

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

TATIANA COSTA MEISTER

**Um algoritmo heurístico aplicado ao
problema de escalonamento de médicos do
Hospital de Clínicas de Porto Alegre**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência da
Computação

Orientadora: Prof. Dra. Luciana Buriol

Porto Alegre
2020

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Meister, Tatiana Costa

Um algoritmo heurístico aplicado ao problema de escalonamento de médicos do Hospital de Clínicas de Porto Alegre / Tatiana Costa Meister. – Porto Alegre: PPGC da UFRGS, 2020.

57 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2020. Orientadora: Luciana Buriol.

1. Problema de Escalonamento de Médicos. 2. Heurísticas. 3. Variable Neighbourhood Search. 4. Branch & Bound. I. Buriol, Luciana. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof^a Luciana Salete Buriol

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

O escalonamento de médicos consiste em definir em quais áreas e turnos do dia cada médico irá atuar para suprir as demandas de um hospital. Essa atividade geralmente é realizada mensalmente e deve obedecer dados contratuais, leis trabalhistas e demandas por serviços de saúde da comunidade. A escolha da escala impacta nas despesas do hospital e na relevância dada às preferências pessoais de cada médico.

Devido à grande quantidade de escalas que podem ser combinadas entre os médicos, definir uma solução adequada demanda tempo e experiência do responsável por essa atividade. Entretanto, o tempo disponível é limitado e executar essa tarefa manualmente tende a acarretar em uma avaliação incompleta das soluções, assim como prover resultados de baixa qualidade. A principal vantagem na aplicação de algoritmos neste tipo de problema é tornar o processo mais ágil, fornecendo soluções melhores do que as do processo manual.

Este trabalho propõe uma heurística *Variable Neighborhood Search* (VNS) para solucionar o problema de escalonamento de médicos aplicado ao Hospital de Clínicas de Porto Alegre (HCPA). O algoritmo de VNS é comparado com a execução de uma formulação exata via o solver COIN Branch and Cut (CBC). A comparação é conduzida em três tipos de instâncias, geradas com base nos dados fornecidos pelo HCPA e classificadas por demanda e disponibilidade dos médicos.

Palavras-chave: Problema de Escalonamento de Médicos. Heurísticas. Variable Neighbourhood Search. Branch & Bound.

A heuristic algorithm applied to the physician rostering problem at *Hospital de Clínicas de Porto Alegre*

ABSTRACT

The physician rostering problem consists of defining which areas and shifts of the day each physician will be assigned to meet the demands of a hospital. This activity is usually done monthly and must comply with contractual data, labor laws and demands for community health services. The choice of an assignment impacts hospital expenses and the relevance given to the personal preferences of each physician.

Due to the large number of assignments that can be combined among physicians, defining an appropriate solution takes time and experience from the responsible for this activity. However, the time available is limited and performing this task manually tends to result in an incomplete evaluation of the solutions, as well as providing low-quality results. The main advantage of applying algorithms in this type of problem is to make the process more agile, providing better solutions than those of the manual process.

This paper presents a heuristic of Variable Neighborhood Search (VNS) to solve a variant of the physician rostering problem applied to *Hospital de Clínicas de Porto Alegre* (HCPA). The VNS algorithm is compared to the execution of an exact formulation via the COIN Branch and Cut (CBC) solver. The comparison is conducted in three types of instances, generated based on data provided by HCPA and classified by demand and availability of physicians.

Keywords: Physician Rostering Problem, Heuristics, Variable Neighbourhood Search, Branch & Bound.

LISTA DE ABREVIATURAS E SIGLAS

HCPA Hospital de Clínicas de Porto Alegre

B&B_T Branch & Bound Truncado

B&B_T^{CO} Branch & Bound Truncado Com Mecanismos de Ordenação

B&B_T^{SO} Branch & Bound Truncado Sem Mecanismos de Ordenação

FM Fase de Melhoramento da Solução Inicial

VNS *Variable Neighborhood Search*

CBC Solver COIN Branch and Cut

LISTA DE FIGURAS

Figura 3.1 Exemplo de uma instância e um escalonamento factível para o problema de escalonamento de médicos com uma janela de 2 dias.	20
Figura 4.1 Fluxo de execução do algoritmo proposto	27
Figura 4.2 Exemplo de uma árvore de soluções do B&B _T	30
Figura 4.3 Exemplo da ordenação das alocações de cada médico em um dia do mês....	32
Figura 4.4 Exemplo de uma modificação válida na solução corrente do VNS pelo tipo de vizinhança <i>CHANGE</i> para um escalonamento de 3 dias.	37
Figura 4.5 Exemplo de uma modificação inválida na solução corrente do VNS pelo tipo de vizinhança <i>CHANGE</i> para um escalonamento de 3 dias.	37
Figura 4.6 Exemplo de uma modificação na solução corrente do VNS pelo tipo de vizinhança <i>SWAP</i> para um escalonamento de 3 dias.	38
Figura 4.7 Exemplo de uma modificação inválida na solução corrente do VNS pelo tipo de vizinhança <i>SWAP</i> para um escalonamento de 3 dias.	38
Figura 4.8 Exemplo de uma modificação na solução corrente do VNS pelo tipo de vizinhança <i>3-CHAIN</i> para um escalonamento de 3 dias.....	39
Figura 4.9 Exemplo de uma modificação inválida na solução corrente do VNS pelo tipo de vizinhança <i>3-CHAIN</i> para um escalonamento de 3 dias.....	40
Figura 5.1 Exemplo de uma instância no formato de entrada gerado	44

LISTA DE TABELAS

Tabela 3.1	Comparação das restrições aplicadas entre as diferentes versões do problema do HCPA e o trabalho atual.....	18
Tabela 3.2	Índices dos dados de entrada utilizados pela formulação matemática	22
Tabela 3.3	Variáveis de decisão utilizadas na formulação matemática.....	23
Tabela 3.4	Variáveis auxiliares utilizadas na formulação matemática	23
Tabela 5.1	Parâmetros do gerador para as instâncias utilizadas nos experimentos.....	45
Tabela 5.2	Instâncias geradas para os experimentos	45
Tabela 6.1	Tempo limite de execução do VNS e do CBC em segundos.....	46
Tabela 6.2	Resultados da solução inicial do B&B _T com uso dos mecanismos de ordenação (B&B _t ^{SO}) e sem uso de tais mecanismos (B&B _t ^{CO})	48
Tabela 6.3	Resultados da solução inicial do B&B _t ^{CO} e após fase de melhoramento (FM)	50
Tabela 6.4	Resultados da solução inicial após fase de melhoramento (FM) com B&B _t ^{SO} e com B&B _t ^{CO}).....	51
Tabela 6.5	Resultados do VNS e CBC.....	53

SUMÁRIO

1 INTRODUÇÃO	9
2 REVISÃO DA LITERATURA.....	10
3 PROBLEMA DE ESCALONAMENTO DE MÉDICOS	14
3.1 Informações de uma Instância do Problema	14
3.2 Restrições de Factibilidade.....	15
3.3 Restrições de Qualidade	16
3.4 Comparação entre Versões do Problema do HCPA.....	18
3.5 Exemplo de um Escalonamento	19
3.6 Formulação Matemática.....	21
4 HEURÍSTICA VNS PROPOSTA	27
4.1 Implementação e Estrutura de Dados.....	27
4.2 Construção da Solução inicial.....	29
4.2.1 Branch & Bound Truncado	29
4.2.1.1 Ordem de Processamento dos Médicos	31
4.2.1.2 Ordem de Processamento das Alocações de um Médico em um Dia.....	31
4.2.1.3 Exemplo do Processo de Ordenação das Alocações	31
4.2.2 Fase de Melhoramento	32
4.3 Heurística VNS.....	35
4.3.1 Algoritmo VNS proposto	35
4.3.2 Tipos de Vizinhança.....	36
4.3.2.1 Vizinhança <i>CHANGE</i>	37
4.3.2.2 Vizinhança <i>SWAP</i>	38
4.3.2.3 Vizinhança <i>3-CHAIN</i>	39
4.3.3 Conclusões sobre o Algoritmo Proposto.....	40
5 GERADOR DE INSTÂNCIAS	41
5.1 Parâmetros de Entrada	41
5.1.1 Limite Superior de Médicos Demandado por Dia/Turno/Área	42
5.2 Formato do Arquivo de Entrada	43
5.3 Instâncias Utilizadas nos Experimentos	44
6 EXPERIMENTOS COMPUTACIONAIS	46
6.1 Ambiente de Execução e Configurações	46
6.2 Análise sobre a Construção da Solução Inicial	47
6.2.1 Qualidade do B&B Truncado x Mecanismos de Ordenação	47
6.2.2 Qualidade do B&B Truncado x Fase de Melhoramento	49
6.2.3 Conclusões sobre a Construção da Solução Inicial.....	52
6.3 Análise sobre VNS e CBC	52
6.3.1 Conclusões sobre o VNS Proposto	54
7 CONCLUSÕES	55
REFERÊNCIAS.....	56

1 INTRODUÇÃO

O escalonamento de tarefas consiste em definir em quais áreas e horários cada profissional irá trabalhar ao longo de um período. O uso de algoritmos no escalonamento de médicos tem ganhado atenção nos últimos anos, devido ao aumento na demanda por saúde (RAIS; VIANA, 2011). O escalonamento de médicos é similar ao de enfermeiras, diferenciando-se pela maior importância dada aos acordos individuais (BRUNNER; BARD; KOLISCH, 2009).

Definir uma escala, na maioria dos hospitais, ainda é uma atividade manual executada por um médico com maior experiência. O processo manual onera o profissional em uma atividade não diretamente relacionada a sua especialidade (BRUNNER; EDENHARTER, 2011), assim como resulta em soluções de baixa qualidade, dado o grande volume de combinações possíveis a serem analisadas.

O problema gera um número exponencial de soluções que se diferenciam por fatores de qualidade determinados por cada hospital. Por exemplo, um hospital que busca reduzir os gastos irá preferir soluções com o menor número de horas alocadas. Outra instituição pode buscar garantir as preferências individuais dos médicos. Habitualmente o problema de escalonamento é uma combinação de diversos objetivos, tanto voltados ao hospital quanto aos próprios médicos.

O uso de ferramentas automatizadas para gerar o escalonamento torna o processo menos oneroso e fornece soluções de melhor qualidade. O presente trabalho tem como objetivo comparar o uso de uma heurística e de um método exato sobre diferentes grupos de instâncias geradas com base nas características do Hospital de Clínicas de Porto Alegre (HCPA). Este trabalho foi publicado em (MEISTER; WICKERT; BURIOL, 2020), no qual foi considerado um conjunto reduzido de instâncias.

O restante do trabalho está organizado da seguinte forma. A Seção 2 apresenta a revisão da literatura sobre o problema e sobre o VNS. A Seção 3 define o problema de escalonamento de médicos, apresentando as restrições consideradas, as diferenças com relação às versões anteriores e a formulação matemática. A Seção 4 apresenta a heurística VNS proposta para resolver o problema do HCPA, incluindo a etapa de construção da solução inicial e os tipos de vizinhança utilizados. A Seção 5 detalha as características das instâncias utilizadas nos testes, assim como informações da implementação do gerador de instâncias. A Seção 6 apresenta os experimentos realizados entre o VNS proposto e o método exato. Finalmente, a Seção 7 traz as conclusões do estudo deste trabalho.

2 REVISÃO DA LITERATURA

Neste capítulo será abordada a literatura existente sobre o problema de escalonamento de médicos e sobre a origem e características da heurística *Variable Neighborhood Search*. Serão apresentadas as diferentes características consideradas no escalonamento de médicos em tais trabalhos e quais métodos foram utilizados para solucionar o problema.

No trabalho de (ERHARD et al., 2018) é fornecida uma revisão de 60 publicações sobre o escalonamento de médicos. São analisadas as características que são comuns na literatura, assim como realizada uma análise sobre o aumento na pesquisa sobre o problema de médicos ao longo das décadas. De acordo com os autores, o aumento do interesse nesse tópico é devido à importância na eficiência do escalonamento de funcionários para a redução de custos dos hospitais. Essa melhor eficiência se deve ao fato do escalonamento automatizado explorar melhor o espaço de busca, sem ser necessário reduzir a qualidade dos serviços prestados. Os autores dividem a literatura em três categorias:

1. Problemas de Força de Trabalho: os quais buscam uma quantidade de funcionários ideal conforme futuras demandas e estratégias de priorização de cada serviço;
2. Problemas de Escalonamento: os quais têm por objetivo identificar o escalonamento de melhor qualidade em uma janela de tempo pré-definida;
3. Problemas de Reescalonamento: os quais tratam de eventos adversos que impactam num escalonamento completo. Estes problemas tratam de alterações em alocações pontuais para que não ocorra a ruptura dos serviços.

Em (MANSINI; ZANOTTI, 2019), o principal foco é a otimização do uso dos recursos humanos, uma vez que o aumento na demanda de serviços de saúde não é proporcional ao aumento na quantidade de médicos. Os autores utilizam características da área cirúrgica do hospital de “*Spedali Civili di Brescia*”, do norte da Itália. As instâncias utilizadas são inspiradas no referido hospital e consideram uma janela de 15 dias para o escalonamento. São propostas duas formulações matemáticas que, posteriormente, são utilizadas na fase de reconstrução da heurística “*Adaptive Large Neighborhood Search*” (ALNS). Os resultados da heurística ALNS foram comparados com um modelo solucionado através do solver exato *Gurobi*, os quais apresentaram melhor qualidade no ALNS proposto pelos autores.

No trabalho de (BRUNI; DETTI, 2014) é apresentada uma formulação MIP

com execução via o solver CPLEX, considerando o problema aplicado ao Centro de Hematologia do hospital da Universidade de Roma, na Itália. Os resultados foram comparados com o escalonamento manual feito pelos responsáveis por essa atividade no hospital. Os autores ainda utilizam parametrizações que permitem ao usuário definir quais restrições deverão ser aplicadas, assim como o impacto dessas no custo final da solução. No presente trabalho foi utilizado o mesmo conceito de pesos para cada restrição, permitindo classificá-las de forma ordenada. Maiores detalhes do uso de pesos é discutido na Seção 3.3.

No estudo conduzido por (GUNAWAN; LAU, 2013) foi implementada uma formulação MIP com execução via solver CPLEX, considerando o caso do hospital do governo de Singapura. O principal foco dos autores se concentra na retenção dos profissionais através de uma abordagem onde é dada maior importância na qualidade de trabalho para os médicos. São propostos três modelos matemáticos que se complementam. O primeiro modelo busca minimizar o número de tarefas que ainda não atingiram o mínimo de médicos necessários. O segundo modelo busca minimizar a diferença entre o escalonamento do primeiro modelo e o escalonamento ideal de cada médico baseado em suas preferências e carga horária. Já o terceiro modelo busca balancear a carga de trabalho dos médicos. Para as instâncias menores, o modelo matemático executado via CPLEX soluciona otimamente o problema, porém o mesmo não foi reproduzível para as instâncias maiores. A fim de solucionar as instâncias com maior quantidade de médicos dentro do tempo limite estipulado, os autores implementaram uma heurística de busca local com uso de vizinhanças.

O trabalho de (ADAMS; O’SULLIVAN; WALKER, 2019) foca na redução da fragmentação do tratamento dos pacientes, ou seja, busca evitar que mais de um médico faça o acompanhamento do mesmo paciente durante sua permanência no hospital. Os autores se inspiraram em um caso real do hospital de Auckland, da Nova Zelândia e propuseram um modelo matemático. Este é um dos poucos trabalhos em que o foco está no atendimento dos pacientes.

