

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RENAN LEON GARCIA

**Abordagem híbrida para o problema de escalonamento de
enfermeiros**

Maringá

2016

RENAN LEON GARCIA

Abordagem híbrida para o problema de escalonamento de enfermeiros

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Ademir Aparecido Constantino

Maringá
2016

Dados Internacionais de Catalogação na Publicação (CIP)
(Biblioteca Central - UEM, Maringá, PR, Brasil)

G216a Garcia, Renan Leon
Abordagem híbrida para o problema de
escalonamento de enfermeiros / Renan Leon Garcia. --
Maringá, 2016.
90 f. : il. (algumas color.), figs., tabs.

Orientador: Prof. Dr. Ademir Aparecido
Constantino.

Dissertação (mestrado) - Universidade Estadual de
Maringá, Centro de Tecnologia, Departamento de
Informática, Programa de Pós-Graduação em Ciência da
Computação, 2016.

1. Problema de escalonamento de enfermeiros -
Atribuição de serviços. 2. Escalonamento -
Enfermeiros - Meta-heurística. 3. Algoritmo
heurístico - Escalonamento - Enfermeiros. 4. Método
de busca local - Escalonamento - Enfermeiros. I.
Constantino, Ademir Aparecido, orient. II.
Universidade Estadual de Maringá. Centro de
Tecnologia. Departamento de Informática. Programa de
Pós-Graduação em Ciência da Computação. III. Título.

CDD 21.ed. 003.1

AMMA-003398

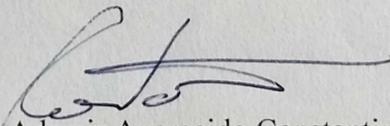
FOLHA DE APROVAÇÃO

RENAN LEON GARCIA

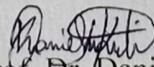
Abordagem híbrida para o problema de escalonamento de enfermeiros

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Banca Examinadora composta pelos membros:

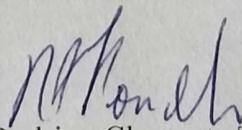
BANCA EXAMINADORA



Prof. Dr. Ademir Aparecido Constantino
Universidade Estadual de Maringá – DIN/UEM



Prof. Dr. Daniel Kikuti
Universidade Estadual de Maringá – DIN/UEM



Prof. Dr. Rodrigo Clemente Thom de Souza
Universidade Federal do Paraná – UFPR-Jandaia do Sul

Aprovada em: 13 de dezembro de 2016.

Local da defesa: Sala 120, Bloco C56, *campus* da Universidade Estadual de Maringá.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me abençoado nesta caminhada e por permitir que tudo fosse realizado.

Agradeço em especial a minha mãe Márcia, meu pai João, meus irmãos Rodolfo e Rafael, e minha esposa Thielle que sempre me incentivaram, apoiaram, acreditaram em mim e não me deixaram desistir nos momentos mais difíceis.

Ao meu orientador Prof. Dr. Ademir Aparecido Constantino pelo apoio, comentários e sugestões no desenvolvimento deste projeto e aos demais professores das disciplinas que cursei.

A secretária do PCC Inês, que sempre me apoiou e ajudou com todas documentações necessárias durante todo o curso.

Ao meu colega de laboratório Alexandre Cassimiro que me ajudou em vários momentos durante todo o projeto e a todos que direta ou indiretamente contribuíram para a realização deste trabalho.

E a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido a este trabalho.

Abordagem híbrida para o problema de escalonamento de enfermeiros

RESUMO

Esta dissertação aborda o Problema de Escalonamento de Enfermeiros (PEE), o qual consiste em construir uma escala de trabalho para enfermeiros de um hospital buscando satisfazer suas preferências e as restrições impostas pelo problema. Para cada enfermeiro é designada uma sequência de turnos que abrange o período todo da escala. O PEE é um problema de otimização classificado como Não Polinomial Difícil (NP-difícil). Ele tem fomentado o desenvolvimento de diversos modelos e algoritmos, heurísticos e meta-heurísticas, devido a grande dificuldade de obter-se uma boa solução. O presente trabalho propõe uma abordagem híbrida para o PEE envolvendo uma variação da meta-heurística *Variable Neighborhood Search* (VNS), combinando método heurístico para explorar a vizinhança, com métodos exatos para gerar solução vizinha. Como procedimentos de busca local foram utilizados três métodos, denominados Processo de Corte e Recombinação (PCR), *k-swap* e *Cover Fit*. Para a validação da proposta é utilizada uma base de dados com 24 instâncias reais do problema. Os resultados obtidos apresentaram que a abordagem proposta trabalha consistentemente bem para as instâncias de tamanho pequeno/médio, no qual os resultados obtidos são comparados com resultados apresentados por outros trabalhos que utilizaram a mesma base de dados. **Palavras-chave:** Problema de Escalonamento de Enfermeiros. Problema de Atribuição. Heurística. Meta-heurística.

Hybrid approach for nurse scheduling problem

ABSTRACT

This dissertation addresses a Nurses Scheduling Problem (NSP), which consists in constructing a working schedule for nurses in a hospital taken account their preferences and the constraints imposed by the problem. For each nurse is assigned a sequence of shifts covering the entire schedule period. The NSP is an optimization problem classified as Non Polynomial Hard (NP-hard). It has fostered the development of several models and algorithms, heuristics and meta-heuristics, due to the great difficulty of getting a good solution. This work proposes a hybrid approach to NSP involving a variation of Variable Neighborhood Search (VNS) meta-heuristic, combining heuristic method to explore the neighborhood, with exact methods to generate neighbor solution. As local search process three methods were used to implement the neighborhood structures, called Cut and Recombination Process (CRP), k-swap and Cover Fit. To validate the proposal is used a database with 24 real-world instances of the problem. The results obtained showed that the proposed approach performs consistently well for small/medium size instances, where the results obtained are compared with results presented by other works that used the same database.

Keywords: Nurse Scheduling Problem. Assignment Problem. Heuristic. Meta-heuristic.

LISTA DE FIGURAS

Figura 3.1	Resolução da matriz de custos utilizando PA.	24
Figura 3.2	Grafo multipartido para 4 enfermeiros em um período de 7 dias. .	26
Figura 3.3	Pseudocódigo meta-heurística VNS.	28
Figura 3.4	Funcionamento básico da meta-heurística VNS.	29
Figura 3.5	Funcionamento básico da meta-heurística VND.	31
Figura 3.6	Funcionamento básico da meta-heurística CNS.	32
Figura 3.7	Pseudocódigo meta-heurística GVNS.	33
Figura 3.8	Grafo com camadas.	33
Figura 3.9	Estrutura da matriz de custos E construída na execução do PCR. .	34
Figura 3.10	Pseudocódigo PCR.	35
Figura 3.11	Solução inicial e possíveis recombinações no corte 3.	35
Figura 3.12	Escala após o corte e recombinação no corte 3.	35
Figura 3.13	Estrutura da matriz de custos construída na execução do k -swap. .	37
Figura 3.14	Pseudocódigo k -swap.	37
Figura 3.15	Solução inicial e possíveis recombinações no corte 4 para 2 -swap. .	38
Figura 3.16	Escala após o corte e recombinação no corte 4 para 2 -swap. . . .	38
Figura 4.1	Estrutura da matriz de custos para os PA da fase construtiva. . .	49
Figura 4.2	Pseudocódigo Fase Construtiva.	50
Figura 4.3	Heurística baseada na técnica <i>best improvement</i>	51
Figura 4.4	Heurística baseada na técnica <i>first improvement</i>	52
Figura 4.5	Procedimento PCR_rand.	54
Figura 4.6	Procedimento k -swap_rand.	54
Figura 4.7	Solução corrente.	56
Figura 4.8	Estrutura da matriz de custos H_D construída na execução do CF no dia 4.	56
Figura 4.9	Procedimento <i>Cover Fit</i>	57
Figura 4.10	Procedimento PF.	58
Figura 4.11	Procedimento KF.	59
Figura 4.12	Procedimento CFF.	60
Figura 5.1	Evolução do custo da solução obtida por C-PFKFCFF para a instância 5.	69
Figura 5.2	Comparativo entre as técnicas VND x CNS para as instâncias de 1 à 7.	71

LISTA DE TABELAS

Tabela 2.1	Escala semanal de enfermeiros.	18
Tabela 2.2	Declaração de preferências dos enfermeiros.	20
Tabela 3.1	Pseudocódigo dos métodos VND e CNS.	30
Tabela 3.2	Características das bases de dados para o PEE.	45
Tabela 3.3	Classificações das exigências do PEE.	46
Tabela 4.1	Pseudocódigo dos métodos VNS e M-VNS.	53
Tabela 4.2	Estruturas de vizinhança para perturbação da escala.	53
Tabela 4.3	Notações para os procedimentos de busca local.	58
Tabela 4.4	Versões do HPAM.	60
Tabela 4.5	Instâncias da base de dados da Universidade de Nottingham.	65
Tabela 5.1	Redução de custo por meio dos procedimentos de melhoria.	67
Tabela 5.2	Melhores resultados obtidos para as instâncias de 1 à 7.	68
Tabela 5.3	Melhores resultados do HPAM e resultados da literatura.	73
Tabela 5.4	Relação entre os resultados do HPAM e os resultados da literatura.	75

LISTA DE SIGLAS E ABREVIATURAS

- AE:** Algoritmo Evolucionário
AG: Algoritmos Genéticos
AM: Algoritmo Memético
ASAP: *Automated Scheduling Optimisation and Planning*
CF: *Cover Fit*
CNS: *Cyclic Neighborhood Search*
EM: *Eletromagnetic Meta-heuristic*
GRASP: *Greedy Random Adaptative Search Procedure*
GVNS: *General Variable Neighborhood Search*
MHS: *Multimodal Homecare Scheduling*
NSP: *Nurse Scheduling Problem*
PA: Problema de Atribuição
PAM: Problema de Atribuição Multinível
PCR: Processo de Corte e Recombinação
PEE: Problemas de Escalonamento de Enfermeiros
PEP: Problemas de Escalonamento de Pessoal
PO: Problemas de Otimização
RF: Restrições Flexíveis
RR: Restrições Rígidas
SAP: *Shortest Augmenting Path*
VDS: *Variable Depth Search*
VND: *Variable Neighborhood Descent*
VNS: *Variable Neighborhood Search*

SUMÁRIO

1	Introdução	11
1.1	Considerações Iniciais	11
1.2	Objetivos e Contribuições	13
1.3	Motivação e Justificativas	13
1.4	Organização do Texto	14
2	Descrição do Problema	15
2.1	Problemas de Otimização	15
2.2	Problema de Escalonamento de Pessoal	16
2.3	Problema de Escalonamento de Enfermeiros	18
3	Revisão da Literatura	22
3.1	Problema de Atribuição	22
3.1.1	Problema de Atribuição Multinível	25
3.2	Grafo Multipartido	25
3.3	Meta-heurística VNS	26
3.3.1	Meta-heurísticas VND e CNS	29
3.4	Procedimento PCR	32
3.5	Procedimento <i>k-swap</i>	36
3.6	Trabalhos Relacionados	39
4	Proposta	47
4.1	Algoritmo Proposto	47
4.1.1	Fase Construtiva	48
4.1.2	Fase de Melhoramento	50
4.1.3	<i>Cover Fit</i>	55
4.1.4	Versões do HPAM	57
4.2	Instâncias Utilizadas	64
5	Experimentos computacionais e análise dos resultados	66
5.1	Desempenho dos procedimentos de melhoria	66
5.2	Resultados das variações do algoritmo HPAM	68
5.3	Comparação com resultados da literatura	72
5.4	Considerações finais	76

6 Conclusão	77
6.1 Trabalhos futuros	79
REFERÊNCIAS	80
A Apêndice A	84
B Apêndice B	89

Introdução

1.1 Considerações Iniciais

Problemas de Escalonamento são problemas de otimização que possuem como objetivo alocar um conjunto de eventos ou recursos ao longo de um período determinado de tempo, atendendo às restrições e exigências impostas por cada tipo de escalonamento, como escalas de pessoas, máquinas, veículos, entre outros.

O Problema de Escalonamento de Pessoal (PEP) é um dos problemas clássicos de escalonamento que vem sendo abordado por pesquisadores a mais de 40 anos (Burke *et al.*, 2004). Ele consiste em construir uma escala de trabalho periódica para trabalhadores de uma empresa ou instituição. O PEP é classificado por (Chuin Lau, 1996) como NP-difícil, indicando que na literatura não são conhecidos algoritmos exatos que resolvem esse problema em tempo polinomial. Essa característica é um incentivo à investigação de novas abordagens que permitam solucionar de forma eficiente e efetiva esses problemas.

Entre as diversas áreas que o PEP abrange, encontra-se o problema de elaboração de escalas de trabalho para enfermeiros de um hospital. O Problema de Escalonamento de Enfermeiros (PEE) tem sido bastante explorado pois, além das características mencionadas de um PEP, ele possui algumas particularidades, como a busca pelo atendimento às preferências de cada enfermeiro por turnos de trabalho. Se tratando de um ambiente de trabalho crítico como um hospital, a satisfação dos empregados é imprescindível, refletindo diretamente na qualidade dos serviços prestados. Dessa maneira, escalas de trabalho com maior atendimento às preferências resultam em uma maior motivação e satisfação dos enfermeiros.

Em geral, os problemas envolvendo a elaboração de escalas de trabalho são resolvidos manualmente, sendo uma tarefa custosa e de difícil execução. Os métodos automatizados para construção de escalas de enfermeiros são uma das maneiras para solucionar esses problemas com maior facilidade e agilidade. Burke *et al.* (2004) afirmam que os métodos automatizados possuem um grande potencial no aprimoramento do processo de construção das escalas de trabalho e na melhoria da qualidade das mesmas. Métodos matemáticos ou heurísticos podem rapidamente gerar um grande número de soluções e informar a qualidade das escalas. Dentre os benefícios da automação do processo de construção de escalas de enfermeiros está o considerável aproveitamento de tempo pelo comitê administrativo responsável pela execução dessa tarefa, bem como a maior satisfação dos enfermeiros para com os turnos designados.

O PEE, também conhecido como *Nurse Scheduling Problem* (NSP), consiste em elaborar escalas de trabalho periódicas para equipes de enfermeiros de um hospital respeitando uma grande variedade de restrições trabalhistas e operacionais. Além disso, a resolução desse problema busca maximizar o atendimento das preferências dos enfermeiros na elaboração de cada escala. Para a construção de uma escala de enfermeiros, um turno de trabalho, incluindo o turno folga, é associado para cada enfermeiro em cada dia da escala, de modo a maximizar a qualidade total dessa escala. A qualidade de uma escala é mesurada pela máxima satisfação das preferências dos enfermeiros e pelo mínimo de violações das restrições impostas pelo problema (Cares e Riff, 2012).

De acordo com Petrovic e Berghe (2008), os objetivos mais comuns buscados para a resolução do PEE são:

- Redução do número de violações e restrições;
- Redução da quantidade de enfermeiros;
- Redução de horas extras;
- Aumento da abrangência da escala de trabalho;
- Aumento da satisfação das preferências;

Contudo, encontrar uma solução para o PEE é uma tarefa árdua que exige tempo e processamento. Segundo Burke *et al.* (2004), a aplicação de métodos exatos de otimização para o PEE é dificultada pela falta de conhecimento sobre a estrutura do problema, bem como pelo seu tamanho. Assim, uma das maneiras de encontrar soluções para esse problema é a utilização de métodos heurísticos. Em geral, não existe a garantia de otimalidade nas soluções obtidas por esses métodos como ocorre nos métodos exatos.

Porém, eles são capazes de alcançar boas soluções em um tempo razoável, podendo assim, ser um tempo aceitável (Burke *et al.*, 2004).

Dessa maneira, este trabalho busca apresentar uma proposta de pesquisa para solucionar o PEE, levando em consideração as restrições exigidas pelo problema e as preferências de cada funcionário. Para isso, é apresentada uma abordagem híbrida, que combina métodos exatos e heurísticos em busca de uma solução.

1.2 Objetivos e Contribuições

O principal objetivo deste trabalho é investigar um algoritmo híbrido para o problema de escalonamento de enfermeiros.

Os objetivos específicos são:

- a) Propor um algoritmo híbrido para a resolução do problema de escalonamento de enfermeiros que utiliza os procedimentos de busca local Processo de Corte e Recombinação (PCR), *k-swap* e *Cover Fit* em conjunto com uma variante da meta-heurística *Variable Neighborhood Search* (VNS);
- b) Investigar o desempenho da abordagem propostas com instâncias da base de dados de Nottingham;
- c) Comparar as soluções obtidas pela abordagem proposta com trabalhos da literatura que utilizaram a mesma base de dados;

Dentre as contribuições deste trabalho está um novo método para a resolução do PEE, o qual faz uso de uma nova metodologia computacional. Além disso, serão disponibilizados resultados da aplicação desse algoritmo em uma base de dados com instâncias do PEE que poderão ser comparados com resultados de novos algoritmos.

1.3 Motivação e Justificativas

Como mencionado anteriormente, a obtenção de soluções viáveis para o PEE em tempo aceitável é importante por se tratar de um problema dinâmico, que envolve custos relacionados à escala de trabalho e considera as restrições trabalhistas e satisfação dos enfermeiros. Assim, torna-se fundamental o desenvolvimento, avaliação e utilização de novos métodos para tal resolução, considerando que não são conhecidos procedimentos na literatura que garantam solução ótima em tempo de processamento aceitável.

Os algoritmos presentes na literatura que objetivam solucionar o PEE apontam que a utilização de métodos exatos combinado com heurísticos podem resultar em modelos promissores, permitindo encontrar soluções para o PEE com baixos custos e em um tempo computacional aceitável.

Na literatura não foram encontrados trabalhos que abordem a implementação da abordagem proposta neste trabalho, o que demonstra ser uma importante lacuna e oportunidade de pesquisa a ser explorada no contexto do Problema de Escalonamento de Enfermeiros.

1.4 Organização do Texto

O texto está organizado como segue: No tópico 2 é definido o Problema de Escalonamento de Enfermeiros abordado neste trabalho, detalhando conceitos básicos e nomenclaturas. Nesse mesmo capítulo também são apresentados alguns trabalhos presentes na literatura relacionados à este. No tópico 3 é apresentada uma revisão de literatura descrevendo o Problema de Atribuição, Problema de Atribuição Multinível, grafo multipartido como forma de representação do PEE, a meta-heurística *Variable Neighborhood Search* (VNS), *Variable Neighborhood Descent* (VND) e *Cyclic Neighborhood Search* (CNS), bem como procedimentos de busca local utilizados em estruturas de vizinhança, denominados Processo de Corte e Recombinação (PCR) e *k-swap*. O tópico 4 descreve a proposta deste trabalho, detalhando a abordagem desenvolvida juntamente com os procedimentos utilizados para sua construção. No tópico 5 são apresentados os experimentos realizados com uma base de dados do PEE e os resultados alcançados, realizando comparações entre as diferentes implementações e os resultados obtidos por algoritmos da literatura. O tópico 6 finaliza com as conclusões e propostas para trabalhos futuros.

Descrição do Problema

Este tópico apresenta a definição, conceitos e nomenclaturas dos Problemas de Otimização (PO), Problema de Escalonamento de Pessoal (PEP) e Problema de Escalonamento de Enfermeiros (PEE) na seção 2.1, 2.2 e 2.3 respectivamente.

2.1 Problemas de Otimização

Segundo Tanomaru (1995), uma grande parte dos problemas científicos podem ser modelados como problemas de busca e otimização, sendo que existe uma dualidade entre os conceitos de busca e otimização, no qual todo problema de busca pode ser considerado um problema de otimização e vice-versa. Existe uma série de fatores que influenciam o desempenho de um sistema, podendo assumir um número limitado ou ilimitado de valores que podem estar sujeitos a algumas restrições. O objetivo principal dos PO é encontrar uma combinação de fatores que proporcione o melhor desempenho possível para o sistema em questão. O conjunto de todas as combinações possíveis para os fatores constitui o espaço de busca pela melhor solução.

Seja uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e um espaço de busca $S \subseteq \mathbb{R}^n$, o problema de otimização pode ser definido como na equação (1).

$$\text{Maximizar } f(x) | x \in S \quad (1)$$

Também pode ser escrito da forma apresentada na equação (2).

$$\text{encontrar } x^* | f(x^*) \geq f(x), \forall x \in S \quad (2)$$

A função $f(\cdot)$ pode ser derivada ou não, uni ou multidimensional, e o espaço de busca S pode ser contínuo ou discreto, finito ou infinito, côncavo ou convexo. Tanomaru (1995) classifica os métodos de otimização em probabilísticos, numéricos e enumerativos, havendo ainda um grande número de métodos híbridos.

Os métodos probabilísticos apontam a ideia de uma busca probabilística. Em contrapartida, os métodos numéricos podem ser divididos em duas categorias: baseados em cálculo numérico ou analíticos. Os métodos baseados em cálculo numérico podem fazer uso de técnicas de pesquisa operacional quando o espaço de busca é linear e técnicas de gradiente ou estatística quando não é linear. Os métodos analíticos são aplicáveis quando a função $f(\cdot)$ é explicitamente conhecida e derivável ou pode ser aproximada por alguma função derivável até o grau desejado de precisão. Por fim, os métodos enumerativos de otimização examinam cada ponto do espaço de busca com o objetivo de encontrar pontos ótimos. Essa ideia pode se tornar inviável quando existe um número infinito ou muito grande de pontos a serem examinados.

Para os PO são necessários métodos com uma boa capacidade de exploração do espaço de busca quando objetiva-se encontrar um mínimo ou máximo global. Na literatura são encontrados diversos métodos para resolução de diferentes PO. Pode-se dizer que o método que apresenta melhor resultado utilizando o mínimo de recursos no menor tempo possível é o método mais eficiente para o problema. Se dois métodos apresentam o mesmo resultado para um problema com a utilização dos mesmos recursos, pode-se considerar o método mais eficiente aquele que encontrou a solução em menor tempo.

Dentre os diversos PO presentes na literatura, o Problema de Escalonamento, classificado como NP-difícil, é bastante abordado por pesquisadores devido a sua complexidade de resolução. O escalonamento ou programação de recursos tem um papel muito importante em processos da manufatura e indústria. Eles ocorrem na medida em que necessita-se alocar recursos, como máquinas e trabalhadores, à tarefas em períodos de tempo pré determinados, almejando otimizar um ou mais objetivos (Pinedo, 2012). Existem diversas ramificações de problemas de escalonamento, dentre as quais encontram-se os Problemas de Escalonamento de Pessoal descrito na próxima seção.

2.2 Problema de Escalonamento de Pessoal

O PEP é um problema de otimização que se baseia na construção de escalas de trabalho periódicas para trabalhadores de uma empresa ou instituição, abrangendo um período de dias determinado pelo problema, podendo ser desde 7 até 365 dias. No processo de construção das escalas devem ser levado em consideração diversas restrições trabalhistas e

operacionais, de modo que a escala gerada atenda os requisitos exigidos por cada problema (Caprara *et al.*, 2003). Devido a grande dificuldade de obter resultados ótimos para o PEP, Chuin Lau (1996) classifica esse problema como NP-difícil.

Esse problema está presente em diferentes atividades organizacionais que necessitam designar turnos de trabalho aos trabalhadores em um horizonte de tempo. Mesmo com diversos mecanismos automatizados para desenvolver escalas de trabalho, ainda é comum encontrar instituições que realizam essa difícil tarefa de forma manual, podendo demorar dias para que fiquem prontas. Além disso, se faz necessário a utilização de um funcionário específico para essa função.

