UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

THOR CASTILHOS SANCHOTENE

Abordagem Heurística para Solução do Problema de Alocação de Médicos do HCPA

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Dra. Luciana Salete Buriol
Coorientador: Alberto Francisco Kummer Neto

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Rui Vicente Oppermann
Vice-Reitora: Prof[a]. Jane Fraga Tutikian
Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento
Diretora do Instituto de Informática: Prof[a]. Carla Maria Dal Sasso Freitas
Coordenador do Curso de Ciência de Computação: Prof. Sérgio Luis Cechin
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

O Problema de Escalonamento de Profissionais possui diversas variações e aplicações em diferentes contextos. A versão específica desse problema tratada neste trabalho é o Problema de Alocação de Médicos do Hospital de Clínicas de Porto Alegre, cujo alto grau de complexidade justifica o interesse pela pesquisa na área de Otimização Combinatória. Este trabalho, em especial, consiste na análise e implementação de um método heurístico para solução do problema do hospital, o qual será comparado com métodos exatos e com um método baseado em programação matemática.

Palavras-chave: Problema de alocação de médicos. otimização. heurística. late acceptance hill climbing. timetabling.

Heuristic Approaches for the HCPA's Physician Rostering Problem

ABSTRACT

The Personnel Scheduling Problem applies to real world problems confronted in a wide variety of areas. The specific version of this problem considered here is the HCPA's Physician Rostering Problem, whose high degree of complexity justifies the interest in Combinatorial Optimization research. This work, specifically, consists in the implementation and analysis of a heuristic-based method for the hospital's problem, which will be compared to exact methods and a mathematical programming method.

Keywords: physician rostering problem. optimization. heuristic. late acceptance hill climbing. timetabling.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

INRC-I	First International Nurse Rostering Competition
INRC-II	Second International Nurse Rostering Competition
HCPA	Hospital de Clínicas de Porto Alegre
PAM	Problema de Alocação de Médicos
LAHC	Late Acceptance Hill Climbing
MIP	Mixed-Integer Programming
CBC	COIN-OR Branch and Cut

SUMÁRIO

1 INTRODUÇÃO

A elaboração de escalas de trabalho de profissionais é um problema complexo, enfrentado por diversas empresas nas mais variadas áreas. De modo geral, a confecção de uma escala consiste em atribuir turnos de trabalho para cada empregado, de acordo com sua carga horária, a fim de suprir a demanda mínima durante todo o período de planejamento, sempre respeitando as leis trabalhistas vigentes, as preferências coletivas e individuais dos funcionários, e as demais regras da instituição. Esse conjunto de restrições torna difícil a construção manual de uma solução de escala.

Esse processo manual torna-se ainda mais complexo na área da saúde, onde a instituição deve prover um serviço de natureza continuada, e a alocação dos profissionais precisa respeitar as habilidades específicas de cada um. Por conta disso, motiva-se sua resolução de forma automatizada, a fim de se agregar eficiência, confiabilidade e justiça no processo, possibilitando ainda redução de custos e aumento da qualidade do serviço prestado. Em especial, mostrou-se interessante a modelagem desse problema como um problema de Otimização Combinatória, geralmente consistindo em um processo de minimização de uma função matemática de múltiplas variáveis discretas, sujeito a restrições na atribuição de valores a essas variáveis. A popularização desse tipo de abordagem deve-se em grande parte às duas edições da *International Nurse Rostering Competition* (INRC-I e INRC-II), que atraiu diversos pesquisadores e profissionais da área de otimização e fomentou a aplicação de diferentes técnicas de solução a um problema científico bem definido de escalonamento de profissionais de enfermagem, que tem diversos requisitos em comum com problemas reais da atualidade.

Inspirando-se no problema objeto da segunda edição da competição citada, e atendendo a um pedido da administração do Hospital de Clínicas de Porto Alegre (HCPA), um dos maiores hospitais do estado, cuja escala dos médicos é feita manualmente no momento, foram propostas uma modelagem e resolução dessa versão do Problema de Alocação de Médicos (PAM).

Na Figura 1.1 é apresentado um exemplo reduzido de uma escala semelhante a soluções de problemas reais de alocação de médicos, tal como o do Hospital de Clínicas. Neste exemplo, a escala tem vinte e oito dias, d_1 a d_{28} e seis médicos, m_1 a m_6. Em cada célula da tabela é indicado o turno de trabalho do médico no dia, dentre os três possíveis, manhã (M), tarde (T) e noite (N), ou “-” em caso de folga do médico naquele dia.

Figura 1.1: Exemplo de escala para um período com vinte e oito dias (d_1 a d_{28}), com seis médicos (m_1 a m_6) e três tipos de turno (manhã (M), tarde (T) e noite (N)).

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	d_{19}	d_{20}	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}	d_{27}	d_{28}
m_1	M	-	T	T	T	T	M	-	-	N	N	-	M	M	M	M	M	-	-	-	-	-	N	N	-	M	T	N
m_2	N	N	-	-	N	N	-	M	M	M	M	M	M	-	-	T	N	-	M	M	M	M	M	-	-	-	N	N
m_3	N	-	M	T	T	M	-	-	M	T	T	M	T	-	-	M	T	T	M	-	M	T	T	M	T	-	-	-
m_4	-	-	-	-	-	-	-	M	T	N	-	N	-	M	M	M	T	N	-	N	-	M	M	M	-	N	-	T
m_5	M	N	N	-	T	M	T	M	-	N	N	-	M	T	M	-	-	M	T	N	-	N	-	M	M	T	T	T
m_6	M	M	M	M	M	-	-	-	-	-	-	-	-	M	T	N	-	N	-	M	M	T	T	T	-	-	N	N

Fonte: Do Autor

Há múltiplos métodos possíveis para a resolução do problema do hospital, os quais podem ser distribuídos em dois grandes grupos: métodos exatos e métodos heurísticos. Os métodos exatos visam encontrar a solução ótima do problema, cujo custo da função objetivo é comprovadamente o melhor possível, ou seja, a solução que melhor atende aos requisitos do hospital. Os métodos heurísticos procuram uma solução suficientemente adequada para o problema, mas não necessariamente ou comprovadamente a melhor.

Levando em conta as constatações anteriores, propõe-se o desenvolvimento de um método baseado em heurísticas, capaz de produzir soluções viáveis para uso em cenários reais no HCPA. O uso deste método pode ser interessante para o hospital por diversos motivos. O principal deles seria o fato de não acarretar custos adicionais à instituição, por não envolver *software* proprietário, como um *solver* comercial, por exemplo. Outro motivo seria a baixa complexidade do método em termos de custos computacionais, uma vez que não há busca por soluções ótimas. Isso pode levar a uma maior velocidade na obtenção de soluções viáveis para o hospital. Essas características são especialmente interessantes se o problema do hospital for expandido ou se o método vier a ser aproveitado em outra instituição com um problema similar. Nesses casos, pode haver um número maior de áreas e de médicos para serem tratados nas escalas, tornando proibitivo o emprego de métodos exatos, em decorrência do tempo e recursos necessários para resolução do problema.