No trabalho de (GENDREAU et al., 2006) é apresentado um problema de escalonamentos de médicos baseado nas restrições de cinco diferentes hospitais de Montreal, no Canadá. Para solucioná-lo, os autores aplicaram uma busca tabu e uma programação inteira. Os autores ainda propõem quatro classificações de uso para as restrições do problema, conforme análise conduzida nos diferentes hospitais. Sendo estas:

1. Restrições de oferta e demanda: que avaliam os requisitos do hospital em cada área;

2. Restrições de carga horária: que avaliam a quantidade de horas trabalhadas por cada profissional de acordo com seu contrato de trabalho;
3. Restrições de justiça: que buscam balancear as alocações menos desejáveis entre todos os médicos, de modo que não fiquem acumuladas em apenas alguns profissionais;
4. Restrições de ergonomia: que buscam aumentar a qualidade do trabalho dos funcionários, reduzindo alocações que são prejudiciais.

Embora cada hospital avaliado na literatura possua características diversas entre si, o objetivo da maioria de suas restrições se encaixa em alguma das classificações propostas por (GENDREAU et al., 2006). Isso permite uma melhor comparação entre a função objetivo dentre os diversos problemas da literatura.

Os autores em (SCHOENFELDER; PFEFFERLEN, 2018) apresentam a aplicação de uma formulação inteira com execução via CPLEX. O problema considera uma janela de um mês e características do departamento de anestesiologia de um hospital de Berlim, na Alemanha. O escalonamento no hospital era realizado manualmente pelo membro com maior experiência da equipe médica. O modelo matemático proposto pelo autores passou a ser utilizado pelo hospital junto de ferramentas que auxiliam em modificações pontuais na solução no dia-a-dia. A qualidade da solução é focada na redução de horas extras, na alocação dos médicos conforme suas preferências e critérios de justiça.

O problema do Hospital de Clínicas de Porto Alegre (HCPA) já foi estudado anteriormente em (WICKERT et al., 2020), no qual são propostos dois modelos de restrições: um básico e um estendido. Os autores propuseram uma formulação inteira (MIP) resolvida através do CPLEX e de um algoritmo de *Fix and Optimize* (F&O). O algoritmo MIP resolveu de forma ótima instâncias de menor tamanho em poucos segundos, enquanto para instâncias maiores a meta-heurística gerou melhores resultados.

O trabalho de (SANCHOTENE; BURIOL, 2018) considera características semelhantes às do modelo básico de restrições do problema tratado em (WICKERT et al., 2020). Para essa variante, é disponibilizada uma heurística de *Late Acceptance Hill Climbing* (LAHC) cujos resultados foram similares aos do F&O, com uma pequena melhora de qualidade para instâncias de maior quantidade de médicos.

Todas as diferenças entre as variantes do problema do HCPA conduzidas nos trabalhos anteriores e no presente trabalho serão detalhadas na Seção 3.4.

O problema de escalonamento de médicos considerado neste trabalho tem por objetivo identificar o escalonamento de melhor qualidade dentro de um limite de tempo

para o problema aplicado ao HCPA. Não é previsto o cenário de reescalonamento ou identificação de um número ótimo de médicos a serem utilizados, que correspondem às categorias 1 e 3 propostas por (ERHARD et al., 2018).

O algoritmo de *Variable Neighborhood Search* (VNS) foi inicialmente proposto em (MLADENOVIC; HANSEN, 1997) e (HANSEN; MLADENOVIC, 1997). Os autores ainda publicaram outros trabalhos sobre a heurística, tais como os princípios e aplicações em (HANSEN; MLADENOVIC, 2001).

O VNS é amplamente utilizado na literatura em uma grande diversidade de problemas de otimização, tais como: o escalonamento de enfermeiras em (BURKE et al., 2003), a coloração de grafos em (AVANTHAY; HERTZ; ZUFFEREY, 2003), o roteamento de veículos em (KYTÖJOKI et al., 2007), o mapeamento de redes virtuais em (INFÜHR; RAIDL, 2013), entre outros.

No presente trabalho foi implementada uma heurística VNS cujos resultados foram comparados com a resolução via um método exato. A análise dos resultados foi conduzida para diferentes tipos de instâncias, classificadas por demanda e disponibilidade de médicos, geradas através da implementação de um gerador de instâncias baseado nas características do HCPA.

3 PROBLEMA DE ESCALONAMENTO DE MÉDICOS

O problema de escalonamento de médicos consiste em definir para cada dia do mês se o médico terá folga ou irá trabalhar. No caso do médico ser alocado para trabalho, se faz necessário determinar a área e o turno do dia no qual irá atuar.

O escalonamento também deve considerar um conjunto de restrições que variam de acordo com cada hospital. Estas restrições podem ser oriundas de leis trabalhistas, dados contratuais, normas do próprio hospital, preferências pessoais de cada médico, entre outros. Independentemente do hospital, todas as restrições podem ser divididas em duas categorias:

- **Restrições de Factibilidade (*Hard Constraints*):** aquelas que obrigatoriamente devem ser respeitadas. Caso contrário, a solução se torna infactível.
- **Restrições de Qualidade (*Soft Constraints*):** as quais são desejáveis de serem atendidas. Entretanto, quando são violadas, associam um custo ao escalonamento que irá reduzir a qualidade desta solução.

No presente trabalho foram replicadas as características e restrições do Hospital de Clínicas de Porto Alegre (HCPA). O escalonamento tem por objetivo determinar as alocações de cada médico dentro de um mês, considerando uma quantidade fixa de áreas e os turnos.

As próximas seções são organizadas conforme a seguir. A Seção 3.1 detalha as informações de uma instância do problema do HCPA. A Seção 3.2 define as restrições de factibilidade, enquanto a Seção 3.3 apresenta as restrições de qualidade. A Seção 3.4 apresenta uma comparação das restrições aplicadas neste trabalho com relação as variantes do mesmo problema aplicado ao HCPA nos trabalhos de (SANCHOTENE; BURIOL, 2018) e (WICKERT et al., 2020). A Seção 3.5 apresenta um exemplo de instância e de escalonamento factível, detalhando como são penalizadas as restrições de qualidade violadas. Finalmente na Seção 3.6 é apresentada a formulação matemática para a variante do problema tratada neste trabalho.

3.1 Informações de uma Instância do Problema

Uma instância do problema consiste em um conjunto de informações que devem ser observadas durante a construção de um escalonamento. São considerados os dados

abaixo para uma instância do problema:

- Competência, indicando o mês e o ano.
- Lista de feriados: dias do mês que serão considerados como dias não úteis, além dos sábados e domingos.
- Lista de áreas do hospital, sendo definido para cada área:
 - Um identificador único e uma descrição;
 - Uma quantidade mínima e máxima de médicos que devem ser alocados para cada dia/turno.
- Lista dos médicos do hospital, sendo definido para cada médico:
 - Um identificador único e um nome;
 - Uma quantidade de horas a serem trabalhadas no mês;
 - Uma quantidade de horas a serem trabalhadas no mês em dias não úteis;
 - Uma lista das áreas em que tem permissão para ser alocado;
 - Uma lista de áreas cuja alocação deve ser evitada (não preferencial);
 - Uma lista de alocações fixas em que o médico obrigatoriamente deve ser alocado no respectivo dia/turno/área indicados;
 - Uma lista de ausências, em que o médico não pode ser alocado no respectivo dia/turno indicados;
 - Uma lista de dias/turnos cuja alocação deve ser evitada (não preferencial).

O problema possui informações recorrentes, que sempre serão iguais para todas as instâncias como, por exemplo, os turnos para alocação: manhã, tarde e noite. Os turnos da manhã e da tarde têm uma duração de seis horas. Já o turno da noite é considerado como um período de “*plantão*”, possuindo doze horas de duração. Outros dados recorrentes constam nas próprias restrições do problema que serão discutidas a seguir.

3.2 Restrições de Factibilidade

As restrições de factibilidade ou *hard constraints* são aquelas que sempre devem ser respeitadas por um escalonamento para que seja considerado válido ou factível. Neste trabalho existem oito restrições deste tipo:

- **H1 - Demanda mínima:** a quantidade de médicos alocados deve ser igual ou su-

perior à demanda mínima em cada dia/turno/área.

- **H2 - Demanda máxima:** a quantidade de médicos alocados deve ser igual ou inferior à demanda máxima em cada dia/turno/área.
- **H3 - Autorização nas áreas:** um médico não pode ser alocado em uma área que não tenha permissão.
- **H4 - Ausências:** um médico não pode ser alocado em um dia/turno em que está ausente.
- **H5 - Alocações fixas:** um médico deve ser alocado em um dia/turno/área em que possui alocação fixa.
- **H6 - Turnos em dias úteis:** um médico deve ser alocado somente em um dos turnos (manhã, tarde ou noite) em dias úteis.
- **H7 - Turnos e áreas em dias não úteis:** um médico deve ser alocado numa única área e somente no turno da noite ou em ambos os turnos do dia (manhã e tarde) em dias não úteis.
- **H8 - Sucessão de turnos proibidos:** um médico alocado no turno da noite não pode ser alocado no turno da manhã ou da tarde do dia imediatamente posterior.

3.3 Restrições de Qualidade

As restrições de qualidade ou *soft constraints* são aquelas que, quando violadas, adicionam um custo sobre a solução, reduzindo sua qualidade. Neste trabalho são consideradas dez restrições:

- **S1 - Mínimo de horas na competência:** a penalização ocorre quando o total de horas alocadas ao médico é inferior à sua carga horária.
- **S2 - Máximo de horas na competência:** a penalização ocorre quando o total de horas alocadas ao médico é superior à sua carga horária.
- **S3 - Mínimo de horas em dias não úteis:** a penalização ocorre quando o total de horas alocadas ao médico em dias não úteis é inferior ao ideal de horas em dias não úteis.
- **S4 - Máximo de horas em dias não úteis:** a penalização ocorre quando o total de horas alocadas ao médico em dias não úteis é superior ao ideal de horas em dias não úteis.

- **S5 - Equilíbrio de horas diurnas e noturnas em dias não úteis:** a penalização ocorre quando o total de horas alocadas ao médico em dias não úteis nos turnos do dia (manhã e tarde) é distinto do total de horas alocadas no turno da noite.
- **S6 - Final de semana incompleto:** a penalização ocorre quando o médico é alocado em somente um dos dias do final do semana (sábado ou domingo, mas não em ambos).
- **S7 - Máximo de finais de semana:** a penalização ocorre quando a quantidade de finais de semana alocados ao médico é superior a dois. Para que um final de semana seja considerado como trabalhado, o médico deve estar alocado em pelo menos um dos dias (sábado ou domingo).
- **S8 - Máximo de noites consecutivas:** a penalização ocorre quando o médico possui mais de três dias consecutivos alocados no turno da noite.
- **S9 - Áreas não preferenciais:** a penalização ocorre quando o médico é alocado em uma área não preferencial, independente do dia/turno da alocação.
- **S10 - Dias/turnos não preferenciais:** a penalização ocorre quando o médico é alocado em um dia/turno não preferencial, independente da área da alocação.

Cada restrição descrita acima é avaliada independentemente das demais e possui um peso de penalização associado. Este peso permite distinguir a qualidade entre dois escalonamentos que sofreram violações em restrições distintas.

Por exemplo, suponhamos que existem dois escalonamentos factíveis. O primeiro possui uma penalização da restrição $S1$, onde faltam 5 horas de trabalho para o médico $M1$. Já o segundo escalonamento possui uma penalização da restrição $S2$, pois o médico $M2$ excedeu 5 horas do seu contrato. Sem o peso associado às restrições, ambos escalonamentos teriam o mesmo custo. Entretanto, se for associado um peso maior à restrição $S1$, então o segundo escalonamento será o de melhor qualidade dentre os dois.

O uso dos pesos neste trabalho observou a relevância de cada restrição dada pelo HCPA. As restrições $S6$ e $S7$ têm um peso associado igual à 30 por penalização. As restrições $S1$ e $S2$ têm um peso 20. As restrições $S9$ e $S10$ têm um peso 1 e as demais tem um peso igual à 15.

O custo de uma solução está diretamente associado com sua qualidade: quando maior o custo, pior será a qualidade do escalonamento. Portanto, o objetivo foi identificar um escalonamento com o menor custo possível dentro de um limite de tempo.

3.4 Comparação entre Versões do Problema do HCPA

As restrições descritas nas Seções 3.2 e 3.3 são baseadas nas características do HCPA conforme análise realizada em 2019. Foram conduzidos outros estudos sobre o problema do HCPA ao longo dos últimos anos. Nesta seção serão comparadas as restrições aplicadas neste trabalho e nos trabalhos de (SANCHOTENE; BURIOL, 2018) e (WICKERT et al., 2020).

Tabela 3.1: Comparação das restrições aplicadas entre as diferentes versões do problema do HCPA e o trabalho atual

Restrição	T18	WB20	WE20	Atual
Demanda mínima	H	H	H	H
Demanda máxima	H	H	H	H
Autorização na área	H	H	H	H
Ausências	H	H	H	H
Alocações fixas	-	-	-	H
Turnos em dias úteis	H	H	H	H
Turnos e áreas em dias não úteis	H	H	H	H
Sucessão de turnos proibidos	H	H	H	H
Mínimo de horas na competência	S	S	S	S
Máximo de horas na competência	S	S	S	S
Mínimo de horas em dias não úteis	-	-	H	S
Máximo de horas em dias não úteis	-	-	H	S
Equilíbrio de horas dia/noite em dias não úteis	-	-	H	S
Final de semana incompleto	S	S	S	S
Máximo de finais de semana	S	S	S	S
Máximo de noites consecutivas	S	S	S	S
Áreas não preferenciais	-	-	S	S
Dias/turnos não preferenciais	S	S	S	S
Dias consecutivos trabalhados	S	S	S	-
Máximo de horas na competência em dias úteis	-	-	S	-
Equilíbrio de horas dia/noite por semana	-	-	S	-

¹(H) tratada como restrição de factibilidade (*hard constraint*), (S) tratada como restrição de qualidade (*soft constraint*) e (-) restrição não foi tratada

A Tabela 3.1 detalha todas as restrições já consideradas no problema do HCPA. Para cada estudo é utilizada a seguinte nomenclatura: (H) indica que a restrição foi tratada como factibilidade (*hard constraint*), (S) que foi tratada como qualidade (*soft constraint*) e (-) indica que não foi considerada no respectivo trabalho. A coluna “T18” indica o trabalho de (SANCHOTENE; BURIOL, 2018). As colunas “WB20” e “WE20” referenciam-se

ao estudo de (WICKERT et al., 2020), considerando o modelo básico e o modelo estendido de restrições, respectivamente. A coluna “*Atual*” indica o presente trabalho.

O presente trabalho tem maior similaridade com as restrições do modelo estendido do trabalho de (WICKERT et al., 2020). Na versão de 2018 ainda não haviam sido incluídas as penalizações sobre horas trabalhadas em dias não úteis. Estas foram incluídas na nova versão após avaliação das soluções geradas em conjunto com o responsável pelos escalonamentos do HCPA.

3.5 Exemplo de um Escalonamento

Nesta seção será apresentado um exemplo de instância e detalhada a penalização das restrições de qualidade sobre um escalonamento factível. Para melhor visualização e compreensão dos exemplos, será considerada uma janela de tempo de dois dias úteis, ao invés de um mês.

A Figura 3.1 apresenta os dados de uma instância e um possível escalonamento para esta. Não existem feriados neste período e são consideradas duas áreas: Cardiologia ($A1$) e COVID19 ($A2$). Os médicos $M1$ até $M6$ têm permissão nas duas áreas do hospital, enquanto os demais têm permissão somente na área $A1$. A carga horária dos médicos $M1$ até $M4$ é de 18 horas para esta janela de escalonamento, enquanto para os demais é de 12 horas. Outras informações contratuais dos médicos podem ser visualizadas na Sub-figura 3.1b. A quantidade mínima e máxima de médicos por dia/turno/área está disponível na Sub-figura 3.1c.