Por se tratar de recursos humanos das empresas, o PEP envolve altos custos, sendo fundamental o desenvolvimento de métodos automatizados que o solucionem de forma eficaz. O grupo de pesquisa *Automated Scheduling Optimisation and Planning* (ASAP, 2014) apresenta algumas vantagens da utilização de métodos automatizados para a construção de escalas de trabalho, conforme listado a seguir:

- **Redução de custos:** em uma escala construída por meio de métodos automatizados não serão alocados funcionários além do necessário, assim os custos dos salários podem ser menores.
- **Redução do absenteísmo e atrasos:** uma escala mal elaborada pode causar insatisfação e fadiga nos funcionários, acarretando em faltas e atrasos, comprometendo assim as atividades da empresa.
- **Atendimento às preferências:** atender às preferências dos funcionários permite que eles programem melhor seus cotidianos, fazendo com que a moral do grupo cresça e a rotatividade diminua.
- **Melhoria na qualidade dos serviços:** com melhores condições de trabalho a produtividade e a qualidade dos serviços aumentam, dessa maneira os funcionários trabalham mais motivados, diminuindo as chances de erros provocados pelo cansaço e estresse.

O PEP é um problema que possui diversas variações, como a elaboração de escalas de trabalho para seguranças de uma empresa, motoristas, atendentes, médicos, enfermeiros e para diversas outras funções. Dentre as variações do PEP, muitos trabalhos da literatura buscam explorar o PEP pois, além da complexidade presente no PEP, se tratar de um problema que envolve uma grande quantidade de restrições e possui diversas abordagens que podem auxiliar a busca pela melhor solução viável. Dessa forma, a próxima seção descreve de forma mais detalhada o PEP.

2.3 Problema de Escalonamento de Enfermeiros

O PEE consiste em elaborar escalas de trabalho para enfermeiros de um hospital de modo que as preferências dos trabalhadores por turnos de trabalho sejam atendidas da melhor maneira possível. O problema envolve diversas restrições que devem ser observadas na elaboração de cada sequência de turnos. Devido a complexidade do problema, assim como o PEP, o PEE é classificado por Osogami e Imai (2000) como NP-difícil.

Abaixo são apresentados alguns conceitos específicos que envolvem o entendimento do PEE:

- **Turno:** se refere ao horário de trabalho e a quantidade de horas seguidas trabalhadas. Geralmente são divididos em turnos de 8 horas, 10 horas, 12 horas ou folga;
- **Tarefa:** é a necessidade de ser atribuído um certo turno a um determinado dia de trabalho;
- **Período:** horizonte de dias coberto pela escala elaborada.
- **Jornada:** sequência de turnos que cobre todo o período da escala de um determinado enfermeiro. São todos os turnos que o enfermeiro terá que cumprir na escala construída.
- **Escala:** conjunto de jornadas de todos os enfermeiros que cobre todo o período de dias de trabalho. Uma escala completa contém os dias e turnos de todos os enfermeiros.

Na Tabela 2.1 é apresentado um exemplo de escala semanal de enfermeiros. Ela contém 4 enfermeiros, distribuídos em três diferentes turnos de 8 horas e turno folga (#) em um período de 7 dias. O turno *D* começa às 07:00 e termina às 15:00, o turno *E* começa às 15:00 e termina às 23:00 e o turno *L* começa às 23:00 e termina às 07:00.

Tabela 2.1: Escala semanal de enfermeiros.

Funcionários	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7
Enfermeiro 1	<i>D</i>	#	<i>L</i>	<i>L</i>	#	<i>D</i>	<i>E</i>
Enfermeiro 2	#	<i>E</i>	<i>E</i>	<i>D</i>	<i>D</i>	#	<i>L</i>
Enfermeiro 3	<i>L</i>	<i>L</i>	#	<i>E</i>	<i>E</i>	<i>L</i>	#
Enfermeiro 4	<i>E</i>	<i>D</i>	<i>D</i>	#	<i>L</i>	#	<i>D</i>

Fonte: Elaborada pelo autor.

As restrições que são envolvidas pelo PEE podem ser divididas em duas categorias: rígidas e flexíveis. As restrições rígidas estão relacionadas a impedimentos firmemente definidos impostos pelo problema, os quais devem ser atendidos em uma escala de trabalho. As restrições flexíveis são exigências potencialmente desejáveis a satisfação na construção da escala, no qual seus valores são ajustáveis de acordo com as características que se almeja encontrar na escala a ser elaborada. Ambos os tipos de restrições são discriminados a seguir.

Restrições Rígidas:

- a) proibição de alguns tipos diferentes de turnos sucessivos;
- b) proibição de atribuir mais que um turno por dia para cada enfermeiro;
- c) número máximo de atribuição de cada tipo de turno para cada enfermeiro;
- d) número mínimo de tempo que cada enfermeiro pode trabalhar na escala;
- e) número máximo de tempo que cada enfermeiro pode trabalhar na escala;
- f) número mínimo de dias consecutivos de trabalho;
- g) número máximo de dias consecutivos de trabalho;
- h) número mínimo de dias consecutivos de folga;
- i) número máximo de finais de semana trabalhados;
- j) dias de folga que não podem ser atribuídos nenhum tipo de turno de trabalho;

Restrições Flexíveis:

- a) preferência para ser atribuído à um tipo de turno em um dia específico;
- b) preferência para não ser atribuído à um tipo de turno em um dia específico;
- c) demanda de enfermeiros exigido em cada dia da escala para cada tipo de turno;

As preferências dos enfermeiros pelos turnos são declaradas antecipadamente. Cada enfermeiro pode escolher os dias que deseja ou não trabalhar, indicando ainda o tipo de turno desejado. Cada turno escolhido é associado a um custo, sendo esse valor atribuído ao custo total da escala se a preferência do enfermeiro não for atendida.

Um exemplo de declaração de preferências é apresentado na Tabela 2.2. Esse exemplo envolve uma escala de 4 enfermeiros, 7 dias da semana e 2 turnos de 8 horas (*D*, *E*). Cada enfermeiro escolhe o dia e o turno que quer ou não trabalhar e atribui um valor

numérico de 1 à 3 de acordo com o peso de preferência, onde o valor 3 possui maior peso. Esse valor numérico será adicionado ao custo total da escala final caso a preferência não seja atendida.

Cada preferência declarada é composta de 3 itens seguindo a seguinte ordem: tipo de turno (D ou E); preferência para trabalhar ou não nesse turno (t = trabalhar; n = não trabalhar); peso da preferência (de 1 à 3).

Tabela 2.2: Declaração de preferências dos enfermeiros.

Funcionários	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7
Enfermeiro 1	$D/t/3$	$D/n/1$			$E/t/1$		$E/t/2$
Enfermeiro 2		$D/t/2$		$E/n/3$	$D/n/1$		
Enfermeiro 3	$D/n/1$	$D/n/2$	$D/t/1$		$E/n/3$		
Enfermeiro 4		$E/t/2$	$E/n/1$		$E/t/3$	$D/n/t$	$E/n/3$

Fonte: Elaborada pelo autor.

Para os dias em que não existe preferência de turno são atribuídos o valor 0 para a designação ou não de qualquer turno nesse dia. Dessa maneira, para o enfermeiro 1 no dia 3 da Tabela 2.2 é indiferente a atribuição do turno D , E ou folga, sendo que qualquer atribuição está relacionada ao custo 0.

Para os dias em que um enfermeiro deseja trabalhar em um turno, a atribuição do mesmo resulta em custo 0, enquanto a atribuição de qualquer outro turno resulta no peso declarado pelo enfermeiro. Por exemplo, se for atribuído o turno D para o enfermeiro 2 no dia 2 da escala não será adicionado nenhum custo à solução. Porém, se for atribuído o turno E ou folga para o mesmo enfermeiro no mesmo dia, será adicionado o custo 2 à solução.

De maneira inversa, para os dias em que um enfermeiro deseja não trabalhar em um turno, a atribuição do mesmo resulta no custo declarado por ele, enquanto a atribuição de qualquer outro turno resulta em custo 0. Assim, se for atribuído o turno D para o enfermeiro 3 no dia 1 da escala, o custo 1 será adicionado à solução. Em contrapartida, a atribuição do turno E ou folga para o mesmo enfermeiro no mesmo dia, não resultará na adição de custo à solução.

Assim como para as preferências, custos são empregados para que determinadas restrições sejam atendidas. Cada restrições (rígida ou flexível) não atendida implica na inclusão do custo associado à essa restrição ao custo total da solução. Dessa maneira, o PEE pode ser definido como um problema de minimização dos custos associados às preferências de cada enfermeiro pelos turnos e dos custos resultantes do não atendimentos às restrições estabelecidas pelo problema.

De maneira mais formal, o PEE é definido como a necessidade de alocar um conjunto N de enfermeiros em um período T de dias, atribuindo uma sequência de turnos a cada trabalhador, formando a jornada de cada um. Em cada dia compreendido pelo período T , os enfermeiros devem ser atribuídos a um tipo de turno pertencente ao conjunto S de turnos. Assim, tem-se:

N: Conjunto de enfermeiros, índice i ($i = 1, \dots, n$), $n = |N|$;

T: Conjunto de dias do período de escalonamento, índice j ($j = 1, \dots, t$), $t = |T|$;

S: Conjunto de turnos, índice l ($l = 1, \dots, s$), $s = |S|$;

Por se tratar de um problema de escalonamento, os conceitos utilizados neste trabalho podem ser facilmente aplicados a outros problemas de escalonamento de pessoal. Dentre as principais aplicações estão incluídos problemas com geração de escalas de atividades dinâmicas, no qual o horário de trabalho varia em cada dia, e problemas em que as sequências de turnos não sejam fixas.

Revisão da Literatura

Neste tópico são apresentados alguns conceitos, modelos matemáticos e algoritmos utilizados na resolução de PEE. Na seção 3.1 é apresentada a descrição do problema de atribuição e do problema de atribuição multinível. Um modo de representar uma escala de trabalho com a utilização de grafo multipartido é apresentado na seção 3.2. Uma descrição da meta-heurística VNS, VND e CNS é apresentada na seção 3.3. Dois procedimentos de busca local, chamados Processo de Corte e Recombinação (PCR) e *k-swap*, são apresentados na seção 3.4 e 3.5, respectivamente. Por fim, na seção 3.6 são descritos alguns dos trabalhos presentes na literatura que buscam encontrar soluções do PEE utilizando diferentes métodos exatos, heurísticos e meta-heurísticos. Além disso, nessa mesma seção são apresentadas duas bases de dados com instâncias do problema de escalonamento de enfermeiros presentes na literatura.

3.1 Problema de Atribuição

O Problema de Atribuição (PA), também conhecido como *Assignment Problem*, é um problema de otimização que objetiva encontrar uma permutação que relacione cada linha de uma matriz de custos a uma coluna específica, de modo que a soma dos custos de todas as designações seja a menor possível. (Hillier, 2001)

Uma matriz de custos é definida como uma matriz quadrada na qual cada célula contém o custo de atribuição entre uma linha e uma coluna referente à célula. O problema de atribuição é amplamente utilizado na modelagem de problemas de escalonamento. O exemplo mais comum é a situação de atribuir tarefas para funcionários. Para esses casos,

as linhas da matriz representam aos trabalhadores, as colunas representam as tarefas, onde o valor de cada célula indica o custo de atribuir uma determinada tarefa para um funcionário. Dessa maneira, como cada trabalhador recebe apenas uma tarefa, o problema possui um único nível.

A solução do PA equivale ao emparelhamento perfeito de um custo mínimo em um grafo bipartido. Assim, seja uma matriz de custos com dimensões $n \times n$, o problema consiste em associar cada linha i a uma coluna j , de modo a minimizar a soma dos custos.

A formulação matemática do PA é modelada por Carpaneto e Toth (1987) por meio das expressões (3) à (6).

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij} \quad (3)$$

$$\text{Sujeito a: } \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (4)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (5)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, n; \quad j = 1, \dots, n \quad (6)$$

Onde se considera:

c_{ij} Custo de se atribuir a coluna j à linha i ;

x_{ij} Atribuição ou não da coluna j à linha i , assumindo 1 ou 0, respectivamente; e

n Ordem da matriz de custos;

A função-objetivo (3) deve ser minimizada, na qual cada designação de uma coluna j a uma linha i é multiplicada pelo respectivo custo da atribuição; a restrição (4) exige que para cada linha da matriz seja associada à uma única coluna; a restrição (5) determina que cada coluna da matriz seja associada a uma única linha; a restrição (6) impõe que a atribuição entre linhas e colunas assuma apenas valores de decisão 0 e 1.

Na Figura 3.1 é exemplificado uma matriz de custos para o problema de associar 4 turnos à 4 enfermeiros. Os elementos em negrito em cada coluna representam as designações que relacionam enfermeiros e tarefas que fazem parte da solução ótima para esse problema.

De maneira mais precisa, Hillier (2001) afirma que o PA deve satisfazer as seguintes hipóteses:

Figura 3.1: Resolução da matriz de custos utilizando PA.

	Turno 1	Turno 2	Turno 3	Turno 4
Enfermeiro 1	2	4	3	7
Enfermeiro 2	11	8	1	3
Enfermeiro 3	5	12	0	4
Enfermeiro 4	7	9	4	6

Fonte: Elaborada pelo autor.

- O número de funcionários e o número de tarefas devem ser iguais, representados por n ;
- Deve ser atribuído exatamente uma tarefa para cada funcionário;
- Cada tarefa deve ser realizada por um único funcionário;
- Existe um custo associado ao funcionário i ($i = 1, \dots, n$) executar a tarefa j ($j = 1, \dots, n$); e
- O objetivo é determinar como as n atribuições devem ser feitas de maneira a minimizar o custo total da solução.

Os problemas que não possuem uma quantidade igual de funcionários e tarefas podem ser facilmente reformulados e balanceados por meio da adição de tarefas fictícias, caso o número de funcionários seja maior que o número de tarefas, ou de funcionários fictícios, em caso contrário.

Para a resolução do PA, Kuhn (1955) desenvolveu um algoritmo de otimização chamado Método Húngaro, o qual é amplamente utilizado por pesquisadores. Com a utilização do algoritmo proposto por Kuhn (1955) combinado com mais um procedimento, Carpaneto e Toth (1987) propõem um novo algoritmo para a resolução do PA. Esse algoritmo combina o Método Húngaro com o procedimento *Shortest Augmenting Path* (SAP) garantindo a obtenção da solução ótima com complexidade assintótica igual a $O(n^3)$.

Os autores afirmam que a diferença entre os métodos está na atualização das variáveis do problema dual e na busca do caminho aumentativo de custo mínimo. O Método Húngaro considera somente elementos com valor igual a zero, enquanto o SAP considera todos os valores relacionados. Mais precisamente, o algoritmo proposto é baseado no SAP que explora as características de procurar caminhos aumentativos de custo reduzido igual a zero do Método Húngaro.

O modelo do PA apresentado anteriormente trabalha com apenas um nível ao relacionar cada linha da matriz de custos a uma coluna. Dessa maneira, ele é utilizado apenas em grafos bipartido, como designar tarefas para trabalhadores em um único dia da escala de trabalho. Contudo, uma adaptação desse mesmo modelo pode ser utilizado para a elaboração de escalas completas que cubram vários dias de trabalho. Essa adaptação é feita empregando o PA de um nível como um subproblema de um problema de vários níveis, denominado Problema de Atribuição Multinível.

3.1.1 Problema de Atribuição Multinível

O Problema de Atribuição Multinível (PAM) é um problema que envolve dois ou mais níveis, os quais são resolvidos com base em resoluções do PA atuando em seus subproblemas. Em uma escala de trabalho, os múltiplos níveis representam os diversos dias que formam as jornadas dos funcionários (Constantino *et al.*, 2014).

Na elaboração das sequências de turnos, o PEE é abordado como um PAM, o qual se baseia em sucessivas resoluções de PA. Cada PA envolve encontrar uma solução para dois níveis do PAM, atuando em um grafo bipartido. Dessa maneira, o modelo se baseia em pares de níveis justapostos que precisam ser interligados. As interligações são obtidas por meio da resolução de uma matriz de custos em que as linhas representam um nível e as colunas, outro. Assim, o PA correspondente é resolvido e um sequenciamento entre os dois níveis é obtido.

Para uma escala de trabalho de 7 dias são necessários 7 sucessivas resoluções de PAs, uma para cada dia da escala. Ao final de todas as resoluções, o PAM está resolvido e a jornada de cada enfermeiro está completa.

3.2 Grafo Multipartido

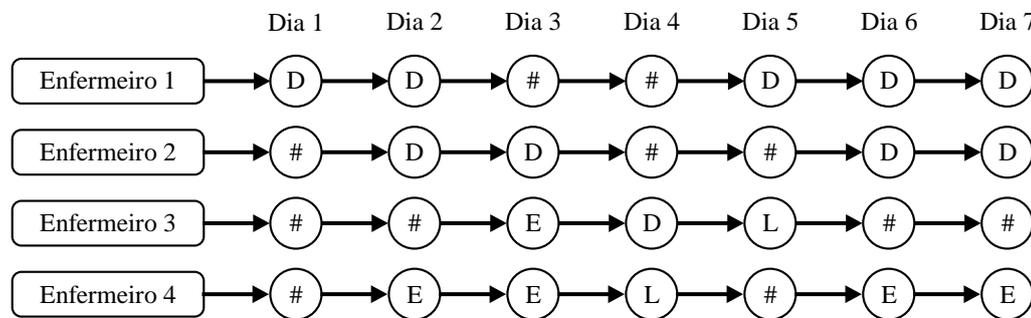
Os grafos multipartido são utilizados no PEE como uma forma de representação das escalas de trabalhos a serem construídas. Dessa maneira, a escala de trabalho pode ser melhor manipulada na aplicação de métodos de busca local que buscam melhorar o custo total da escala. Constantino *et al.* (2014) representam a solução do PEE como grafos multipartido acíclico com $t + 1$ partições, no qual os vértices da primeira partição correspondem ao conjunto de enfermeiros e os vértices das partições restantes correspondem ao conjunto das tarefas, sendo uma partição para cada dia da escala.

Seja um grafo $G = (V, A)$, no qual V representa o conjunto dos vértices e A o conjunto de arestas. O conjunto P é formado pelas partições $(P_0, P_1, P_2, \dots, P_t)$, no qual P_0 é

composto pelo conjunto de vértices que representam os enfermeiros, enquanto as partições P_1 à P_t são compostas pelo conjunto de vértices que representam os turnos em cada dia. O objetivo é encontrar n caminhos, da primeira à última partição, de modo que o custo total seja minimizado, sendo que a jornada de cada enfermeiro é representada por um caminho.

Na Figura 3.2 é apresentado um exemplo de representação do PEE com $n = 4$ enfermeiros e $t = 7$ dias em um grafo multipartido, no qual cada coluna de vértices representa uma partição. Cada aresta representa uma possível atribuição de um turno para um enfermeiro em um determinado dia, no qual não existe uma aresta conectando dois vértices de uma mesma partição. Cada vértice da segunda à última partição representa um turno de trabalho, onde as letras (D, E, L) representam diferentes tipos de turno e o símbolo especial (#) representa o turno folga. Uma sequência de arestas que conectam um vértice n da primeira partição (correspondente ao enfermeiro n) a um vértice na última partição indica uma sequência de turnos atribuídos à esse enfermeiro. Cada turno atribuído à um enfermeiro possui um custo relacionado às suas preferências e às restrições impostas pelo problema.

Figura 3.2: Grafo multipartido para 4 enfermeiros em um período de 7 dias.



Fonte: Elaborada pelo autor.

3.3 Meta-heurística VNS

A meta-heurística *Variable Neighborhood Search* (VNS), também conhecida como Pesquisa em Vizinhança Variável, é uma abordagem que faz uso de um método de busca local explorando o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhança. Proposto por Mladenović e Hansen (1997), esse método é amplamente utilizado para resolução de problemas de otimização combinatória.

Segundo Hansen *et al.* (2010), a meta-heurística VNS explora sistematicamente a ideia de mudança de estrutura de vizinhança com inclinação para mínimos locais, de modo a fugir dos valores que os contêm. A VNS se baseia nas seguintes observações:

- a) um mínimo local obtido por meio de uma estrutura de vizinhança não é necessariamente o mesmo obtido por outra estrutura de vizinhança;
- b) o mínimo global é um mínimo local obtido por todas as possíveis estruturas de vizinhança;
- c) para muitos problemas, os mínimos locais obtidos por uma ou várias estruturas de vizinhança são relativamente próximos uns dos outros;

A observação c) implica que um ótimo local pode fornecer algum tipo de informação sobre o ótimo global. Dessa maneira, um estudo mais detalhado sobre a vizinhança desse ótimo local pode ser realizado até que um possível resultado melhor seja encontrado. Diferentemente de outras meta-heurísticas baseadas em métodos de busca local, o método VNS não segue uma trajetória simples, mas explora as vizinhanças gradativamente mais distantes da solução corrente e focaliza a busca em torno de uma nova solução somente se um movimento de melhora é realizado. Esse processo funciona como um mecanismo de diversificação da busca (Carvalho *et al.*, 2003).

O procedimento VNS parte de uma solução inicial criada anteriormente. Então, é feita uma perturbação da solução corrente por meio de diferentes estruturas de vizinhança, o que pode resultar na degradação da solução. Porém, esse mecanismo permite escapar de ótimos locais. Em seguida, é executado um processo de busca local que intensifica a busca por melhoria na solução corrente. Sempre que um procedimento de busca local é aplicado e uma nova solução melhorada é encontrada, essa nova solução passa a ser a solução corrente. Em contrapartida, se o procedimento de busca local não consegue encontrar melhorias na solução corrente, altera-se o mecanismo de perturbação utilizando outra estrutura de vizinhança. Assim, o VNS é executado até que algum critério de parada seja satisfeito.

Seja s^* uma solução inicial e r o número de estruturas de vizinhança N_k , ($k = 0, 1, \dots, r - 1$), na Figura 3.3 é apresentado o pseudocódigo do método VNS.

É importante observar que a solução s_1 gerada na linha 6 é um vizinho qualquer da solução corrente s_0 dentro da estrutura de vizinhança N_k , onde a solução s_1 não é necessariamente melhor que a solução s_0 . A solução s_2 é obtida por meio da aplicação de um método de busca local na solução s_1 . O método de busca local trabalha explorando a solução s_1 de maneira a encontrar melhorias no custo total da solução.