Em linhas gerais, o método proposto para resolver o problema pode ser dividido em duas partes. A primeira consiste em uma heurística construtiva, que visa, idealmente, a confecção incremental de uma solução que atenda aos requisitos mais rígidos do hospital. A segunda parte, a heurística de busca, é responsável por refinar a solução retornada pelo método construtivo, via busca local, a fim de se entregar uma solução que atenda melhor aos requisitos definidos.

O conteúdo do trabalho está dividido como segue. No Capítulo 2 é apresentada uma revisão de literatura sobre a origem do problema em questão, métodos utilizados para problemas similares e sobre a meta-heurística usada. No Capítulo 3 a versão do problema do hospital tratada no trabalho é mostrada com detalhes. O método proposto para resolução do problema é descrito no Capítulo 4. No Capítulo 5 são apresentados os resultados obtidos para algumas instâncias do problema, bem como uma comparação do desempenho do algoritmo implementado com o de outros métodos. As considerações finais e os trabalhos futuros são abordados no Capítulo 6.

2 REVISÃO DE LITERATURA

Este capítulo fornece uma visão geral sobre os trabalhos relacionados a problemas de escalonamento de profissionais, em especial na área da saúde, destacando as meta-heurísticas mais usadas para resolução desses problemas. Também apresenta detalhes sobre a origem do problema tratado neste trabalho e, por último, descreve o funcionamento da meta-heurística utilizada no método de busca proposto no Capítulo 4.

2.1 Trabalhos Relacionados

Considerados como uma subárea dos problemas de agendamento de horários, conhecidos como *timetabling problems*, os problemas de escalonamento de profissionais da área da saúde possuem uma vasta gama de abordagens estudadas e testadas ao longo das últimas décadas. Tratando-se exclusivamente desses problemas, foram implementados métodos dos mais variados tipos, tanto exatos como heurísticos, alguns dos quais são citados ao longo desta seção. Considerando o objetivo principal deste trabalho, o foco da revisão bibliográfica apresentada aqui será em métodos que utilizam meta-heurísticas para resolução de problemas similares ao Problema de Alocação de Médicos do HCPA, detalhado no Capítulo 3. Antes de apresentar as referências ao Problema de Alocação de Médicos, será feita uma revisão geral do Problema de Escalonamento de Enfermagem, devido a sua maior divulgação e similaridade com o problema dos médicos.

2.1.1 Problema de Escalonamento de Enfermagem

Na literatura pesquisada, como já dito, foram encontradas com mais facilidade e com maior nível de detalhamento soluções para problemas de escalonamento de enfermeiros, em comparação com problemas de alocação de médicos, provavelmente por sua maior demanda em hospitais e também por seu reconhecimento na área de Otimização Combinatória. Nos últimos anos, esse reconhecimento deve-se especialmente à *International Nurse Rostering Competition*, que teve grande importância no decorrer do desenvolvimento deste trabalho e é descrita a seguir.

Uma revisão bibliográfica genérica do problema de escalonamento de enfermagem pode ser encontrada em Burke et al. (2004), onde são comparadas inúmeras publicações de acordo com vários critérios, como propriedades do problema, variações na definição e métodos de resolução empregados. Esses métodos incluem, entre outros, programação matemática, programação com restrições, meta-heurísticas, e também algumas heurísticas específicas.

Foram realizadas duas edições da *International Nurse Rostering Competition*, referenciadas como INRC-I e INRC-II, nas quais foram definidas formalmente duas versões do problema de escalonamento de enfermagem, uma em cada edição.

2.1.1.1 INRC-I

Na primeira edição da competição foi proposta a resolução do problema em uma única etapa, ou seja, sem subdividir o problema em semanas. A avaliação dos competidores foi feita a partir de instâncias agrupadas em três conjuntos diferentes com base em seu tamanho e no tempo limite de execução (HASPESLAGH et al., 2010). Das propostas de resolução apresentadas na competição que fizeram uso de meta-heurísticas conhecidas, destacam-se duas, que ficaram entre as primeiras colocadas na competição.

A primeira, descrita por Bilgin et al. (2010), consiste em uma hiper-heurística, isto é, uma heurística de busca que automatiza, via técnicas de inteligência artificial, o processo de seleção e combinação de outras heurísticas mais simples ou partes delas (BURKE et al., 2010). A técnica usada na competição pode ser dividida em duas partes: a seleção de um método heurístico e a aplicação de um critério de aceitação baseado no *Simulated Annealing*.

Já Nonobe (2010) utilizou programação com restrições, com um *solver* genérico, proposto por Nonobe and Ibaraki (1998), para resolver o problema. Esse *solver*, usa como base a meta-heurística Busca Tabu, com movimentos simples e eficientes, segundo os autores.

2.1.1.2 INRC-II

Sobre a segunda edição, ela apresenta, em contraste com a primeira, um problema multiestágios, onde a solução para cada semana tem que ser produzida sozinha, uma após a outra, sem informações sobre as semanas seguintes (CESCHIA et al., 2015).

Uma proposta heurística bem-sucedida para o problema da INRC-II, que teve seu método divulgado, foi a de Kheiri et al. (2016), que propôs um método de solução que também usa hiper-heurística como base, analisando e produzindo sequências de heurísticas durante o processo de otimização.

Já após o término da competição, Dang et al. (2016) implementou um método para resolver o problema da INRC-II baseado na meta-heurística *Simulated Annealing*, que envolveu uma composição de vizinhanças. Segundo os autores, os resultados foram semelhantes aos dos finalistas, embora nem todas as regras da competição tenham sido respeitadas.

2.2 Problema de Alocação de Médicos

Assim como o Problema de Escalonamento de Enfermagem, vem sendo estudado também o Problema de Alocação de Médicos, devido a sua crescente demanda, dada a dificuldade de tratar-se manualmente o problema. Em especial, o trabalho de Wickert, Neto and Buriol (2018) apresenta uma proposta de definição formal para o Problema de Alocação de Médicos do HCPA, referenciado como PRP (*Physician Rostering Problem*), o qual é fortemente inspirado no problema objeto da segunda edição da *International Nurse Rostering Competition*, com leves alterações nas restrições a fim de se atender melhor aos requisitos do hospital.