Um exemplo de escalonamento factível para a instância da 3.1 é apresentado na Sub-figura 3.1d. Cada linha apresenta as atribuições de cada médico indicando em qual área foi alocado no dia/turno. Todas as restrições de factibilidade são atendidas, entretanto existem violações, as quais são detalhadas a seguir:

Figura 3.1: Exemplo de uma instância e um escalonamento factível para o problema de escalonamento de médicos com uma janela de 2 dias.

Janela de tempo: 01/02/2021 até 02/02/2021 (2 dias úteis)
Feriados: não possui
Áreas: A1 - Cardiologia e A2 - COVID19

(a) Competência e áreas

Informação	Médico M1	Médico M2	Médico M3	Médico M4	Médico M5	Médico M6	Médico M7	Médico M8
Horas por mês	18	18	18	18	12	12	12	12
Horas em dias não úteis	12	12	0	0	0	12	12	0
Áreas com permissão	A1, A2	A1, A2	A1, A2	A1, A2	A1, A2	A1, A2	A1	A1
Áreas não preferenciais	-	A1	-	-	-	-	-	-
Alocações fixas	-	-	-	-	-	-	-	-
Ausências	-	-	-	-	-	-	-	-
Dia/Turno não preferenciais	-	-	-	Dia 1 e 2 (noite)	-	-	-	-

(b) Dados contratuais por médico

Turno	Área	Dia 1		Dia 2	
		Mínimo	Máximo	Mínimo	Máximo
Manhã	A1	2	3	2	2
Tarde	A1	1	2	1	2
Noite	A1	1	1	1	1
Manhã	A2	2	3	2	2
Tarde	A2	1	2	1	2
Noite	A2	1	1	1	1

(c) Demanda mínima e máxima por dia/turno/área

Médico	Dia 1			Dia 2		
	Manhã	Tarde	Noite	Manhã	Tarde	Noite
M1	A1	-	-	A1	-	-
M2	A1	-	-	-	A1	-
M3	-	A2	-	A2	-	-
M4	-	-	A2	-	-	A2
M5	A2	-	-	A2	-	-
M6	A2	-	-	A1	-	-
M7	-	A1	-	A1	-	-
M8	-	-	A1	-	-	A1

Para cada dia e turno é indicado se o médico terá folga (-) ou em qual área será alocado

(d) Exemplo de escalonamento factível

- A restrição $S1$ é violada para os médicos $M1$, $M2$, $M3$. O total de horas alocadas (12 horas para cada médico) é inferior à carga horária de seus contratos (18 horas). O custo associado a esta restrição será de 360, uma vez que o peso desta restrição é igual à 20 e o total de horas faltantes é igual à 18.
- A restrição $S2$ é violada para os médicos $M4$ e $M8$. O total de horas alocadas (24 horas para cada médico) excede o total de horas do contrato (18 e 12 horas, respectivamente). O custo associado a esta restrição será de 360, uma vez que o peso desta é igual à 20 e o total de horas excedentes é igual à 18 (6 horas do médico $M4$ e 12 horas do médico $M8$).
- A restrição $S9$ é violada para o médico $M2$. O médico não tem preferência em ser alocado na área $A1$, porém existem duas alocações nesta área. O custo associado a esta restrição será de 2, uma vez que o peso é igual à 1 e o total de alocações não preferenciais é igual a 2.
- A restrição $S10$ é violada para o médico $M4$. O médico não tem preferência em ser alocado no turno da noite nos dois dias, porém foi alocado em ambos. O custo associado a esta restrição será de 2, uma vez que o peso é igual à 1 e o total de alocações não preferenciais é igual a 2.

3.6 Formulação Matemática

Nesta seção será apresentada a formulação matemática para o problema do HCPA. Esta formulação é baseada na formulação do trabalho de (WICKERT et al., 2020), considerando o modelo estendido das restrições e sendo realizadas adaptações para as diferenças entre as restrições de cada versão do problema.

A Tabela 3.2 apresenta os índices para os dados de entrada que serão referenciados no modelo matemático. Na Tabela 3.3 são demonstrados os índices das variáveis de decisão e na Tabela 3.4, os índices das variáveis auxiliares para o cálculo da função objetivo.

Tabela 3.2: Índices dos dados de entrada utilizados pela formulação matemática

Símbolo	Definição
$m \in M$	m é o índice do médico e M é o conjunto de médicos do hospital
$d \in D$	d é o índice do dia e D é o conjunto de dias do mês
$\tilde{d} \in \tilde{D}$	\tilde{d} é o índice do dia e \tilde{D} são os dias não úteis do mês
$t \in T$	t é o índice do turno e T é o conjunto de turnos, onde t_M é o turno da manhã, t_T é o turno da tarde e t_N é o turno da noite
$a \in A$	a é o índice da área e A é o conjunto de áreas do hospital
$w \in W$	w é o índice do sábado e W é o conjunto de todos os sábados, com exceção do último sábado, se for o último dia do mês
$w \in \widetilde{W}$	w é o índice da semana e \widetilde{W} são as semanas do mês
noH_t	Quantidade de horas do turno t
$dMin_{dta}$	Mínimo de médicos para o turno t e área a no dia d
$dMax_{dta}$	Máximo de médicos para o turno t e área a no dia d
$p_{ma} \in 0, 1$	1 se o médico m tem permissão na área a , 0 caso contrário
$r_{ma} \in 0, 1$	1 se o médico m não tem preferência na área a , 0 caso contrário
$(m, d, t) \in S$	Ausências do médico m no dia d e turno
$(m, d, t, a) \in F$	Alocações fixas do médico m no dia d , turno t e área a
$(m, d, t) \in N$	Alocações não preferenciais do médico m no dia d e turno t
ch_m	Carga horária do médico m
chN_m	Ideal de horas trabalhadas em dias não úteis do médico m
ω^i	Peso associado à restrição de custo $i \in \{1...10\}$

Tabela 3.3: Variáveis de decisão utilizadas na formulação matemática

Símbolo	Definição
$x_{mdta} \in \{0, 1\}$	1 se o médico m está alocado no turno t e área a no dia d 0 caso contrário
$o_{md} \in \{0, 1\}$	1 se o médico m possui alocação no dia d 0 caso contrário
$z_{m\tilde{d}} \in \{0, 1\}$	1 se o médico m possui alocação no dia não útil \tilde{d} em ambos turnos do dia (manhã e tarde) 0 caso contrário
$y_{mw} \in \{0, 1\}$	1 se o médico m possui alocação no final de semana w 0 caso contrário

Tabela 3.4: Variáveis auxiliares utilizadas na formulação matemática

Símbolo	Definição
$vioS_m^i \in \mathbb{N}^*$	Número de violações restrição $i \in \{1, 2, 3, 4, 5, 7\}$ para médico m
$vioS_{mw}^6 \in \mathbb{N}^*$	Número de violações restrição 6 para médico m e final de semana w
$vioS_{md}^8 \in \mathbb{N}^*$	Número de violações restrição 8 para médico m e dia d
$vioS_{ma}^9 \in \mathbb{N}^*$	Número de violações restrição 9 para médico m e área a
$vioS_{mdt}^{10} \in \mathbb{N}^*$	Número de violações restrição 10 para médico m , dia d e turno t

A função objetivo, definida na Equação 3.1, busca minimizar o custo da solução considerando todas as restrições de qualidade. No primeiro somatório constam as restrições $S1$, $S2$, $S3$, $S4$, $S5$ e $S7$, as quais são penalizadas por hora excedente ou faltante de cada médico. Os demais somatórios apresentam as demais restrições de qualidade, cada qual com diferentes critérios por médico para penalização. A restrição $S6$ é penalizada por final de semana; a restrição $S8$ por dia do mês; a restrição $S9$, por área do hospital e a restrição $S10$ por dia e turno do mês.

$$\begin{aligned}
\min \quad & \sum_{i=1..5,7} \sum_{m \in M} (vioS_m^i * \omega^i) + \sum_{m \in M} (vioS_{mw}^6 * \omega^6) + \sum_{m \in M} \sum_{d \in D} (vioS_{md}^8 * \omega^8) + \\
& \sum_{m \in M} \sum_{a \in A} (vioS_{ma}^9 * \omega^9) + \sum_{m \in M} \sum_{d \in D} \sum_{t \in T} (vioS_{mdt}^{10} * \omega^{10})
\end{aligned} \tag{3.1}$$

Sujeito a

H1 - Demanda mínima

$$\sum_{m \in M} x_{mdta} \geq dMin_{dta} \quad \forall d \in D, t \in T, a \in A \tag{3.2}$$

H2 - Demanda máxima

$$\sum_{m \in M} x_{mdta} \leq dMax_{dta} \quad \forall d \in D, t \in T, a \in A \tag{3.3}$$

H3 - Autorização nas áreas

$$(1 - p_{ma}) * x_{mdta} = 0 \quad \forall m \in M, d \in D, t \in T, a \in A \tag{3.4}$$

H4 - Ausências

$$\sum_{a \in A} x_{mdta} = 0 \quad \forall (m, d, t) \in S \tag{3.5}$$

H5 - Alocações fixas

$$x_{mdta} = 1 \quad \forall (m, d, t, a) \in F \tag{3.6}$$

H6 - Turnos em dias úteis

$$\sum_{t \in T} \sum_{a \in A} x_{mdta} \leq 1 \quad \forall m \in M, d \in D \setminus \tilde{D} \tag{3.7}$$

H7 - Turnos e Áreas em dias não-úteis

$$\sum_{a \in A} (x_{m\tilde{d}t_Ma} + x_{m\tilde{d}t_Ta}) = 2z_{m\tilde{d}} \quad \forall m \in M, \tilde{d} \in \tilde{D} \tag{3.8}$$

$$\sum_{a \in A} x_{m\tilde{d}t_Na} + 2z_{m\tilde{d}} \leq 1 \quad \forall m \in M, \tilde{d} \in \tilde{D} \tag{3.9}$$

H8 - Sucessão turnos proibidos

$$\sum_{a \in A} x_{m\tilde{d}t_Na} + \sum_{t=M,T} \sum_{a \in A} x_{m(d+1)ta} \leq 1 \quad \forall m \in M, d \in \{1, \dots, |D| - 1\} \tag{3.10}$$

S1 - Mínimo de horas na competência

$$\sum_{d \in D} \sum_{t \in T} \sum_{a \in A} (noH_t * x_{mdta}) + vioS_m^1 \geq ch_m \quad \forall m \in M \tag{3.11}$$

S2 - Máximo de horas na competência

$$\sum_{d \in D} \sum_{t \in T} \sum_{a \in A} (noH_t * x_{mdta}) - vioS_m^2 \leq ch_m \quad \forall m \in M \quad (3.12)$$

S3 - Mínimo de horas em dias não úteis

$$\sum_{\tilde{d} \in \tilde{D}} \sum_{t \in T} \sum_{a \in A} (noH_t * x_{m\tilde{d}ta}) + vioS_m^3 \geq chN_m \quad \forall m \in M \quad (3.13)$$

S4 - Máximo de horas em dias não úteis

$$\sum_{\tilde{d} \in \tilde{D}} \sum_{t \in T} \sum_{a \in A} (noH_t * x_{m\tilde{d}ta}) - vioS_m^4 \leq chN_m \quad \forall m \in M \quad (3.14)$$

S5 - Equilíbrio horas em dias não úteis

$$\left| \sum_{\tilde{d} \in \tilde{D}} \sum_{a \in A} (6 * x_{m\tilde{d}t_Ma} + 6 * x_{m\tilde{d}t_Ta} - 12 * x_{m\tilde{d}t_Na}) \right| - vioS_m^5 = 0 \quad \forall m \in M \quad (3.15)$$

S6 - Final de semana incompleto

$$\sum_{t \in T} \sum_{a \in A} x_{mdta} \leq 2o_{md} \quad \forall m \in M, d \in D \quad (3.16)$$

$$o_{mw} + o_{m(w+1)} + vioS_{mw}^6 = 2y_{mw} \quad \forall m \in M, w \in W \quad (3.17)$$

S7 - Máximo finais de semana

$$\sum_{w \in W} y_{mw} - vioS_m^7 \leq 2 \quad \forall m \in M \quad (3.18)$$

S8 - Máximo de noites consecutivas

$$\sum_{d'=d}^{d+3} \sum_{a \in A} x_{md'ta} - vioS_{md}^8 \leq 3 \quad \forall m \in M, d \in \{1..|D| - 3\} \quad (3.19)$$

S9 - Áreas não preferenciais

$$r_{ma} * \left(\sum_{d \in D} \sum_{t \in T} x_{mdta} \right) - vioS_{ma}^9 = 0 \quad \forall m \in M, a \in A \quad (3.20)$$

S10 - Dias/turnos não preferenciais

$$\sum_{a \in A} x_{mdta} \leq vioS_{mdt}^{10} \quad \forall (m, d, t) \in N \quad (3.21)$$

As restrições de factibilidade são detalhadas da Restrição 3.2 até a Restrição 3.10. A Restrição 3.2 garante que o número de médicos alocados em cada dia/turno/área atenda à demanda mínima. A Restrição 3.3 garante que o número de médicos alocados em cada dia/turno/área não ultrapasse a demanda máxima. A Restrição 3.4 garante que nenhum médico será alocado em uma área em que não tenha permissão. A Restrição 3.5 garante

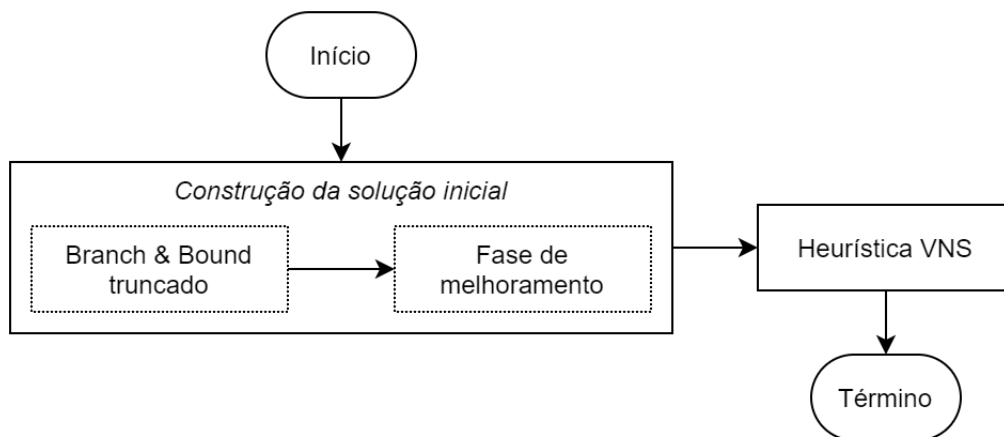
que nenhum médico será alocado em um dia e turno em que está ausente. A Restrição 3.6 garante que todas as alocações fixas serão contempladas na solução final. A Restrição 3.7 garante que existirá somente um turno alocado por dia para cada médico. As Restrições 3.8 e 3.9 garantem que o médico seja alocado em ambos os turnos do dia (manhã e tarde) ou então somente no turno da noite em dias não úteis. A Restrição 3.10 garante que, se o médico for alocado no turno da noite, não seja alocado nos turnos da manhã ou da tarde no dia imediatamente posterior.

As restrições 3.11 até 3.21 modelam as restrições de qualidade do problema. A Restrição 3.11 calcula a quantidade de violações da $S1$ em horas que estão abaixo do contrato de trabalho. A Restrição 3.12 calcula as violações da $S2$ em horas que estão acima do contrato. A Restrição 3.13 calcula as violações da $S3$ em horas alocadas abaixo do ideal em dias não úteis. A Restrição 3.14 calcula as violações da $S4$ em horas alocadas acima do ideal em dias não úteis. A restrição 3.15 calcula as violações da $S5$ em diferença de horas alocadas entre os turnos do dia (manhã e tarde) e o turno da noite. As Restrições 3.16 e 3.17 calculam as violações da $S6$ em quantidade de finais de semana incompletos, em que somente um dos dias foi alocado. A Restrição 3.18 calcula as violações da $S7$ em quantidade de finais de semana alocados acima do máximo permitido. A Restrição 3.19 calcula as violações da $S8$ em quantidade de dias consecutivos alocados no turno da noite acima do máximo permitido. A Restrição 3.20 calcula as violações da $S9$ em quantidade de alocações em áreas não preferenciais. E a Restrição 3.21 calcula as violações da $S10$ em quantidade de alocações em dia/turno não preferenciais.