Figura 3.3: Pseudocódigo meta-heurística VNS.

```

1: Procedimento Variable-Neighborhood-Search( $s^*$ )
2:    $s_0 \leftarrow s^*$ 
3:   Enquanto condição de parada não satisfeita faça
4:      $k \leftarrow 0$ 
5:     Enquanto  $k < r$  faça
6:       Gere um vizinho qualquer  $s_1 \in N_k$  a partir de  $s_0$ 
7:        $s_2 \leftarrow$  procedimento Busca-Local ( $s_1$ )
8:       Se custo da solução  $s_2 <$  custo da solução  $s_0$  então
9:          $s_0 \leftarrow s_2$ 
10:         $k \leftarrow 0$ 
11:       Senão
12:          $k \leftarrow k + 1$ 
13:   Retorne  $s_0$ 

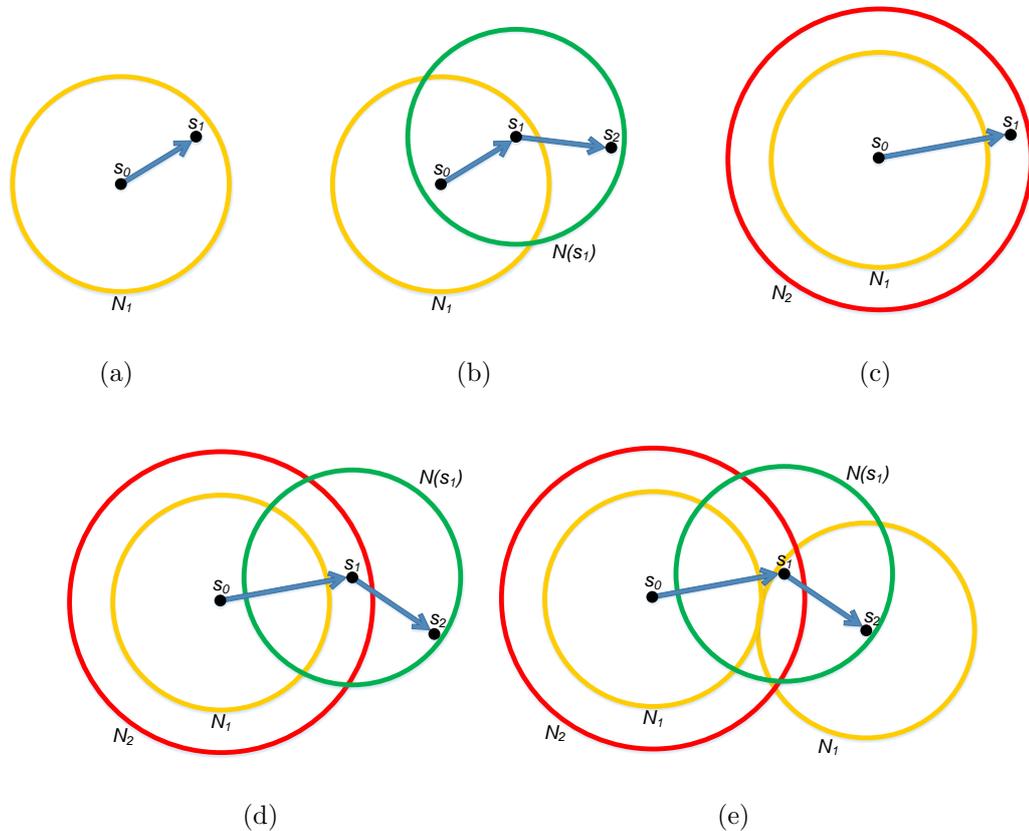
```

Fonte: Mladenović e Hansen (1997).

Na Figura 3.4 é apresentado um exemplo do funcionamento básico da meta-heurística VNS. Na Figura 3.4(a) o método recebe como entrada uma solução inicial que passa a ser a solução corrente s_0 , em seguida é gerado um vizinho qualquer s_1 a partir de s_0 dentro da primeira estrutura de vizinhança N_1 . Na Figura 3.4(b) é aplicado um procedimento de busca local na solução s_1 de modo a encontrar o melhor vizinho s_2 a partir de s_1 . Nesse exemplo, o custo da solução s_2 não é menor que o custo da solução corrente s_0 , então o método continua com a solução s_0 como a solução corrente e utiliza a próxima estrutura de vizinhança N_2 para encontrar um novo vizinho qualquer s_1 a partir de s_0 , assim como mostrado na Figura 3.4(c). Na Figura 3.4(d) é aplicado um procedimento de busca local na solução s_1 buscando encontrar o melhor vizinho s_2 . Nesse caso, a solução s_2 encontrada possui um custo menor que o custo da solução corrente s_0 , assim a solução s_2 passa a ser a solução corrente e o método retorna para a primeira estrutura de vizinhança N_1 , como mostrado na Figura 3.4(e). Esse procedimento é executado até que todas as estruturas de vizinhanças sejam exploradas sem encontrar melhoria na solução corrente.

Uma variação da meta-heurística VNS, chamada *Variable Neighborhood Descent* (VND), foi proposta por Hansen e Mladenović (2001) de modo que o VND explora soluções em busca de um ótimo local, enquanto o VNS busca um ótimo global. Além disso, um outro método que explora soluções em busca de um ótimo local é apresentado por Deng e Bard (2011), chamado *Cyclic Neighborhood Search* (CNS).

Figura 3.4: Funcionamento básico da meta-heurística VNS.



Fonte: Elaborada pelo autor.

3.3.1 Meta-heurísticas VND e CNS

O método VND, assim como o método VNS, explora o espaço de soluções por meio de trocas sistemáticas de vizinhança, buscando encontrar a melhor solução local. Ele trabalha com um conjunto de estruturas de vizinhança no qual é utilizado uma estrutura a cada iteração, chamada estrutura de vizinhança corrente.

O procedimento inicia com uma estrutura de vizinhança principal que objetiva melhorar a solução corrente. Caso a estrutura de vizinhança não consiga encontrar melhorias na solução corrente, ela é trocada por uma estrutura de vizinhança mais distante, seguindo uma ordem de estruturas de vizinhança predefinidas. Sempre que uma solução melhor é encontrada por uma estrutura de vizinhança, o procedimento retorna para a estrutura principal. O fim da execução do VND é dado após utilizar todas as estruturas de vizinhança sem apresentar melhoria na solução corrente (Hansen e Mladenović, 2001).

O método CNS trabalha de maneira similar ao VND. Ele tem como principal objetivo a busca pela melhor solução local por meio de trocas sistemáticas de vizinhanças. Assim como o VND, o CNS trabalha com um conjunto de estruturas de vizinhanças, sendo utilizado uma por iteração. Porém, independente se a estrutura de vizinhança corrente encontrou melhorias ou não, ela é sempre trocada pela próxima estrutura, trabalhando de maneira cíclica até que não seja possível melhorar a solução corrente.

Seja a solução s_1 e 7 diferentes estruturas de vizinhança ($r = 7$). De maneira a enfatizar as diferenças entre os métodos, na Tabela 3.1 é apresentado o pseudocódigo das meta-heurísticas VND e CNS.

Tabela 3.1: Pseudocódigo dos métodos VND e CNS.

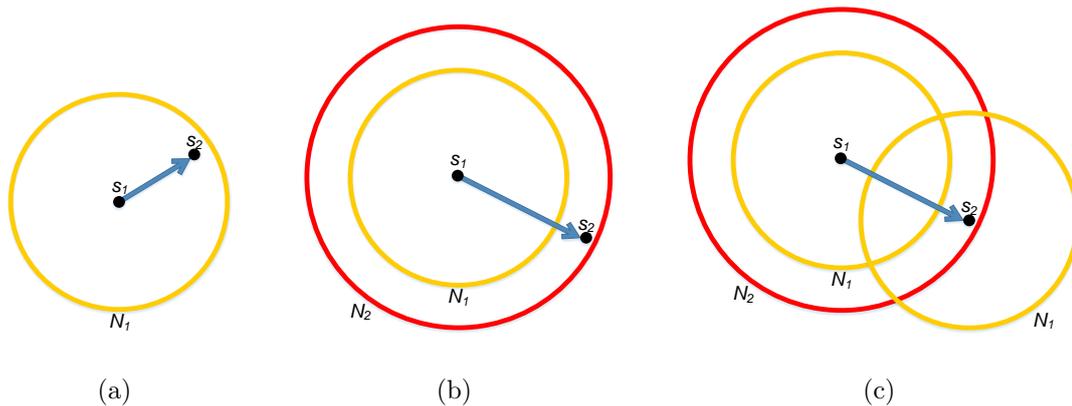
<i>Variable Neighborhood Descent</i>	<i>Cyclic Neighborhood Search</i>
1: Procedimento VND(s_1)	1: Procedimento CNS(s_1)
2: $k \leftarrow 1$	2: $k \leftarrow 1$
3: Enquanto $k \leq r$ faça	3: $melhora \leftarrow f$
4: Encontre o melhor vizinho $s_2 \in N_k$ a partir de s_1	4: Enquanto $k \leq r$ faça
5: Se custo da solução $s_2 <$ custo da solução s_1 então	5: Encontre o melhor vizinho $s_2 \in N_k$ a partir de s_1
6: $s_1 \leftarrow s_2$	6: Se custo da solução $s_2 <$ custo da solução s_1 então
7: $k \leftarrow 1$	7: $s_1 \leftarrow s_2$
8: Senão	8: $melhora \leftarrow t$
9: $k \leftarrow k + 1$	9: $k \leftarrow k + 1$
10: Retorne s_1	10: Se ($k > r$) e ($melhora = t$) então
	11: $k \leftarrow 1$
	12: $melhora \leftarrow f$
	13: Retorne s_1

Fonte: Hansen e Mladenović (2001) e Deng e Bard (2011).

Na Figura 3.5 é apresentado um exemplo do funcionamento básico da meta-heurística VND. Na Figura 3.5(a) o método recebe como entrada uma solução inicial utilizada como a solução corrente s_1 , em seguida é aplicado um procedimento de busca local nessa solução corrente de modo a encontrar o melhor vizinho s_2 dentro da primeira estrutura de vizinhança N_1 . Nesse exemplo, o custo da solução s_2 não é menor que o custo da solução corrente s_1 , então o método continua com a solução s_1 como a solução corrente e utiliza a próxima estrutura de vizinhança N_2 para encontrar o melhor vizinho s_2 dentro dessa vizinhança, como mostrado na Figura 3.5(b). Nesse caso, a solução s_2 encontrada possui um custo menor que o custo da solução corrente s_1 , assim a solução s_2 passa a ser

a solução corrente e o método retorna para a primeira estrutura de vizinhança N_1 , como mostrado na Figura 3.5(c). Esse procedimento é executado até que todas as estruturas de vizinhanças sejam exploradas sem encontrar melhoria na solução corrente.

Figura 3.5: Funcionamento básico da meta-heurística VND.

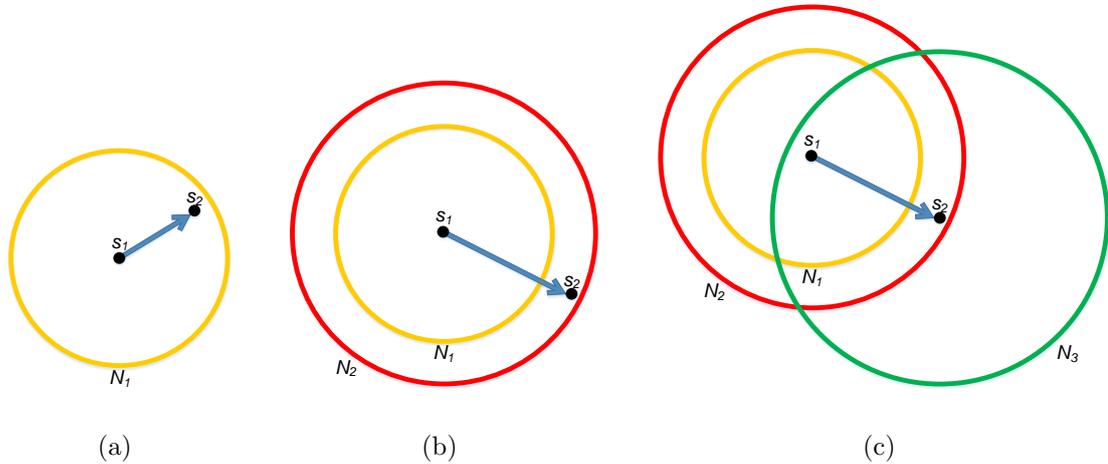


Fonte: Elaborada pelo autor.

Assim como descrito anteriormente, a principal diferença entre o método VND e o método CNS está no modo como cada método trabalha a troca de estruturas de vizinhanças. Na Figura 3.6 é apresentado um exemplo do funcionamento básico da meta-heurística CNS. Na Figura 3.6(a) o método recebe como entrada uma solução inicial utilizada como a solução corrente s_1 , em seguida é aplicado um procedimento de busca local nessa solução corrente de modo a encontrar o melhor vizinho s_2 dentro da primeira estrutura de vizinhança N_1 . Nesse exemplo, o custo da solução s_2 não é menor que o custo da solução corrente s_1 , então o método continua utilizando a solução s_1 como a solução corrente. No método CNS, após encontrar o melhor vizinho dentro de uma estrutura de vizinhança N_k , essa estrutura de vizinhança é sempre trocada pela próxima estrutura de vizinhança N_{k+1} (com exceção da última vizinhança que é trocada pela primeira). Assim, na Figura 3.6(b) é utilizado a próxima estrutura de vizinhança N_2 para encontrar o melhor vizinho s_2 dentro dessa vizinhança. Nesse caso, a solução s_2 encontrada possui um custo menor que o custo da solução corrente s_1 , assim a solução s_2 passa a ser a solução corrente e o método utiliza a próxima estrutura de vizinhança N_3 , como mostrado na Figura 3.6(c). Esse procedimento é executado até que todas um ciclo de estruturas de vizinhanças (da primeira à última estrutura) seja executado sem encontrar melhoria na solução corrente.

A aplicação de diferentes operadores de busca local para a implementação das estruturas de vizinhanças (Linha 4, VND; Linha 5, CNS) é de grande importância. Dessa

Figura 3.6: Funcionamento básico da meta-heurística CNS.



Fonte: Elaborada pelo autor.

maneira, a combinação do VND ou CNS com esses operadores tende a aumentar as chances de encontrar melhorias na solução corrente.

Com o objetivo de intensificar a busca por melhorias na solução corrente, Hansen *et al.* (2010) introduziram uma nova meta-heurística, chamada *General Variable Neighborhood Search* (GVNS). Essa meta-heurística utiliza o VNS como procedimento principal, aplicando o VND como método de busca local do VNS (Figura 3.3, Linha 7), no qual o VND trabalha como intensificador da busca por um ótimo global.

Seja s^* uma solução inicial e r o número de estruturas de vizinhança presentes em N_k , ($k = 0, 1, \dots, r - 1$). Na Figura 3.7 é apresentado o pseudocódigo da meta-heurística GVNS.

Além do método VND, o CNS também pode ser aplicado como método de busca local utilizado pelo VNS, intensificando a busca por melhorias na solução corrente (Rendl *et al.*, 2012).

3.4 Procedimento PCR

O Processo de Corte e Recombinação (PCR) proposto por Constantino *et al.* (2014) é um procedimento de busca local que visa o melhoramento de uma solução inicial para problemas de escalonamento. Para um problema de escalonamento qualquer, seja uma solução inicial descrita como um grafo $G = (V, A)$, onde V é o conjunto de vértices que representam as tarefas e A o conjunto de arcos a_{ij} que indicam a possibilidade de um

Figura 3.7: Pseudocódigo meta-heurística GVNS.

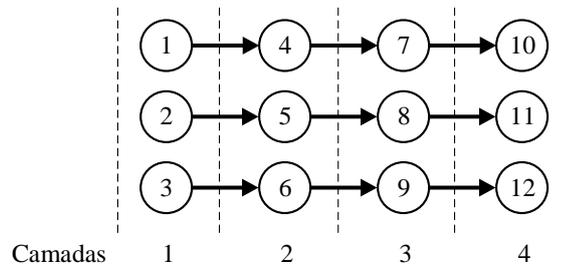
```

1: Procedimento General-Variable-Neighborhood-Search( $s^*$ )
2:    $s_0 \leftarrow s^*$ 
3:   Enquanto condição de parada não satisfeita faça
4:      $k \leftarrow 0$ 
5:     Enquanto  $k < r$  faça
6:       Gere um vizinho qualquer  $s_1 \in N_k$  a partir de  $s_0$ 
7:        $s_2 \leftarrow$  procedimento VND ( $s_1$ )
8:       Se custo da solução  $s_2 <$  custo da solução  $s_0$  então
9:          $s_0 \leftarrow s_2$ 
10:         $k \leftarrow 0$ 
11:       Senão
12:          $k \leftarrow k + 1$ 
13:   Retorne  $s_0$ 

```

Fonte: Hansen *et al.* (2010).

funcionário realizar uma tarefa j após a tarefa i . Os vértices estão dispostos em camadas de maneira que, cada camada é composta por vértices que só podem ser sucedidos por vértices da camada seguinte, conforme mostrado na Figura 3.8.

Figura 3.8: Grafo com camadas.

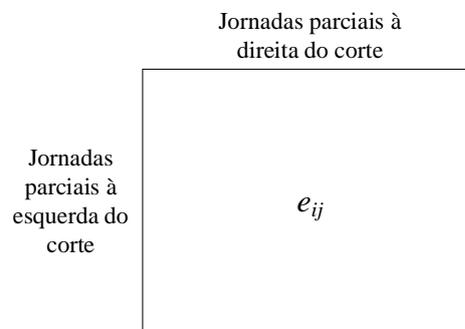
Fonte: Elaborada pelo autor.

Considerando uma escala com m camadas, o PCR consiste em efetuar $m - 1$ cortes entre duas camadas da escala, dividindo as jornadas dos funcionários em duas partes, jornada parcial à esquerda e jornada parcial à direita. A camada 1 é composta pelos vértices que apresentam os funcionários, enquanto as camadas de 2 à m são formadas por vértices que representam as tarefas dispostas nos dias da escala. Dessa maneira, um corte no dia j pode ser definido como um corte realizado antes do dia j . Após cada corte realizado na escala, é construído uma matriz de custos $E = [e_{ij}]$ de dimensão $n \times n$

referente às recombinações das n jornadas parciais à esquerda com as n jornadas parciais à direita.

Na matriz E , as jornadas parciais à esquerda são representadas pelas linhas, enquanto as jornadas parciais à direita são representadas pelas colunas da matriz. Cada elemento e_{ij} representa o custo de se associar uma jornada parcial à esquerda i com uma jornada parcial à direita j . O custo dos elementos da matriz são calculados de acordo com as restrições impostas por cada problema de escalonamento. A Figura 3.9 ilustra a estrutura da matriz de custos E construída na execução do método PCR.

Figura 3.9: Estrutura da matriz de custos E construída na execução do PCR.



Fonte: Constantino *et al.* (2014).

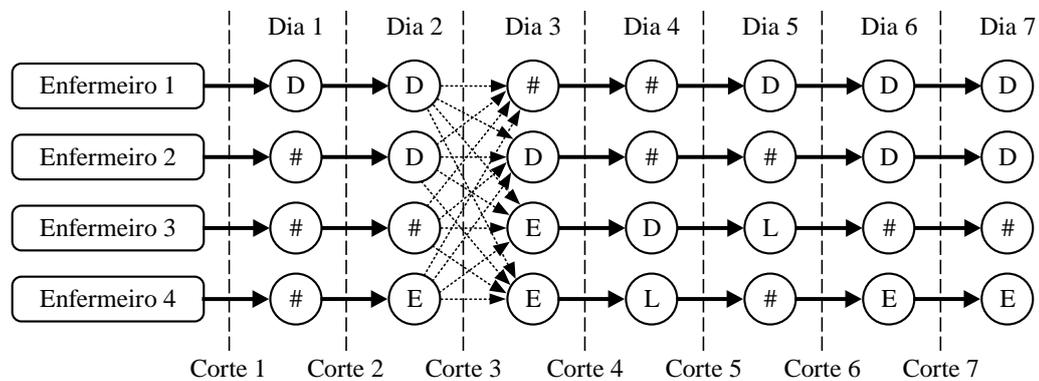
Obtida a matriz de custos E , o PA correspondente é resolvido e as jornadas são recombinadas, formando novas jornadas completas. Conforme afirmado por Carpaneto e Toth (1987), como o algoritmo que resolve o PA entre dois níveis é exato e garante solução ótima, o custo da solução diminui ou se mantém igual em cada corte e recombinação, nunca piora. Cada iteração do PCR consiste em efetuar t cortes e recombinações, um entre os funcionários e suas jornadas e $t - 1$ cortes e recombinações entre os dias da escala, considerando uma escala com período de t dias. O PCR pode ser implementado em duas direções, do dia 1 ao dia t ou do dia t ao dia 1. Seja s_0 uma escala construída anteriormente, a seqüência de operações do PCR executadas do dia 1 ao dia t é descrita na Figura 3.10.

O PCR foi utilizado por Constantino *et al.* (2014) e Volpato *et al.* (2013) na fase de melhoramento de seus algoritmos para a resolução do PEE, apontando uma grande importância do método PCR aplicado à esse problema. No exemplo a seguir é apresentado o funcionamento do PCR em uma escala de trabalho para enfermeiros. A Figura 3.11 apresenta uma escala de 7 dias construída para 4 enfermeiros. As linhas pontilhadas verticais antes de cada dia indicam onde o PCR realiza seus cortes. Nesse exemplo

Figura 3.10: Pseudocódigo PCR.

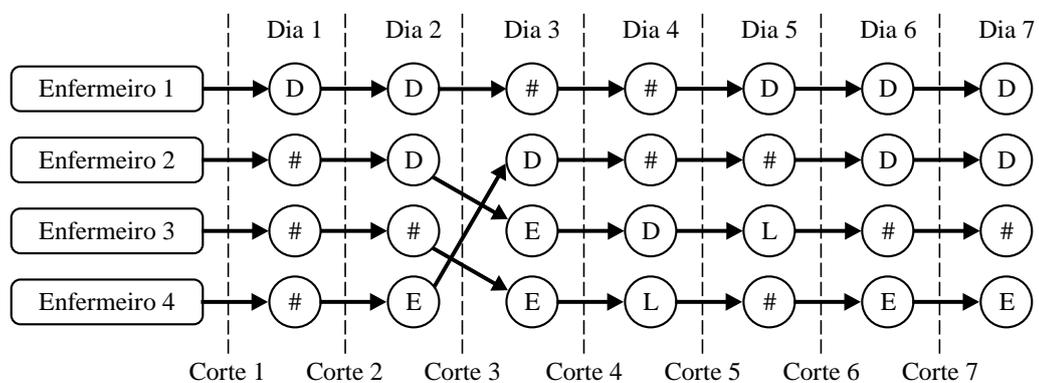
-
- 1: **Procedimento** PCR(s_0)
 - 2: **Para** $j \leftarrow 1$ até t **faça**
 - 3: Efetue um corte antes do dia j
 - 4: Gere uma matriz de custos E
 - 5: Resolva o PA correspondente à matriz E
 - 6: Realoque em s_0 as jornadas parciais de acordo com a solução do PA
 - 7: **Retorne** s_0
-

Fonte: Constantino *et al.* (2014).

Figura 3.11: Solução inicial e possíveis recombinações no corte 3.

Fonte: Elaborada pelo autor.

é realizado um corte entre os dias 2 e 3, no qual as setas pontilhadas indicam as possibilidades de recombinações das jornadas parciais para esse corte.

Figura 3.12: Escala após o corte e recombinação no corte 3.

Fonte: Elaborada pelo autor.

Após realizar o corte entre os dias 2 e 3, uma matriz de custo é gerada, conforme descrito anteriormente, calculando o custo de todas as possíveis recombinações de jornadas parciais à esquerda com jornadas parciais à direita. Em seguida, o PA correspondente é resolvido e as jornadas são re combinadas de acordo com a melhor solução encontrada. Na Figura 3.12 é apresentado a escala de trabalho após a recombinação das jornadas.

3.5 Procedimento k-swap

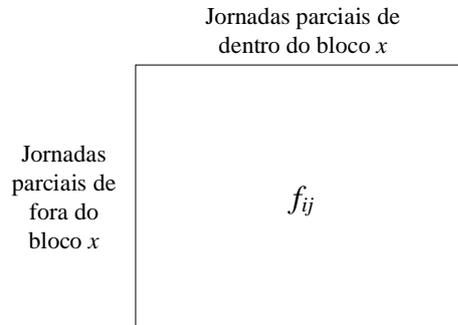
O *k-swap* é um método de busca local proposto por Volpato *et al.* (2013) que investiga a possibilidade de redistribuir os turnos entre os enfermeiros dentro de uma escala parcial de k dias. Esse método tem como objetivo diminuir o custo total de uma solução corrente. A variável k representa em dias o tamanho do bloco x da escala parcial que será utilizado para a redistribuição, podendo variar desde as tarefas de um único dia, quando $k = 1$, até as tarefas da escala completa, quando $k = t$.