Além da *International Nurse Rostering Competition*, os autores também citam diversos outros trabalhos como referência, incluindo algumas propostas de resolução de problemas reais de escalonamento de médicos ao redor do mundo. Bruni and Detti (2014) propuseram uma formulação matemática flexível, em programação inteira mista, a qual pode ser facilmente modificada para representar diferentes cenários, e também implementaram diferentes soluções para uma versão real do problema, as quais se mostraram mais efetivas que as soluções manuais adotadas na época. Já Brunner, Bard and Kolisch (2009) propuseram um modelo com turnos flexíveis de trabalho, isto é, turnos com duração e horário de início variáveis, o que permite maior autonomia no processo de construção da escala. As soluções apresentadas também se mostraram superiores, quando comparadas a soluções adotadas por uma instituição médica. Vale destacar também o trabalho de Rousseau, Pesant and Gendreau (2002), onde é apresentada uma das primeiras abordagens genéricas para problemas de alocações de médicos, a qual pode ser considerada como a base de diversos trabalhos posteriores da área.

Após apresentar a definição, os autores também comparam, uma a uma, as restrições do problema do hospital com as do problema da competição, verificando quais são aplicáveis a ambos. Também é proposta uma formulação matemática em programação inteira para o problema dos médicos, considerando tais restrições.

Por último, propôs-se um método para resolução do problema, baseado na *matheuristic Fix-and-optimize*, o qual teve seus resultados comparados com o *solver* CPLEX, após ambos serem aplicados a instâncias fictícias do problema, geradas de forma sistemática, com base nos dados reais do hospital. Essas instâncias foram divididas em três grupos (instâncias pequenas, médias e grandes), de acordo com o número de médicos de cada uma (50, 100 e 150, respectivamente). As execuções respeitaram limites pré-definidos de tempo, de acordo com o método e com o grupo ao qual a instância pertencia.

As conclusões apresentadas nesse trabalho consideraram esses diferentes grupos de instâncias. Para as instâncias pequenas, o *solver* CPLEX para problemas de programação inteira mista (MIP) atingiu a otimalidade em todas as instâncias, mesmo tendo sido executado dentro de um limite relativamente curto de tempo (20 minutos). Nas instâncias grandes, o CPLEX só alcançou soluções factíveis ao ser executado por um período mais longo de tempo (diversas horas). Já a heurística proposta obteve bons resultados para todos os grupos de instâncias.

O artigo de Wickert, Neto and Buriol (2018) é a base deste trabalho, que propõe um método independente de *solver* para a resolução do problema proposto pelos autores, considerando a definição formal e a formulação matemática que foram apresentadas para o mesmo. Também são usados alguns dos resultados divulgados, a fim de contextualizar melhor o desempenho do método proposto neste trabalho.

2.3 *Late Acceptance Hill Climbing*

Para a definição da meta-heurística a ser usada no processo de resolução do PAM, foram levadas em conta diversas técnicas heurísticas propostas na literatura, em especial a proposta de Burke and Bykov (2008), onde a técnica estocástica *Late Acceptance Hill Climbing* é definida e aplicada a um problema de *timetabling*. Especificamente, o algoritmo escolhido para compor a base do método de busca desenvolvido neste trabalho é a proposta de Burke and Bykov (2012), considerada pelos próprios autores uma versão melhorada de sua proposta inicial, de 2008. Ambas as versões foram definidas implicitamente para problemas de minimização, mas podem ser facilmente adaptadas para problemas de maximização.

A ideia principal da meta-heurística é permitir que sejam aceitas soluções piores que a melhor encontrada até o momento, mas mantendo a simplicidade e eficiência do *Hill Climbing* guloso, e sem fazer uso de mecanismos artificiais de resfriamento, como no *Simulated Annealing*. Para tanto, além da variável contendo a solução atual, retornada no final do método, o algoritmo mantém também uma lista circular com o custo das últimas soluções aceitas, similar à Busca Tabu, mas com a diferença de apenas os custos serem armazenados. Essa lista é inicializada com o custo de uma solução inicial, geralmente aleatória, e seu tamanho é o único parâmetro necessário para o método.

De forma geral, para entendimento do funcionamento do LAHC, essa lista pode ser vista como uma fila onde, a cada iteração, o primeiro elemento dessa fila (aquele que foi inserido nela há mais tempo) é retirado e comparado com o custo da nova solução gerada na iteração (candidata). Então, há duas possibilidades: se os custos forem iguais ou se o menor custo da comparação for o da solução candidata, seu custo é inserido no fim da fila e essa solução passa a ser a atual; caso contrário, se o menor custo for aquele do início da fila, então é inserido no final da fila o custo da solução atual. Vale ressaltar que a solução candidata mencionada é sempre produzida a partir da solução atual (no trabalho, foi usada a estrutura de vizinhança descrita na Seção 4.4.1).

O nome da meta-heurística é justificado pelo fato da solução gerada em uma iteração vir a ser usada no procedimento de aceitação de novas soluções somente várias iterações depois, ao contrário do *Hill Climbing* simples, onde a solução candidata é sempre comparada com a última solução aceita (BURKE; BYKOV, 2012).

No meio desse processo todo, algo indesejado pode ocorrer quando o custo presente no início da fila é menor que o custo da solução atual. Basta que o custo da solução candidata gerada na iteração esteja entre esses dois custos, fazendo com que a solução seja rejeitada, mesmo sendo melhor que a atual, já que a comparação é feita apenas com o custo na lista. Para evitar esse problema, na proposta de Burke and Bykov (2012), escolhida para este trabalho, foi adicionada uma segunda comparação entre o custo da solução candidata e o da solução atual.

A Figura 2.1 apresenta o pseudocódigo do algoritmo LAHC descrito em Burke and Bykov (2012). Considera-se L a lista circular do algoritmo, t o tamanho dessa lista (entrada do método), $C()$ a função de custo do problema, s_0 a solução inicial, s a solução atual (saída do método) e s' a solução candidata. Já w, i e v são variáveis auxiliares. Nas duas primeiras linhas, uma solução inicial s_0 é gerada e atribuída à variável contendo a solução corrente, s. Então, nas linhas 3 à 5, os t elementos da lista L são definidos como sendo o custo dessa solução ($C(s)$). Da linha 7 à linha 15 ocorre a parte principal do método: um laço que é executado até ser atingida uma condição de parada pré-definida (número de iterações, convergência do algoritmo, valor aceitável para a solução, tempo esgotado, etc.). A cada iteração, é calculada a posição do elemento da lista circular que será usado para comparação nessa iteração (linha 8). Então, uma solução candidata s' é gerada (linha 9) e avaliada pelo critério de aceitação (linha 10). Uma solução candidata é aceita se seu custo for igual ou menor que o custo $L[v]$ da lista, ou igual ou menor que o custo da solução corrente. Neste caso, se s' é aceita, s é atualizada com seu conteúdo (linha 11). Em seguida, aquela posição da lista (v, no caso) é atualizada com o custo da solução corrente s (linha 13) e o laço recomeça, a menos que a condição de parada tenha sido alcançada. No término do laço, a solução corrente é devolvida como saída (linha 16).