4 HEURÍSTICA VNS PROPOSTA

Este capítulo detalha a heurística *Variable Neighborhood Search* (VNS) desenvolvida para solucionar o problema de escalonamento de médicos do HCPA. A Figura 4.1 apresenta o fluxo de execução do algoritmo proposto. A primeira etapa consiste na construção da solução inicial que será utilizada pelo VNS, a qual é dividida em dois estágios. Inicialmente é gerada uma solução factível via um algoritmo de Branch & Bound truncado (B&B_T). Posteriormente a solução passa por uma fase de melhoramento através de uma busca local descendente. A segunda etapa consiste na execução do VNS, utilizando-se da solução inicial de duas fases.

Figura 4.1: Fluxo de execução do algoritmo proposto



As próximas seções são organizadas conforme a seguir. A Seção 4.1 detalha a linguagem de programação e estruturas de dados utilizadas para a implementação do algoritmo. A Subseção 4.2.1 detalha o comportamento do primeiro estágio da construção da solução inicial através do B&B_T. Nesta seção também é discutida a importância dos cortes aplicados e da ordem de execução das alocações para que seja identificada uma solução factível de forma ágil. A Subseção 4.2.2 apresenta a fase de melhoramento aplicada sobre a solução inicial. E a Seção 4.3 traz detalhes do comportamento do VNS, tais como os tipos de vizinhança utilizados e pseudo-código.

4.1 Implementação e Estrutura de Dados

A implementação do algoritmo proposto foi realizada em Java. Essa linguagem foi selecionada devido à sua aplicação no HCPA em outros softwares, garantindo que o

uso do algoritmo de VNS fosse mais acessível ao hospital.

A avaliação do custo de uma solução deste problema é bastante onerosa devido ao volume de médicos e dias do mês. Em virtude disso, o principal foco da implementação foi o fácil acesso de informações utilizadas pelas restrições do problema. A estrutura de dados salva não somente as alocações de cada médico, mas também mantém informações totalizadoras sobre a solução, sendo estas:

- Número de horas alocadas no mês para cada médico;
- Número de horas alocadas em dias não úteis para cada médico;
- Número de horas alocadas nos turnos da manhã e da tarde em dias não úteis para cada médico;
- Número de horas alocadas no turno da noite em dias não úteis para cada médico;
- Número de finais de semana alocados para cada médico;
- Número de alocações consecutivas no turno da noite até o dia anterior;
- O custo associado de cada restrição de qualidade por médico e alocação.

Com os dados acima, a avaliação do custo final da solução se torna mais ágil. Por exemplo: é possível calcular a violação da restrição $S2$ com uma única comparação entre o total de horas alocadas e a carga horária do médico. Já sem essa informação, seria necessário avaliar a alocação de todos os dias para o médico, identificando qual turno foi alocado e totalizar as horas a cada avaliação do custo da solução.

A atualização desses dados é realizada toda vez que uma alocação é atribuída (no caso do $B\&B_T$) ou modificada (no caso da fase de melhoramento da solução inicial e do VNS). Por exemplo: se o médico possuir uma alocação no turno da manhã alterada para o turno da noite, o total de horas no mês será decrementado com a diferença de horas entre o novo turno e o antigo turno. Neste caso, será acrescido ao total um valor de 6 horas, dado que o turno da noite tem 12 horas e o turno da manhã tem 6 horas.

O custo associado por restrição auxilia na atualização do custo das soluções que são avaliadas dentro da fase de melhoramento e do VNS. Nestas etapas, já existe uma solução completa e os algoritmos apenas alteram uma alocação pré-existente por uma nova. Com a informação do custo associado por alocação não é necessário recalculer este valor para a alocação que está sendo substituída, apenas decrementar seu valor do custo total da solução.

4.2 Construção da Solução inicial

A solução inicial é gerada em dois estágios. Primeiro é executado um algoritmo de Branch & Bound truncado ($B\&B_T$) que constrói uma solução considerando todas as restrições de factibilidade. Assim que o $B\&B_T$ identifica a primeira solução factível, o algoritmo é finalizado e inicia-se a fase de melhoramento. A fase de melhoramento, por sua vez, aplica uma busca local a fim de reduzir o custo da solução do $B\&B_T$ até que seja interrompida por um mínimo local.

As próximas subseções são organizadas conforme a seguir. A Subseção 4.2.1 apresenta o algoritmo de $B\&B_T$ e os mecanismos de ordenação implementados. E a Subseção 4.2.2 detalha a busca local da fase de melhoramento.

4.2.1 Branch & Bound Truncado

O Branch & Bound ($B\&B$) é um algoritmo exato que se baseia no algoritmo de *Backtracking* com aplicação de cortes. O algoritmo de *Backtracking* consiste na enumeração de todas as soluções de um problema. A fim de reduzir o espaço de busca, o $B\&B$ aplica cortes durante a enumeração utilizando-se do custo da solução corrente e o melhor custo conhecido (*lower bound*).

Devido à grande quantidade de soluções e à dificuldade de aplicação de cortes para este problema, o $B\&B$ original não conseguiu ser finalizado em nenhuma das instâncias testadas, mesmo com tempos de execução superiores a 12 horas. Entretanto, o $B\&B$ identifica uma primeira solução factível com poucos segundos de execução.

Na literatura, é comum o uso de algoritmos que constroem a solução inicial de uma heurística através de alocações aleatórias. Porém, como possuem um caráter aleatório, se torna inviável gerarem uma solução factível. Para solucionar este problema são aplicadas penalizações elevadas quando alguma restrição de factibilidade é violada. Entretanto, essa abordagem onera parte do processamento da heurística para que primeiro seja tratada a infactibilidade da solução.

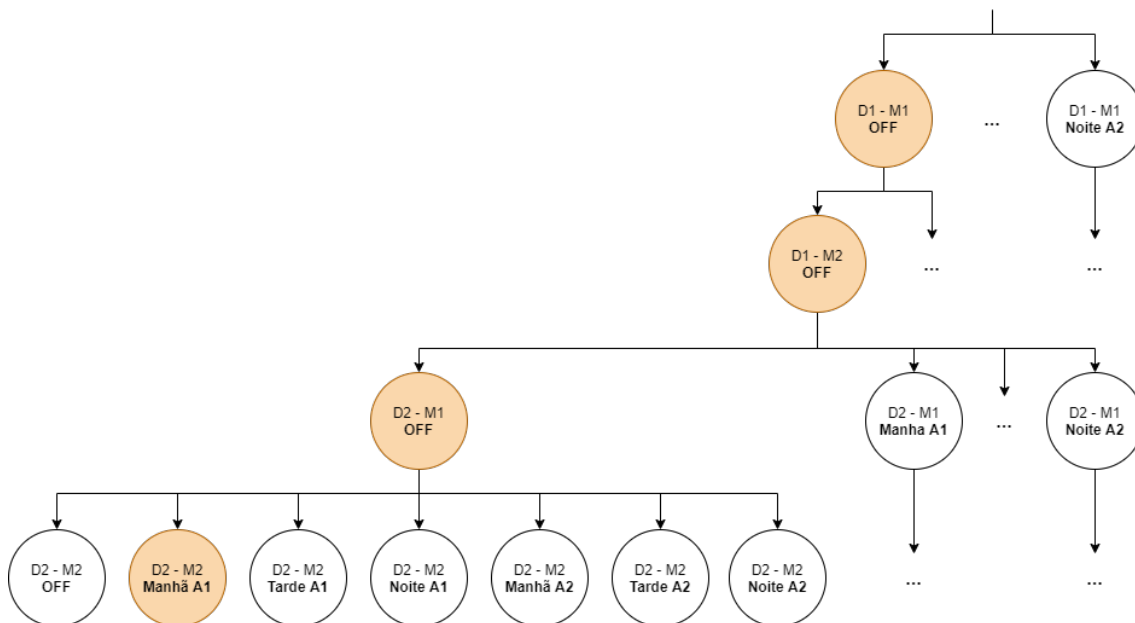
Dado que o $B\&B$ identifica rapidamente uma solução factível, este foi adaptado para que seja truncado no momento em que a primeira solução válida é encontrada. Dessa forma o tempo de processamento da heurística ficará concentrado na melhoria da qualidade da solução.

O algoritmo de $B\&B_T$ constrói uma solução através de uma árvore de escolhas,

onde cada nodo representa um dia do mês e um médico do hospital e determina se haverá uma alocação em um turno/área ou se haverá folga. Assim que todos os médicos foram alocados em todos os dias do mês, a solução é dada por completa. A ordem de construção da solução é feita por dia do mês. Isso significa que primeiro serão avaliadas as atribuições de todos os médicos no dia 1 para então passar a serem avaliadas as do dia 2, até que o último dia do mês seja alcançado.

A Figura 4.2 apresenta parte de uma árvore de soluções do B&B_T para um problema com 2 médicos e 2 dias. Os nodos apresentam a atribuição realizada para o dia d e o médico m , podendo ser uma folga (*OFF*) ou uma alocação em um turno e a área A_i , onde i é o índice da área.

Figura 4.2: Exemplo de uma árvore de soluções do B&B_T



No primeiro nível da árvore da Figura 4.2 existem as possíveis atribuições para o dia 1 e o médico 1. Já no segundo nível, existem as atribuições do dia 1 para o médico 2 e assim por diante. A árvore é percorrida em profundidade, onde o segundo caminho percorrido seleciona as seguintes atribuições: OFF, OFF, OFF e Manhã A1.

A escolha das alocações de cada médico em um dia pode influenciar negativamente o desempenho do algoritmo de enumeração. Quando um médico é alocado sem que seja considerada a demanda e a disponibilidade dos demais médicos naquele dia, o número de soluções infactíveis percorridas aumenta. Esse aumento é oriundo da restrição de demanda mínima, em que uma alocação sem análise pode acarretar a falta deste médico em outro turno/área desse dia. Para evitar esse problema, foram implementados mecanismos para definir uma ordem de processamento dos médicos e das atribuições disponíveis

conforme os dados atuais da solução.

4.2.1.1 Ordem de Processamento dos Médicos

Para cada dia do mês, o $B\&B_T$ define uma ordem de processamento dos médicos. Primeiro é verificado quais são as alocações válidas para cada médico. Então, para cada turno/área com alocação válida, é avaliada a quantidade de médicos faltante para atingir a demanda mínima. A ordenação dará prioridade para aqueles médicos que tem alocações em turno/área com maior demanda não atendida.

4.2.1.2 Ordem de Processamento das Alocações de um Médico em um Dia

A ordem de atribuição de cada uma das alocações válidas de um médico em um dia do mês segue o mesmo modelo da ordem de processamento dos médicos. São primeiro avaliadas as alocações cuja demanda ainda não foi atendida, em ordem decrescente de número de médicos faltante. No caso de empate na quantidade de médicos, é utilizada aquela com menor custo associado. Para a atribuição de folga sempre é assumido que a demanda mínima é 0.

4.2.1.3 Exemplo do Processo de Ordenação das Alocações

Um exemplo do processo de ordenação das alocações dos médicos em um dia do problema é apresentado na Figura 4.3. Na primeira iteração, o algoritmo determina a ordem das alocações do médico $M1$. Nesse momento ainda não foram realizadas alocações para esse dia, portanto o valor $DifMin$, que representa o número de médicos faltante para atender à demanda mínima, é igual ao valor da demanda mínima para todos turnos/áreas.

A maior falta de médicos ($DifMin$) está no turno da tarde (2), seguido do turno da noite (1) e da folga (0). Considerando que seja selecionada a alocação *Tarde A1* para o médico $M1$, o algoritmo passa para a segunda iteração e busca a ordem das alocações do médico $M2$. Agora, o turno da tarde tem 1 médico alocado, portanto o valor $DifMin$ será 1, igual à diferença do turno da noite. Como a alocação no turno da noite tem menor custo, a ordem deste médico ficará: *Noite A1*, *Tarde A1* e *Folga*.

Na última iteração, quando está sendo avaliado o médico $M5$, o turno da manhã fica em terceira posição, mesmo tendo menor custo que o turno da noite, pois a demanda deste último já foi excedida em um médico (-1 em $DifMin$).

Figura 4.3: Exemplo da ordenação das alocações de cada médico em um dia do mês

Turno	Área	Mínimo
Manhã	A1	0
Tarde	A1	2
Noite	A1	1

(a) Demanda mínima

Iteração	Médico	Informação	Folga	Manhã A1	Tarde A1	Noite A1
1	M1	<i>DifMin</i>	0	NP	2	1
		<i>Custo</i>	30	NP	20	15
		<i>Ordem</i>	3	-	1	2
2	M2	<i>DifMin</i>	0	NP	1	1
		<i>Custo</i>	30	NP	20	15
		<i>Ordem</i>	3	-	2	1
3	M3	<i>DifMin</i>	0	0	1	NP
		<i>Custo</i>	30	40	35	NP
		<i>Ordem</i>	2	3	1	-
4	M4	<i>DifMin</i>	0	0	1	NP
		<i>Custo</i>	30	5	10	NP
		<i>Ordem</i>	3	1	2	-
5	M5	<i>DifMin</i>	0	-1	NP	0
		<i>Custo</i>	10	10	NP	15
		<i>Ordem</i>	1	3	-	2

(b) Processamento do ordenamento

¹Os médicos já estão ordenados, conforme indicado na coluna “Iteração”. Cada iteração apresenta o número de médicos faltante para a alocação (*DifMin*); o custo associado a esta alocação (*Custo*), podendo ser *NP* quando não permitido e a ordem final de processamento (*Ordem*).

4.2.2 Fase de Melhoramento

Nesta subseção será detalhado o comportamento da busca local aplicada sobre a solução do B&B_T, assim como discutidos os motivos e benefícios de sua aplicação antes

da heurística de VNS.

O objetivo da fase de melhoramento é reduzir as violações mais comuns e de fácil ajuste que ocorrem sobre a solução gerada pelo B&B_T. Devido ao comportamento construtivo do B&B_T, o cálculo da penalização de algumas restrições somente pode ocorrer quando a solução está completa. O B&B_T dá preferência em alocar o mínimo de médicos demandado por dia/turno/área. Isso ocasiona uma alta penalização sobre as restrições de mínimo de horas (*S1*, *S3* e *S5*).

A fase de melhoramento utiliza uma busca local descendente que executa cinco tipos de vizinhança distintos. As vizinhanças são focadas nas restrições com maior penalização e cada uma modifica uma alocação de um médico em um dia do mês por outra alocação que irá reduzir o custo da restrição associada a esta vizinhança.

O Algoritmo 1 apresenta um pseudo-código da fase de melhoramento da solução inicial. O processo se inicia com a solução inicial do B&B_T (linha 1) e com a vizinhança 1 (linha 2). Para cada vizinhança N^k é selecionado o melhor vizinho s' com base na solução incumbente s (linha 4). Se o custo do vizinho s' for menor que o da incumbente s , então o vizinho passa a ser a solução corrente s (linha 8). Caso contrário, a busca local passa a utilizar a próxima vizinhança k (linha 6). O algoritmo é finalizado quando o tempo limite de execução é atingido ou quando todas as vizinhanças disponíveis já foram utilizadas (linha 3).