Cada vez que é realizado um corte antes de um dia na escala, paralelamente é realizado outro corte k dias à frente, formando o bloco x composto das tarefas dos k dias. Seja $k = 2$, para um corte realizado no dia 2 de uma escala é feito outro corte no dia 4, formando um bloco com de tamanho 2 (dias 2 e 3), lembrando que os cortes são realizados antes dos dias na escala. Como o PEE envolve custos relacionados às preferências dos enfermeiros, uma mesma tarefa ou conjunto de tarefas executadas por enfermeiros diferentes pode incorrer em custos diferentes. (Volpato *et al.*, 2013)

Para cada valor de k , são realizados sucessivos cortes na escala antes de cada dia j até o dia $j - k + 1$, no qual o último corte realizado forma um bloco com os últimos k dias. Para realizar a recombinação das jornadas, as jornadas parciais à esquerda e à direita do bloco não sofrem alterações, sendo re combinado apenas as jornadas parciais que estão dentro do bloco x , entre os dois cortes realizados.

Para cada corte realizado na escala, é calculado uma matriz de custos $F = [f_{ij}]$ que relaciona cada uma das n jornadas fora do bloco x com as n jornadas de dentro do bloco. Cada elemento f_{ij} da matriz F é obtido calculando-se o custo de associar a jornada parcial do enfermeiro i que está fora do bloco x com a jornada parcial do enfermeiro j que está dentro do bloco x . Assim como no PCR, cada célula da matriz F representa uma jornada completa.

Na Figura 3.13 é apresentado a estrutura da matriz de custos F obtida por meio da execução do procedimento *k-swap*, na qual as linhas representam as jornadas parciais fora do bloco x e as colunas representam as jornadas parciais de dentro do bloco.

Figura 3.13: Estrutura da matriz de custos construída na execução do k -swap.

Fonte: Volpato *et al.* (2013).

A recombinação é obtida por meio da resolução do PA, o qual retorna as recombinações que produzem menor custo na escala. Por se basear na resolução do PA, a execução do k -swap permite que o custo da solução inicial seja reduzido ou, no mínimo, mantido. Cada iteração do k -swap consiste em efetuar $t - k + 1$ cortes e recombinações, um entre os funcionários e suas jornadas e $t - k$ cortes e recombinações entre os dias da escala, considerando uma escala com período de t dias. O k -swap pode ser implementado em duas direções, do dia 1 ao dia $t - k + 1$ ou do dia $t - k + 1$ ao dia 1. Seja s_0 uma escala construída inicialmente, a sequência de operações do k -swap executadas do dia 1 ao dia $t - k + 1$ é descrita na Figura 3.14.

Figura 3.14: Pseudocódigo k -swap.

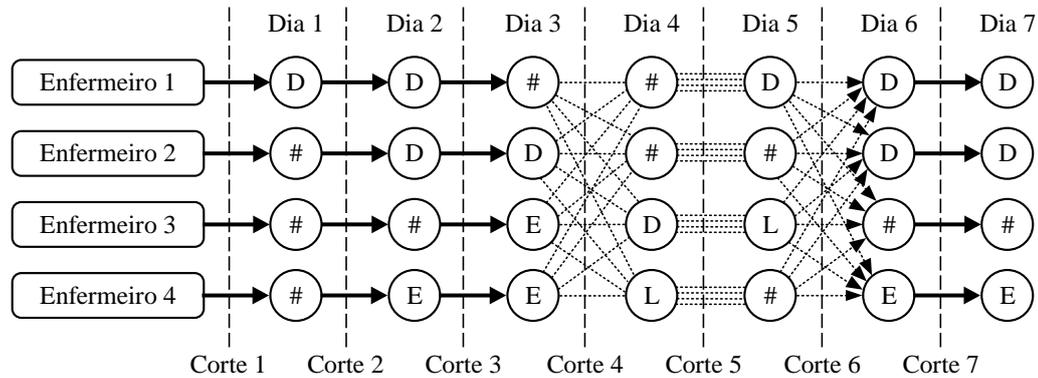
-
- 1: **Procedimento** k -swap(s_0)
 - 2: **Para** $j \leftarrow 1$ até $(t - k + 1)$ **faça**
 - 3: Efetue um corte antes do dia j e antes do dia $j + k$
 - 4: Gere uma matriz de custos F para as partições do dia j até o dia $(j + k - 1)$
 - 5: Resolva o PA da matriz F
 - 6: Realoque em s_0 as jornadas parciais de acordo com a solução do PA
 - 7: **Retorne** s_0
-

Fonte: Volpato *et al.* (2013).

Na Figura 3.15 é exemplificado a execução do procedimento 2 -swap ($k = 2$) em uma escala com 7 dias para 4 enfermeiros construída na fase inicial. As linhas pontilhadas verticais antes de cada dia indicam onde o k -swap realiza seus cortes. Nesse exemplo, é realizado um corte no dia 4, assim, o bloco x é composto pelas jornadas parciais dos enfermeiros nos dias 4 e 5. As setas pontilhadas indicam as possibilidades de

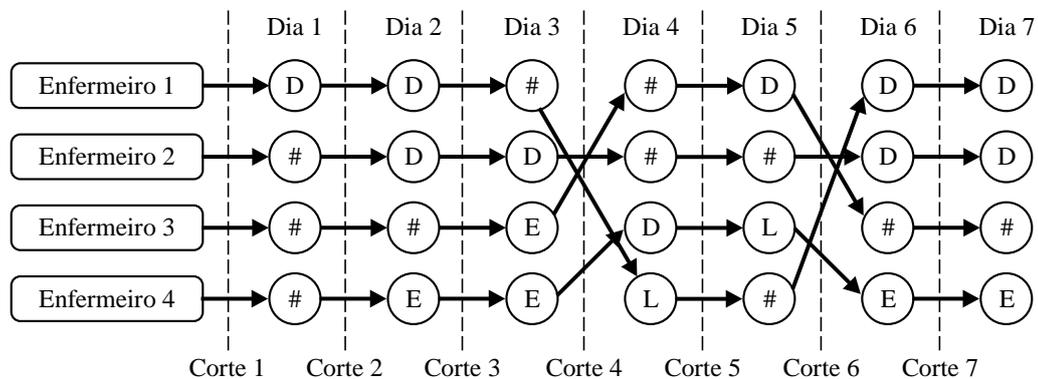
recombinações das jornadas parciais fora do bloco x com as jornadas parciais de dentro do bloco x .

Figura 3.15: Solução inicial e possíveis recombinações no corte 4 para 2 -swap.



Fonte: Elaborada pelo autor.

Figura 3.16: Escala após o corte e recombinação no corte 4 para 2 -swap.



Fonte: Elaborada pelo autor.

Após realizar o corte e gerar a matriz de custos da recombinação das jornadas parciais, o PA correspondente é resolvido e as jornadas são recombinadas de acordo com a melhor solução encontrada para esse PA.

Na Figura 3.16 é apresentada a solução após a recombinação das jornadas parciais. Pode-se notar que não houve alteração nas jornadas parciais antes do bloco x e nas jornadas parciais depois do bloco x , pois no k -swap as jornadas parciais antes e depois do bloco x de um enfermeiro i são tratadas como uma única jornada, não havendo troca entre elas. Somente as jornadas parciais compostas pelas tarefas dos dias 4 e 5 são recombinadas.

3.6 Trabalhos Relacionados

Devido a complexidade inerente à resolução de um problema de escalonamento de enfermeiros, muitos pesquisadores da área de computação possuem diversos trabalhos que buscam encontrar soluções viáveis para o problema de maneira mais eficiente com relação ao tempo de processamento. Nesta seção são descritos alguns dos principais métodos encontrados na literatura que objetivam encontrar uma solução para problemas de escalonamento.

Uma metodologia bastante explorada para a resolução do PEE são os Algoritmos Genéticos (AG). Esses algoritmos são definidos como métodos computacionais de busca baseados nos mecanismos de evolução natural e na genética. Em um AG, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua (Tanomaru, 1995).

Kawanaka *et al.* (2003) exploram um AG para o PEE que evita geração de soluções infactíveis no processo de cruzamento. A permuta de genes é realizada de modo que, apesar dos cromossomos filhos terem o máximo de características dos pais, os mesmos não incorporam as violações que seriam herdadas em um cruzamento sem tal tratamento. O método restringe a área de busca do AG e permite que boas soluções sejam encontradas.

Segundo Aickelin e Dowsland (2004), um AG clássico não possui grande capacidade de lidar com o PEE devido aos fortes conflitos entre a função-objetivo e as várias restrições do problema. Mesmo os AG que trabalham com subpopulações e que incorporam métodos de fuga de ótimos locais, apesar de chegarem a bons resultados, não possuem robustez para manter o desempenho quando pequenas alterações são feitas no problema. Assim, AG indireto podem apresentar melhores resultados que os obtidos por outras meta-heurísticas.

Tomando como justificativa a afirmação anterior, Aickelin e Dowsland (2004) apresentam um AG indireto para o PEE, no qual os indivíduos da população não representam diretamente as soluções. Escalas factíveis ou com o mínimo de violações das restrições são geradas a partir de soluções incompletas. Para tanto, o algoritmo utiliza um decodificador que constrói soluções a partir de permutações com os enfermeiros disponíveis, preenchendo as lacunas da escala de modo que os custos sejam mínimos.

Uma outra variação do AG é apresentada por Ohki *et al.* (2006), no qual é implementado um AG cooperativo em que cada indivíduo representa a jornada de um enfermeiro, e a população total, formada por todos os indivíduos, representa a escala completa. Nesse método é utilizado um operador de mutação e um operador de cruzamento, tornando esse

algoritmo diferente dos AG cooperativos que utilizam apenas operador de cruzamento. O algoritmo também faz uso de um outro operador que evita a estagnação em ótimos locais maiores, um problema recorrente nos AG cooperativos, dos quais os operadores de cruzamento e mutação sozinhos não escapariam. As mutações são realizadas de modo que a validade das escalas não seja perdida, sendo responsáveis pela evasão de pequenos ótimos locais. Foram utilizadas instâncias reais do PEE para a realização dos experimentos, no qual os resultados indicam a eficiência do método proposto.

O mesmo princípio de AG cooperativo é utilizado por Ohki *et al.* (2008). Entretanto, o método proposto trabalha com uma vizinhança de busca expandida, sendo utilizado duas CPUs que se comunicam para trocar informações a respeito das melhores soluções encontradas por cada uma. Com a utilização desse recurso, muito mais possibilidades são exploradas em um tempo razoavelmente menor. Os resultados obtidos pelo AG cooperativo com processamento paralelo em duas CPUs chegaram a ser melhores que os resultados do processamento convencional, atingindo um tempo de execução até cinco vezes menor.

Maenhout e Vanhoucke (2008) apresentam um AG para o PEE que utiliza diferentes operadores para a melhoria da solução. As soluções iniciais são obtidas por meio da resolução de um Problema de Fluxo de Custo Mínimo para cada jornada associada a cada enfermeiro. Posteriormente, os indivíduos são selecionados para que os operadores de cruzamento e mutação sejam empregados. Nesse trabalho, os autores realizam comparações extensivas entre os diversos operadores de cruzamento, realizando testes em um conjunto de dados padrão.

Um algoritmo com princípio de dispersão foi proposto por Maenhout e Vanhoucke (2006). Essa metodologia envolve a geração de diversas soluções iniciais, as quais são utilizadas em um processo de intensificação de busca enquanto certa diversificação é mantida. Para isso, dois grupos de soluções são reservados, um composto pelas melhores soluções e o outro por soluções que visam garantir a diversidade. Dessa maneira, um método iterativo de geração, recombinação e melhoramento de soluções é empregado até que o critério de parada seja alcançado.

Segundo Maenhout e Vanhoucke (2006), esse algoritmo contribui de três maneiras diferentes para a literatura. Em primeiro lugar, os autores foram os primeiros a apresentarem um algoritmo de busca que utiliza o princípio de dispersão. Em segundo lugar, eles investigaram dois tipos diferentes de métodos de combinação de soluções no contexto de busca por dispersão. E por último, eles apresentaram experimentos computacionais detalhados sobre um conjunto de dados de uma base de dados e resolveram esses casos problemáticos sob diferentes suposições. Eles afirmam ainda que esse procedimento

executa consistentemente bem em muitas circunstâncias diferentes e, portanto, pode ser considerado como robusto contra restrições de casos específicos.

Maenhout e Vanhoucke (2007) desenvolveram uma meta-heurística para o PEE com técnicas de eletromagnetismo, também conhecida como *Electromagnetic Meta-heuristic* (EM). A EM opera com diversas soluções distribuídas pelo espaço, sendo que cada uma exerce certas forças sobre as demais. O algoritmo proposto por Maenhout e Vanhoucke (2007) faz uso da EM para realizar melhorias na solução inicial. São realizadas recombinações que trabalham com a alteração da solução inicial, sendo conduzidas por soluções guiadas. Para esse fim são executados movimentos envolvendo a vizinhança de um dia ou de um enfermeiro. As soluções criadas são avaliadas e podem substituir membros do grupo das melhores soluções ou do grupo das soluções que propiciam diversificação. A abordagem de eletromagnetismo é baseada na teoria da física, que simula o movimento de atração e repulsão de pontos de amostragem, a fim de avançar para uma solução promissora.

Constantino *et al.* (2014) propuseram um algoritmo baseado no problema de atribuição multinível. O algoritmo possui duas fases: uma fase de construção e uma fase de melhoria. Na fase de construção é elaborado uma escala completa por meio de resoluções sucessivas do problema de atribuição entre duas partes da escala, levando em consideração as restrições rígidas, flexíveis e as preferências de cada enfermeiro. Para uma melhor manipulação da escala na fase de melhoria, os autores utilizam o grafo multipartido para a representação da escala.

Na segunda fase são utilizados dois procedimentos de busca local com o objetivo de reduzir o número de violações das restrições e atender uma maior quantidade de preferência dos enfermeiros. O algoritmo foi aplicado à base de dados NSPLib (Vanhoucke e Maenhout, 2007) e, segundo Constantino *et al.* (2014), o algoritmo proposto apresentou melhores resultados em relação a outros métodos que foram testados na mesma base de dados.

A utilização da meta-heurística VNS (*Variable Neighborhood Search*) foi explorada por Burke *et al.* (2003) e testada em uma antiga base de dados da Universidade de Nottingham. Nesse trabalho os autores apresentam e combinam diversos operadores de vizinhança. Mesmo com a alta quantidade de restrições o algoritmo consegue facilmente fugir de ótimos locais e gerar soluções melhores quando comparado a outras meta-heurísticas. Os experimentos apontam que o uso de uma busca local intensiva reflete diretamente na solução.

Burke *et al.* (2008) apresentam um algoritmo heurístico híbrido que também faz uso da meta-heurística VNS, combinada com uma heurística de ordenação. Nessa proposta,

uma solução inicial é construída utilizando a heurística de ordenação, que tem como objetivo classificar todos os turnos por ordem de dificuldade de alocação ou probabilidade de causar elevadas penalidades. Em seguida, o algoritmo faz uso do método VNS em busca de melhorar o custo da solução.

Duas meta-heurísticas são apresentadas por Freitas e Montané (2008) para uma variante do Problema de Roteamento de Veículos: GRASP-VND e VNS-VND. A meta-heurística GRASP-VND é baseada na meta-heurística GRASP (*Greedy Random Adaptive Search Procedure*) que utiliza o método VND (*Variable Neighborhood Descent*) como procedimento de busca local. Cada iteração da GRASP-VND consiste em duas fases: uma fase de construção, na qual é construída uma solução inicial viável; e uma fase de busca local, na qual é utilizado o procedimento VND na busca por melhorias na qualidade da solução inicial.

A meta-heurística VNS-VND se baseia na meta-heurística VNS utilizando o método VND como procedimento de busca local. O VNS-VND trabalha perturbando sistematicamente a solução corrente com o uso de diferentes estruturas de vizinhança, podendo resultar na degradação da solução. Em seguida é executado o VND que trabalha como intensificador da busca por melhorias na solução inicial (Freitas e Montané, 2008).

Volpato *et al.* (2013) propõe a utilização da meta-heurística VNS conjuntamente com dois métodos de busca local. Uma solução inicial é construída por meio de resoluções sucessivas de problemas de atribuição. Posteriormente, o algoritmo busca melhorar o custo da escala inicial com a aplicação da meta-heurística VNS. O algoritmo foi testado na base de dados NSPLib (Vanhoucke e Maenhout, 2007) que apresentou desempenho superior quando aplicado nas instâncias de grande porte do problema.

Uma adaptação do método de ejeção em cadeia, conhecido como *Variable Depth Search* (VDS), e o método *Branch and Price* são desenvolvidos por Burke e Curtois (2014) separadamente em busca de soluções para o PEE. Ambos os métodos foram testados em uma base de dados da Universidade de Nottingham, no qual os resultados demonstram que o algoritmo *Branch and Price* superou o VDS, especialmente em grandes e medias instâncias.

Gonsalves e Kuwata (2015) propõem um Algoritmo Memético (AM) que consiste de um AG combinado com um método de busca local. Uma abordagem bastante utilizada em AM é a aplicação de um métodos de busca local na população do AG depois do processo de cruzamento e mutação, com o objetivo de explorar as melhores regiões de busca. O algoritmo proposto resolve o PEE em dois estágios. No primeiro estágio, soluções aleatórias são evoluídas até que elas se tornem viáveis (satisfazendo todas as restrições rígidas). No segundo estágio, as soluções provenientes do primeiro estágio são evoluídas

de maneira a minimizar as violações das restrições flexíveis. Por fim, o algoritmo produz solução ótima para o PEE do *Queen's Medical Centre* (Nottingham, UK) atendendo por completo as restrições rígidas e flexíveis.

Uma abordagem heurística híbrida é apresentada por Rendl *et al.* (2012). O método proposto busca encontrar soluções para o problema de escalonamento de atendimento domiciliar, também conhecido como *Multimodal Homecare Scheduling* (MHS), que objetiva encontrar uma escala ótima para enfermeiros que realizam diversos atendimentos domiciliares viajando de paciente em paciente com a utilização de diferentes meios de transportes. O algoritmo trabalha em duas fases: fase construtiva e fase de melhoramento. Na primeira fase é gerado uma solução inicial utilizando a programação de restrições, também conhecido como *Constraint Programming* (CP). O CP consiste em gerar uma solução inicial com base nas restrições rígidas exigidas pelo problema, buscando gerar uma solução inicial sem violações rígidas. Na segunda fase, a solução inicial é melhorada utilizando uma entre as cinco meta-heurísticas apresentadas pelos autores, sendo elas: *Variable Neighborhood Descent* (VND), VNS, Algoritmo Evolucionário (AE), *Scatter Search* e *Simulated Annealing Hyper Heuristic*.

Para a validação do método proposto, os autores utilizaram uma seleção aleatória de instâncias reais do problema fornecidas por um projeto parceiro chamado SoGL. Dessa maneira, os autores afirmam que o AE produziu os melhores resultados, seguido da abordagem VNS que encontrou resultados similarmente bons como os do AE porém em maior tempo.

Os mesmos autores produziram uma pesquisa mais aprofundada de maneira que o novo trabalho apresentado por Hiermann *et al.* (2015) complementasse a pesquisa apresentada anteriormente (Rendl *et al.*, 2012). Os autores Hiermann *et al.* (2015) afirmam que o primeiro trabalho focou na hibridização com o estágio de inicialização da escala com base no CP, enquanto o trabalho mais recente foi focado na formulação detalhada do problema e uma descrição aprofundada da abordagens apresentadas.

Os autores utilizaram uma abordagem semelhante ao primeiro trabalho, porém com a utilização de um método diferente. Assim como a abordagem anterior, ela trabalha em duas fases: fase construtiva e fase de melhoramento. Na primeira fase, eles geram uma solução inicial com base no CP ou de maneira aleatória. Durante a segunda fase, a solução inicial é melhorada iterativamente com a utilização de uma entre as quatro meta-heurísticas: VNS, Algoritmo Memético (AM), *Scatter Search* e *Simulated Annealing Hyper Heuristic*. Os autores afirmam que todas as abordagens são capazes de resolver instâncias do mundo real em um tempo razoável, bem como produzem soluções válidas em alguns segundos.

Com base nos testes realizados, os melhores resultados foram encontrados pelo AM independente do mecanismo utilizado na fase construtiva. O método VNS foi considerado a segunda melhor abordagem quando aplicado em soluções iniciais construídas com base no CP.

Dellaert e Jeunet (2016) apresentam um algoritmo baseado na meta-heurística VNS com o objetivo de encontrar soluções para o problema de planejamento tático de cirurgias, o qual busca minimizar o excesso e a falta de utilização dos recursos de um hospital, como salas de cirurgias, macas, enfermeiros, entre outros. Para isso, os autores realizam algumas mudanças na meta-heurística VNS para construir o algoritmo. Além disso, o algoritmo utiliza 5 diferentes operadores de busca local para realizar a busca por melhorias na solução. Foram utilizadas instâncias reais do problema de planejamento tático de cirurgias de um centro cardiorádico de Dutch para comparar o desempenho do VNS clássico e do VNS alterado pelos autores. Os resultados dos experimentos apontaram a habilidade do algoritmo que utiliza o VNS alterado de alcançar soluções de grande qualidade em um rápido tempo de execução.

Pesquisas realizadas por Cheang *et al.* (2003) e Burke *et al.* (2004) apontam a necessidade de um conjunto de base de dados com instâncias do PEE para facilitar a comparação de muitos algoritmos propostos para esse problema. Dessa maneira, Vanhoucke e Maenhout (2007) propuseram uma grande base de dados chamada NSPLib, que inclui também um gerador de instâncias para o problema. A NSPLib é composta de 248.640 instâncias do PEE, reais e geradas aleatoriamente, sendo elas classificadas de acordo com seu tamanho e complexidade. São utilizados dois subconjuntos para a classificação das instâncias, chamados de realista e variado. O subconjunto realista é formado por instâncias de escalas com período de 28 dias, enquanto o subconjunto variado possui instâncias de escalas com período de 7 dias. A base de dados NSPLib está disponível em: <http://www.projectmanagement.ugent.be/nsp.php>.

Uma base de dados mais recente para o PEE foi desenvolvida por pesquisadores da Universidade de Nottingham. Eles disponibilizam uma base de dados com 24 instâncias do problema de escalonamento de pessoal incluindo escalonamento de enfermeiros (Curtois e Qu, 2014). Diferentemente da NSPLib, as instâncias são todas baseadas em problemas do mundo real, as quais possuem diversas dificuldades. A instância mais simples possui uma escala de 14 dias, 8 enfermeiros e 1 tipo de turno, enquanto a instância mais elaborada possui uma escala de 364 dias, 150 enfermeiros e 32 tipos de turno. A base de dados está disponível em: <http://www.cs.nott.ac.uk/~psztc/NRP/>.

A principal diferença entre a base de dados de Nottingham e a NSPLib é como cada uma aborda a quantidade de enfermeiros exigidos em cada dia da escala para cada

turno. Para a NSPLib, a demanda de enfermeiros exigida é tratada como um restrição rígida, na qual é exigido um número mínimo de enfermeiros em cada dia da escala para cada turno. Assim, a demanda deve obrigatoriamente ser obedecida, de modo que a solução final seja viável. Essa restrição rígida é violada somente se não forem atribuídos a demanda mínima de enfermeiros exigida.

Para a base de dados de Nottingham, a demanda de enfermeiros exigida pelo problema é tratada como uma restrição flexível, sendo exigido um número exato de enfermeiros em cada dia da escala para cada turno. Dessa maneira, o atendimento da demanda é desejável, porém não obrigatório para a obtenção de uma solução viável. Essa restrição flexível é violada quando são atribuídos menos ou mais enfermeiros em comparação com a quantidade exigida pela demanda, sendo atribuído um custo adicional ao custo total da escala para cada restrição não atendida.