Figura 2.1: Pseudocódigo da meta-heurística LAHC utilizada.

Algoritmo `LAHC` (t)

1: Gera solução inicial s_0
2: $s \leftarrow s_0$
3: **for** $w \in 0...t-1$ **do**
4: $\quad L[w] \leftarrow C(s)$
5: **end for**
6: $i \leftarrow 0$
7: **repeat**
8: $\quad v \leftarrow i \,\%\, t$
9: $\quad$ Gera uma solução candidata s'
10: $\quad$ **if** $C(s') \leq L[v]$ *ou* $C(s') \leq C(s)$ **then**
11: $\quad\quad s \leftarrow s'$
12: $\quad$ **end if**
13: $\quad L[v] \leftarrow C(s)$
14: $\quad i \leftarrow i+1$
15: **until** condição de parada atingida
16: **return** s

3 DEFINIÇÃO DO PROBLEMA

Neste capítulo será descrita em detalhes a definição formalizada do Problema de Alocação de Médicos (PAM) presente no trabalho de Wickert, Neto and Buriol (2018), a ser usada ao longo deste trabalho. Primeiramente, uma visão geral do PAM é apresentada na Seção 3.1. Em seguida, as restrições consideradas no problema são classificadas e enumeradas na Seção 3.2. Na Seção 3.3 é apresentada a formulação matemática do problema, de acordo com o artigo citado. Por último, na Seção 3.4, são apresentados, em alto nível de abstração, um exemplo de instância do problema e uma solução factível para essa instância.

3.1 Visão Geral do PAM

O objetivo do problema é construir uma escala para um conjunto específico de médicos, considerando um determinado período de escalonamento, e um conjunto pré-definido de áreas de trabalho. Cada dia do período é subdividido em um conjunto de turnos pré-estabelecidos: manhã (M), das 8h às 14h; tarde (T), das 14h às 20h; e noite (N), das 20h às 8h. Assim sendo, o processo de produção da escala consiste na alocação dos médicos em combinações de dias, turnos e áreas, de modo a suprir a demanda mínima de médicos de cada uma dessas combinações, obedecendo, ao máximo, às regras acordadas.

3.2 Restrições

O problema em questão considera dois grupos disjuntos de restrições: restrições fortes e restrições fracas. As restrições fortes devem ser obrigatoriamente satisfeitas para que uma escala possa ser utilizada na prática. Já violações das restrições fracas levam a uma solução de pior qualidade. Se ao menos uma restrição forte for violada, a solução é considerada *infactível*. As restrições de ambos os tipos são apresentadas a seguir, com leves alterações com relação à definição presente no artigo, a fim de retratar mais adequadamente o modelo matemático proposto no mesmo.

Restrições Fortes:

H1: A demanda mínima de cada combinação de dias, turnos e áreas deve ser suprida.

H2: A demanda máxima de cada combinação de dias, turnos e áreas não pode ser ultrapassada.

H3: Um médico não pode trabalhar em mais de um turno no mesmo dia útil.

H4: Em dias não úteis, cada médico alocado deve trabalhar ou em ambos os turnos diurnos (manhã e tarde), ou no turno da noite.

H5: Para cada médico, a sucessão de turnos trabalhados em dias consecutivos deve ser válida (exemplo: é proibido o mesmo médico trabalhar no turno da noite em um dia e no turno da tarde no dia seguinte).

H6: Um médico não pode trabalhar em um turno em que esteja ausente.

H7: Um médico não pode ser alocado em mais de uma área no mesmo dia.

H8: Um médico não pode trabalhar em uma área para a qual não tem autorização.

Restrições Fracas:

S1: O número de noites consecutivas trabalhadas por cada médico deve ser inferior ou igual ao máximo permitido.

S2: O número de dias de trabalho consecutivos de cada médico deve ser inferior ou igual ao máximo permitido.

S3: Um médico não deve trabalhar em um turno indesejado.

S4: Um médico deve trabalhar ou nos dois dias de um final de semana ou em nenhum deles, caso tenha esta preferência.

S5: O número de horas totais trabalhadas por cada médico deve ser maior ou igual a sua carga horária mínima.

S6: O número de horas totais trabalhadas por cada médico deve ser menor ou igual a sua carga horária máxima (não deve haver horas extras).

S7: O número de finais de semana trabalhados por cada médico deve ser inferior ou igual ao máximo permitido para ele.

3.3 Modelagem Matemática

Esta seção apresenta a formulação matemática do problema do HCPA em programação inteira, proposta por Wickert, Neto and Buriol (2018), que busca refletir fielmente as restrições descritas na seção anterior. Foram feitas pequenas modificações a fim de considerar as restrições de demanda como fortes. Antes da formulação em si, na Tabela 3.1 são apresentados os índices, conjuntos e variáveis usados.

Tabela 3.1: Índices, conjuntos e variáveis usados na formulação matemática.

Símbolo	Definição
Entrada	
$n \in N$	n é o índice do médico, e N o conjunto de médicos;
$d \in D$	d é o índice do dia, e D o conjunto de dias;
$d \in \tilde{D}$	d é o índice do dia não útil, e $\tilde{D}$ o conjunto de dias não úteis;
$s \in S$	s é o índice do turno, e S o conjunto de turnos;
$k \in K$	k é o índice da área, e K o conjunto de áreas;
$l_{nk} \in \{0,1\}$	l_{nk} indica se alocações do médico n na área k são permitidas (igual a 1) ou proibidas (igual a 0);
$(n,d,s) \in R$	conjunto contendo triplas com o dia d e turno s para o médico n que não está disponível para trabalhar;
$(n,d,s) \in U$	conjunto contendo triplas com o dia d e turno s indesejados para o médico n;
$(s',s'') \in \hat{S}$	conjunto contendo os pares de sucessões inválidas de turnos em dias adjacentes (s' não pode ser seguido por s'');
$w \in W$	w é o índice de um sábado e W o conjunto de todos os índices de sábados da escala;
α^i_{dsk}	limite da restrição forte $i \in 1,2$;
β^i_n	limite da restrição fraca $i \in 1,2,5,6,7$;
ω^i_n	peso por violar o limite da restrição fraca i para o médico n.
Variáveis de decisão	
$x_{ndsk} \in \{0,1\}$	1 se o médico n está alocado no dia d, turno s e área k, 0 caso contrário;
$y_{nw} \in \{0,1\}$	1 se o médico n trabalha no fim de semana w, 0 caso contrário;
$z_{nd} \in \{0,1\}$	1 se o médico n trabalha no dia d, durante ambos os turnos diurnos, 0 caso contrário;

$o_{nd} \in \{0,1\}$	1 se o médico n está alocado para trabalhar no dia d, 0 caso contrário.