Algoritmo 1: Pseudo-código da Fase de Melhoramento da Construção da Solução Inicial

```

1   $s$  = solução inicial gerada pelo B&BT;
2   $k = 1$ ;
3  while (Tempo de execução < Tempo limite e  $k \leq 5$ ) do
4      Selecione o melhor vizinho  $s'$  da vizinhança  $N^k(s)$ ;
5      if  $f(s') \geq f(s)$  then
6           $k = k + 1$ ;
7      else
8           $s = s'$ ;
9          Se mantém na mesma vizinhança;
10     end
11 end

```

Os tipos de vizinhança são aplicados em ordem decrescente de custo associado, conforme avaliação de testes preliminares. São utilizadas as restrições, nesta ordem: *S1*, *S5*, *S3*, *S2* e *S4*. As restrições *S1*, *S5* e *S3* são as que apresentam maior custo associado. Já as restrições *S2* e *S4* foram incluídas não somente por seu custo, mas para que o algo-

ritmo busque equilibrar a quantidade de horas com as restrições $S1$ e $S3$, respectivamente.

Os critérios para seleção da atribuição a ser modificada e da nova variam por vizinhança conforme descrito abaixo:

- **Vizinhança S1:** poderão ser modificadas as atribuições em que o médico tem saldo de horas abaixo de sua carga horária e cujo dia o médico foi atribuído com folga. A substituição poderá ser feita por uma alocação em turno/área que não torne a solução infactível.
- **Vizinhança S5:** poderão ser modificadas as atribuições em que o médico tem saldo distinto de horas em dias não úteis entre os turnos do dia e da noite e cujo dia não útil o médico foi atribuído no turno com maior saldo. A substituição poderá ser feita por uma alocação no turno inverso (manhã ou tarde se estava alocado à noite e vice-versa) e mesma área, desde que não torne a solução infactível.
- **Vizinhança S3:** poderão ser modificadas as atribuições em que o médico tem saldo de horas em dias não úteis abaixo do ideal e cujo dia não útil o médico foi atribuído com folga. A substituição poderá ser feita por uma alocação em turno/área que não torne a solução infactível.
- **Vizinhança S2:** poderão ser modificadas as atribuições em que o médico tem saldo de horas acima de sua carga horária e cujo dia o médico foi atribuído com um turno/área. A substituição poderá ser feita por uma folga, desde que não torne a solução infactível.
- **Vizinhança S4:** poderão ser modificadas as atribuições em que o médico tem saldo de horas em dias não úteis acima do ideal e cujo dia não útil o médico foi atribuído com um turno/área. A substituição poderá ser feita por uma folga, desde que não torne a solução infactível.

A fase de melhoramento é baseada no VNS, porém distingui-se por não retornar para a primeira vizinhança quando a solução incumbente é atualizada. O uso da fase de melhoramento resultou em melhor qualidade na solução final do VNS em testes preliminares. Portanto, esta fase foi mantida no algoritmo para os experimentos finais e comparação com o método exato.

4.3 Heurística VNS

A heurística VNS consiste de uma busca local que utiliza diferentes vizinhanças para a atualização da solução incumbente. O método explora uma vizinhança até que um mínimo local seja identificado, para então passar para a próxima vizinhança. Quando todas as vizinhanças são utilizadas, o algoritmo realiza uma alteração denominada *shake*, que é responsável por modificar a solução incumbente de modo que seja possível sair do mínimo local.

As próximas subseções são organizadas conforme a seguir. Na Subseção 4.3.1 é apresentado o pseudo-código do VNS e detalhado seu comportamento. A Subseção 4.3.2 apresenta as vizinhanças utilizadas e exemplos de sua aplicação. E a Subseção 4.3.3 apresenta as conclusões sobre a implementação do algoritmo proposto.

4.3.1 Algoritmo VNS proposto

O Algoritmo 2 apresenta um pseudo-código da implementação da heurística VNS. O algoritmo começa com a construção da solução inicial através do B&B_T e da fase de melhoramento (linha 1). Em cada iteração é selecionado o vizinho de menor custo da vizinhança k sobre a solução incumbente s (linha 6). Se o vizinho tiver um custo melhor que o da incumbente, então o vizinho passa a ser a solução corrente (linha 8), volta-se a avaliar a primeira vizinhança (linha 9) e o número de iterações sem melhora no custo é zerado (linha 10). Caso contrário, o algoritmo passa a utilizar a próxima vizinhança disponível (linha 12) e é incrementado o número de iterações sem melhora da solução (linha 13). Após todas as vizinhanças terem sido utilizadas, a solução incumbente é atualizada por uma função *shake* (linhas 16 e 17) que tem por objetivo modifica-la para que o algoritmo possa ser novamente executado sem cair no mesmo mínimo local.

A seleção do vizinho s' (linha 6) busca dentre todos os vizinhos existentes aquele de menor custo associado. Se existir mais de um vizinho com o mesmo custo, então o vizinho s' é selecionado de forma aleatória dentre os de menor custo.

A avaliação de custo dos vizinhos utiliza as estruturas de dados mencionados na Seção 4.1, sem atualizar diretamente a solução incumbente. Não é levado em consideração o cálculo do custo da restrição $S8$ durante a seleção do melhor vizinho s' , devido ao alto consumo de tempo para sua avaliação. Entretanto, quando a solução corrente é atualizada, o custo desta restrição é calculado, garantindo a real qualidade da nova solução.

Algoritmo 2: Pseudo-código VNS

```

1   $s$  = solução inicial de duas fases;
2  while (Tempo de execução < Tempo limite e
3  Nro iterações sem melhora da solução < Máximo iterações) do
4       $k = 1$ ;
5      while (Tempo de execução < Tempo limite e  $k \leq 3$ ) do
6          Seleciona o melhor vizinho  $s'$  da vizinhança  $N^k(s)$ ;
7          if  $f(s') < f(s)$  then
8               $s = s'$ ;
9               $k = 1$ ;
10             Zera número de iterações sem melhora da solução;
11         else
12              $k = k + 1$ ;
13             Incrementa número de iterações sem melhora da solução;
14         end
15     end
16     Seleciona vizinho  $s''$  da vizinhança shake( $s$ );
17      $s = s''$ ;
18 end

```

O movimento *shake* altera a atribuição de todos os médicos considerando um dia aleatório do mês para cada médico. A modificação não considera as restrições de factibilidade $H1$, $H2$ e $H8$, podendo tornar a solução corrente infactível. Essa medida onera o início do processamento do VNS nas primeiras iterações, dado que primeiro será necessário remover a infactibilidade. Entretanto, como a solução corrente parou em um mínimo local, se faz necessário um ajuste denso da solução para percorrer outros espaços de busca. Quando uma restrição de factibilidade é violada, é associada uma penalização 1.000 vezes maior que o peso das restrições de qualidade. Com isso, ao iniciar uma nova iteração com uma solução inválida, o algoritmo dará preferência para os vizinhos que tornam a incumbente factível.

O VNS é interrompido por limite de tempo de execução ou limite de iterações sem melhora no custo da solução. Esses limitantes foram definidos conforme a quantidade de médicos de cada instância e são apresentados no Capítulo 6.

4.3.2 Tipos de Vizinhança

No VNS os tipos de vizinhança têm por objetivo explorar diferentes espaços de soluções. Para isso, foram implementadas três vizinhanças: *CHANGE*, *SWAP* e *3-CHAIN*,

as quais serão definidas e exemplificadas nas subseções a seguir.

4.3.2.1 Vizinhaça *CHANGE*

Na vizinhaça *CHANGE*, os vizinhos alteram a atribuição de um dia e de um médico da solução corrente por outra alocação ou folga, desde que esta não torne a solução infactível. A Figura 4.4 apresenta um exemplo de vizinho para o escalonamento de um médico, considerando uma janela de três dias. A primeira linha exhibe as atribuições do médico *M1* na solução corrente e na segunda linha são apresentadas as atribuições deste médico após ser modificada a alocação do dia 1 por um vizinho do tipo *CHANGE*.

Figura 4.4: Exemplo de uma modificação válida na solução corrente do VNS pelo tipo de vizinhaça *CHANGE* para um escalonamento de 3 dias.

	Médico	Dia 1 (útil)	Dia 2 (não útil)	Dia 3 (não útil)
<i>Solução atual</i>	M1	Noite A1	Folga	Manhã/Tarde A1
<i>Vizinho CHANGE válido</i>	M1	Noite A2	Folga	Manhã/Tarde A1

¹A modificação foi feita no dia 1, alterando a área de alocação do médico *M1* de *A1* para *A2*.

Já a Figura 4.5 apresenta um exemplo de movimento *CHANGE* inválido, pois torna a solução infactível ao tentar alocar o turno da noite no dia anterior a uma alocação no turno da manhã e tarde.

Figura 4.5: Exemplo de uma modificação inválida na solução corrente do VNS pelo tipo de vizinhaça *CHANGE* para um escalonamento de 3 dias.

	Médico	Dia 1 (útil)	Dia 2 (não útil)	Dia 3 (não útil)
<i>Solução atual</i>	M1	Noite A1	Folga	Manhã/Tarde A1
<i>Vizinho CHANGE inválido</i>	M1	Noite A1	Noite A1	Manhã/Tarde A1

¹A tentativa de modificação foi feita no dia 2, porém é inválida, pois no dia 3 o médico está alocado nos turnos da manhã e tarde. Portanto, não pode ser alocado no turno da noite no dia 2.

O processo de seleção do vizinho considera todas as trocas válidas entre todos os médicos e dias do mês. A ordem de execução é crescente por número de dias e número identificador do médico. Caso exista mais de um vizinho com o menor custo associado, então a seleção ocorre de forma aleatória.

4.3.2.2 Vizinhança SWAP

Na vizinhança *SWAP*, os vizinhos invertem a atribuição de dois médicos de um dia da solução corrente, desde que esta não torne a solução infactível. A Figura 4.6 apresenta um exemplo de vizinho para o escalonamento de dois médicos, considerando uma janela de três dias. A primeira e segunda linhas exibem as atribuições dos médicos *M1* e *M2* na solução corrente e na terceira e quarta linhas são apresentadas como ficam as atribuições após a inversão.

Figura 4.6: Exemplo de uma modificação na solução corrente do VNS pelo tipo de vizinhança *SWAP* para um escalonamento de 3 dias.

	Médico	Dia 1 (útil)	Dia 2 (não útil)	Dia 3 (não útil)
Solução atual	M1	Noite A1	Folga	Manhã/Tarde A1
	M2	Manhã A1	Noite A1	Noite A1
Vizinho SWAP válido	M1	Manhã A1	Folga	Manhã/Tarde A1
	M2	Noite A1	Noite A1	Noite A1

¹A modificação foi feita no dia 1, invertendo as alocações dos médicos *M1* e *M2*.

Já a Figura 4.7 apresenta um exemplo de movimento *SWAP* inválido, pois torna a solução infactível ao tentar alocar o turno da noite no dia anterior a uma alocação no turno da manhã e tarde.

Figura 4.7: Exemplo de uma modificação inválida na solução corrente do VNS pelo tipo de vizinhança *SWAP* para um escalonamento de 3 dias.

	Médico	Dia 1 (útil)	Dia 2 (não útil)	Dia 3 (não útil)
Solução atual	M1	Noite A1	Folga	Manhã/Tarde A1
	M2	Manhã A1	Noite A1	Noite A1
Vizinho SWAP inválido	M1	Noite A1	Noite A1	Manhã/Tarde A1
	M2	Manhã A1	Folga	Noite A1

¹A tentativa de modificação foi feita no dia 2. A inversão é inválida, pois no dia 3 o médico *M1* está alocado nos turnos da manhã e da tarde, logo não pode ser alocado no turno da noite.

O processo de seleção do vizinho considera todas as trocas válidas entre todos os dias do mês e um limite de 50 médicos escolhidos aleatoriamente. A ordem de execução

é crescente por número de dias e número identificador dos médicos selecionados. Caso exista mais de um vizinho com o menor custo associado, então a seleção ocorre de forma aleatória.

4.3.2.3 Vizinhaça 3-CHAIN

Na vizinhaça 3-CHAIN, os vizinhos invertem a atribuição de um dia entre três médicos da solução corrente, desde que esta não torne a solução infactível. A Figura 4.8 apresenta um exemplo de vizinho para o escalonamento de três médicos considerando uma janela de três dias. As três primeiras linhas exibem as atribuições dos médicos *M1*, *M2* e *M3* na solução corrente e nas três últimas linhas são apresentadas como ficam as atribuições após a inversão.

Figura 4.8: Exemplo de uma modificação na solução corrente do VNS pelo tipo de vizinhaça 3-CHAIN para um escalonamento de 3 dias.

	Médico	Dia 1 (útil)	Dia 2 (não útil)	Dia 3 (não útil)
Solução atual	M1	Noite A1	Folga	Manhã/Tarde A1
	M2	Manhã A1	Noite A1	Noite A1
	M3	Tarde A1	Manhã/Tarde A1	Folga
Vizinho 3-CHAIN válido	M1	Noite A1	Folga	Noite A1
	M2	Manhã A1	Noite A1	Folga
	M3	Tarde A1	Manhã/Tarde A1	Manhã/Tarde A1

¹A modificação foi feita no dia 3, onde o médico *M1* recebeu a alocação do médico *M2*; o médico *M2* recebeu do *M3* e o médico *M3* recebeu a alocação do *M1*.

Já a Figura 4.9 apresenta um exemplo de movimento 3-CHAIN inválido, pois torna a solução infactível ao tentar alocar o turno da noite no dia anterior a uma alocação no turno da manhã e tarde.

Da mesma forma que na vizinhaça SWAP, o processo de seleção do vizinho considera todas as trocas válidas entre todos os dias do mês e um limite de 50 médicos escolhidos aleatoriamente. A ordem de execução é crescente por número de dias e número identificador dos médicos selecionados. Caso exista mais de um vizinho com o menor custo associado, então a seleção ocorre de forma aleatória.

Figura 4.9: Exemplo de uma modificação inválida na solução corrente do VNS pelo tipo de vizinhança *3-CHAIN* para um escalonamento de 3 dias.

	Médico	Dia 1 (útil)	Dia 2 (não útil)	Dia 3 (não útil)
<i>Solução atual</i>	M1	Noite A1	Folga	Manhã/Tarde A1
	M2	Manhã A1	Noite A1	Noite A1
	M3	Tarde A1	Manhã/Tarde A1	Folga
<i>Vizinho 3-CHAIN inválido</i>	M1	Noite A1	Noite A1	Manhã/Tarde A1
	M2	Manhã A1	Manhã/Tarde A1	Noite A1
	M3	Tarde A1	Folga	Folga

¹A tentativa de modificação foi feita no dia 2. A inversão é inválida, pois no dia 3 o médico *M1* está alocado nos turno da manhã e tarde, logo não pode ser alocado no turno da noite.

4.3.3 Conclusões sobre o Algoritmo Proposto

O algoritmo proposto utiliza-se de uma solução inicial já factível através do $B\&B_T$ e a heurística de VNS aborda os principais tipos de vizinhança utilizados na literatura.

Para as vizinhanças *SWAP* e *3-CHAIN*, o número de médicos influencia significativamente no tempo de execução devido ao número de combinações realizadas para as inversões. A fim de evitar o consumo excessivo de tempo, foi limitada a avaliação em 50 médicos selecionados aleatoriamente. Foram realizados experimentos com um limite maior de médicos, porém o número de combinações reduz a quantidade de iterações da heurística e a qualidade final ficou comprometida.

Em testes preliminares também foi conduzido o impacto do uso da fase de melhoramento da solução inicial sobre a qualidade final do VNS. Na maioria dos experimentos a qualidade das soluções do VNS foi superior quando o custo inicial da solução já havia sido reduzido.

A implementação do algoritmo, contendo o $B\&B_T$, fase de melhoramento e heurística VNS, está disponível no site <<https://github.com/taticm13/PRP-HCPA-2020.git>>.

5 GERADOR DE INSTÂNCIAS

Nesta seção será detalhada a implementação do gerador de instâncias, as configurações disponíveis e o formato do arquivo de entrada. Também serão apresentados os diferentes grupos de instâncias gerados para os experimentos sobre o VNS proposto.