Na Tabela 3.2 é apresentado as principais características da base de dados da Universidade de Nottingham e da base de dados NSPLib.

Tabela 3.2: Características das bases de dados para o PEE.

Características	Universidade de Nottingham	NSPLib
Número de instâncias	24	248.640
Tipo de instâncias	Reais	Aleatórias e Reais
Formato das instâncias	XML e TXT	TXT
Menor período	14 dias	7 dias
Maior período	365 dias	28 dias
Menor equipe	8 enfermeiros	25 enfermeiros
Maior equipe	150 enfermeiros	100 enfermeiros
Menor conjunto de turnos	1 turno	4 turnos
Maior conjunto de turnos	32 turno	4 turnos

Fonte: Curtois e Qu (2014) e Vanhoucke e Maenhout (2007).

Na Tabela 3.3 é apresentado uma comparação do modo como cada base de dados classifica as exigências do PEE, classificando como restrições flexíveis (RF) ou restrições rígidas (RR). Para as exigências que não possuem classificação, a base de dados não as considera como uma restrição.

Analisando as tabelas apresentadas é possível observar que a base de dados de Nottingham possui instâncias de maior dificuldade em relação à NSPLib, as quais possuem maior equipe máxima de enfermeiros, turnos, dias e restrições rígidas.

Tabela 3.3: Classificações das exigências do PEE.

Exigência	Universidade de Nottingham	NSPLib
Não atribuir mais que um turno por dia para cada enfermeiro	RR	RR
Rotação de turno	RR	RR
Máximo de cada tipo de turno por enfermeiro	RR	RF
Mínimo de cada tipo de turno por enfermeiro	-	RF
Máximo de tempo que cada enfermeiro deve trabalhar na escala	RR	RF
Mínimo de tempo que cada enfermeiro deve trabalhar na escala	RR	RF
Máximo de dias consecutivos trabalhados	RR	RF
Mínimo de dias consecutivos trabalhados	RR	RF
Máximo de dias consecutivos de folga	-	RF
Mínimo de dias consecutivos de folga	RR	RF
Máximo de dias consecutivos de cada tipo de turno por enfermeiro	-	RF
Mínimo de dias consecutivos de cada tipo de turno por enfermeiro	-	RF
Máximo de fins de semana	RR	-
Dias de folga predefinidos	RR	-
Preferência dos enfermeiros	RF	RF
Máximo de enfermeiros em cada dia da escala	RF	-
Mínimo de enfermeiros em cada dia da escala	RF	RR

Fonte: Curtois e Qu (2014) e Vanhoucke e Maenhout (2007).

Proposta

No presente capítulo é proposto um algoritmo que utiliza o problema de atribuição multinível PAM para a resolução do PEE.

Primeiramente são detalhados os procedimentos do algoritmo proposto na seção 4.1 juntamente com suas diferentes implementações que exploram de diversas maneiras os métodos de busca local. Por fim, na seção 4.2 são apresentadas as instâncias utilizadas para a realização dos experimentos.

4.1 Algoritmo Proposto

O método proposto neste trabalho, chamado HPAM, é um algoritmo **H**íbrido para o PEE que se baseia no **P**roblema de **A**tribuição **M**ultinível, combinando três métodos de busca local com uma variação da meta-heurística VNS proposta por Mladenović e Hansen (1997). Assim como feito por Constantino *et al.* (2014), o algoritmo aborda o PEE como um PAM, resolvendo seus subproblemas como PA. Para a resolução desses PA é implementado o algoritmo de Carpaneto e Toth (1987), que garante a obtenção da solução ótima para esses problemas, possuindo complexidade assintótica igual a $O(n^3)$.

O PEE é modelado pelo HPAM como um grafo multipartido acíclico com $t + 1$ partições, no qual os vértices da primeira partição correspondem ao conjunto de enfermeiros e os vértices das partições seguintes correspondem ao conjunto de turnos, sendo cada dia da escala representado por uma partição.

O HPAM trabalha em duas fases: fase construtiva e fase de melhoramento. Na fase construtiva é gerado uma primeira solução por meio de resoluções sucessivas do PA,

buscando maximizar o atendimento das restrições rígidas impostas pelo problema e as preferências dos enfermeiros.

Na fase de melhoramento é aplicado uma variação da meta-heurística VNS em conjunto com três procedimentos de busca local: PCR (Constantino *et al.*, 2014), *k-swap* (Volpato *et al.*, 2013) e *Cover Fit* (CF). Essa segunda fase tem como objetivo reduzir o custo total da solução construída na fase anterior, buscando atender uma maior quantidade de preferências dos enfermeiros e reduzir o número de violações às restrições. Para isso, os métodos PCR e *k-swap* buscam essas melhorias por meio de trocas de turnos ou sequência de turnos entre os enfermeiros. O método CF aumenta ou reduz, de acordo com a necessidade, a quantidade de turnos em um dia onde a demanda não é atendida, objetivando minimizar a diferença entre o número de enfermeiros escalados e a demanda de enfermeiros exigida, diminuindo o custo total da escala.

4.1.1 Fase Construtiva

A fase construtiva consiste na alocação das tarefas para cada enfermeiro em cada dia da escala. Para isso, é gerado um grafo multipartido acíclico que representa a escala inicial construída nessa fase, no qual cada jornada de trabalho corresponde a um caminho do primeiro ao último nível.

As alocações das tarefas em cada dia da escala é feita por meio da resolução do PA correspondente. Assim, a escala completa é obtida por meio de t resoluções sucessivas do PA. Para cada PA é gerado uma matriz de custos $C = [c_{il}]$ de dimensão $n \times n$, no qual cada elemento c_{il} representa o custo de atribuir uma tarefa l para um enfermeiro i em um dia da escala em questão.

Cada linha da matriz representa um enfermeiro, enquanto cada coluna representa um turno. Nas instâncias do PEE utilizadas nesse trabalho, o número de enfermeiros (n) é sempre maior ou igual a quantidade de turnos diferentes (s), sendo $n \geq s$. Assim, cada coluna de 1 à s representa um tipo diferente de turno (incluindo o turno folga), enquanto as colunas de $s + 1$ à n representam os turnos de menor custo para cada enfermeiro, podendo representar um turno diferente para cada um. Dessa maneira, obtêm-se uma matriz quadrada $n \times n$.

Nessa fase, a quantidade de enfermeiros escalados em cada dia pode ser diferente da quantidade exigida pela demanda, levando em consideração que o PEE abordado classifica a demanda de enfermeiros exigida como uma restrição flexível. Na Figura 4.1 é apresentado a estrutura da matriz de custos C gerada em cada dia da escala.

Figura 4.1: Estrutura da matriz de custos para os PA da fase construtiva.

	Bloco I	Bloco II
Enfermeiros	$c_{il} = f'(i,j,l)$	$c_{il} = \text{Min } f'(i,j,l);$ $l = 1, \dots, s$

Fonte: Elaborada pelo autor.

O Bloco I é formado por todos os tipos de turnos presentes no problema, enquanto o Bloco II contém os turnos adicionados para formar a matriz quadrada C . Com todos os turnos sendo representados no Bloco I, é permitido qualquer tipo de turno no Bloco II, no qual, para cada enfermeiro, é escolhido o turno de menor custo para ser apresentados também pelas colunas do Bloco II. Dessa maneira, cada coluna no Bloco II não representa necessariamente um único turno para todos os enfermeiro, mas sim o turno de menor custo de cada enfermeiro.

Os valores dos elementos c_{il} são obtidos por meio da função $f'(i, j, l)$, a qual soma o custo da preferência $cp(i, j, l)$ do enfermeiro i no dia j pelo turno l , e o número de violações às restrições rígidas que tal atribuição acrescenta (nRR), multiplicado pela penalidade de cada violação à restrição rígida ($penRR$). A função calculada em cada elemento da matriz é expressa na equação (7).

$$f'(i, j, l) = cp(i, j, l) + nRR \cdot penRR \quad (7)$$

O problema abordado neste trabalho fornece os valores que devem ser acrescentados ao custo total da escala para cada restrição flexível não atendida, variando de 1 à 100, os quais são utilizados como penalidade para cada violação flexível. Porém, como as restrições rígidas devem obrigatoriamente ser atendidas para que a escala seja viável, o problema não fornece um custo para as violações rígidas. Assim, para um tratamento das restrições rígidas pela abordagem proposta, cada violação rígida acrescentará um valor de 1000 ao custo total da escala ($penRR = 1000$). Dessa maneira, por possuir um valor

de penalidade maior, a abordagem proposta sempre prioriza o atendimento das restrições rígidas.

No início da fase de construção todas as posições formadas pelos turnos de trabalho estão vazias. Para atribuir as tarefas aos enfermeiros no primeiro dia da escala, a matriz de custos C é gerada e o PA é resolvido. Ao final desse processo, um turno é designado para cada um dos enfermeiros no primeiro dia. Em seguida, o mesmo processo é executado para os dias subsequentes até que todas as jornadas de cada enfermeiro estejam completas. Porém, para os dias 2 até t , os turnos atribuídos para cada enfermeiro antes do dia em questão devem ser considerados para o cálculo da função de custo. Após resolver os t PA, obtêm-se uma escala inicial para o conjunto de n enfermeiros. Seja s_0 uma escala inicial gerada na fase de construção, uma visão geral da fase construtiva do HPAM é apresentada na Figura 4.2.

Figura 4.2: Pseudocódigo Fase Construtiva.

```

1: Procedimento FASE-CONSTRUTIVA
2:   Para  $j \leftarrow 1$  até  $t$  faça
3:     Gere uma matriz de custos  $C$ 
4:     Resolva o PA da matriz  $C$ 
5:     Aloque em  $s_0$  os turnos aos enfermeiros de acordo com o resultado obtido
6:   Retorne  $s_0$ 

```

Fonte: Elaborada pelo autor.

Na Figura 3.2 é apresentado um exemplo de escala gerada na primeira fase do algoritmo HPAM, representada como um grafo multipartido. Cada vértices da primeira partição representa um enfermeiro diferente, enquanto os vértices das partições seguintes indicam o turno atribuído. As letras D, E e L representam diferentes tipos de turno, como por exemplo turno da manhã, tarde e noite, e o símbolo especial # representa o turno folga.

4.1.2 Fase de Melhoramento

A fase de melhoramento consiste na aplicação de uma variação da meta-heurística VNS, que utiliza o método VND ou CNS, em conjunto com três procedimentos de busca local (PCR, k -swap e *Cover Fit*) objetivando melhorar a qualidade da solução construída na fase inicial por meio da redução do custo total da escala.

Seja uma solução inicial s_0 , uma heurística busca local deve se mover de s_0 para um mínimo local s' . Hansen *et al.* (2010) apresentam duas estratégias de execução utilizadas por heurísticas de busca local:

- **Best Improvement:** essa técnica consiste em explorar toda a vizinhança e se mover para o vizinho com menor valor de função objetivo enquanto ele for menor que a solução corrente. Partindo de uma solução inicial $s' = s_0$, a cada iteração substitui s' pela solução $s = \min\{s'' \in N(s')\}$ enquanto $f(s) < f(s')$.
- **First Improvement:** essa técnica consiste em, a cada iteração, mover-se para o primeiro vizinho que tenha valor de função objetivo menor que a solução corrente. Dessa maneira, dependendo do tamanho da vizinhança, essa técnica pode consumir menor tempo de CPU para explorá-la completamente. Partindo de uma solução inicial $s' = s_0$, a cada iteração, s' é substituída pela primeira solução s_i encontrada em $N(s')$ que satisfaça $f(s_i) < f(s')$. O processo é repetido enquanto existe $s_i \in N(s')$ tal que $f(s_i) < f(s')$.

Os pseudocódigos das técnicas *best improvement* e *first improvement* são apresentados na Figura 4.3 e Figura 4.4, respectivamente.

Figura 4.3: Heurística baseada na técnica *best improvement*.

```

1: Procedimento Best-Improvement-Heuristic( $s_0, N$ )
2:    $s = s_0$ 
3:   Repita
4:      $s' = s$ 
5:      $s = \min\{s'' \in N(s)\}$ 
6:   Até  $f(s) \geq f(s')$ 
7:   Retorne  $s'$ 

```

Fonte: Hansen *et al.* (2010).

Neste trabalho é adotado a técnica *first improvement* por ser mais rápida e apresentar bons resultados em trabalhos da literatura (Cares e Riff, 2012; Dellaert e Jeunet, 2016; Hiermann *et al.*, 2015; Rendl *et al.*, 2012).

Dellaert e Jeunet (2016) apresentam uma variação da meta-heurística GVNS, chamada R-VNS, aplicada ao problema de planejamento tático de cirurgias. O R-VNS diferencia-se do GVNS clássico na aplicação da perturbação da escala e do método de busca local. No GVNS é feito uma perturbação na escala corrente no início de cada iteração, utilizando o método VND como procedimento de busca local. Em contrapartida, no R-VNS a perturbação da escala é aplicada somente quando a busca local não consegue melhorar a escala corrente, sendo que a sequência dos operadores de busca local é aplicada repetidamente na solução corrente e cada operador realiza apenas uma modificação

Figura 4.4: Heurística baseada na técnica *first improvement*.

```

1: Procedimento First-Improvement-Heuristic( $s_0, N$ )
2:    $s = s_0$ 
3:   Repita
4:      $s' = s$ 
5:      $i = 0$ 
6:     Repita
7:        $i = i + 1$ 
8:        $s = \min\{s, s_i\}, s_i \in N(s')$ 
9:     Até  $f(s) < f(s')$  ou  $i = |N(s')|$ 
10:  Até  $f(s) \geq f(s')$ 
11:  Retorne  $s'$ 

```

Fonte: Hansen *et al.* (2010).

aleatória que são aceitas enquanto a solução é viável. Segundo os autores, o R-VNS encontrou resultados melhores que o GVNS para todas as instâncias utilizadas.

Baseado nas modificações realizadas por Dellaert e Jeunet (2016) e em suas conclusões, neste trabalho propomos uma variação da meta-heurística VNS na fase de melhoramento do HPAM, denominada M-VNS. O que diferencia o VNS clássico do M-VNS é a aplicação do método de busca local na solução inicial, antes de realizar a primeira perturbação na escala, até que não seja encontrado um ótimo local.

Seja s^* uma solução inicial e r o número de estruturas de vizinhança presentes em N_k , ($k = 0, 1, \dots, r - 1$). De modo a enfatizar as diferenças entre as meta-heurísticas, na Tabela 4.1 é apresentado o pseudocódigo do VNS clássico e do M-VNS.

Neste trabalho, a condição de parada do algoritmo M-VNS é definida como sendo uma execução sem melhorias. Assim, caso o algoritmo percorra todas as estruturas de vizinhança e não consiga melhorar a solução corrente, a execução é finalizada.

Os métodos PCR e *k-swap* são utilizados para implementar as estruturas de vizinhança aplicadas para realizar a perturbação da escala (Figura 3.3, Linha 6), de maneira a gerar uma solução aleatória à partir da solução corrente. Dessa maneira, é utilizado um conjunto de 6 estruturas N_k ($r_{vns} = 6$) conforme apresentado na Tabela 4.2.

Para realizar a perturbação da solução corrente, os procedimentos descritos na Tabela 4.2 realizam uma recombinação aleatória das jornadas após cada corte realizado na escala, independente da matriz de custos. Esses procedimentos executam cortes uma única vez em cada dia no mesmo sentido da escala, do dia 1 ao dia t . A principal função dessas estruturas é embaralhar a escala de trabalho de modo a distribuir de maneira diferente

Tabela 4.1: Pseudocódigo dos métodos VNS e M-VNS.

VNS clássico	M-VNS
1: Procedimento VNS(s^*)	1: Procedimento M-VNS(s^*)
2: $s_0 \leftarrow s^*$	2: $s_0 \leftarrow s^*$
3: Enquanto condição de parada não satisfeita faça	3: $s_2 \leftarrow$ Busca-Local (s_0)
4: $k \leftarrow 0$	4: Se custo da solução $s_2 <$ custo da solução s_0 então
5: Enquanto $k < r$ faça	5: $s_0 \leftarrow s_2$
6: Gere um vizinho qualquer $s_1 \in N_k$ a partir de s_0	6: Enquanto condição de parada não satisfeita faça
7: $s_2 \leftarrow$ Busca-Local (s_1)	7: $k \leftarrow 0$
8: Se custo da solução $s_2 <$ custo da solução s_0 então	8: Enquanto $k < r$ faça
9: $s_0 \leftarrow s_2$	9: Gere um vizinho qualquer $s_1 \in N_k$ a partir de s_0
10: $k \leftarrow 0$	10: $s_2 \leftarrow$ Busca-Local (s_1)
11: Senão	11: Se custo da solução $s_2 <$ custo da solução s_0 então
12: $k \leftarrow k + 1$	12: $s_0 \leftarrow s_2$
13: Retorne s_0	13: $k \leftarrow 0$
	14: Senão
	15: $k \leftarrow k + 1$
	16: Retorne s_0

Fonte: Mladenović e Hansen (1997) e o autor.

Tabela 4.2: Estruturas de vizinhança para perturbação da escala.

Valor de k	Procedimento
0	PCR_rand
1	1-swap_rand
2	2-swap_rand
3	3-swap_rand
4	4-swap_rand
5	5-swap_rand

Fonte: Elaborada pelo autor.

os turnos atribuídos. Dessa maneira, o M-VNS realiza a perturbação na solução corrente explorando gradativamente vizinhanças mais distantes. Na Figura 4.5 e Figura 4.6 são apresentados o funcionamento das estruturas de vizinhança PCR_rand e k -swap_rand, respectivamente.

Figura 4.5: Procedimento PCR_rand.

```

1: Procedimento PCR_RANDOM( $s_0$ )
2:   Para  $j \leftarrow 1$  até  $t$  faça
3:     Efetue um corte antes do dia  $j$ 
4:     Realoque em  $s_0$  as jornadas parciais de modo aleatório
5:   Retorne  $s_0$ 

```

Fonte: Elaborada pelo autor.

Figura 4.6: Procedimento k -swap_rand.

```

1: Procedimento  $k$ -swap_RANDOM( $s_0$ )
2:   Para  $j \leftarrow 1$  até  $t - k + 1$  faça
3:     Efetue um corte antes do dia  $j$  e antes do dia  $j + k$ 
4:     Realoque em  $s_0$  as jornadas parciais de modo aleatório
5:   Retorne  $s_0$ 

```

Fonte: Elaborada pelo autor.

Após a perturbação da solução corrente, é iniciado uma heurística de busca local, VND ou CNS, que objetiva encontrar melhorias na qualidade da escala com a aplicação de diferentes operadores de busca local. No HPAM, essas heurísticas fazem uso dos procedimentos PCR, k -swap e um novo método de busca local desenvolvido nesse trabalho, de maneira a intensificar a busca pela redução do custo da solução.

O PEE abordado neste trabalho considera a demanda de enfermeiros em cada dia da escala como uma restrição flexível, sendo que cada enfermeiro escalado à mais ou à menos representa uma violação à essa restrição. Dessa maneira, se faz necessário um método que busque aproximar da demanda exigida a quantidade de enfermeiros escalados em um dia. Os métodos PCR e k -swap buscam melhorias na escala sem alterar a quantidade de enfermeiros escalados, realizando apenas troca dos turnos já atribuídos entre os enfermeiros. Com isso, este trabalho introduz um novo procedimento de busca local, chamado *Cover Fit*, que busca aproximar da demanda exigida a quantidade de enfermeiros escalados em cada dia.

4.1.3 Cover Fit

O CF é um método de busca local que investiga a possibilidade de redistribuir os turnos de um dia j alterando a quantidade de turnos atribuídos aos enfermeiros para uma quantidade mais próxima da exigida pela demanda.

Seja uma solução corrente na qual q_l representa o número de turnos l atribuídos em um dia j e e_l a quantidade de turnos l exigidos pelo problema para o mesmo dia da escala. Para cada dia j da escala em que a demanda exigida não é atendida para todos os turnos, é calculado uma matriz de custos $H_l = [h_{iv}]$ de dimensão $n \times n$ para cada turno l onde $q_l \neq e_l$ nesse dia. As colunas da matriz representam os tipos de turno (incluindo turno l) enquanto as linhas representam os enfermeiros. Cada elemento h_{iv} representa o custo de se atribuir um turno v a um enfermeiro i , levando em consideração os turnos já alocados nos outros dias da escala para esse mesmo enfermeiro i .

Cada turno v é representado por uma quantidade fixa de colunas igual à quantidade de atribuição desse turno no dia j da solução corrente, com exceção dos turnos folga e l . Se $q_l > e_l$, então o turno l será representado por $q_l - 1$ colunas, senão, ele é representado por $q_l + 1$ colunas. Após estabelecer a quantidade de colunas para cada turno, as colunas restantes passam a representar o turno folga.

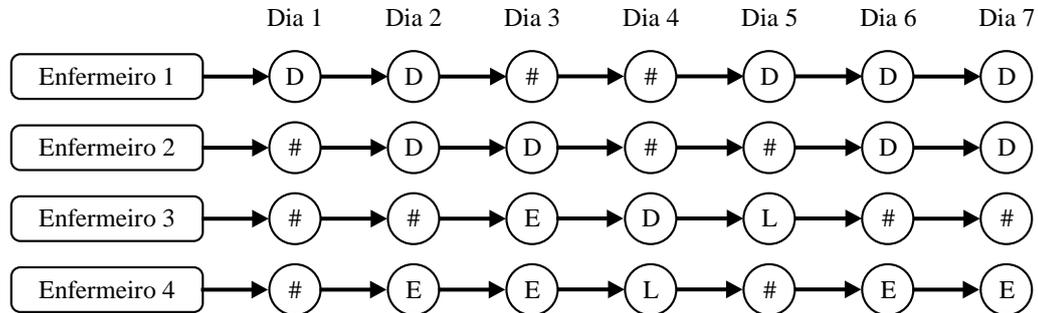
Os valores dos elementos h_{iv} são obtidos por meio da função $f''(i, j, v)$, a qual soma-se o número de vezes que cada restrição flexível é violada (n_x , onde $r =$ número total de restrições flexíveis), multiplicado pela penalidade de cada violação à respectiva restrição flexível (pen_x), somado com o custo da preferência $cp(i, j, v)$ do enfermeiro i no dia j pelo turno v e o número de violações às restrições rígidas que tal atribuição acrescenta (nRR), multiplicado pela penalidade de cada violação à restrição rígida ($penRR$). A função calculada em cada elemento da matriz é expressa na equação (8).

$$f''(i, j, v) = \sum_{x=1}^r n_x \cdot pen_x + cp(i, j, v) + nRR \cdot penRR \quad (8)$$

Assim como definido na função (7) da fase de construção, para o tratamento das restrições rígidas no procedimento *Cover Fit*, cada violação rígida acrescentará um valor de 1000 ao custo total da escala, assim $penRR = 1000$ na função (8).

Seja a Figura 4.7 uma solução corrente na qual a demanda exigida do turno D para o dia 4 é de 2 enfermeiros. Na Figura 4.8 é apresentado a estrutura da matriz de custos H_D obtida por meio da execução do procedimento CF para o dia 4 dessa escala corrente.

Obtida a matriz de custos, o PA é resolvido e os turnos são redistribuídos no dia j da escala, formando uma nova escala. Se o custo total da nova escala for menor que o custo

Figura 4.7: Solução corrente.

Fonte: Elaborada pelo autor.

Figura 4.8: Estrutura da matriz de custos H_D construída na execução do CF no dia 4.