Variáveis auxiliares

$a^i_{nd} \in \mathbb{N}^*$	número de violações da restrição fraca $i \in (1,2)$ para o médico n no dia d;
$c^3_{nds} \in \mathbb{N}^*$	número de violações da restrição fraca 3 para o médico n no dia d e turno s;
$g^4_{nw} \in \mathbb{N}^*$	número de violações da restrição fraca 4 para o médico n no fim de semana w;
$h^i_n \in \mathbb{N}^*$	número de violações da restrição fraca $i \in (5..7)$ para o médico n.

A formulação matemática começa com a definição da função objetivo a ser minimizada, onde, para cada violação de uma restrição fraca, existe um peso para sua penalização. Em seguida, é feita a modelagem de todas as restrições do problema.

$$\begin{aligned}\textbf{Min} \quad & \left[\sum_{n\in N}\sum_{d\in D}\sum_{i=1,2} a^i_{nd}\omega^i_n\right] + \left[\sum_{n\in N}\sum_{d\in D}\sum_{s\in S} c^3_{nds}\omega^3_n\right] + \\ & \left[\sum_{n\in N}\sum_{w\in W} g^4_{nw}\omega^4_n\right] + \left[\sum_{n\in N}\sum_{i=5..7} h^i_n\omega^i_n\right]\end{aligned} \tag{3.1}$$

Sujeito a

$$\sum_{n\in N} x_{ndsk} \geq \alpha^1_{dsk} \qquad \forall d \in D, s \in S, k \in K \tag{3.2}$$

$$\sum_{n\in N} x_{ndsk} \leq \alpha^2_{dsk} \qquad \forall d \in D, s \in S, k \in K \tag{3.3}$$

$$\sum_{s\in S}\sum_{k\in K} x_{ndsk} \leq 1 \qquad \forall n \in N, d \in D \setminus \tilde{D} \tag{3.4}$$

$$\sum_{k\in K} (x_{ndsk} + x_{nd(s+1)k}) = 2z_{n\tilde{d}} \qquad \forall n \in N, d \in \tilde{D}, s = 1 \tag{3.5}$$

$$\sum_{k\in K} x_{ndsk} + z_{nd} \leq 1 \qquad \forall n \in N, d \in \tilde{D}, s = 3 \tag{3.6}$$

$$\sum_{k\in K} (x_{nds'k} + x_{n(d+1)s''k}) \leq 1 \qquad \forall n \in N, d \in D \setminus \{|D|\}, (s', s'') \in \hat{S} \tag{3.7}$$

$$\sum_{k\in K} x_{ndsk} = 0 \qquad \forall (n,d,s) \in R \tag{3.8}$$

$$x_{ndsk} - x_{nd(s+1)k} = 0 \qquad \forall n \in N, d \in \tilde{D}, s = 1, k \in K \tag{3.9}$$

$$(1 - l_{nk})x_{ndsk} = 0 \qquad \forall n \in N, d \in D, s \in S, k \in K \tag{3.10}$$

$$\sum_{d'=d}^{\beta^1_n+d}\sum_{k\in K} x_{nd'sk} - a^1_{nd} \leq \beta^1_n \qquad \forall n \in N, d \in \{1,\ldots,|D| - \beta^1_n\}, s = 3 \tag{3.11}$$

$$\sum_{s\in S}\sum_{k\in K} x_{ndsk} \leq 2o_{nd} \qquad \forall n \in N, d \in D \tag{3.12}$$

$$\sum_{d'=d}^{\beta_n^2+d} o_{nd'} - a_{nd}^2 \leq \beta_n^2 \qquad \forall n \in N, d \in \{1,\ldots,|D|-\beta_n^2\} \tag{3.13}$$

$$\sum_{k\in K} x_{ndsk} \leq c_{nds}^3 \qquad \forall (n,d,s) \in U \tag{3.14}$$

$$o_{nw} + o_{n(w+1)} + g_{nw}^4 = 2y_{nw} \qquad \forall n \in N, w \in W \tag{3.15}$$

$$\sum_{d\in D}\sum_{s\in\{1,2\}}\sum_{k\in K} x_{ndsk} + \sum_{d\in D}\sum_{s\in\{3\}}\sum_{k\in K} 2x_{ndsk} + h_n^5 \geq \beta_n^5 \qquad \forall n \in N \tag{3.16}$$

$$\sum_{d\in D}\sum_{s\in\{1,2\}}\sum_{k\in K} x_{ndsk} + \sum_{d\in D}\sum_{s\in\{3\}}\sum_{k\in K} 2x_{ndsk} - h_n^6 \leq \beta_n^6 \qquad \forall n \in N \tag{3.17}$$

$$\sum_{w\in W} y_{nw} - h_n^7 \leq \beta_n^7 \qquad \forall n \in N \tag{3.18}$$

$$x_{ndsk} \in \{0,1\} \qquad \forall n \in N, d \in D, s \in S, k \in K \tag{3.19}$$

$$y_{nw} \in \{0,1\} \qquad \forall n \in N, w \in W \tag{3.20}$$

$$z_{nd} \in \{0,1\} \qquad \forall n \in N, d \in D \tag{3.21}$$

$$o_{nd} \in \{0,1\} \qquad \forall n \in N, d \in D \tag{3.22}$$

As restrições (3.2) à (3.10) modelam as restrições fortes. Especificamente, as restrições (3.2) asseguram o cumprimento da restrição H1 do problema. As restrições (3.3) correspondem à H2. As restrições (3.4) garantem H3. Já os conjuntos de restrições (3.5) e (3.6) correspondem à H4, o conjunto (3.7) à H5, (3.8) à H6, (3.9) à H7 e (3.10) à H8.

As restrições (3.11) à (3.18) modelam as restrições fracas do problema. O conjunto de restrições (3.11) calcula o número de violações da restrição S1. As restrições (3.12) calculam o valor de variáveis auxiliares usadas para cálculo das restrições S2 e S4. As restrições (3.13) correspondem à S2, (3.14) à S3, (3.15) à S4, (3.16) à S5, (3.17) à S6 e (3.18) à S7. As restrições (3.19) à (3.22) definem as variáveis de decisão como sendo binárias.

3.4 Exemplo de Instância

Cada instância do problema pode ser dividida logicamente em 3 partes: Cenário, Dados Semanais e Histórico. O Cenário contém as informações gerais sobre a instância, em especial sobre cada um dos médicos. A informação da demanda de cada combinação de dias, turnos e áreas, bem como as ausências e pedidos de folga dos médicos constam nos Dados Semanais. Já o Histórico contém os dados relativos à solução adotada no período anterior que são importantes para a confecção de uma escala para o novo período.