O gerador foi implementado na linguagem Java, considerando as características do problema aplicado ao HCPA. Os parâmetros de entrada refletem as informações que podem variar mês-a-mês e são detalhados na Seção 5.1. O formato do arquivo de entrada, que representa uma instância, é definido e exemplificado na Seção 5.2. Já as instâncias geradas para os testes sobre o VNS são apresentadas na Seção 5.3.

5.1 Parâmetros de Entrada

Os parâmetros de entrada do gerador de instâncias representam informações que tipicamente variam mensalmente no hospital. Abaixo são definidos todos os parâmetros e como devem ser informados.

- **Competência:** o usuário informa o mês e ano para escalonamento. O escalonamento sempre irá ocorrer do primeiro até o último dia do mês.
- **Feriados:** o usuário informa uma lista de dias da competência que serão considerados como feriados.
- **Áreas:** o usuário informa a quantidade de áreas para escalonamento. A identificação de cada área é feita de forma sequencial de 1 até N , sendo N o número de áreas informada.
- **Médicos:** o usuário informa a quantidade de médicos do hospital e qual o percentual aplicado para cada um dos tipos de contrato listados abaixo:
 - Contrato de *tempo integral*, onde o médico terá 200 horas mensais e um ideal de 48 horas trabalhadas em dias não úteis.
 - Contrato de *tempo parcial*: onde o médico terá 120 horas mensais e um ideal de 24 horas trabalhadas em dias não úteis.
- **Permissão nas áreas:** o usuário informa o percentual de áreas que serão permitidas para todos os médicos e a probabilidade de um médico ter permissão nas demais áreas que são restritas. Se o parâmetro informado como 100%, todos os médicos

terão permissão em todas as áreas do hospital. Caso contrário, as áreas restritas serão selecionadas aleatoriamente.

- **Áreas não preferenciais:** o usuário informa a probabilidade do médico não ter preferência de ser alocado em uma área.
- **Ausências:** o usuário informa a probabilidade do médico estar indisponível em algum dia/turno do mês. A seleção dos dias e turnos com ausência é realizada aleatoriamente para cada médico.
- **Alocações fixas:** o usuário informa a probabilidade do médico possuir uma alocação fixa em um dia do mês. A atribuição do turno e área é aleatória, obedecendo as permissões do médico e descartando dias em que esteja ausente. A seleção do dia em que haverá uma alocação fixa também é realizada aleatoriamente para cada médico.
- **Dia/Turno não preferenciais:** o usuário informa a probabilidade do médico não ter preferência em ser alocado em um dia/turno do mês. A seleção dos dias não preferencias é aleatória, sendo descartados os dias em que o médico tem uma alocação fixa ou ausência.
- **Demanda por dia/turno/área:** o usuário informa a taxa de aproximação com relação ao limite superior de médicos demandado por dia/turno/área (α). A demanda mínima e máxima de cada dia/turno/área (DSA) varia conforme esta taxa de aproximação e se o dia é útil ou não. Na Subseção 5.1.1 é detalhado como é realizado cálculo desta função e seu significado.

5.1.1 Limite Superior de Médicos Demandado por Dia/Turno/Área

O limite superior de médicos alocados por dia/turno/área ($limSupDSA$) representa o máximo de médicos que um dia/turno/área pode ter como demanda mínima de modo que a instância seja factível. Isto é, que exista uma solução que atenda à demanda mínima de todos os dias/turnos/áreas da instância, atendendo à restrição $H1$.

A Equação 5.1 apresenta o cálculo de $limSupDSA$, onde α representa a taxa de aproximação com relação ao limite superior; $noMed$ representa o número de médicos disponíveis do hospital; $noArea$, a quantidade de áreas do hospital e $noTurno$, a quantidade de turnos válidos no dia. Para dias úteis são considerados 3 turnos (Manhã, Tarde

ou Noite) e para dias não úteis são considerados 2 turnos (Manhã/Tarde ou Noite).

$$\limSupDSA = (noMed / (noArea * noTurno)) * \alpha \quad (5.1)$$

Esse limite superior compara a quantidade de médicos disponíveis e a quantidade de *slots* (turno/área) que terão médicos alocados no mesmo dia. Por exemplo: em um hospital que possui 4 áreas irão existir 12 *slots* para alocação em um dia útil (produto entre o número de turnos do dia e o número de áreas do hospital). Se cada um desses *slots* ter uma demanda mínima de 2 médicos, a quantidade de médicos disponível no hospital deverá ser igual ou superior à 24. Caso contrário, não será possível atender à demanda mínima, uma vez que cada médico só pode ser alocado em um turno/área do mesmo dia conforme restrições *H6* e *H7*.

Quanto maior a taxa de aproximação do limite superior, maior é a quantidade de soluções infactíveis que essa instância irá possuir. Isso ocorre devido à pequena margem para variação das alocações dentro de um dia, ocasionando na infactibilidade da maioria das combinações que seriam válidas só observando os dados contratuais dos médicos.

5.2 Formato do Arquivo de Entrada

Uma instância é representada através de um arquivo de entrada cujo formato é similar ao utilizado na Segunda Competição Internacional de Escalonamento de Enfermeiras (INRC-II) (CESCHIA et al., 2015). A Figura 5.1 apresenta um exemplo de uma instância no formato de entrada utilizado pelo gerador. O arquivo é gerado em "TXT" e deve possuir os seguidos dados:

- Uma linha com o ano, mês, primeiro e último dias do mês (identificador *MONTH*);
- Uma linha com a quantidade de feriados, seguida por uma linha para cada dia considerado como feriado (identificador *HOLIDAYS*);
- Uma linha com a quantidade de áreas, seguida por uma linha para cada área, indicando seu ID e descrição (identificador *LOCATIONS*);
- Uma linha com a quantidade de médicos, seguida por uma linha para cada médico, indicando seu ID, nome, horas mensais, horas alocadas em dias não úteis e se tem permissão (1) ou não (0) em cada área do hospital (identificador *PHYSICIANS*);
- Uma linha com a quantidade de alocações fixas, seguida por uma linha para cada alocação fixa, indicando o ID do médico, dia, turno (1-manhã, 2-tarde e 3-noite) e

Figura 5.1: Exemplo de uma instância no formato de entrada gerado

INÍCIO DO DOCUMENTO TXT	FINAL DO DOCUMENTO TXT
MONTH = 2020 1 1 31	FIXED ASSIGNMENTS = 2
HOLIDAYS = 1	9 24 2 1
1	8 3 1 1
LOCATIONS = 1	LOCKS = 1
1 Location1	5 3 1
PHYSICIANS = 10	NOT PREFERENCE PER LOCATION = 2
1 Physician1 150 24 1	5 1 1
2 Physician2 200 48 1	6 1 1
3 Physician3 150 24 1	PENALTY PER ASSIGN = 2
4 Physician4 200 48 1	10 31 1 1
5 Physician5 200 48 1	10 31 3 1
6 Physician6 200 48 1	REQUIREMENTS = 124
7 Physician7 200 48 1	1 1 1 1 2
8 Physician8 200 48 1	1 2 1 1 3
9 Physician9 200 48 1	1 3 1 0 2
10 Physician10 200 48 1	< CONTINUA >

ID da área (identificador *FIXED ASSIGNMENTS*);

- Uma linha com a quantidade de ausências, seguida por uma linha para cada ausência, indicando o ID do médico, dia e turno (1-manhã, 2-tarde e 3-noite) (identificador *LOCKS*);
- Uma linha com a quantidade de áreas não preferenciais, seguida por uma linha para cada área não preferencial, indicando o ID do médico, ID da área e peso associado (identificador *NOT PREFERENCE PER LOCATION*);
- Uma linha com a quantidade de alocações não preferenciais, seguida por uma linha para cada alocação não preferencial, indicando o ID do médico, dia, ID do turno e peso associado (identificador *PENALTY PER ASSIGN*);
- Uma linha com a quantidade de combinações entre dias, turnos e áreas, seguida por uma linha para cada DSA, indicando o dia, o ID do turno, o ID da área, o mínimo e máximo de médicos (identificador *REQUIREMENTS*);

5.3 Instâncias Utilizadas nos Experimentos

Os experimentos conduzidos no Capítulo 6 comparam o comportamento da heurística proposta com o de um método exato. O principal foco da análise está sobre o impacto da taxa de aproximação do *limSupDSA* entre os diferentes métodos. Para esta avaliação, foram fixados todos os demais parâmetros, com exceção da taxa de aproxima-

ção e do número de médicos.

Tabela 5.1: Parâmetros do gerador para as instâncias utilizadas nos experimentos

Parâmetro	Detalhamento
Competência	Janeiro de 2020
Feriados	Feriados nacionais
Áreas	Quantidade: 4
Médicos	Quantidades: 50, 100, 150, 250, 500 80% dos médicos com contrato integral 20% dos médicos com contrato parcial
Permissão nas áreas	100% das áreas sem restrição
Áreas não preferenciais	30%
Ausências	20% para grupo de Baixa Demanda 10% para grupo de Média Demanda 0% para grupo de Alta Demanda
Alocações fixas	0%
Alocações não preferenciais	30%
Taxa de aproximação (α)	10% para grupo de Baixa Demanda
do <i>limSupDSA</i>	45% para grupo de Baixa Demanda 90% para grupo de Baixa Demanda

As instâncias foram divididas em três grupos: baixa, média e alta demanda. A Tabela 5.1 apresenta os parâmetros utilizados. Para cada um dos grupos foram geradas 3 instâncias com 50, 100, 150, 250 e 500 médicos, todas com 4 áreas. Ao total foram utilizadas 45 instâncias, listadas na Tabela 5.2.

Tabela 5.2: Instâncias geradas para os experimentos

Nome	Grupo	#Médicos	#Áreas	#Instâncias
I_BD_050P_4L_ID1 até ID3	BD	050	4	3
I_BD_100P_4L_ID1 até ID3	BD	100	4	3
I_BD_150P_4L_ID1 até ID3	BD	150	4	3
I_BD_250P_4L_ID1 até ID3	BD	250	4	3
I_BD_500P_4L_ID1 até ID3	BD	500	4	3
I_MD_050P_4L_ID1 até ID3	MD	050	4	3
I_MD_100P_4L_ID1 até ID3	MD	100	4	3
I_MD_150P_4L_ID1 até ID3	MD	150	4	3
I_MD_250P_4L_ID1 até ID3	MD	250	4	3
I_MD_500P_4L_ID1 até ID3	MD	500	4	3
I_AD_050P_4L_ID1 até ID3	AD	050	4	3
I_AD_100P_4L_ID1 até ID3	AD	100	4	3
I_AD_150P_4L_ID1 até ID3	AD	150	4	3
I_AD_250P_4L_ID1 até ID3	AD	250	4	3
I_AD_500P_4L_ID1 até ID3	AD	500	4	3

O gerador implementado e todas as instâncias utilizados neste trabalho estão disponíveis no site <https://github.com/taticm13/PRP-HCPA_202011.git>.

6 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo são apresentados os experimentos conduzidos sobre os diferentes grupos de instâncias e algoritmos implementados. O principal objetivo foi identificar em quais circunstâncias o uso da heurística proposta é mais adequado quando comparado com um método exato.

Para a comparação foi utilizada uma formulação matemática com execução via o solver COIN Branch & Cut (CBC). Uma vez que o HCPA busca reduzir seus custos financeiros com o uso de algoritmos no escalonamento, optou-se por um solver gratuito ao invés do CPLEX, o qual é necessário realizar o pagamento de uma licença para utilização.

As próximas seções são organizadas conforme a seguir. A Seção 6.1 apresenta as características do ambiente de execução e as configurações utilizadas. A Seção 6.2 apresenta os experimentos conduzidos sobre a fase de construção da solução inicial. E na Seção 6.3 são analisados os resultados entre o VNS e o CBC.

6.1 Ambiente de Execução e Configurações

Todos os experimentos foram realizados em um computador Intel Core i7, com 16GB de RAM e 2,20GHz. Os testes realizados com VNS apresentados na Seção 6.3 e com a Fase de Melhoramento na Seção 6.2.2 correspondem à média de 10 execuções por instância. Já nos testes com CBC e B&B_T, por serem métodos exatos, foram executados somente uma vez para cada instância.

Tabela 6.1: Tempo limite de execução do VNS e do CBC em segundos

Método	Número de médicos da instância				
	050	100	150	250	500
VNS	60s	180s	180s	300s	300s
CBC	900s	1800s	1800s	3600s	3600s

O tempo limite de execução para o VNS e o CBC é variável conforme o número de médicos e pode ser consultado na Tabela 6.1. O número máximo de iterações sem melhoria do custo da solução corrente utilizado no VNS é 10 vezes o número de médicos da respectiva instância.

A comparação de qualidade entre as soluções geradas no experimentos apresen-

tados nas próximas seções é realizada por GAP (%) com relação ao *lower bound* (LB) de cada instância. O cálculo do GAP é apresentado em percentual e é apurado conforme a Equação 6.1, onde “*custoGerado*” representa o custo da solução em avaliação e “*LB*” representa o *lower bound* da instância.

$$GAP = (custoGerado - LB) / custoGerado \quad (6.1)$$

Todos os resultados apresentados nos experimentos das próximas seções, assim como os arquivos de solução e algoritmos implementados estão disponíveis no site <https://github.com/taticm13/PRP-HCPA_202011.git>.

6.2 Análise sobre a Construção da Solução Inicial

Nesta seção serão apresentados os experimentos realizados sobre a fase de construção da solução inicial. Na Subseção 6.2.1 é avaliado o impacto na qualidade da solução inicial gerada pelo B&B_T com e sem o uso dos mecanismos de ordenação apresentados na Subseção 4.2.1. Na Subseção 6.2.2 é comparada a qualidade entre a solução inicial do B&B_T e da fase de melhoramento. Por fim, a Subseção 6.2.3 apresenta as conclusões sobre os experimentos conduzidos sobre a solução inicial.

6.2.1 Qualidade do B&B Truncado x Mecanismos de Ordenação

O primeiro experimento tem por objetivo identificar o impacto sobre a qualidade da solução inicial quando utilizados os mecanismos de ordenação no algoritmo de B&B_T.

A Tabela 6.2 apresenta para cada instância a diferença entre a qualidade da solução gerada sem os mecanismos de ordenação (B&B^{SO}_T) e com tais mecanismos (B&B^{CO}_T). Além do GAP da solução com relação ao *lower bound* (LB) da instância, também são apresentados o custo e o tempo de execução em segundos. Para cada grupo (BD, MD e AD) é apresentada uma linha “*média*”, que demonstra a média do GAP de todas as instâncias desse grupo.

Em todas as instâncias testadas, o uso dos mecanismos de ordenação apresentou soluções com melhor qualidade. A redução do GAP com relação à execução sem o uso das ordenações é mais perceptível nas instâncias do grupo de média demanda (MD).