	D	D	L	#
Enfermeiro 1	h_{1D}	h_{1D}	h_{1L}	$h_{1\#}$
Enfermeiro 2	h_{2D}	h_{2D}	h_{2L}	$h_{2\#}$
Enfermeiro 3	h_{3D}	h_{3D}	h_{3L}	$h_{3\#}$
Enfermeiro 4	h_{4D}	h_{4D}	h_{4L}	$h_{4\#}$

Fonte: Elaborada pelo autor.

total da escala corrente, ela passa a ser a escala corrente, senão, o procedimento é aplicado novamente à escala corrente com uma quantidade diferente de colunas representando o turno l . Esse procedimento é aplicado ao mesmo turno l no mesmo dia da escala até que seja comparado com uma solução onde $q_l = e_l$. Dessa maneira, o método é repetido em todos os dias da escala para cada turno que não atende à demanda exigida, buscando assim, minimizar a distância entre a quantidade de enfermeiros escalados para um turno e a quantidade de enfermeiros exigidos.

Cada iteração CF consiste em realizar o procedimento em todos os dias da escala de trabalho. A sequência de execução dos procedimentos pode ser do dia 1 ao dia t quanto do dia t ao dia 1. Seja s_0 uma solução corrente, o funcionamento do procedimento CF é descrito na Figura 4.9.

Assim como descrito anteriormente, as heurísticas de busca local VND e CNS utilizadas pela meta-heurística M-VNS fazem uso de diferentes estruturas de vizinhança para realizar buscas por melhorias na solução corrente. Os métodos PCR, k -swap e *Cover Fit* são utilizados para implementar essas estruturas de vizinhança. Assim, é utilizado um

Figura 4.9: Procedimento *Cover Fit*.

```

1: Procedimento Cover-Fit( $s_0$ )
2:   Para  $j \leftarrow 1$  até  $t$  faça
3:     Para cada  $l$  onde  $q_l \neq e_l$  faça
4:        $x \leftarrow q_l$ 
5:       Enquanto  $q_l \neq e_l$  faça
6:         Se  $q_l < e_l$  então
7:            $i \leftarrow i + 1$ 
8:         Senão
9:            $i \leftarrow i - 1$ 
10:        Gere uma matriz de custos  $H_l$  com  $i$  colunas representando o turno  $l$ 
11:        Resolva o PA correspondente à matriz  $H_l$ 
12:        Realoque os turnos de acordo com a resolução do PA, gerando  $s_1$ 
13:        Se custo da solução  $s_1 <$  custo da solução  $s_0$  então
14:           $s_0 \leftarrow s_1$ 
Retorne  $s_0$ 

```

Fonte: Elaborada pelo autor.

conjunto de 7 estruturas N_k ($r = 7$) na fase de melhoria, sendo PCR, 1-*swap*, 2-*swap*, 3-*swap*, 4-*swap*, 5-*swap* e *Cover Fit*.

Neste trabalho as estruturas de vizinhanças são aplicadas em diferentes sequências de execução, de modo a encontrar a melhor sequência de estruturas de vizinhança a ser aplicada no algoritmo.

4.1.4 Versões do HPAM

Os procedimentos de busca local PCR, *k-swap* e CF podem ser executados em duas direções, do dia 1 ao dia t e do dia t ao dia 1 da escala. Cada iteração dos procedimentos PCR e *k-swap* corresponde na execução das duas direções, independente da ordem das mesmas. Para o procedimento CF, cada iteração corresponde na execução de uma única direção.

Dessa maneira, esses métodos foram implementados de duas formas em relação à sequência de execução na escala:

- Para os procedimentos PCR e *k-swap*:

Forward: do dia 1 ao dia t e do dia t ao dia 1

Backward: do dia t ao dia 1 e do dia 1 ao dia t

- Para o procedimento *Cover Fit*:

Forward: do dia 1 ao dia t

Backward: do dia t ao dia 1

Quando definido uma sequência de execução para o procedimento *k-swap*, essa sequência é implementada para qualquer valor de k . Na Tabela 4.3 é apresentado as notações referentes a cada possibilidade de implementação dos procedimentos PCR, *k-swap* e *Cover Fit*.

Tabela 4.3: Notações para os procedimentos de busca local.

Notação	Procedimento	Sequência de execução
PF	PCR	<i>Forward</i>
PB	PCR	<i>Backward</i>
KF	<i>k-swap</i>	<i>Forward</i>
KB	<i>k-swap</i>	<i>Backward</i>
CFF	<i>Cover Fit</i>	<i>Forward</i>
CFB	<i>Cover Fit</i>	<i>Backward</i>

Fonte: Elaborada pelo autor.

Considere uma solução corrente s_0 , na Figura 4.10 é apresentado a sequência de passos para a execução do procedimento PF.

Figura 4.10: Procedimento PF.

-
- 1: **Procedimento** PF(s_0)
 - 2: **Para** $j \leftarrow 1$ até t **faça**
 - 3: Efetue um corte antes do dia j
 - 4: Gere uma matriz de custos E
 - 5: Resolva o PA correspondente à matriz E
 - 6: Realoque em s_0 as jornadas parciais de acordo com a solução do PA
 - 7: **Para** $j \leftarrow t$ até 1 **faça**
 - 8: Efetue um corte antes do dia j
 - 9: Gere uma matriz de custos E
 - 10: Resolva o PA correspondente à matriz E
 - 11: Realoque em s_0 as jornadas parciais de acordo com a solução do PA
 - 12: **Retorne** s_0
-

Fonte: Elaborada pelo autor.

Para a versão *Backward* do PCR, substitui-se a linha 2 pela linha 7 e a linha 7 pela linha 2.

Na Figura 4.11 é apresentado a sequência de passos para a execução do procedimento KF.

Figura 4.11: Procedimento KF.

-
- 1: **Procedimento** KF(s_0)
 - 2: **Para** $j \leftarrow 1$ até $(t - k + 1)$ **faça**
 - 3: Efetue um corte antes do dia j e antes do dia $j + k$
 - 4: Gere uma matriz de custos F para as partições do dia j até o dia $(j + k - 1)$
 - 5: Resolva o PA da matriz F
 - 6: Realoque em s_0 as jornadas parciais de acordo com a solução do PA
 - 7: **Para** $j \leftarrow (t - k + 1)$ até 1 **faça**
 - 8: Efetue um corte antes do dia j e antes do dia $j + k$
 - 9: Gere uma matriz de custos F para as partições do dia j até o dia $(j + k - 1)$
 - 10: Resolva o PA da matriz F
 - 11: Realoque em s_0 as jornadas parciais de acordo com a solução do PA
 - 12: **Retorne** s_0
-

Fonte: Elaborada pelo autor.

Para a versão *Backward* do k -swap, substitui-se a linha 2 pela linha 7 e a linha 7 pela linha 2. Na Figura 4.12 é apresentado a sequência de passos para a execução do procedimento CFF.

A versão *Backward* do procedimento *Cover Fit* é obtida com a substituição da linha 2 por: **Para** $j \leftarrow t$ até 1 **faça**

Assim como citado anteriormente, o HPAM faz uso de uma das duas heurísticas de busca local aplicadas no M-VNS, sendo elas VND e CNS. Além disso foram testadas diferentes sequências de aplicação dos métodos de busca local utilizados pelos procedimentos VND e CNS. Dessa maneira, neste trabalho foram implementadas 96 versões diferentes do algoritmo HPAM, variando a heurística de busca local, a sequência dos procedimentos de busca local e a sequência de execução de cada um desses procedimentos. Na Tabela 4.4 é apresentado cada uma das versões implementadas juntamente com suas características.

A coluna **Notação** (coluna 1) representa a notação utilizada para a variação do algoritmo HPAM que utiliza a sequência de procedimentos descritos nas colunas seguintes. Na coluna **Heurística** é apresentado a heurística de busca local utilizado pela versão do algoritmo. As colunas **P1**, **P2**, **P3**, **P4**, **P5**, **P6** e **P7** (colunas de 3 à 9) representam os procedimentos de melhoria aplicados na respectiva ordem, sendo utilizado valores

Figura 4.12: Procedimento CFF.

```

1: Procedimento CFF( $s_0$ )
2:   Para  $j \leftarrow 1$  até  $t$  faça
3:     Para cada  $l$  onde  $q_l \neq e_l$  faça
4:        $x \leftarrow q_l$ 
5:       Enquanto  $q_l \neq e_l$  faça
6:         Se  $q_l < e_l$  então
7:            $i \leftarrow i + 1$ 
8:         Senão
9:            $i \leftarrow i - 1$ 
10:        Gere uma matriz de custos  $H_l$  com  $i$  colunas representando o turno  $l$ 
11:        Resolva o PA correspondente à matriz  $H_l$ 
12:        Realoque os turnos de acordo com a resolução do PA, gerando  $s_1$ 
13:        Se custo da solução  $s_1 <$  custo da solução  $s_0$  então
14:           $s_0 \leftarrow s_1$ 
Retorne  $s_0$ 

```

Fonte: Elaborada pelo autor.

de 1 à 5 na implementação do k -swap: 1-swap-Forward (1F), 1-swap-Backward (1B), 2-swap-Forward (2F), 2-swap-Backward (2B), 3-swap-Forward (3F), 3-swap-Backward (3B), 4-swap-Forward (4F), 4-swap-Backward (4B), 5-swap-Forward (5F), 5-swap-Backward (5B).

Tabela 4.4: Versões do HPAM.

Notação	Heurística	P1	P2	P3	P4	P5	P6	P7
V-PFKFCFF	VND	PF	1F	2F	3F	4F	5F	CFF
V-PFKFCFB	VND	PF	1F	2F	3F	4F	5F	CFB
V-PFKBCFF	VND	PF	1B	2B	3B	4B	5B	CFF
V-PFKBCFB	VND	PF	1B	2B	3B	4B	5B	CFB
V-PBKFCFF	VND	PB	1F	2F	3F	4F	5F	CFF
V-PBKFCFB	VND	PB	1F	2F	3F	4F	5F	CFB
V-PBKBCFF	VND	PB	1B	2B	3B	4B	5B	CFF
V-PBKBCFB	VND	PB	1B	2B	3B	4B	5B	CFB
C-PFKFCFF	CNS	PF	1F	2F	3F	4F	5F	CFF

Notação	Heurística	P1	P2	P3	P4	P5	P6	P7
C-PFKFCFB	CNS	PF	1F	2F	3F	4F	5F	CFB
C-PFKBCFF	CNS	PF	1B	2B	3B	4B	5B	CFF
C-PFKBCFB	CNS	PF	1B	2B	3B	4B	5B	CFB
C-PBKFCFF	CNS	PB	1F	2F	3F	4F	5F	CFF
C-PBKFCFB	CNS	PB	1F	2F	3F	4F	5F	CFB
C-PBKBCFF	CNS	PB	1B	2B	3B	4B	5B	CFF
C-PBKBCFB	CNS	PB	1B	2B	3B	4B	5B	CFB
V-PFCFFKF	VND	PF	CFF	1F	2F	3F	4F	5F
V-PFCFFKB	VND	PF	CFF	1B	2B	3B	4B	5B
V-PFCFBKF	VND	PF	CFB	1F	2F	3F	4F	5F
V-PFCFBKB	VND	PF	CFB	1B	2B	3B	4B	5B
V-PBCFFKF	VND	PB	CFF	1F	2F	3F	4F	5F
V-PBCFFKB	VND	PB	CFF	1B	2B	3B	4B	5B
V-PBCFBKF	VND	PB	CFB	1F	2F	3F	4F	5F
V-PBCFBKB	VND	PB	CFB	1B	2B	3B	4B	5B
C-PFCFFKF	CNS	PF	CFF	1F	2F	3F	4F	5F
C-PFCFFKB	CNS	PF	CFF	1B	2B	3B	4B	5B
C-PFCFBKF	CNS	PF	CFB	1F	2F	3F	4F	5F
C-PFCFBKB	CNS	PF	CFB	1B	2B	3B	4B	5B
C-PBCFFKF	CNS	PB	CFF	1F	2F	3F	4F	5F
C-PBCFFKB	CNS	PB	CFF	1B	2B	3B	4B	5B
C-PBCFBKF	CNS	PB	CFB	1F	2F	3F	4F	5F
C-PBCFBKB	CNS	PB	CFB	1B	2B	3B	4B	5B
V-KFPFCFF	VND	1F	2F	3F	4F	5F	PF	CFF
V-KFPFCFB	VND	1F	2F	3F	4F	5F	PF	CFB

Notação	Heurística	P1	P2	P3	P4	P5	P6	P7
V-KFPBCFF	VND	1F	2F	3F	4F	5F	PB	CFF
V-KFPBCFB	VND	1F	2F	3F	4F	5F	PB	CFB
V-KBPFCFF	VND	1B	2B	3B	4B	5B	PF	CFF
V-KBPFCFB	VND	1B	2B	3B	4B	5B	PF	CFB
V-KBPBCFF	VND	1B	2B	3B	4B	5B	PB	CFF
V-KBPBCFB	VND	1B	2B	3B	4B	5B	PB	CFB
C-KFPFCFF	CNS	1F	2F	3F	4F	5F	PF	CFF
C-KFPFCFB	CNS	1F	2F	3F	4F	5F	PF	CFB
C-KFPBCFF	CNS	1F	2F	3F	4F	5F	PB	CFF
C-KFPBCFB	CNS	1F	2F	3F	4F	5F	PB	CFB
C-KBPFCFF	CNS	1B	2B	3B	4B	5B	PF	CFF
C-KBPFCFB	CNS	1B	2B	3B	4B	5B	PF	CFB
C-KBPBCFF	CNS	1B	2B	3B	4B	5B	PB	CFF
C-KBPBCFB	CNS	1B	2B	3B	4B	5B	PB	CFB
V-KFCFFPF	VND	1F	2F	3F	4F	5F	CFF	PF
V-KFCFFPB	VND	1F	2F	3F	4F	5F	CFF	PB
V-KFCFBPF	VND	1F	2F	3F	4F	5F	CFB	PF
V-KFCFBPB	VND	1F	2F	3F	4F	5F	CFB	PB
V-KBCFFPF	VND	1B	2B	3B	4B	5B	CFF	PF
V-KBCFFPB	VND	1B	2B	3B	4B	5B	CFF	PB
V-KBCFBPF	VND	1B	2B	3B	4B	5B	CFB	PF
V-KBCFBPB	VND	1B	2B	3B	4B	5B	CFB	PB
C-KFCFFPF	CNS	1F	2F	3F	4F	5F	CFF	PF
C-KFCFFPB	CNS	1F	2F	3F	4F	5F	CFF	PB
C-KFCFBPF	CNS	1F	2F	3F	4F	5F	CFB	PF

Notação	Heurística	P1	P2	P3	P4	P5	P6	P7
C-KFCFBPB	CNS	1F	2F	3F	4F	5F	CFB	PB
C-KBCFFPF	CNS	1B	2B	3B	4B	5B	CFF	PF
C-KBCFFPB	CNS	1B	2B	3B	4B	5B	CFF	PB
C-KBCFBPF	CNS	1B	2B	3B	4B	5B	CFB	PF
C-KBCFBPB	CNS	1B	2B	3B	4B	5B	CFB	PB
V-CFFPFKF	VND	CFF	PF	1F	2F	3F	4F	5F
V-CFFPFKB	VND	CFF	PF	1B	2B	3B	4B	5B
V-CFFPBKF	VND	CFF	PB	1F	2F	3F	4F	5F
V-CFFPBKB	VND	CFF	PB	1B	2B	3B	4B	5B
V-CFBPFKF	VND	CFB	PF	1F	2F	3F	4F	5F
V-CFBPFKB	VND	CFB	PF	1B	2B	3B	4B	5B
V-CFBPBKF	VND	CFB	PB	1F	2F	3F	4F	5F
V-CFBPBKB	VND	CFB	PB	1B	2B	3B	4B	5B
C-CFFPFKF	CNS	CFF	PF	1F	2F	3F	4F	5F
C-CFFPFKB	CNS	CFF	PF	1B	2B	3B	4B	5B
C-CFFPBKF	CNS	CFF	PB	1F	2F	3F	4F	5F
C-CFFPBKB	CNS	CFF	PB	1B	2B	3B	4B	5B
C-CFBPFKF	CNS	CFB	PF	1F	2F	3F	4F	5F
C-CFBPFKB	CNS	CFB	PF	1B	2B	3B	4B	5B
C-CFBPBKF	CNS	CFB	PB	1F	2F	3F	4F	5F
C-CFBPBKB	CNS	CFB	PB	1B	2B	3B	4B	5B
V-CFFKFPF	VND	CFF	1F	2F	3F	4F	5F	PF
V-CFFKFPB	VND	CFF	1F	2F	3F	4F	5F	PB
V-CFFKBPF	VND	CFF	1B	2B	3B	4B	5B	PF
V-CFFKBPB	VND	CFF	1B	2B	3B	4B	5B	PB

Notação	Heurística	P1	P2	P3	P4	P5	P6	P7
V-CFBKFPF	VND	CFB	1F	2F	3F	4F	5F	PF
V-CFBKFPB	VND	CFB	1F	2F	3F	4F	5F	PB
V-CFBKBPF	VND	CFB	1B	2B	3B	4B	5B	PF
V-CFBKBPB	VND	CFB	1B	2B	3B	4B	5B	PB
C-CFFKFPF	CNS	CFF	1F	2F	3F	4F	5F	PF
C-CFFKFPB	CNS	CFF	1F	2F	3F	4F	5F	PB
C-CFFKBPF	CNS	CFF	1B	2B	3B	4B	5B	PF
C-CFFKBPB	CNS	CFF	1B	2B	3B	4B	5B	PB
C-CFBKFPF	CNS	CFB	1F	2F	3F	4F	5F	PF
C-CFBKFPB	CNS	CFB	1F	2F	3F	4F	5F	PB
C-CFBKBPF	CNS	CFB	1B	2B	3B	4B	5B	PF
C-CFBKBPB	CNS	CFB	1B	2B	3B	4B	5B	PB

Fonte: Elaborada pelo autor.

4.2 Instâncias Utilizadas

Para a validação do algoritmo foi utilizado uma base de dados com 24 instâncias fornecidas por pesquisadores da Universidade de Nottingham. Devido a grande variedade de problemas de escalonamento de enfermeiros, Curtois e Qu (2014) propuseram uma base de dados com instâncias reais para o problema de escalonamento de pessoal, incluindo escalonamento de enfermeiros, de modo a padronizar os parâmetros do problema. Com essa base de dados é possível realizar comparações entre os diferentes métodos de resolução para o problema.

A base de dados é composta de 24 instâncias reais com diferentes níveis de dificuldade. A instância mais simples consiste em escalar 1 tipo de turno para 8 enfermeiros em um período de 2 semanas, enquanto a instância mais elaborada consiste em designar 32 tipos de turnos para 150 enfermeiros em um período de 52 semanas. Na Tabela 4.5 é apresentado as 24 instâncias juntamente com as características de cada uma, contendo o número de semanas, enfermeiros e tipos de turnos.

Tabela 4.5: Instâncias da base de dados da Universidade de Nottingham.

Instâncias	Semanas	Enfermeiros	Tipos de turnos
Instância 1	2	8	1
Instância 2	2	14	2
Instância 3	2	20	3
Instância 4	4	10	2
Instância 5	4	16	2
Instância 6	4	18	3
Instância 7	4	20	3
Instância 8	4	30	4
Instância 9	4	36	4
Instância 10	4	40	5
Instância 11	4	50	6
Instância 12	4	60	10
Instância 13	4	120	18
Instância 14	6	32	4
Instância 15	6	45	6
Instância 16	8	20	3
Instância 17	8	32	4
Instância 18	12	22	3
Instância 19	12	40	5
Instância 20	26	50	6
Instância 21	26	100	8
Instância 22	52	50	10
Instância 23	52	100	16
Instância 24	52	150	32

Fonte: Curtois e Qu (2014).

Os autores disponibilizam a base de dados, uma interface gráfica para análise de novas soluções e um conjunto de soluções para os problemas no endereço: <http://www.cs.nott.ac.uk/psztc/NRP/>. Essa base de dados foi escolhida por ser uma base de dados recente que possui apenas instâncias reais do PEE.

Experimentos computacionais e análise dos resultados

Neste tópico são apresentados os resultados de experimentos realizados com os algoritmos propostos no tópico 4.1 utilizando as instâncias descritas na seção 4.2.

O algoritmo foi implementado em Pascal, no ambiente Lazarus V.1.4.2. Os experimentos foram realizados em um servidor Dell PowerEdge R910 com processador Intel(R) Xeon(R) CPU E7-4860 de 2.27GHz utilizando uma máquina virtual com 64 GB de RAM e sistema operacional Linux. Os tempos de processamento das versões do HPAM apresentados nessa seção se referem às execuções nesse equipamento.

5.1 Desempenho dos procedimentos de melhoria

Para verificar a eficácia de cada procedimento de busca local empregado na fase de melhoria do HPAM, foram realizados testes com as instâncias de 1 à 20 presentes na Tabela 4.5. Foi gerada uma solução para cada instância na fase construtiva do algoritmo e em seguida empregado isoladamente cada procedimento de melhoria apresentado na Tabela 4.4.

Na Tabela 5.1 é apresentado a porcentagem de redução referente ao custo da solução inicial em que cada procedimento alcançou, destacando os melhores percentuais para cada instância. A primeira coluna indica as instâncias utilizadas, sendo de 1 à 20. A segunda coluna indica o custo da solução inicial. As colunas restantes indicam a redução percentual que cada método de busca local propiciou.

Tabela 5.1: Redução de custo por meio dos procedimentos de melhoria.

Inst.	Custo Inicial	PF (%)	PB (%)	1F (%)	1B (%)	2F (%)	2B (%)	3F (%)	3B (%)	4F (%)	4B (%)	5F (%)	5B (%)	CFF (%)	CFB (%)
1	3116	0,03	0,03	0	0	0,03	0,03	0,03	0,03	0	0	0	0	0	0
2	6868	0,06	0,06	0,06	0,04	0,06	0,04	0	0	0,04	0,04	0	0	30,63	33,57
3	10982	0	0	0,02	0,02	0,02	0,05	0	0,07	0	0,01	0	0	22,00	33,82
4	19979	45,05	30,05	35,04	35,03	50,05	50,05	45,06	50,06	60,09	65,06	50,08	45,06	30,67	29,13
5	19119	20,90	20,7	15,70	31,37	20,93	15,69	26,16	26,17	20,93	26,15	10,48	26,11	34,60	45,10
6	17623	11,33	11,38	22,70	11,40	11,37	17,06	17,09	22,75	11,38	5,67	11,37	5,74	31,37	25,64
7	22958	30,44	17,39	21,73	39,16	17,40	26,12	34,79	39,10	30,41	30,48	26,12	39,09	32,73	25,79
8	28578	0,01	0,02	3,54	14,05	0,05	7,04	10,55	7,05	7,06	7,00	10,53	21,00	13,71	14,80
9	39007	15,37	17,96	15,46	7,73	15,40	17,98	15,45	17,96	25,67	15,41	17,98	15,42	46,01	45,50
10	44169	4,54	11,32	15,87	13,59	18,13	18,13	18,11	13,59	11,34	9,09	9,05	6,82	19,32	14,99
11	59097	6,71	10,12	5,08	3,39	1,70	5,08	1,70	3,39	5,09	5,10	5,08	6,78	13,97	15,60
12	75601	7,93	3,98	7,99	5,31	7,96	4,00	9,30	9,30	10,62	13,26	9,26	6,65	15,78	21,33
13	176408	23,79	22,09	21,58	19,88	26,10	22,15	24,40	24,96	26,08	27,79	28,91	25,52	35,01	28,39
14	36519	5,48	8,21	19,25	10,99	24,66	13,69	21,90	24,67	16,47	13,72	19,20	24,66	30,30	29,20
15	81430	3,68	4,91	19,65	15,97	11,07	7,38	9,84	8,61	9,84	9,83	8,59	7,36	35,55	24,16
16	47710	0,01	0,01	14,70	16,77	23,05	27,23	16,79	29,34	18,87	27,24	25,16	27,25	26,42	24,97
17	59379	3,37	3,37	3,41	6,77	3,42	3,42	5,11	3,44	5,11	5,12	6,80	6,79	12,51	13,05
18	66868	0,01	0,01	19,46	20,94	17,97	22,44	10,48	17,97	16,46	17,94	22,42	23,93	25,99	27,35
19	139024	2,17	2,89	18,72	22,32	20,90	19,44	21,61	22,32	20,88	20,17	18,00	22,33	31,79	28,78
20	256616	2,34	2,34	1,60	1,59	2,77	1,21	2,38	2,77	3,16	2,76	3,55	3,93	16,72	12,92

Fonte: Elaborada pelo autor.