Assim sendo, seguindo a especificação deste capítulo, a Figura 3.1 apresenta, de maneira abstrata, uma instância simples do problema, composta por apenas uma semana, tendo naturalmente cinco dias úteis (d_1 a d_5) e dois dias não úteis (d_6 e d_7), seis médicos (m_1 a m_6) e duas áreas de trabalho (a_1 e a_2). Nesta figura, cada médico é associado a um contrato, contendo informações úteis para solucionar o problema. Cada médico também é associado a um conjunto de áreas em que possui as habilidades necessárias para trabalhar. Além disso, também consta uma tabela com os números mínimo e máximo de médicos permitidos para cada combinação de dias, turnos e áreas. As demais informações são consideradas autoexplicativas. Assumiu-se que os pesos (custos) das violações, para cada um dos tipos de restrição, são pré-definidos.

Figura 3.1: Instância com um período de apenas uma semana, seis médicos e duas áreas.

Cenário:

- Número de semanas: 1
- Áreas: a_1, a_2
- Sucessões proibidas de turnos: (N, M), (N, T)
- Contratos:
  - C1
    - Carga horária mínima (em turnos de 6 horas): 6
    - Carga horária máxima (em turnos de 6 horas): 8
    - Número máximo de dias consecutivos de trabalho: 5
    - Número máximo de noites consecutivas de trabalho: 3
    - Número máximo de fins de semana trabalhados: 1
    - Preferência por trabalhar nos dois dias dos fins de semana: Sim
  - C2
    - Carga horária mínima (em turnos de 6 horas): 4
    - Carga horária máxima (em turnos de 6 horas): 6
    - Número máximo de dias consecutivos de trabalho: 5
    - Número máximo de noites consecutivas de trabalho: 3
    - Número máximo de fins de semana trabalhados: 1
    - Preferência por trabalhar nos dois dias dos fins de semana: Não

- Contrato e áreas de trabalho permitidas para cada médico:

Médico	Contrato	Áreas permitidas
m_1	C1	a_1
m_2	C1	a_1
m_3	C1	a_1
m_4	C2	a_2
m_5	C2	a_2
m_6	C2	a_1, a_2

Dados Semanais:

- Demanda:

Turno	Área	d_1		d_2		d_3		d_4		d_5		d_6		d_7	
		mín	máx	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx	mín	máx
M	a_1	1	2	1	2	1	2	1	2	1	2	2	3	2	3
T	a_1	1	2	1	2	1	2	1	2	1	2	2	3	2	3
N	a_1	1	2	1	2	1	2	1	2	1	2	2	3	2	3
M	a_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	a_2	1	1	1	1	1	1	1	1	1	1	0	0	0	0
N	a_2	1	1	1	1	1	1	1	1	1	1	2	3	2	3

- Ausências:
  - Médico m_1 no dia d_3 no turno M
  - Médico m_2 no dia d_5 no turno T

- Pedidos de folga:
  - Médico m_6 no dia d_1 em todos os turnos

Histórico:

- Os médicos m_1, m_2 e m_5 trabalharam nos 4 últimos dias do período anterior.
- O médico m_4 trabalhou no último dia do período anterior.
- O médico m_2 trabalhou nos 2 últimos dias do período anterior no turno da noite.

Fonte: Do Autor

A Figura 3.2 apresenta um exemplo de uma solução factível para a instância da Figura 3.1. Deve-se lembrar que, para uma solução ser considerada factível, ela deve apenas respeitar todas as restrições fortes, mas pode ignorar as fracas. Na figura abaixo, cada célula da tabela apresenta a área em que o médico da linha foi alocado no turno e dia da coluna, ou um espaço em branco, caso o médico não trabalhe naquele turno. Por exemplo, no dia d_1 o médico m_1 trabalha apenas no turno da manhã (M), na área a_1, e tem folga nos demais turnos.

Figura 3.2: Escala com sete dias (d_1 a d_7), seis médicos (m_1 a m_6) e duas áreas de trabalho (a_1 e a_2).

	d_1			d_2			d_3			d_4			d_5			d_6			d_7		
	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N
m_1	a_1			a_1						a_1				a_1		a_1	a_1		a_1	a_1	
m_2			a_1			a_1			a_1				a_1					a_1			a_1
m_3		a_1			a_1			a_1				a_1			a_1			a_1			a_1
m_4		a_2				a_2			a_2			a_2						a_2			a_2
m_5			a_2					a_2			a_2				a_2			a_2			a_2
m_6					a_2		a_1				a_1			a_2		a_1	a_1		a_1	a_1	

Fonte: Do Autor

4 MÉTODO PROPOSTO

Este capítulo destina-se a caracterização do método proposto para resolução do problema descrito no capítulo anterior. Em linhas gerais, o método implementado divide-se em duas partes: um algoritmo para geração de uma solução inicial para uma determinada instância do problema, e um algoritmo para refinamento de uma dada solução.

4.1 Estrutura de Dados da Solução

Para armazenar uma solução, foi usada uma tabela *hash*, na qual cada chave do *hash* representa uma dentre as múltiplas combinações de médicos, dias, turnos e áreas. Para isso, foi usado um simples *string* na forma de 4 números separados entre si por *underscores*, onde cada número representa um dos índices da combinação. Por exemplo, assumindo-se que os índices começam em zero, a palavra *"2_14_1_2"* refere-se à combinação formada pelo terceiro médico, pelo décimo quinto dia da escala, pelo segundo turno (tarde) e pela terceira área, considerando a ordem em que aparecem na instância. Neste caso, se a combinação fizer parte da solução em questão, o valor correspondente a essa chave na estrutura deve ser 1. Caso não faça parte, o valor pode ser zero ou então a chave pode simplesmente não aparecer no *hash*.

4.2 Validador de Soluções

Para facilitar a caracterização das soluções, isto é, seus pontos fracos e fortes, foi implementado um conjunto de funcionalidades auxiliares, chamado aqui de um validador ou verificador de soluções. Dada uma instância, além de sua função básica de retornar o custo de uma solução, o validador permite a visualização detalhada de todas as violações da solução fornecida para essa instância, incluindo um resumo contendo o número de violações e o custo total para cada um dos diversos tipos de restrição. O funcionamento do validador bem como o formato de sua saída seguem as ideias básicas do validador disponível na segunda competição do problema de enfermagem (INRC-II).