Tabela 6.2: Resultados da solução inicial do B&B_T com uso dos mecanismos de ordenação (B&B_t^{SO}) e sem uso de tais mecanismos (B&B_t^{CO})

Instância	LB	B&B _t ^{SO}			B&B _t ^{CO}		
		Custo	GAP	Tempo	Custo	GAP	Tempo
I_BD_050P_4L_ID1	222400	222400	0.0%	0.01s	222400	0.0%	0.03s
I_BD_050P_4L_ID2	222400	222400	0.0%	0.00s	222400	0.0%	0.00s
I_BD_050P_4L_ID3	222400	222400	0.0%	0.00s	222400	0.0%	0.00s
I_BD_100P_4L_ID1	372320	429362	13.3%	0.01s	379995	2.0%	0.03s
I_BD_100P_4L_ID2	372320	434750	14.4%	0.00s	379729	2.0%	0.01s
I_BD_100P_4L_ID3	372320	431479	13.7%	0.00s	380058	2.0%	0.01s
I_BD_150P_4L_ID1	594720	657263	9.5%	0.01s	602229	1.2%	0.04s
I_BD_150P_4L_ID2	594720	667306	10.9%	0.00s	600785	1.0%	0.03s
I_BD_150P_4L_ID3	594720	659142	9.8%	0.00s	603084	1.4%	0.02s
I_BD_250P_4L_ID1	894560	1089978	17.9%	0.01s	924809	3.3%	0.05s
I_BD_250P_4L_ID2	894560	1087701	17.8%	0.01s	923111	3.1%	0.04s
I_BD_250P_4L_ID3	894560	1063756	15.9%	0.01s	923167	3.1%	0.05s
I_BD_500P_4L_ID1	1849600	2089631	11.5%	0.02s	1887103	2.0%	0.10s
I_BD_500P_4L_ID2	1849600	2121168	12.8%	0.01s	1888640	2.1%	0.07s
I_BD_500P_4L_ID3	1849600	2121160	12.8%	0.01s	1888238	2.0%	0.08s
<i>média do grupo BD</i>			10.7%			1.7%	
I_MD_050P_4L_ID1	65440	131783	50.3%	0.00s	79520	17.7%	0.01s
I_MD_050P_4L_ID2	65440	133563	51.0%	0.00s	76354	14.3%	0.01s
I_MD_050P_4L_ID3	65440	134466	51.3%	0.00s	74808	12.5%	0.01s
I_MD_100P_4L_ID1	70974	349566	79.7%	0.00s	114194	37.8%	0.02s
I_MD_100P_4L_ID2	70917	332376	78.7%	0.00s	112825	37.1%	0.01s
I_MD_100P_4L_ID3	71025	344090	79.4%	0.00s	107840	34.1%	0.02s
I_MD_150P_4L_ID1	191034	587560	67.5%	0.00s	259194	26.3%	0.02s
I_MD_150P_4L_ID2	190969	590602	67.7%	0.00s	246093	22.4%	0.02s
I_MD_150P_4L_ID3	190925	592404	67.8%	0.00s	246054	22.4%	0.03s
I_MD_250P_4L_ID1	262028	934005	71.9%	0.00s	357833	26.8%	0.04s
I_MD_250P_4L_ID2	261960	946632	72.3%	0.01s	363181	27.9%	0.05s
I_MD_250P_4L_ID3	260971	946300	72.4%	0.01s	368635	29.2%	0.04s
I_MD_500P_4L_ID1	449455	1840110	75.6%	0.01s	661220	32.0%	0.08s
I_MD_500P_4L_ID2	449240	1849581	75.7%	0.01s	670698	33.0%	0.14s
I_MD_500P_4L_ID3	449214	1856366	75.8%	0.01s	666438	32.6%	0.08s
<i>média do grupo MD</i>			69.1%			27.1%	
I_AD_050P_4L_ID1	39229	214431	81.7%	0.00s	68943	43.1%	0.01s
I_AD_050P_4L_ID2	39185	214301	81.7%	0.00s	68798	43.0%	0.01s
I_AD_050P_4L_ID3	38230	214485	82.2%	0.00s	67900	43.7%	0.01s
I_AD_100P_4L_ID1	6171	406005	98.5%	0.00s	96138	93.6%	0.02s
I_AD_100P_4L_ID2	5982	405851	98.5%	0.00s	102007	94.1%	0.02s
I_AD_100P_4L_ID3	5993	406348	98.5%	0.00s	96604	93.8%	0.02s
I_AD_150P_4L_ID1	40115	643225	93.8%	0.00s	175324	77.1%	0.03s
I_AD_150P_4L_ID2	39802	642920	93.8%	0.00s	171848	76.8%	0.03s
I_AD_150P_4L_ID3	39556	643367	93.9%	0.00s	179470	78.0%	0.03s
I_AD_250P_4L_ID1	73370	1076936	93.2%	0.00s	300216	75.6%	0.05s
I_AD_250P_4L_ID2	94422	1077783	91.2%	0.00s	295445	68.0%	0.05s
I_AD_250P_4L_ID3	72002	1077128	93.3%	0.00s	303382	76.3%	0.06s
I_AD_500P_4L_ID1	185858	2140614	91.3%	0.01s	623676	70.2%	0.11s
I_AD_500P_4L_ID2	185784	2140831	91.3%	0.01s	629648	70.5%	0.11s
I_AD_500P_4L_ID3	185809	2140606	91.3%	0.01s	635290	70.8%	0.17s
<i>média do grupo AD</i>			91.6%			71.6%	
<i>média de todos os grupos</i>			57.1%			33.5%	

¹Os tempos são apresentados em segundos.

6.2.2 Qualidade do B&B Truncado x Fase de Melhoramento

O segundo experimento conduzido foi sobre a melhoria na qualidade da solução inicial com o uso da fase de melhoramento. Também foi avaliada a diferença entre o custo final da solução inicial com o uso da fase de melhoramento quando a solução do B&B_T foi gerada sem os mecanismos de ordenação.

A Tabela 6.3 apresenta para cada instância a diferença entre a qualidade da solução gerada no B&B_T (com uso das ordenações) e a solução final após fase de melhoramento. Além do GAP da solução com relação ao *lower bound* (LB) da instância, também são apresentados o custo e o tempo de execução em segundos. Para cada grupo (BD, MD e AD) é apresentada uma linha "*média*", que demonstra a média do GAP de todas as instâncias desse grupo.

A fase de melhoramento tem impacto gradativo conforme o aumento da taxa de aproximação da instância. Para as instâncias de baixa demanda, a redução do custo é pequena, porém para as instâncias de alta demanda há uma redução de, em média, 22,1% no GAP. A melhor performance é gradual devido ao menor número de soluções factíveis que as instâncias de alta demanda possuem. A alta quantidade de soluções infactíveis nessas instâncias dificulta a seleção de uma boa solução inicial no B&B_T, que passa a utilizar os mecanismos de ordenação não mais para redução do custo da solução inicial, mas para viabilizar uma solução factível. Por outro lado, a fase de melhoramento recebe uma solução factível e a quantidade de vizinhos válidos fica reduzida, facilitando a identificação de melhores soluções.

A Tabela 6.4 apresenta para cada instância a diferença entre a qualidade da solução final gerada na fase de melhoramento considerando as duas versões do B&B_T: com o uso dos mecanismos de ordenação (B&B_t^{CO}) e sem tais mecanismos (B&B_t^{SO}).

Mesmo com o uso da fase de melhoramento, o GAP da solução inicial construída com o B&B_t^{SO} não é reduzido ao mesmo GAP daquelas que passaram pelo ordenamento. Além de apresentar piores resultados, o tempo de processamento da fase de melhoramento sem a ordenação também aumenta significativamente.

Tabela 6.3: Resultados da solução inicial do $B\&B_t^{CO}$ e após fase de melhoramento (FM)

Instância	LB	$B\&B_t^{CO}$			FM sobre $B\&B_t^{CO}$		
		Custo	GAP	Tempo	Custo	GAP	Tempo
I_BD_050P_4L_ID1	222400	222400	0.0%	0.01s	222400.0	0.0%	0.00s
I_BD_050P_4L_ID2	222400	222400	0.0%	0.00s	222400.0	0.0%	0.00s
I_BD_050P_4L_ID3	222400	222400	0.0%	0.00s	222400.0	0.0%	0.00s
I_BD_100P_4L_ID1	372320	379995	2.0%	0.02s	379995.0	2.0%	0.00s
I_BD_100P_4L_ID2	372320	379729	2.0%	0.01s	379729.0	2.0%	0.00s
I_BD_100P_4L_ID3	372320	380058	2.0%	0.01s	380058.0	2.0%	0.00s
I_BD_150P_4L_ID1	594720	602229	1.2%	0.02s	602229.0	1.2%	0.00s
I_BD_150P_4L_ID2	594720	600785	1.0%	0.01s	600785.0	1.0%	0.00s
I_BD_150P_4L_ID3	594720	603084	1.4%	0.02s	603084.0	1.4%	0.00s
I_BD_250P_4L_ID1	894560	924809	3.3%	0.03s	914579.6	2.2%	0.07s
I_BD_250P_4L_ID2	894560	923111	3.1%	0.03s	913328.5	2.1%	0.06s
I_BD_250P_4L_ID3	894560	923167	3.1%	0.03s	913913.8	2.1%	0.06s
I_BD_500P_4L_ID1	1849600	1887103	2.0%	0.06s	1875221.5	1.4%	0.14s
I_BD_500P_4L_ID2	1849600	1888640	2.1%	0.07s	1876341.8	1.4%	0.15s
I_BD_500P_4L_ID3	1849600	1888238	2.0%	0.06s	1873247.2	1.3%	0.18s
<i>média do grupo BD</i>			1.7%			1.3%	
I_MD_050P_4L_ID1	65440	79520	17.7%	0.01s	77410.0	15.5%	0.00s
I_MD_050P_4L_ID2	65440	76354	14.3%	0.01s	74432.0	12.1%	0.00s
I_MD_050P_4L_ID3	65440	74808	12.5%	0.01s	73729.0	11.2%	0.00s
I_MD_100P_4L_ID1	70974	114194	37.8%	0.02s	93383.2	24.0%	0.03s
I_MD_100P_4L_ID2	70917	112825	37.1%	0.02s	95855.7	26.0%	0.02s
I_MD_100P_4L_ID3	71025	107840	34.1%	0.02s	90868.8	21.8%	0.02s
I_MD_150P_4L_ID1	191034	259194	26.3%	0.02s	230514.6	17.1%	0.06s
I_MD_150P_4L_ID2	190969	246093	22.4%	0.02s	218864.6	12.7%	0.06s
I_MD_150P_4L_ID3	190925	246054	22.4%	0.02s	218779.2	12.7%	0.06s
I_MD_250P_4L_ID1	262028	357833	26.8%	0.04s	314683.6	16.7%	0.13s
I_MD_250P_4L_ID2	261960	363181	27.9%	0.04s	316503.3	17.2%	0.16s
I_MD_250P_4L_ID3	260971	368635	29.2%	0.04s	321049.8	18.7%	0.16s
I_MD_500P_4L_ID1	449455	661220	32.0%	0.09s	554460.4	18.9%	0.66s
I_MD_500P_4L_ID2	449240	670698	33.0%	0.09s	549982.4	18.3%	0.79s
I_MD_500P_4L_ID3	449214	666438	32.6%	0.09s	556509.1	19.3%	0.67s
<i>média do grupo MD</i>			27.1%			17.5%	
I_AD_050P_4L_ID1	39229	68943	43.1%	0.01s	54002.8	27.4%	0.00s
I_AD_050P_4L_ID2	39185	68798	43.0%	0.01s	56921.2	31.2%	0.00s
I_AD_050P_4L_ID3	38230	67900	43.7%	0.01s	55453.2	31.1%	0.00s
I_AD_100P_4L_ID1	6171	96138	93.6%	0.02s	30116.0	79.5%	0.05s
I_AD_100P_4L_ID2	5982	102007	94.1%	0.02s	33651.6	82.2%	0.05s
I_AD_100P_4L_ID3	5993	96604	93.8%	0.02s	32558.7	81.6%	0.04s
I_AD_150P_4L_ID1	40115	175324	77.1%	0.03s	80965.9	50.5%	0.11s
I_AD_150P_4L_ID2	39802	171848	76.8%	0.03s	76751.2	48.1%	0.10s
I_AD_150P_4L_ID3	39556	179470	78.0%	0.03s	80318.2	50.8%	0.11s
I_AD_250P_4L_ID1	73370	300216	75.6%	0.05s	139671.1	47.5%	0.28s
I_AD_250P_4L_ID2	94422	295445	68.0%	0.05s	142117.0	33.6%	0.26s
I_AD_250P_4L_ID3	72002	303382	76.3%	0.05s	137361.6	47.6%	0.29s
I_AD_500P_4L_ID1	185858	623676	70.2%	0.10s	337858.8	45.0%	1.08s
I_AD_500P_4L_ID2	185784	629648	70.5%	0.11s	334305.2	44.4%	1.08s
I_AD_500P_4L_ID3	185809	635290	70.8%	0.11s	319371.8	41.8%	1.22s
<i>média do grupo AD</i>			71.6%			49.5%	
<i>média de todos os grupos</i>			33.5%			22.8%	

¹Os tempos são apresentados em segundos.²Os valores de custo e tempo das soluções de *FM sobre $B\&B_T^{CO}$* correspondem à média de 10 execuções.

Tabela 6.4: Resultados da solução inicial após fase de melhoramento (FM) com $B\&B_t^{SO}$ e com $B\&B_t^{CO}$

Instância	LB	FM sobre $B\&B_t^{SO}$			FM sobre $B\&B_t^{CO}$		
		Custo	GAP	Tempo	Custo	GAP	Tempo
I_BD_050P_4L_ID1	222400	222400.0	0.0%	0.00s	222400.0	0.0%	0.00s
I_BD_050P_4L_ID2	222400	222400.0	0.0%	0.00s	222400.0	0.0%	0.00s
I_BD_050P_4L_ID3	222400	222400.0	0.0%	0.00s	222400.0	0.0%	0.00s
I_BD_100P_4L_ID1	372320	429362.0	13.3%	0.00s	379995.0	2.0%	0.00s
I_BD_100P_4L_ID2	372320	434750.0	14.4%	0.00s	379729.0	2.0%	0.00s
I_BD_100P_4L_ID3	372320	431479.0	13.7%	0.00s	380058.0	2.0%	0.00s
I_BD_150P_4L_ID1	594720	657263.0	9.5%	0.00s	602229.0	1.2%	0.00s
I_BD_150P_4L_ID2	594720	667306.0	10.9%	0.00s	600785.0	1.0%	0.00s
I_BD_150P_4L_ID3	594720	659142.0	9.8%	0.00s	603084.0	1.4%	0.00s
I_BD_250P_4L_ID1	894560	913890.4	2.1%	0.26s	914579.6	2.2%	0.07s
I_BD_250P_4L_ID2	894560	910617.9	1.8%	0.25s	913328.5	2.1%	0.06s
I_BD_250P_4L_ID3	894560	909391.5	1.6%	0.24s	913913.8	2.1%	0.06s
I_BD_500P_4L_ID1	1849600	1873182.8	1.3%	0.65s	1875221.5	1.4%	0.14s
I_BD_500P_4L_ID2	1849600	1879176.7	1.6%	0.70s	1876341.8	1.4%	0.15s
I_BD_500P_4L_ID3	1849600	1879701.6	1.6%	0.69s	1873247.2	1.3%	0.18s
<i>média do grupo BD</i>			5.4%			1.3%	
I_MD_050P_4L_ID1	65440	117341.8	44.2%	0.00s	77410.0	15.5%	0.00s
I_MD_050P_4L_ID2	65440	118900.7	45.0%	0.00s	74432.0	12.1%	0.00s
I_MD_050P_4L_ID3	65440	118862.4	44.9%	0.00s	73729.0	11.2%	0.00s
I_MD_100P_4L_ID1	70974	116742.2	39.2%	0.14s	93383.2	24.0%	0.03s
I_MD_100P_4L_ID2	70917	116451.2	39.1%	0.14s	95855.7	26.0%	0.02s
I_MD_100P_4L_ID3	71025	116234.9	38.9%	0.14s	90868.8	21.8%	0.02s
I_MD_150P_4L_ID1	191034	237757.7	19.7%	0.32s	230514.6	17.1%	0.06s
I_MD_150P_4L_ID2	190969	242553.4	21.3%	0.32s	218864.6	12.7%	0.06s
I_MD_150P_4L_ID3	190925	237918.5	19.8%	0.32s	218779.2	12.7%	0.06s
I_MD_250P_4L_ID1	262028	357318.5	26.7%	0.90s	314683.6	16.7%	0.13s
I_MD_250P_4L_ID2	261960	352117.5	25.6%	0.92s	316503.3	17.2%	0.16s
I_MD_250P_4L_ID3	260971	356939.1	26.9%	0.88s	321049.8	18.7%	0.16s
I_MD_500P_4L_ID1	449455	658450.2	31.7%	3.90s	554460.4	18.9%	0.66s
I_MD_500P_4L_ID2	449240	656956.5	31.6%	3.67s	549982.4	18.3%	0.79s
I_MD_500P_4L_ID3	449214	665706.2	32.5%	3.72s	556509.1	19.3%	0.67s
<i>média do grupo MD</i>			32.5%			17.5%	
I_AD_050P_4L_ID1	39229	99524.2	60.6%	0.04s	54002.8	27.4%	0.00s
I_AD_050P_4L_ID2	39185	99556.2	60.6%	0.04s	56921.2	31.2%	0.00s
I_AD_050P_4L_ID3	38230	99481.2	61.6%	0.04s	55453.2	31.1%	0.00s
I_AD_100P_4L_ID1	6171	143779.3	95.7%	0.18s	30116.0	79.5%	0.05s
I_AD_100P_4L_ID2	5982	142344.8	95.8%	0.21s	33651.6	82.2%	0.05s
I_AD_100P_4L_ID3	5993	144132.4	95.8%	0.18s	32558.7	81.6%	0.04s
I_AD_150P_4L_ID1	40115	235354.4	83.0%	0.46s	80965.9	50.5%	0.11s
I_AD_150P_4L_ID2	39802	232703.4	82.9%	0.50s	76751.2	48.1%	0.10s
I_AD_150P_4L_ID3	39556	231051.4	82.9%	0.46s	80318.2	50.8%	0.11s
I_AD_250P_4L_ID1	73370	392887.8	81.3%	1.37s	139671.1	47.5%	0.28s
I_AD_250P_4L_ID2	94422	389084.0	75.7%	1.40s	142117.0	33.6%	0.26s
I_AD_250P_4L_ID3	72002	387493.1	81.4%	1.27s	137361.6	47.6%	0.29s
I_AD_500P_4L_ID1	185858	768094.6	75.8%	5.50s	337858.8	45.0%	1.08s
I_AD_500P_4L_ID2	185784	767144.7	75.8%	5.29s	334305.2	44.4%	1.08s
I_AD_500P_4L_ID3	185809	770922.5	75.9%	5.27s	319371.8	41.8%	1.22s
<i>média do grupo AD</i>			79.0%			49.5%	
<i>média de todos os grupos</i>			39.0%			22.8%	

