2014.
- HADDAD, M. N. **Algoritmos Heurísticos Híbridos para o Problema de Sequenciamento em Máquinas Paralelas Não-Relacionadas com Tempo de Preparação Dependentes da Sequência**. Dissertação de Mestrado - Programa de Pós-graduação em Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2012.
- HERTZ, A.; WIDMER, M. Guidelines for the use of meta-heuristics in combinatorial optimization. **European Journal of Operational Research**, v. 151, p. 247-252, 2003.
- HUANG, G. et al. Surgical scheduling based on hybrid flow-shop scheduling. **Third International Conference on Engineering Design and Optimization**, v. 201-202, p. 1004-1007, 2012.
- JAIN, A. S.; MEERAN, S. Deterministic job-shop scheduling: past, present and future. **European Journal of Operational Research**, v. 113, p. 390-434, 1999.
- JEBALI, A.; HADJALOUANE, A.; LADET, P. Operating rooms scheduling. **International Journal Production Economics**, v. 99, p. 52-62, 2006.

KHANNA, S. et al. An intelligent approach to surgery scheduling. **Lecture Notes in Computer Science**, v. 7057, p. 535-550, 2012.

KIM, S.; HOROWITZ, I. Scheduling hospital services: the efficacy of elective-surgery quotas. **International Journal of Management Science**, v. 30, p. 335-346, 2002.

LEUNG, J. Introduction and notation. In: _____. **Handbook of scheduling: algorithms, models, and performance analysis**. Boca Raton: CRC Press, 2004.

LOURENÇO, H. R.; MARTIN, O.; STÜTZLE, T. Iterated Local Search. Handbook of Metaheuristics. Kluwer Academic Publishers, **International Series in Operations Research & Management Science**, v. 57, p. 321-353, 2003.

MACARIO, A. et al. Where are the costs in perioperative care: analysis of hospital costs and charges for inpatient surgical care. **Anesthesiology**, v. 83, p.1138-1144, 1995.

MARCHESI, J. F. **Planejamento da programação de salas de cirurgia**: aplicação ao caso de hospital público. Dissertação de Mestrado - Programa de Pós-graduação em Engenharia de Produção, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2015.

MLADENOVIC, N.; HANSEN, P. A Variable Neighborhood Search. **Computers & Operations Research**, v. 24, p. 1097-1100, 1997.

MANCILLA, C.; STORER, R. H. Stochastic sequencing of surgeries for a single surgeon operating in parallel operating rooms. **Industrial and Systems Engineering**, Lehigh University, 2011.

MARQUES, I; CAPTIVO, M. E.; PATO, M. V. An integer programming approach to elective surgery scheduling. **OR Spectrum**, v. 34, p. 407-427, 2012.

MARTÍ, R.; REINELT, G. **The linear ordering problem: exact and heuristic methods in combinatorial optimization**. Editora Springer-Verlag Berlin Heidelberg, v. 175, 2011.

MAY, J. H.; SPANGLER, W. E.; STRUM, D. P.; VARGAS L. G. The surgical scheduling problem: current research and future opportunities. **Production and Operations Management**, v. 20, p. 392-405, 2011.

MIN, D.; YIH, Y. An elective surgery scheduling problem considering patient priority. **Computers & Operations Research**, v. 37, p. 1091-1099, 2010.

MINE, M. T. **Um algoritmo heurístico híbrido para o problema de roteamento de veículos com coleta e entrega simultânea**. Dissertação de Mestrado - Programa de Pós-graduação em Ciência da Computação, Universidade Federal Fluminense, Rio de Janeiro, 2009.

NASHEFA, S. A. M. et al. EuroSCORE II. **European Journal of Cardio-Thoracic Surgery**, v. 41, p. 734-745, 2012.

OOSTRUM, J. M. et al. A master surgical scheduling approach for cyclic scheduling in operating room departments. **OR Spectrum**, v. 30, p. 355-374, 2008.

PHAM, D. N.; KLINKERT, A. Surgical case scheduling as a generalized job-shop scheduling problem. **European Journal of Operational Research**, v. 185, p. 1011-1025, 2008.

PINEDO, M. L. **Planning and Scheduling in Manufacturing and Services**. Springer, New York, 2005.

PINEDO, M. L. **Scheduling: theory, algorithms, and systems**. Springer, New York, 2008.

PROENÇA, I. M. **Planeamento de Cirurgias Electivas**. Tese de Doutoramento - Universidade de Lisboa, Portugal, 2010.

SAADOULI, H. et al. An optimization and simulation approach for Operating room scheduling under stochastic durations. **Control, Decision and Information Technologies**, p. 257-262, 2014.

SILVA, T. C. B. **GENILS-TS-CL-PR**: um algoritmo heurístico para resolução do Problema de Roteamento de Veículos com Coleta e Entrega Simultânea. Dissertação de Mestrado - Programa de Pós-graduação em Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2012.

SOUZA, M. J. F. Busca Tabu. In: **Manual de Computação Evolutiva e Metaheurística**. Editora da Universidade Federal de Minas Gerais, 2012, v. 2, p. 177-202.

SOUZA, M. J. F. **Inteligência computacional para otimização: notas de aula**. Departamento de Computação, Universidade Federal de Ouro Preto, 2009. Disponível em: <<http://www.decom.ufop.br/marcone/Disciplinas/InteligenciaComputacional/Introducao.pdf>> Acesso em: 15 de setembro de 2016.

SPERANDIO, F. R. **Large scale elective surgery scheduling under uncertainty**. Tese de Doutoramento - Universidade do Porto, Faculdade de Engenharia, Portugal, 2015.

STÜTZLE, T.; HOOS, H. H. Analyzing the run-time behaviour of iterated local search for the TSP. **Third Metaheuristics International Conference**, p. 449-453, 1999.

SUBRAMANIAN, A. et al. Um Algoritmo Heurístico Baseado em Iterated Local Search para Problemas de Roteamento de Veículos. In: **Meta-Heurísticas em Pesquisa Operacional**. [s.L]: Editora Omnipax, 2013, v. 1, c. 11, p. 167-180.

VONDER, S. V.; DEMEULEMEESTER, E.; HERROELEN, W. Proactive heuristic procedures for robust project scheduling: An experimental analysis. **European Journal of Operational Research**, v. 189, p. 723-733, 2008.

YANG, Y. et al. A surgical scheduling method considering surgeon's preferences. **Springer Science Business**, New York, 2015.

ZHONG, L. et al. A two-stage approach for surgery scheduling. **Springer Science Business**, v. 27, p. 545-556, 2012.