A existência de um validador como esse permite avaliar a qualidade do método sendo aplicado e, mais importante que isso, conduz as mudanças a serem efetuadas em vários pontos do método, a fim de evoluí-lo. Em especial, neste trabalho, a visualização das violações das soluções foi de extrema importância para a determinação de quais estruturas de vizinhança poderiam ser úteis na heurística de busca.

4.3 Heurística Construtiva

A meta-heurística de busca descrita na Seção 2.3 necessita como entrada uma solução inicial para ser refinada. A geração dessa solução inicial se dá a partir de um algoritmo pseudoaleatório, que recebe uma instância e visa retornar uma solução para ela o mais próximo possível da factibilidade, isto é, que satisfaça ao máximo as restrições fortes impostas.

O pseudocódigo do algoritmo implementado consta na Figura 4.1. Para simplificação, chamou-se de turno Diurno a junção dos turnos Manhã e Tarde, em conformidade com a restrição H4 do problema (ver a Seção 3.2). Assim sendo, ao se alocar um médico em um turno Diurno de um dia, está na verdade se alocando esse médico nos turnos Manhã e Tarde daquele dia. Também foi considerada a existência de uma função *booleana* $demandaNaoAtingida()$, que recebe, na ordem, uma instância do problema, uma solução, um dia, um turno e uma área e retorna $true$ quando a solução dada não possui médicos suficientes alocados na combinação formada pelo dia, turno e área fornecidos, segundo informações presentes na instância. As demais considerações são autoexplicativas.

Figura 4.1: Pseudocódigo simplificado da heurística construtiva implementada.

Algoritmo `heuristicaConstrutiva` (*instancia*)

```
for (sab, dom) ∈ FinsDeSemana do
    for area ∈ Areas do
        for turno ∈ {Diurno, Noite} do
            while demandaNaoAtingida(instancia, sol, sab, turno, area) and
            demandaNaoAtingida(instancia, sol, dom, turno, area) do
                medico ← medicoAleatorio()
                sol.adiciona(medico, sab, turno, area)
                sol.adiciona(medico, dom, turno, area)
                if not sol.factivel then
                    sol.remove(medico, sab, turno, area)
                    sol.remove(medico, dom, turno, area)
                end if
            end while
        end for
    end for
end for
for dia ∈ DiasDaEscala do
    for turno ∈ {Manha, Tarde, Noite} do
        for area ∈ Areas do
            while demandaNaoAtingida(instancia, sol, dia, turno, area) do
                medico ← medicoAleatorio()
                sol.adiciona(medico, dia, turno, area)
                if not sol.factivel then
                    sol.remove(medico, dia, turno, area)
                end if
            end while
        end for
    end for
end for
return sol
```

Fonte: Do Autor

Baseado em relato do responsável pela confecção das escalas atualmente no hospital, começar as alocações de médicos pelos fins de semana, em especial, alocando-se um único médico nos dois dias do mesmo fim de semana no mesmo turno (diurno ou noturno), mostra-se eficaz no desenvolvimento das soluções finais. Isso pode ser explicado pelo fato de haver menos violações relativas às restrições S4 e S7, restrições custosas, que tratam, respectivamente, de fins de semana completos e de número máximo de fins de semana trabalhados por cada médico. Esse processo está representado da linha 1 à linha 15 do pseudocódigo. De fato, a incorporação dessa estratégia ao processo puramente aleatório (linha 16 à 28) resultou em um melhor desempenho do método de busca aplicado posteriormente, embora a solução gerada pelo método construtivo tenha custos semelhantes com e sem essa estratégia.

Vale ressaltar que o pseudocódigo mostrado contém algumas simplificações para facilitar a visualização da lógica geral do algoritmo. Em primeiro lugar, nos testes de factibilidade (linhas 8 e 22) a restrição de demanda mínima (H1) não é verificada, uma vez que a escala ainda está em construção. Em segundo lugar, na segunda parte do algoritmo (a partir da linha 16), as alocações em dias não úteis, se houver, são feitas respeitando-se a restrição H4, ou seja, ao ser alocado em um turno diurno (Manhã ou Tarde), o médico é automaticamente alocado no outro turno diurno, na mesma área. Por último, frisa-se que os laços condicionais das linhas 4 e 19 possuem um limite máximo de iterações a serem executadas a cada vez, a fim de ser identificada a impossibilidade ou grande dificuldade de geração de uma solução factível a partir de um determinado momento. Nesse caso, o processo atual inteiro é abandonado e reinicia-se tudo. Também se limitou o número de vezes que o procedimento pode ser reiniciado.

Esse algoritmo possui complexidade assintótica de tempo polinomial, proporcional ao número de dias da escala, ao número de áreas de trabalho e aos limitantes artificiais dos laços, citados acima.

4.4 Heurística de Busca

O método de busca proposto aqui é basicamente a meta-heurística LAHC descrita na Seção 2.3, onde a solução candidata de cada iteração é escolhida aleatoriamente dentre os múltiplos vizinhos da solução atual, definidos a partir de uma estrutura de vizinhança, descrita a seguir. A complexidade do método está diretamente associada à complexidade de geração de um vizinho.

4.4.1 Estrutura de Vizinhança

Levando-se em conta a ideia da heurística construtiva sempre visar a confecção de uma solução factível, a estrutura de vizinhança usada foi definida com o objetivo primordial de conservar a factibilidade da solução, ou seja, alterar levemente a solução de maneira a manter altas as chances da nova solução ser factível. Para isso, foram sendo observadas, progressivamente, as soluções geradas pela heurística construtiva aleatória e também pela versão corrente da heurística de busca. Por exemplo, pôde ser facilmente percebido que as soluções geradas sempre apresentavam um problema com relação à demanda mínima exigida: esta ser maior do que a soma total das cargas horárias dos médicos (violação da restrição de carga horária máxima (S6) para todos os médicos). Desse modo, não se mostraram eficazes estruturas de vizinhança que certamente diminuem o número de alocações de alguma combinação de dias, turnos e áreas, pois esse número tende a já estar em seu limite inferior (por exemplo, dar uma folga a um médico ou trocar sua área de trabalho em determinado turno).

Assim sendo, a estrutura escolhida para este trabalho consiste em trocar as alocações de dois médicos entre si, dentro de um determinado bloco de dias consecutivos do período. Isto é, a formação de um membro da vizinhança inclui escolher aleatoriamente um par de médicos, um tamanho para o bloco e o dia de início desse bloco, e, então, transferir todas as alocações, dentro do bloco, de um médico para o outro (e vice-versa). Em especial, a escolha do tamanho do bloco é um componente ajustável do algoritmo, de modo a permitir que os tamanhos de bloco que se mostrem mais eficazes tenham mais chances de ser escolhidos (ver Seção 5.5). Também vale salientar que a escolha do dia de início do bloco é feita tomando-se cuidado para que o bloco inteiro esteja contido dentro do período de escalonamento. As figuras 4.2, 4.3 e 4.4 mostram possíveis aplicações dessa estrutura para gerar novas soluções s' a partir da solução atual s. As regiões da solução afetadas pelos movimentos encontram-se hachuradas.