Os valores expressos na coluna 2 da Tabela 5.1 e os valores utilizados para calcular o percentual de redução dos procedimentos foram obtidos por meio da função $f'''(e)$ definida em (9), a qual soma-se o custo final da escala $cs(e)$ e o número de violações rígidas (nRR) multiplicado pela penalidade atribuída a cada violação ($penRR$), sendo $penRR = 1000$.

$$f'''(e) = cs(e) + nRR \cdot penRR \quad (9)$$

Analisando a Tabela 5.1 é possível constatar que à partir da instância 4 todas as variações conseguem reduzir o custo da solução inicial. Além disso, os métodos possuem menor eficiência quando aplicados em problemas que abrangem um período de 2 semanas (instancias 1, 2 e 3), onde a fase de construção do algoritmo consegue gerar uma escala inicial com custo próximo da melhor solução local.

A maioria dos percentuais de maior redução em cada instância são obtidos pelas variações do método *Cover Fit* (CFF e CFB). Considerando que a construção da solução inicial é baseada nas restrições rígidas e preferência dos enfermeiros, as variações dos métodos PCR e *k-swap* tendem a encontrar um número menor de melhorias em relação às variações do método *Cover Fit*. Isso se deve ao fato de que o *Cover Fit* busca melhorias na escala com foco na redução das violações flexíveis referentes à demanda de enfermeiros, as

quais não são consideradas na construção da solução inicial. Porém, mesmo as variações dos métodos PCR e *k-swap* buscarem melhorias com foco no atendimento das preferências, as quais são consideradas na construção da solução inicial, em alguns casos essas variações conseguem obter uma redução significativa no custo total, reduzindo mais que 50% em algumas instâncias (linha 4 da Tabela 5.1).

5.2 Resultados das variações do algoritmo HPAM

Buscando identificar a variação do HPAM que consiga encontrar bons resultados para todas as instâncias do problema abordado, foram realizados experimentos com todas as versões apresentadas na Tabela 4.4 utilizando as 7 primeiras instâncias do PEE descritas na seção 4.2, sendo executado 10 vezes cada experimento.

Na Tabela 5.2 é apresentado os melhores resultados obtidos pelo algoritmo para as 7 menores instâncias do PEE presentes na base de dados de Nottingham. As soluções apresentadas representam o custo da escala obtida pelo algoritmo baseado nas restrições flexíveis e preferências os enfermeiros. Todas as soluções apresentadas são consideradas viáveis, as quais na escala final não há violação às restrições rígidas.

A coluna **Versão** (coluna 2) representa a variação do algoritmo HPAM que obteve a melhor solução para cada uma das instâncias. Na coluna **Custo** (coluna 3) é apresentado o custo da escala final das melhores soluções obtidas. Os valores apresentados na coluna **Iterações** (coluna 4) representam o número de vezes que a heurística de busca local correspondente é executada até que a solução se estabilize. Na coluna **Tempo** é expresso o tempo que a versão do algoritmo levou para encontrar a solução, seguindo o padrão de horas, minutos e segundos (hh:mm:ss).

Tabela 5.2: Melhores resultados obtidos para as instâncias de 1 à 7.

Instância	Versão	Custo	Iterações	Tempo
1	C-CFBPBKB	607	9	00:00:51
2	V-CFFPFKB	1032	14	00:06:13
3	C-CFBKFPB	1210	16	00:14:49
4	V-CFFPBKF	2140	11	00:08:15
5	C-PFKFCFF	2154	31	00:31:16
6	C-PFKFCFF	4244	42	01:13:23
7	C-CFFPFKB	1677	17	00:35:08

Fonte: Elaborada pelo autor.

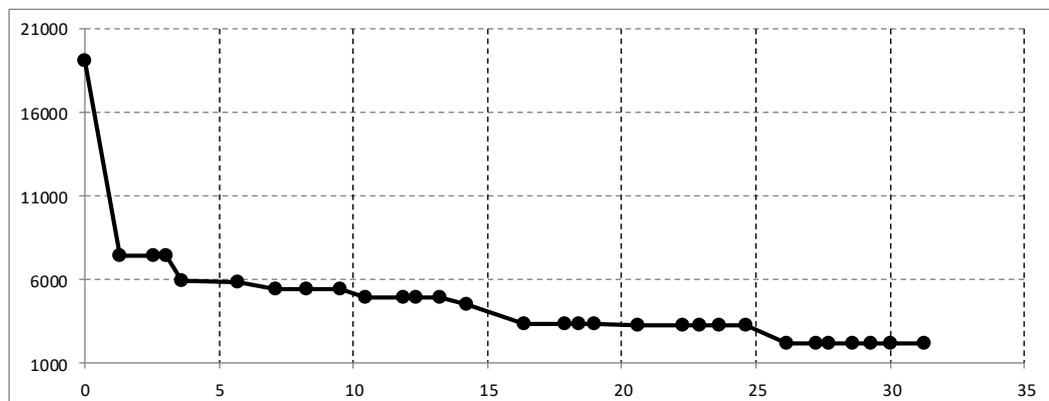
Os melhores resultados para as 7 instâncias utilizadas não foram obtidos por uma única versão do algoritmo. Porém, podemos observar que:

- O algoritmo conseguiu encontrar uma solução viável para cada instância utilizada.
- A heurística de busca local CNS encontrou melhores resultados para cinco das sete instâncias utilizadas.
- Em nenhuma versão o procedimento *k-swap* não aparece como primeiro método de melhoria.
- Para as instâncias com período maior que 2 semanas (instâncias 4 à 7), as versões que utilizam a técnica CFF como primeiro ou último procedimento tendem a encontrar soluções melhores.
- A versão C-PFKFCFF encontrou a melhor solução para duas das sete instancias utilizadas, sendo a versão que encontrou a maior quantidade de melhores soluções.

Para ilustrar a redução do custo de uma solução conforme as iterações ocorrem, na Figura 5.1 é apresentado a evolução do custo da solução gerada pela variação C-PFKFCFF para a instância 5. O eixo vertical representa os custos das soluções, enquanto o eixo horizontal representa o tempo de execução em minutos.

Os valores apresentados foram obtidos por meio da equação (9), os quais representam o custo da solução corrente. Cada ponto do gráfico representa uma iteração da heurística de busca local CNS, havendo um total de 31 iterações em um período de 31 minutos e 16 segundos, conforme apresentado na Tabela 5.2.

Figura 5.1: Evolução do custo da solução obtida por C-PFKFCFF para a instância 5.



Fonte: Elaborada pelo autor.

Nesse exemplo, o algoritmo gerou uma solução inicial com custo de 19119, representado pelo primeiro ponto no gráfico. É possível observar que ao final da primeira iteração o custo da solução é reduzido consideravelmente, alcançando uma redução de 60,9 % do custo da solução. Em geral, a primeira iteração é capaz de realizar a maior redução de custo dentre todas as iterações. Isso ocorre devido à construção da solução inicial ser baseada apenas nas restrições rígidas e preferências dos enfermeiros.

Nas últimas 6 iterações o algoritmo não consegue obter nenhuma melhoria. Nesse caso, a meta-heurística M-VNS utilizada pelo algoritmo percorreu por todas as 6 estruturas de vizinhanças apresentadas na Tabela 4.2 sem obter sucesso, então a execução é encerrada.

A evolução do custo das soluções obtidas para o restante das instâncias testadas apresentaram a mesma característica. Uma grande quantidade de melhoria é encontrada na primeira iteração reduzindo significativamente o custo da solução. As diminuições dos custos são gradativamente menores a cada iteração até que a solução se estabiliza.

Além dos melhores resultados obtidos para cada instância de 1 à 7, utilizando a ferramenta *Boxplot*, na Figura 5.2 é apresentado os resultados obtidos com as variações do algoritmo presentes na Tabela 4.4 que utilizam a heurística VND e as variações presentes na Tabela 4.4 que utilizam a heurística CNS.

Os custos das escalas finais obtidas por cada versão do algoritmo são calculados com base na equação (9).

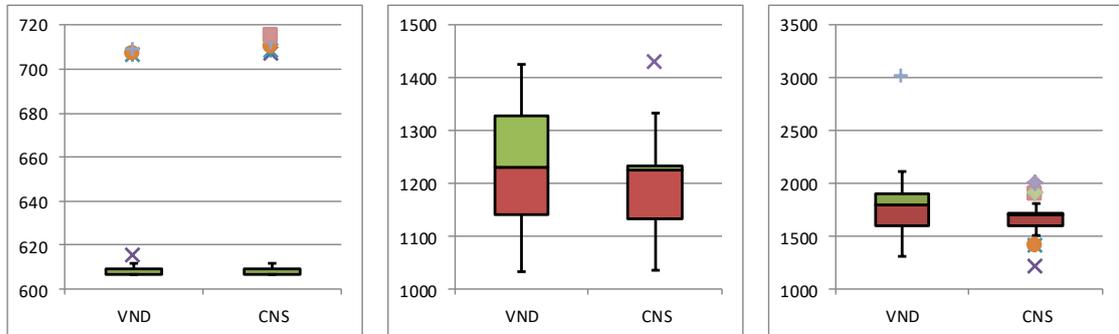
Com base na Figura 5.2 é possível observar que somente para a instância 1 representada na Figura 5.3(a) os resultados obtidos pelas versões de ambas as técnicas são consideravelmente semelhantes. Nesse caso, o melhor resultado obtido pelo algoritmo possui um custo de 607, o que foi obtido por varias versões de ambas as técnicas.

As duas técnicas apresentaram as mesmas características de simetria dos resultados em todas as instâncias utilizadas. Nas instâncias 1, 4, 5 e 7, representadas pela Figura 5.3(a), Figura 5.3(d), Figura 5.3(e) e Figura 5.3(g) respectivamente, os resultados apresentados são positivamente assimétricos. Porém, nas instâncias 2, 3 e 6, representadas pela Figura 5.3(b), Figura 5.3(c) e Figura 5.3(f) respectivamente, os resultados obtidos são negativamente assimétricos.

Além disso, as versões que utilizam a técnica CNS apresentam melhores resultados para todas as instâncias restantes, com exceção das instâncias 2 e 4, apresentadas na Figura 5.3(b) e Figura 5.3(d) respectivamente.

Os resultados obtidos pelas variações do algoritmo que utilizam a técnica CNS apresentaram menos esparsidade em relação aos resultados obtidos pelas versões que utilizam o VND, com exceção das instâncias 1 e 4. Para a instância 1 as técnicas

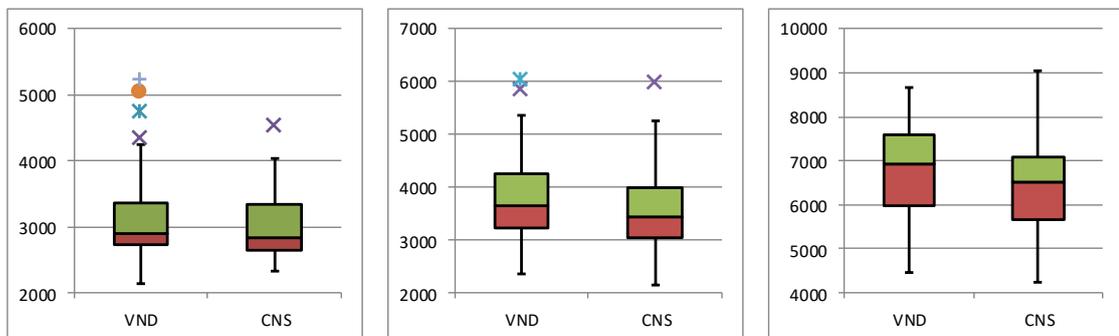
Figura 5.2: Comparativo entre as técnicas VND x CNS para as instâncias de 1 à 7.



(a) Instância 1.

(b) Instância 2.

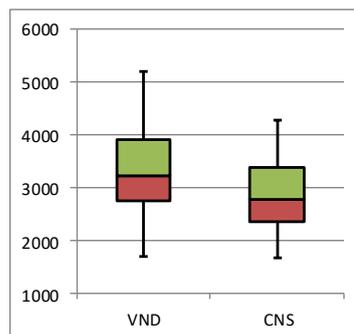
(c) Instância 3.



(d) Instância 4.

(e) Instância 5.

(f) Instância 6.



(g) Instância 7.

Fonte: Elaborada pelo autor.

apresentaram o mesmo valor de esparsidade, no qual metade dos custos das soluções obtidas variam de 607 à 612, representadas pelas caixas verdes na Figura 5.3(a).

Os melhores resultados encontrados por cada variação do algoritmo HPAM em 10 execuções para cada instância de 1 à 7 é apresentado no Apêndice A, no qual os valores apresentados foram obtidos por meio da equação (9).

Assim como feito por Demirovic *et al.* (2016), foi selecionado a melhor variação do algoritmo proposto com base na soma dos melhores resultados obtidos para cada instância utilizada. Assim, a melhor variação foi utilizada para a realização dos testes finais que serão comparados com os resultados da literatura na seção seguinte.

No Apêndice B é apresentado a soma dos melhores resultados de cada uma das variações do algoritmo, no qual a implementação C-PFKFCFF obteve a menor soma. Assim, os resultados obtidos por essa variação serão utilizados para comparar com os resultados presentes na literatura.

5.3 Comparação com resultados da literatura

Nessa seção é apresentado um comparativo dos resultados obtidos pela melhor variação do algoritmo HPAM com os melhores resultados presentes na literatura.

Burke e Curtois (2014) apresentam um algoritmo exato para o PEE baseado na abordagem *Branch and Price*. Esse algoritmo trabalha resolvendo diversos subproblemas relacionados ao escalonamento de cada enfermeiro. Para isso, o problema principal é modelado como um conjunto de problemas de atribuição (*branch*) baseado na estrutura de árvore, os quais são resolvidos por meio de programação linear, como o método *simplex*. O problema *pricing* é modelado baseado no problema do caminho mais curto, sendo utilizado uma abordagem de programação dinâmica para resolvê-lo. Esse algoritmo é aplicado na base de dados de Nottingham pelos autores da base dados (Curtois e Qu, 2014). Além disso, eles aplicaram ainda um programa chamado Gurobi 5.6.3 (Gurobi Optimization, 2014). Esse programa é um solucionador de problemas de otimização baseado em programação linear.

No trabalho de Demirovic *et al.* (2016) os autores modelam uma coleção de instâncias de escalonamento de pessoal como um problema de máxima satisfação parcialmente ponderado (maxSAT). Assim, eles comparam quatro diferentes implementações das restrições com base em seu modelo. Dessa maneira, os autores aplicam os algoritmos exatos WPM3 (Ansótegui *et al.*, 2015) e Optiriss na base de dados de Nottingham.

Na Tabela 5.3 é apresentado os resultados para as 24 instâncias da base de dados de Nottingham obtidos pela melhor versão do algoritmo HPAM proposto juntamente com os resultados presentes nos trabalhos citados acima. O algoritmo HPAM foi executado 10 vezes cada teste com um tempo limite de 96 horas cada um. O Gurobi foi limitado à um

thread com tempo máximo de execução de uma hora. Além disso, os algoritmos WPM3 e Optiriss foram executados com um tempo limite de quatro horas.

A primeira coluna da tabela representa as instâncias utilizadas nos experimentos, de 1 à 24. Na segunda e terceira coluna é apresentado os custos e tempos dos resultados obtidos pelo algoritmo HPAM. Assim como nas colunas 4-5 e 6-7 são apresentados os resultados obtidos pelo algoritmo *Branch and Price* e pelo software Gurobi, respectivamente. Na oitava e nona coluna é apresentado o custo das soluções encontradas pelos algoritmo WPM3 e Optiriss, respectivamente. Os custos destacados em negrito representam as melhores soluções encontradas para cada instância. Os valores apresentados dentro de parênteses em alguns dos resultados obtidos pelo HPAM indicam a quantidade de violações rígidas presentes na soluções, o que torna essas soluções inviáveis.

Tabela 5.3: Melhores resultados do HPAM e resultados da literatura.

Inst.	HPAM		Curtois e Qu (2014)				Demirovic <i>et al.</i> (2016)	
	Custo	Tempo	<i>Branch and Price</i>		Gurobi		WPM3	Optiriss
			Custo	Tempo	Custo	Tempo	Custo	Custo
1	607	00:01:37	607	00:00:00.27	607	00:00:01	607	607
2	1325	00:07:51	828	00:00:00.13	828	00:00:05	828	835
3	1701	00:21:54	1001	00:00:00.45	1001	00:00:13	1009	3475
4	2621	00:06:23	1716	00:00:01	1716	00:02:38	3102	3608
5	2154	00:31:16	1160	00:00:25	1143	00:25:20	4037	3645
6	4244	01:13:23	1952	00:00:10	1950	00:07:20	6450	6941
7	2368	01:48:52	1058	00:01:33	1056	00:35:52	4596	5421
8	8061(1)	02:11:45	1308	03:17:11	1323	00:59:59	11018	7617
9	1075	20:54:10	439	00:01:16	439	00:59:59	10949	6394
10	8678	38:04:27	4631	00:01:53	4631	00:04:04	16435	15350
11	21662	25:03:00	3443	00:00:19	3443	00:01:49	12183	15420
12	32961	34:58:29	4046	00:22:16	4040	00:38:23	18770	28598
13	69340(1)	73:24:30	-	-	3109	01:00:00	6110163	69203
14	8013(1)	04:39:42	-	-	1280	01:00:00	16303	9776
15	18266(3)	18:26:08	-	-	4964	01:00:00	30833	16506
16	14081(9)	00:52:11	3323	00:04:25	3233	00:59:59	10292	13026
17	26808(2)	11:11:21	-	-	5851	01:00:00	22002	22073
18	19270(2)	06:40:42	-	-	4760	00:59:59	18498	14433
19	44142(3)	28:40:33	-	-	5420	01:00:00	1698538	50274
20	117214(7)	66:57:10	-	-	-	-	5519316	147325
21	281843(17)	86:32:59	-	-	-	-	14715064	-
22	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor.

Com base na Tabela 5.3 é possível observar que o HPAM e os algoritmos da literatura não foram capazes de encontrar uma solução para as instâncias que abrangem um período de 52 semanas (instâncias 22-24) nos limites de tempo predeterminados. Além disso, os métodos apresentados por Burke e Curtois (2014) e Gurobi Optimization (2014) também não foram capazes de encontrar solução para as instâncias 13-21, com exceção da instância 16, e 20-21, respectivamente. Contudo, nossa abordagem foi capaz de encontrar uma solução para as instâncias 1-21.

Embora o HPAM não tenha encontrado novas soluções ótimas, ele foi capaz de obter a solução ótima para instância 1. Além disso, o algoritmo obteve uma solução viável para as 12 primeiras instâncias com exceção da instância 8.

Uma comparação entre os tempos de execução da abordagem proposta com os tempos de execução apresentados por Curtois e Qu (2014) e Demirovic *et al.* (2016) pode não ser considerada justa, uma vez que as plataformas computacionais utilizadas para os experimentos são diferentes. No entanto, neste trabalho realizamos algumas comparações superficiais. Os experimentos executados por Curtois e Qu (2014) foram realizados em uma máquina Intel Core 2 Duo 3.16GHz e 8 GB de RAM, enquanto os experimentos executados por Demirovic *et al.* (2016) foram realizados em uma máquina Intel Xeon E5345 2.33GHz com um total de 48 GB de RAM.

Comparando os resultados do HPAM com os resultados presentes em Curtois e Qu (2014) (colunas 4-5 e 6-7), é possível notar que o HPAM requer um tempo maior que o algoritmo *Branch and Price* e o software Gurobi para obter as soluções, exceto para as instâncias 8 e 16. Além disso, o HPAM não encontrou soluções com custos menores para nenhuma instância, porém obteve uma solução com mesmo custo para um problema (instância 1).

Com relação aos algoritmos apresentados em Demirovic *et al.* (2016), o HPAM obteve melhores resultados nas instâncias 4-7, 9 e 10 em comparação aos resultados obtidos pelo WPM3, o qual demandou mais tempo que o HPAM para obter soluções para as instâncias 1-7 e 16.

Além disso, o HPAM superou o Optiriss nas instâncias 3-7, 9 e 10, encontrando soluções com custos menores, assim como obteve soluções para as instâncias 1-7 e 16 em menor tempo.

Na Tabela 5.4 é apresentado o *gap* dos custos das melhores soluções obtidos pelo HPAM em comparação aos resultados da literatura. Os valores expressos na tabela foram obtidos com base na equação (10). As soluções para as instâncias 8 e 13-21 não são

comparadas com os resultados da literatura, considerando que essas soluções obtidas pelo HPAM possuem violações rígidas.

$$gap = \frac{(Resultado\ HPAM) - (Resultado\ literatura)}{(Resultado\ literatura)} \cdot 100 \quad (10)$$

Na primeira coluna são apresentadas as instâncias para as quais o HPAM encontrou soluções viáveis. As colunas de 2 à 5 representam o *gap* dos resultados obtidos pelo HPAM em comparação aos resultados obtidos pelos algoritmos *Branch and Price*, Gurobi, WPM3 e Optiriss, respectivamente. Os *gaps* negativos indicam uma redução no custo das soluções encontradas pelo HPAM em relação ao algoritmo relacionado, e os *gaps* positivos indicam um aumento no custo das soluções.

Tabela 5.4: Relação entre os resultados do HPAM e os resultados da literatura.

Instância	<i>Gap</i> <i>Branch and Price</i>	<i>Gap</i> Gurobi	<i>Gap</i> WPM3	<i>Gap</i> Optiriss
1	0	0	0	0
2	60,02	60,02	60,02	58,68
3	69,93	69,93	68,58	-51,05
4	52,73	52,73	-15,50	-27,35
5	85,68	88,45	-46,64	-40,90
6	117,41	117,64	-34,20	-38,85
7	123,81	124,24	-48,47	-56,31
9	144,87	144,87	-90,18	-83,18
10	87,38	87,38	-47,19	-43,46
11	529,16	529,16	77,80	40,47
12	714,65	715,86	75,60	15,25

Fonte: Elaborada pelo autor.

Com base na Tabela 5.4 é possível observar que, com exceção da instância 1, em todos os casos os algoritmos *Branch and Price* e Gurobi superaram o HPAM, principalmente nos problemas maiores (instâncias 11 e 12). Porém, na instância 1, a solução do HPAM se igualou aos resultados da literatura. Além disso, o HPAM superou os algoritmos WPM3 e Optiriss em 50% e 58% dos casos respectivamente.

Dessa maneira é possível afirmar que, apesar de alguns algoritmos exatos serem capazes de encontrar melhores soluções, o HPAM é um algoritmo híbrido promissor capaz de encontrar boas soluções para problemas de escalonamento de enfermeiros.

5.4 Considerações finais

Neste capítulo foram apresentados e analisados os resultados obtidos pelo algoritmo proposto neste trabalho. Primeiramente foi realizado uma comparação entre o desempenho das diferentes estruturas de busca local utilizados pelo algoritmo. Em seguida, foram realizadas algumas comparações entre as diversas variações do algoritmo proposto e escolhido a melhor versão. Assim, foram comparados os resultados obtidos pela melhor versão do HPAM com os resultados de algoritmos presentes na literatura.