¹Os tempos são apresentados em segundos.

²Os valores de custo e tempo correspondem à média de 10 execuções.

6.2.3 Conclusões sobre a Construção da Solução Inicial

Conforme testes preliminares sobre o VNS proposto, quanto melhor a qualidade da solução inicial, menor o custo associado ao final do processamento da heurística. Com base dos resultados apresentados nas subseções anteriores foi possível definir qual a melhor combinação dos algoritmos propostos fornece a melhor solução inicial.

Essa decisão está refletida no detalhamento da Seção 4.2 e nos testes conduzidos na comparação entre o VNS e CBC da próxima seção, os quais utilizam a versão do B&B_T com uso dos mecanismos de ordenação seguida da fase de melhoramento.

6.3 Análise sobre VNS e CBC

Nesta seção são comparados os resultados do VNS e do CBC para cada um dos grupos de instâncias gerados.

A Tabela 6.5 apresenta para cada instância o custo da melhor solução encontrada para o VNS e para o CBC. A comparação é realizada pelo GAP (%) de cada método com relação ao *lower bound* (BD) de cada instância. Para cada grupo (BD, MD e AD) é apresentada uma linha "*média*" que demonstra a média do GAP de todas as instâncias desse grupo.

Embora o número de médicos da instância influencie na quantidade de iterações que cada método precisa executar, a maior diferença de custo entre o VNS e o CBC se dá pelo tipo de demanda da instância. Nas instâncias de baixa demanda, a diferença de qualidade entre o VNS e o CBC é insignificante. Porém nas instâncias de média e alta demanda, a qualidade das soluções do VNS aumenta gradativamente com relação à taxa de aproximação do *limSupDSA*.

Para as instâncias de alta demanda, o VNS encontra soluções com um GAP 55,8% menor, em média, do que o CBC. Neste grupo, o solver tem maior dificuldade para identificar uma solução factível, o que onera seu processamento e a geração de soluções de boa qualidade.

Tabela 6.5: Resultados do VNS e CBC

Instância	LB	CBC			VNS		
		Custo	GAP	Tempo	Custo	GAP	Tempo
I_BD_050P_4L_ID1	222400	222400	0.0%	2s	222400.0	0.0%	60s
I_BD_050P_4L_ID2	222400	222400	0.0%	2s	222400.0	0.0%	60s
I_BD_050P_4L_ID3	222400	222400	0.0%	2s	222400.0	0.0%	60s
I_BD_100P_4L_ID1	372320	372560	0.1%	1795s	372707.8	0.1%	180s
I_BD_100P_4L_ID2	372320	372560	0.1%	1795s	372692.7	0.1%	180s
I_BD_100P_4L_ID3	372320	372560	0.1%	1795s	372694.8	0.1%	180s
I_BD_150P_4L_ID1	594720	594964	0.0%	1798s	595311.9	0.1%	180s
I_BD_150P_4L_ID2	594720	594964	0.0%	1798s	595333.9	0.1%	180s
I_BD_150P_4L_ID3	594720	594964	0.0%	1798s	595316.8	0.1%	180s
I_BD_250P_4L_ID1	894560	898303	0.4%	3568s	896634.8	0.2%	300s
I_BD_250P_4L_ID2	894560	898303	0.4%	3568s	896537.8	0.2%	300s
I_BD_250P_4L_ID3	894560	898303	0.4%	3568s	896614.0	0.2%	300s
I_BD_500P_4L_ID1	1849600	1853610	0.2%	3548s	1853952.9	0.2%	300s
I_BD_500P_4L_ID2	1849600	1853610	0.2%	3548s	1853846.9	0.2%	300s
I_BD_500P_4L_ID3	1849600	1853610	0.2%	3548s	1853996.6	0.2%	300s
<i>média do grupo BD</i>			0.1%			0.1%	
I_MD_050P_4L_ID1	65440	65692	0.4%	897s	66098.8	1.0%	60s
I_MD_050P_4L_ID2	65440	65692	0.4%	897s	65905.0	0.7%	60s
I_MD_050P_4L_ID3	65440	65692	0.4%	897s	66048.9	0.9%	60s
I_MD_100P_4L_ID1	70974	78299	9.4%	1785s	71972.3	1.4%	180s
I_MD_100P_4L_ID2	70917	78299	9.4%	1785s	72101.6	1.6%	180s
I_MD_100P_4L_ID3	71025	78299	9.3%	1785s	72116.8	1.5%	180s
I_MD_150P_4L_ID1	191034	200536	4.7%	1782s	192728.7	0.9%	180s
I_MD_150P_4L_ID2	190969	200536	4.8%	1782s	192837.5	1.0%	180s
I_MD_150P_4L_ID3	190925	200536	4.8%	1782s	192876.4	1.0%	180s
I_MD_250P_4L_ID1	262028	272405	3.8%	3543s	265537.6	1.3%	300s
I_MD_250P_4L_ID2	261960	272405	3.8%	3543s	265254.0	1.2%	300s
I_MD_250P_4L_ID3	260971	272405	4.2%	3543s	265430.1	1.7%	300s
I_MD_500P_4L_ID1	449455	512816	12.4%	3459s	460309.9	2.4%	300s
I_MD_500P_4L_ID2	449240	512816	12.4%	3459s	459875.1	2.3%	300s
I_MD_500P_4L_ID3	449214	512816	12.4%	3459s	460323.3	2.4%	300s
<i>média do grupo MD</i>			6.2%			1.4%	
I_AD_050P_4L_ID1	39229	NE ¹	100.0%	900s	41854.2	6.3%	60s
I_AD_050P_4L_ID2	39185	NE ¹	100.0%	900s	42010.0	6.7%	60s
I_AD_050P_4L_ID3	38230	NE ¹	100.0%	900s	41845.1	8.6%	60s
I_AD_100P_4L_ID1	6171	36094	82.9%	1753s	13449.3	54.1%	180s
I_AD_100P_4L_ID2	5982	36094	83.4%	1753s	13584.6	56.0%	180s
I_AD_100P_4L_ID3	5993	36094	83.4%	1753s	13701.4	56.3%	180s
I_AD_150P_4L_ID1	40115	80745	50.3%	1768s	46948.1	14.6%	180s
I_AD_150P_4L_ID2	39802	80745	50.7%	1768s	46816.6	15.0%	180s
I_AD_150P_4L_ID3	39556	80745	51.0%	1768s	47220.7	16.2%	180s
I_AD_250P_4L_ID1	73370	133031	44.8%	3404s	84377.2	13.0%	300s
I_AD_250P_4L_ID2	94422	133031	29.0%	3404s	83954.7	-12.5%	300s
I_AD_250P_4L_ID3	72002	133031	45.9%	3404s	83999.2	14.3%	300s
I_AD_500P_4L_ID1	185858	NE ¹	100.0%	3600s	212033.0	12.3%	300s
I_AD_500P_4L_ID2	185784	NE ¹	100.0%	3600s	211230.1	12.0%	300s
I_AD_500P_4L_ID3	185809	NE ¹	100.0%	3600s	211148.9	12.0%	300s
<i>média do grupo AD</i>			74.8%			19.0%	
<i>média de todos os grupos</i>			27.0%			6.9%	

¹NE indica que não foi encontrada solução factível dentro do tempo limite.²Os tempos são apresentados em segundos.³Os valores de custo e tempo do VNS correspondem à média de 10 execuções.

6.3.1 Conclusões sobre o VNS Proposto

Com base nos resultados apresentados, o uso da heurística proposta garante melhores resultados para instâncias com alta taxa de aproximação do *limSupDSA*. Para instâncias com menor taxa de aproximação, os resultados apresentados pelo solver são equiparáveis aos do VNS. Entretanto, o método exato não garante a solução ótima e demanda um tempo de execução superior ao da heurística.

Sugere-se que o uso de um método exato ou de heurística deve observar duas constantes durante o escalonamento: o tempo disponível e a taxa de aproximação do *limSupDSA*. Se há disponibilidade para uma execução com um ou mais dias de duração e a taxa de aproximação é baixa, vale-se executar o método exato a fim de se identificar a solução ótima para a instância. Porém, se necessária uma solução de boa qualidade em poucos minutos ou, se a instância é muito restrita, recomendá-se o uso da heurística.

7 CONCLUSÕES

Algoritmos que automatizem o processo de escalonamento de médicos têm sido cada vez mais aplicados em situações práticas dos hospitais ao redor do mundo. Este trabalho teve por objetivo fornecer uma solução heurística para o problema de escalonamento de médico aplicado ao HCPA. As principais contribuições são: (i) a implementação e distribuição de um gerador de instâncias, (ii) o estudo sobre outra variante das restrições do HCPA, (iii) o desenvolvimento de uma heurística VNS e (iv) a comparação entre o VNS proposto e um método exato para diferentes tipos de instâncias.

O gerador de instâncias considera características que variam mensalmente no HCPA, permitindo ao usuário determinar alocações fixas, ausências, número de médicos, entre outros parâmetros. Todas as instâncias foram geradas no mesmo formato de entrada, sendo similar ao utilizado no INRC-II (CESCHIA et al., 2015).

O VNS proposto usa três vizinhanças e utiliza estruturas de dados que agilizam o cálculo da função objetivo. Sua solução inicial é construída em duas fases a fim de garantir uma solução factível e de boa qualidade através de um B&B truncado, seguido por uma fase de melhoramento.

Os experimentos avaliaram o impacto na qualidade da melhor solução encontrada entre o VNS proposto e um método exato com execução via CBC. A análise foi conduzida sobre o impacto de cada método para três diferentes grupos de instâncias que se distinguem pela taxa de aproximação do limite superior de médicos demandado por dia/turno/área ($limSupDSA$). A qualidade das soluções via VNS foi superior ao do CBC nas instâncias com maior taxa de aproximação do $limSupDSA$ (média e alta demanda). Enquanto as instâncias de baixa demanda apresentaram resultados sem diferença significativa entre os dois métodos.

REFERÊNCIAS

- ADAMS, T.; O’SULLIVAN, M.; WALKER, C. Physician rostering for workload balance. **Operations Research for Health Care**, Elsevier, v. 20, p. 01–10, 2019.
- AVANTHAY, C.; HERTZ, A.; ZUFFEREY, N. A variable neighborhood search for graph coloring. **European Journal of Operational Research**, Elsevier, v. 151, n. 2, p. 379–388, 2003.
- BRUNI, R.; DETTI, P. A flexible discrete optimization approach to the physician scheduling problem. **Operations Research for Health Care**, Elsevier, v. 3, n. 4, p. 191–199, 2014.
- BRUNNER, J. O.; BARD, J. F.; KOLISCH, R. Flexible shift scheduling of physicians. **Health care management science**, Springer, v. 12, n. 3, p. 285–305, 2009.
- BRUNNER, J. O.; EDENHARTER, G. M. Long term staff scheduling of physicians with different experience levels in hospitals using column generation. **Health care management science**, Springer, v. 14, n. 2, p. 189–202, 2011.
- BURKE, E. et al. Variable neighborhood search for nurse rostering problems. In: **Metaheuristics: computer decision-making**. [S.l.]: Springer, 2003. p. 153–172.
- CESCHIA, S. et al. Second international nurse rostering competition (inrc-ii)—problem description and rules—. **arXiv preprint arXiv:1501.04177**, 2015.
- ERHARD, M. et al. State of the art in physician scheduling. **European Journal of Operational Research**, Elsevier, v. 265, n. 1, p. 01–18, 2018.
- GENDREAU, M. et al. Physician scheduling in emergency rooms. In: SPRINGER. **International Conference on the Practice and Theory of Automated Timetabling**. [S.l.], 2006. p. 53–66.
- GUNAWAN, A.; LAU, H. C. Master physician scheduling problem. **Journal of the Operational Research Society**, Taylor & Francis, v. 64, n. 3, p. 410–425, 2013.
- HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search for the p-median. **Location Science**, Elsevier, v. 5, n. 4, p. 207–226, 1997.
- HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. **European journal of operational research**, Elsevier, v. 130, n. 3, p. 449–467, 2001.
- INFÜHR, J.; RAIDL, G. R. Grasp and variable neighborhood search for the virtual network mapping problem. In: SPRINGER. **International Workshop on Hybrid Metaheuristics**. [S.l.], 2013. p. 159–173.
- KYTÖJOKI, J. et al. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. **Computers & operations research**, Elsevier, v. 34, n. 9, p. 2743–2757, 2007.
- MANSINI, R.; ZANOTTI, R. Optimizing the physician scheduling problem in a large hospital ward. **Journal of Scheduling**, Springer, p. 01–25, 2019.

MEISTER, T. C.; WICKERT, T. I.; BURIOL, L. S. Solving a physician rostering problem. In: IEEE. **2020 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2020. p. 01–07.

MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997.

RAIS, A.; VIANA, A. Operations research in healthcare: a survey. **International transactions in operational research**, Wiley Online Library, v. 18, n. 1, p. 01–31, 2011.

SANCHOTENE, T.; BURIOL, L. S. Abordagem heurística para solução do problema de alocação de médicos do hcpa. **Universidade Federal do Rio Grande do Sul**, 2018.

SCHOENFELDER, J.; PFEFFERLEN, C. Decision support for the physician scheduling process at a german hospital. **Service Science**, INFORMS, v. 10, n. 3, p. 215–229, 2018.

WICKERT, T. I. et al. An integer programming approach for the physician rostering problem. **Annals of Operations Research**, Springer, p. 01–28, 2020.