Como observado nas seções anteriores, o algoritmo heurístico proposto não foi capaz de superar os melhores algoritmos exatos presentes na literatura. Contudo, os resultados alcançados por ele apontam que esse pode ser um algoritmo promissor para ser utilizado em diversos problemas de escalonamento de pessoal.

Conclusão

Nesta dissertação, abordamos o Problema de Escalonamento de Enfermeiros (PEE), o qual é de grande importância para os gestores dos hospitais, pois se trata de um problema que envolve uma grande quantidade de restrições que influenciam diretamente na satisfação e comprometimento dos enfermeiros no ambiente de trabalho.

Este trabalho teve como principal objetivo propor um novo algoritmo híbrido para o PEE, com a utilização de métodos presentes na literatura em conjunto com um novo método de busca local proposto.

Para isso, propomos uma nova abordagem híbrida, chamada HPAM, para a resolução do problema relacionado, que combina uma variação da meta-heurística VNS com os métodos exatos PCR e *k-swap* presentes na literatura e o método *Cover Fit* proposto neste trabalho. Essa abordagem é baseada em procedimentos exatos de complexidade polinomial na resolução dos subproblemas.

Primeiro, foram realizados testes com os métodos de busca local utilizados na fase de melhoria do algoritmo de modo a analisar a eficiência desses métodos nas instâncias do PEE utilizadas. Assim, foi construída uma escala inicial para cada instância de 1 à 20 da base de dados da Universidade de Nottingham e aplicado isoladamente cada procedimento de melhoria apresentado na Tabela 4.4.

Dessa maneira, foi possível observar que, em geral, todos os métodos obtiveram uma melhoria significativa no custo das soluções iniciais. Porém, eles possuem menor eficiência quando aplicados em problemas relativamente pequenos (instâncias 1, 2 e 3), onde a solução inicial possui um custo próximo do custo da melhor solução local. Além disso, as variações do método *Cover Fit* obtiveram melhor desempenho por buscar melhorias com

base nas restrições relacionadas à demanda de enfermeiros, as quais não são consideradas na construção da solução inicial.

A abordagem proposta está sujeita à diversas variações em relação à sequência dos procedimentos de busca local aplicados na fase de melhoria, bem como à ordem de execução que cada método de busca local emprega. Dessa maneira, foram implementadas 96 variações do algoritmo proposto com o objetivo de encontrar a configuração capaz de obter os melhores resultados. Uma versão do algoritmo proposto se destacou por ter encontrado o melhor resultado para 2 das 7 instâncias testadas e obteve a menor soma dos custos das soluções.

Em seguida, foram realizados testes com a melhor variação do HPAM na base de dados de Nottingham. A abordagem proposta foi capaz de encontrar soluções para 21 das 24 instâncias presentes na base de dados. Além disso, a abordagem obteve soluções viáveis para as instâncias 1-12, com exceção da instância 8, encontrando uma solução ótima para a instância 1.

Em comparação com os resultados das técnicas presentes na literatura, a abordagem proposta não foi capaz de superar os métodos exatos *Branch and Price* e Gurobi (Curtois e Qu, 2014), no qual o primeiro apresentou soluções com custos menores para as instâncias 2-12 e 16, e o segundo para as instâncias 2-19. Porém o HPAM superou significativamente os métodos WPM3 e Optiriss (Demirovic *et al.*, 2016) para as instâncias de tamanho médio. Em comparação ao método WPM3, o HPAM obteve melhores resultados em 50% das instâncias em que encontrou soluções viáveis, reduzindo em 90,18% o custo da solução encontrada pelo WPM3 para a instância 9. Em comparação ao método Optiriss, o algoritmo proposto obteve melhores resultados em 58% dos problemas em que encontrou soluções viáveis, no qual atingiu uma redução de 83,18% do custo da solução encontrada pelo Optiriss para a instância 9.

Como conclusão geral, os resultados obtidos neste trabalho apontam que o HPAM é um algoritmo híbrido promissor capaz de obter boas soluções para diversos problemas de escalonamento de enfermeiros.

Este trabalho tem como contribuição a apresentação de um novo algoritmo híbrido para o PEE, bem como um novo método de busca local. Além disso, este trabalho fornece novos resultados computacionais relacionados com o escalonamento de enfermeiros que podem ser utilizados para comparação por futuros trabalhos.

6.1 Trabalhos futuros

Como trabalhos futuros, pretende-se avaliar o algoritmo proposto de modo a melhorar seu desempenho nos problemas abordados com a implantação de melhorias nos procedimentos utilizados. Além disso, pretende-se investigar as causas do algoritmo não conseguir encontrar uma solução para os problemas que abrangem um período de 52 semanas dentro do tempo predeterminado, bem como a possibilidade de aplicar outros métodos de busca local na fase melhoria.

Aplicar o HPAM em outros problemas de escalonamento de modo a verificar a eficiência do algoritmo em diferentes tipos de problemas.

REFERÊNCIAS

- AICKELIN, U.; DOWSLAND, K. A. An indirect Genetic Algorithm for a nurse-scheduling problem. *Computers & Operations Research*, v. 31, n. 5, p. 761–778, 2004.
- ANSÓTEGUI, C.; DIDIER, F.; GABAS, J. Exploiting the structure of unsatisfiable cores in maxsat. In: *Proceedings of the 24th International Conference on Artificial Intelligence*, AAAI Press, 2015, p. 283–289.
- ASAP Automated scheduling, optimisation and planning research group. 2014. Disponível em <http://www.asap.cs.nott.ac.uk/>
- BURKE, E.; CAUSMAECKER, P. D.; PETROVIC, S.; BERGHE, G. V. Variable Neighborhood Search for Nurse Rostering Problems. In: *Metaheuristics: Computer Decision-Making*, n. 86 in Applied Optimization, Springer US, p. 153–172, 2003.
- BURKE, E. K.; CAUSMAECKER, P. D.; BERGHE, G. V.; LANDEGHEM, H. V. The State of the Art of Nurse Rostering. *Journal of Scheduling*, v. 7, n. 6, p. 441–499, 2004.
- BURKE, E. K.; CURTOIS, T. New approaches to nurse rostering benchmark instances. *European Journal of Operational Research*, v. 237, n. 1, p. 71–81, 2014.
- BURKE, E. K.; CURTOIS, T.; POST, G.; QU, R.; VELTMAN, B. A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research*, v. 188, n. 2, p. 330–341, 2008.
- CAPRARA, A.; MONACI, M.; TOTH, P. Models and algorithms for a staff scheduling problem. *Mathematical Programming*, v. 98, n. 1-3, p. 445–476, 2003.
- CARES, J.; RIFF, M. A Collaborative Approach to Solve a Nurse Scheduling Problem. In: *2012 IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2012, p. 344–350.

- CARPANETO, G.; TOTH, P. Primal-dual algorithms for the assignment problem. *Discrete Applied Mathematics*, v. 18, n. 2, p. 137–153, 1987.
- CARVALHO, D. B. F.; DE OLIVEIRA, G. A.; SOUZA, M. J. F. Método de Pesquisa em Vizinhança Variável Aplicado à Resolução do Problema de Roteamento de Veículos. *Aceito para publicação nos Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional-SBPO*, 2003.
- CHEANG, B.; LI, H.; LIM, A.; RODRIGUES, B. Nurse rostering problems—a bibliographic survey. *European Journal of Operational Research*, v. 151, n. 3, p. 447–460, 2003.
- CHUIN LAU, H. On the complexity of manpower shift scheduling. *Computers & Operations Research*, v. 23, n. 1, p. 93–102, 1996.
- CONSTANTINO, A. A.; LANDA-SILVA, D.; DE MELO, E. L.; DE MENDONÇA, C. F. X.; RIZZATO, D. B.; ROMÃO, W. A heuristic algorithm based on multi-assignment procedures for nurse scheduling. *Annals of Operations Research*, 2014.
- CURTOIS, T.; QU, R. Computational results on new staff scheduling benchmark instances. In: *Relatório Técnico 06-10-2014*, ASAP Research Group, School of Computer Science, University of Nottingham, 2014.
- DELLAERT, N.; JEUNET, J. A variable neighborhood search algorithm for the surgery tactical planning problem. *Computers & Operations Research*, 2016.
- DEMIROVIC, E.; MUSLIU, N.; WINTER, F. Modeling and Solving Staff Scheduling with Partial Weighted maxSAT. *11th International Conference on the Practice and Theory of Automated Timetabling*, 2016.
- DENG, Y.; BARD, J. F. A reactive GRASP with path relinking for capacitated clustering. *Journal of Heuristics*, v. 17, n. 2, p. 119–152, 2011.
- FREITAS, L. M. B.; MONTANÉ, F. A. T. Metaheurísticas vns-vnd e grasp-vnd para problemas de roteamento de veículos com coleta e entrega simultânea. *XI Simpósio de Pesquisa Operacional e Logística da Marinha*, 2008.
- GONSALVES, T.; KUWATA, K. Memetic Algorithm for the Nurse Scheduling Problem. *International Journal of Artificial Intelligence & Applications (IJAA)*, v. 6, n. 4, p. 44–52, 2015.

GUROBI OPTIMIZATION, I. Gurobi optimizer reference manual. 2014.

Disponível em <http://www.gurobi.com>

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. *European journal of operational research*, v. 130, n. 3, p. 449–467, 2001.

HANSEN, P.; MLADENOVIĆ, N.; MORENO PÉREZ, J. A. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, v. 175, n. 1, p. 367–407, 2010.

HIERMANN, G.; PRANDTSTETTER, M.; RENDL, A.; PUCHINGER, J.; RAIDL, G. R. Metaheuristics for solving a multimodal home-healthcare scheduling problem. *Central European Journal of Operations Research*, v. 23, n. 1, p. 89–113, 2015.

HILLIER, F. S. *Introduction to operations research*. 7th ed ed. Boston: McGraw-Hill, 2001.

KAWANAKA, H.; YOSHIKAWA, T.; SHINOGI, T.; TSURUOKA, S. Constraints and search efficiency in nurse scheduling problem. In: *2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2003. Proceedings, 2003*, p. 312–317 vol.1.

KUHN, H. W. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, v. 2, n. 1-2, p. 83–97, 1955.

MAENHOUT, B.; VANHOUCKE, M. New Computational Results for the Nurse Scheduling Problem: A Scatter Search Algorithm. In: GOTTlieb, J.; RAIDL, G. R., eds. *Evolutionary Computation in Combinatorial Optimization*, n. 3906 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, p. 159–170, 2006.

MAENHOUT, B.; VANHOUCKE, M. An electromagnetic meta-heuristic for the nurse scheduling problem. *Journal of Heuristics*, v. 13, n. 4, p. 359–385, 2007.

MAENHOUT, B.; VANHOUCKE, M. Comparison and hybridization of crossover operators for the nurse scheduling problem. *Annals of Operations Research*, v. 159, n. 1, p. 333–353, 2008.

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

- OHKI, M.; MORIMOTO, A.; MIYAKE, K. Nurse Scheduling by Using Cooperative GA with Efficient Mutation and Mountain-Climbing Operators. In: *2006 3rd International IEEE Conference on Intelligent Systems*, 2006, p. 164–169.
- OHKI, M.; UNEME, S.; KAWANO, H. Parallel processing of cooperative genetic algorithm for nurse scheduling. In: *Intelligent Systems, 2008. IS '08. 4th International IEEE Conference*, 2008, p. 10–36–10–41.
- OSOGAMI, T.; IMAI, H. Classification of various neighborhood operations for the nurse scheduling problem. In: *Algorithms and Computation*, Springer, p. 72–83, 2000.
- PETROVIC, S.; BERGHE, G. V. Comparison of algorithms for nurse rostering problems, p. 1–18. 2008.
- PINEDO, M. L. *Scheduling. Theory, Algorithms, and Systems*. Boston, MA: Springer US, 2012.
- RENDL, A.; PRANDTSTETTER, M.; HIERMANN, G.; PUCHINGER, J.; RAIDL, G. Hybrid Heuristics for Multimodal Homecare Scheduling. In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: 9th International Conference, CPAIOR 2012, Nantes, France, May 28 – June 1, 2012. Proceedings*, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 339–355, 2012.
- TANOMARU, J. Motivação, fundamentos e aplicações de algoritmos genéticos. In: *II Congresso Brasileiro de Redes Neurais*, 1995, p. 373–403.
- VANHOUCKE, M.; MAENHOUT, B. NSPLib—a nurse scheduling problem library: a tool to evaluate (meta-) heuristic procedures. In: *Operational research for health policy: making better decisions. Proceedings of the 31st annual meeting of the working group on operations research applied to health services*, 2007, p. 151–165.
- VOLPATO, L. T.; CONSTANTINO, A. A.; ROMÃO, W.; SAVINIEC, L. Meta-heurística VNS aplicada ao Problema de Escalonamento de Enfermeiros. In: *XLV Simpósio Brasileiro de Pesquisa Operacional*, 2013, p. 2791–2801.

Apêndice A

Melhores resultados obtidos por cada variação do HPAM em 10 execuções para as instâncias de 1 à 7 da base de dados de Nottingham.

Versão	Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7
V-PFKFCFF	607	1323	1612	2537	3438	4581	3164
V-PFKFCFB	607	1325	1906	2438	3147	4761	3277
V-PFKBCFF	615	1327	1900	2531	2941	4855	2863
V-PFKBCFB	707	1424	1902	2326	3042	4663	2968
V-PBKFCFF	607	1328	1804	2829	3954	6888	3683
V-PBKFCFB	607	1225	2102	2842	2840	6373	4072
V-PBKBCFF	609	1329	1903	2832	3052	6068	4373
V-PBKBCFB	607	1327	1803	2631	3453	5552	4168
C-PFKFCFF	607	1325	1701	2621	2154	4244	2368
C-PFKFCFB	607	1328	1705	2744	3550	4771	2762
C-PFKBCFF	608	1322	1702	2727	3437	4453	2655
C-PFKBCFB	607	1228	1708	2537	2542	4357	3163
C-PBKFCFF	607	1135	1705	2841	3645	5590	3171
C-PBKFCFB	609	1227	1903	2538	3240	6680	3081
C-PBKBCFF	607	1325	1700	2842	3250	6055	2774
C-PBKBCFB	609	1235	1811	2642	3046	5388	3472
V-PFCFFKF	609	1126	1712	3032	2355	4453	1892

Versão	Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7
V-PFCFFKB	615	1228	1401	3130	2356	6255	1877
V-PFCFBKF	707	1125	1511	3328	2752	5960	1698
V-PFCFBKB	607	1036	1409	4030	2956	4755	2177
V-PBCFFKF	607	1141	1411	2340	2969	7259	2089
V-PBCFFKB	709	1131	1311	2728	4253	7058	2085
V-PBCFBKF	607	1131	1604	2841	3262	6555	2779
V-PBCFBKB	607	1041	1702	2947	3966	5956	3083
C-PFCFFKF	607	1327	1504	2331	2640	5560	1973
C-PFCFFKB	707	1124	1504	2426	3435	4346	1977
C-PFCFBKF	715	1224	1505	2747	2946	4755	2371
C-PFCFBKB	708	1224	1602	3457	3046	4757	1995
C-PBCFFKF	608	1223	1610	2343	2643	5769	2188
C-PBCFFKB	707	1128	1702	2741	3243	6273	2866
C-PBCFBKF	608	1127	1604	2841	3443	6058	2482
C-PBCFBKB	609	1125	1510	2947	3053	5756	2594
V-KFPFCFF	706	1423	1803	2637	4377	6873	4065
V-KFPFCFB	608	1426	1800	2732	3552	8477	4985
V-KFPBCFF	607	1422	1903	3035	4163	5782	4368
V-KFPBCFB	607	1326	2104	2840	4555	7570	5201
V-KBPFCFF	607	1229	1705	2848	4557	6470	3776
V-KBPFCFB	607	1322	1902	4037	3255	6167	4777
V-KBPBCFF	607	1324	1807	2843	4557	6470	3885
V-KBPBCFB	609	1328	1902	3242	3249	7079	3579
C-KFPFCFF	612	1223	1707	3429	5260	5367	2779
C-KFPFCFB	607	1325	1701	2629	4254	6580	4287

Versão	Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7
C-KFPBCFF	607	1224	1605	2530	3149	7867	2668
C-KFPBCFB	607	1227	1803	2744	4560	6582	3378
C-KBPFCFF	607	1224	1605	3243	5968	6374	3567
C-KBPFCFB	607	1333	1605	2640	3360	5677	3387
C-KBPBCFF	607	1325	1906	2739	5968	6183	3277
C-KBPBCFB	607	1233	1703	2739	2974	5677	3375
V-KFCFFPF	607	1324	1605	2729	4149	7387	3466
V-KFCFFPB	607	1225	1907	2851	3651	5969	3969
V-KFCFBPF	607	1225	1802	2933	3447	7670	4599
V-KFCFBPB	607	1224	1803	3243	3474	5975	4081
V-KBCFFPF	607	1228	1709	2636	4261	5178	3388
V-KBCFFPB	607	1327	1806	2747	3731	6470	4479
V-KBCFBPF	607	1327	1707	3140	3944	6979	3787
V-KBCFBPB	607	1425	1708	2742	3656	6979	3291
C-KFCFFPF	607	1226	1902	2725	3845	6491	3478
C-KFCFFPB	607	1225	1507	2753	3450	6676	2871
C-KFCFBPF	607	1228	1702	2937	2566	6580	3771
C-KFCFBPB	607	1222	1709	3244	3669	6580	3162
C-KBCFFPF	607	1225	1705	2846	3559	8474	3996
C-KBCFFPB	607	1324	1600	2635	3643	7179	3269
C-KBCFBPF	608	1326	1707	3242	3358	5677	2992
C-KBCFBPB	607	1225	1804	2738	3959	5677	4008
V-CFFPFKF	607	1141	1909	3131	2658	7678	2672
V-CFFPFKB	609	1032	1604	3433	3851	8467	2966
V-CFFPBKF	607	1139	1507	2140	4166	7371	2791

Versão	Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7
V-CFFPBKB	607	1331	1707	2542	4253	8675	2971
V-CFBPFKF	709	1129	1413	4134	2448	6460	2072
V-CFBPFKB	607	1237	1414	5232	3246	7855	2777
V-CFBPBKF	609	1227	3021	3339	4245	8372	2275
V-CFBPBKB	607	1230	1611	3630	3553	7763	3190
C-CFFPFKF	611	1133	1409	3335	3355	7065	1984
C-CFFPFKB	712	1128	1505	2523	3252	6560	1677
C-CFFPBKF	607	1228	1710	2949	3855	8578	2475
C-CFFPBKB	609	1231	2006	3534	4445	6273	2261
C-CFBPFKF	710	1036	1701	2940	2855	7369	2700
C-CFBPFKB	707	1037	1707	3343	2846	7377	2291
C-CFBPBKF	610	1139	1606	3339	4063	7969	2403
C-CFBPBKB	607	1124	1610	3341	3664	7078	3788
V-CFFKFPF	607	1331	2106	3759	4958	7365	3707
V-CFFKFPB	607	1236	1708	4331	4757	7365	2197
V-CFFKBPF	608	1140	2113	2633	4945	7169	3669
V-CFFKBPB	607	1229	2010	3632	5368	8567	3276
V-CFBKFPF	612	1130	1510	5032	5367	7762	2695
V-CFBKFPB	607	1130	1909	3133	6035	8073	3185
V-CFBKBPF	607	1230	2005	4743	3577	7576	3102
V-CFBKBPB	607	1232	1503	4243	5853	7576	2290
C-CFFKFPF	607	1430	1601	4526	4450	6658	3071
C-CFFKFPB	607	1231	1602	3848	3757	6658	2292
C-CFFKBPF	607	1227	1807	3735	5040	8659	2767
C-CFFKBPB	607	1140	1702	3833	3953	9047	2191

Versão	Inst.1	Inst.2	Inst.3	Inst.4	Inst.5	Inst.6	Inst.7
C-CFBKFPF	607	1129	1415	2840	4061	7262	1987
C-CFBKFPB	607	1042	1210	3835	4559	7076	3466
C-CFBKBPF	607	1130	1808	3555	2864	6567	3867
C-CFBKBPB	607	1137	1706	4040	4181	8074	2282

Fonte: Elaborada pelo autor.

Apêndice B

Soma dos melhores resultados obtidos por cada variação do HPAM para as instâncias de 1 à 7 da base de dados da Universidade de Nottingham.

Versão	Soma	Versão	Soma	Versão	Soma
V-PFKFCFF	17262	V-KFPFCFF	21884	V-CFFPFKF	19796
V-PFKFCFB	17461	V-KFPFCFB	23580	V-CFFPFKB	21962
V-PFKBCFF	17032	V-KFPBCFF	20980	V-CFFPBKF	19721
V-PFKBCFB	17032	V-KFPBCFB	24203	V-CFFPBKB	22086
V-PBKFCFF	21093	V-KBPFCFF	21192	V-CFBPFKF	18365
V-PBKFCFB	20061	V-KBPFCFB	22067	V-CFBPFKB	22368
V-PBKBCFF	20166	V-KBPBCFF	21493	V-CFBPBKF	23088
V-PBKBCFB	19541	V-KBPBCFB	20988	V-CFBPBKB	21584
C-PFKFCFF	15020	C-KFPFCFF	20377	C-CFFPFKF	18892
C-PFKFCFB	17467	C-KFPFCFB	21383	C-CFFPFKB	17357
C-PFKBCFF	16904	C-KFPBCFF	19650	C-CFFPBKF	21402
C-PFKBCFB	16142	C-KFPBCFB	20901	C-CFFPBKB	20359
C-PBKFCFF	18694	C-KBPFCFF	22588	C-CFBPFKF	19311
C-PBKFCFB	19278	C-KBPFCFB	18609	C-CFBPFKB	19308
C-PBKBCFF	18553	C-KBPBCFF	22005	C-CFBPBKF	21129
C-PBKBCFB	18206	C-KBPBCFB	18308	C-CFBPBKB	21212
V-PFCFFKF	15179	V-KFCFFPF	21267	V-CFFKFPF	23833
V-PFCFFKB	16862	V-KFCFFPB	20179	V-CFFKFPB	22201
V-PFCFBKF	17081	V-KFCFBPF	22283	V-CFFKBPF	22277
V-PFCFBKB	16970	V-KFCFBPB	20407	V-CFFKBPB	24689
V-PBCFFKF	17816	V-KBCFFPF	19007	V-CFBKFPF	24108
V-PBCFFKB	19275	V-KBCFFPB	21167	V-CFBKFPB	24072
V-PBCFBKF	18779	V-KBCFBPF	21491	V-CFBKBPF	22840

Versão	Soma	Versão	Soma	Versão	Soma
V-PBCFBKB	19302	V-KBCFBPB	20408	V-CFBKBPB	23304
C-PFCFFKF	15942	C-KFCFFPF	20274	C-CFFKFPF	22343
C-PFCFFKB	15519	C-KFCFFPB	19089	C-CFFKFPB	19995
C-PFCFBKF	16263	C-KFCFBPF	19391	C-CFFKBPF	23842
C-PFCFBKB	16789	C-KFCFBPB	20193	C-CFFKBPB	22473
C-PBCFFKF	16384	C-KBCFFPF	22412	C-CFBKFPF	19301
C-PBCFFKB	18660	C-KBCFFPB	20257	C-CFBKFPB	21795
C-PBCFBKF	18270	C-KBCFBPF	18910	C-CFBKBPF	20398
C-PBCFBKB	16983	C-KBCFBPB	20018	C-CFBKBPB	22027

Fonte: Elaborada pelo autor.