Também foram testadas outras estruturas de vizinhança que, assim como a escolhida, respeitam a restrição da demanda, como, por exemplo, fazer trocas de alocações envolvendo mais de dois médicos. No entanto, nenhuma obteve resultados satisfatórios em comparação com a estrutura escolhida, geralmente por motivos de aumento na complexidade total do método.

Figura 4.2: Exemplo de geração de um vizinho, escolhendo-se os médicos m_2 e m_3, tamanho de bloco 1, e d_3 como início do bloco.

$s =$

	d_1			d_2			d_3			d_4			d_5			d_6			d_7		
	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N
m_1	a_1			a_1						a_1				a_1		a_1	a_1		a_1	a_1	
m_2		a_2				a_2			a_2			a_2						a_2			a_2
m_3					a_2		a_1				a_1			a_2		a_1	a_1		a_1	a_1	

$s' =$

	d_1			d_2			d_3			d_4			d_5			d_6			d_7		
	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N
m_1	a_1			a_1						a_1				a_1		a_1	a_1		a_1	a_1	
m_2		a_2				a_2	a_1					a_2						a_2			a_2
m_3					a_2				a_2		a_1			a_2		a_1	a_1		a_1	a_1	

Fonte: Do Autor

Figura 4.3: Exemplo de geração de um vizinho, escolhendo-se os médicos m_1 e m_2, tamanho de bloco 3, e d_4 como início do bloco.

$s =$

	d_1			d_2			d_3			d_4			d_5			d_6			d_7		
	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N
m_1	a_1			a_1						a_1				a_1		a_1	a_1		a_1	a_1	
m_2		a_2				a_2			a_2			a_2						a_2			a_2
m_3					a_2		a_1				a_1			a_2		a_1	a_1		a_1	a_1	

$s' =$

	d_1			d_2			d_3			d_4			d_5			d_6			d_7		
	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N
m_1	a_1			a_1								a_2						a_2	a_1	a_1	
m_2		a_2				a_2			a_2	a_1				a_1		a_1	a_1				a_2
m_3					a_2		a_1				a_1			a_2		a_1	a_1		a_1	a_1	

Fonte: Do Autor

Figura 4.4: Exemplo de geração de um vizinho, escolhendo-se os médicos m_1 e m_3, tamanho de bloco 7, e d_1 como início do bloco.

$s =$

	d_1			d_2			d_3			d_4			d_5			d_6			d_7		
	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N
m_1	a_1			a_1						a_1				a_1		a_1	a_1		a_1	a_1	
m_2		a_2				a_2			a_2			a_2						a_2			a_2
m_3					a_2		a_1				a_1			a_2		a_1	a_1		a_1	a_1	

$s' =$

	d_1			d_2			d_3			d_4			d_5			d_6			d_7		
	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N	M	T	N
m_1					a_2		a_1				a_1			a_2		a_1	a_1		a_1	a_1	
m_2		a_2				a_2			a_2			a_2						a_2			a_2
m_3	a_1			a_1						a_1				a_1		a_1	a_1		a_1	a_1	

Fonte: Do Autor

5 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo constam os experimentos computacionais realizados. Primeiramente, na Seção 5.1 são descritos brevemente os objetivos da experimentação. Na Seção 5.2 são apresentadas as instâncias usadas para teste. As configurações dos ambientes de execução são detalhadas na Seção 5.3. Na Seção 5.4 são definidas as configurações necessárias para a realização dos experimentos. Já na Seção 5.5, é descrita a forma como os parâmetros do método foram ajustados, e é apresentada a configuração escolhida. Finalmente, na Seção 5.6 são apresentados os resultados obtidos, com uma breve análise.

5.1 Objetivos

O objetivo principal dos testes apresentados no final deste capítulo é dar uma noção do desempenho do método proposto neste trabalho para resolução do PAM definido no Capítulo 3, mediante a comparação com o método proposto por Wickert, Neto and Buriol (2018), baseado na heurística *Fix-and-optimize*, e com um dos *solvers* MIP (*Mixed-Integer Programming*) mais reconhecidos do mercado, o do CPLEX. Os resultados usados para essa comparação foram retirados diretamente do artigo citado.

Devido ao custo financeiro de ambos os métodos usados na comparação citada, também se viu interessante comparar o método implementado com *solvers* MIP de código aberto, que poderiam ser adotados pelo hospital, assim como o método proposto neste trabalho. Para a escolha do *solver* a ser usado na comparação, foi levado em conta o trabalho de Meindl and Templ (2013), que compara o desempenho de diversos *solvers* aplicados a problemas de programação linear. Dos *solvers* cobertos por esse trabalho, foram considerados somente aqueles que permitem uso comercial irrestrito de suas funcionalidades, e, dentre esses, o CBC (*COIN-OR Branch and Cut*) foi escolhido para representar os demais *solvers* gratuitos, por ter apresentado os melhores resultados nos testes realizados.

5.2 Instâncias

Conforme explicado na Seção 2.2, para avaliar o algoritmo proposto foram utilizadas as mesmas instâncias usadas no trabalho de Wickert, Neto and Buriol (2018), disponibilizadas *online*[1], as quais foram geradas de maneira sistemática, buscando a máxima semelhança com a realidade do HCPA. Os resultados também foram organizados da mesma forma que no artigo, com as instâncias divididas por tamanho em três grupos de dez, de acordo com o número de médicos a serem escalonados. Foram consideradas dez instâncias com 50 médicos, dez com 100 médicos e dez com 150 médicos. Essas instâncias são chamadas, para simplificação, de instâncias pequenas, médias e grandes, respectivamente. A Tabela 5.1 mostra informações sobre cada um desses grupos, através de valores médios de algumas características das instâncias.

Tabela 5.1: Detalhes de cada um dos grupos de instâncias considerados.

Grupo	Número de médicos	Número de dias	Número de áreas	Ausências (em dias)	Pedidos de folga (em turnos)	Demanda mínima em dias úteis	Demanda mínima em dias não úteis
Pequenas	50	28	3	140	307	5	4
Médias	100	28	3	280	568	9	7
Grandes	150	28	3	420	850	14	12

Semelhante ao que foi feito na INRC, no nome dado a cada instância são indicados, na ordem, o número de médicos do grupo a que pertence a instância, e sua posição nesse grupo. Por exemplo, a instância $p100\_inst\_01$ é a primeira do grupo de instâncias com 100 médicos.

5.3 Ambientes de Execução

Os experimentos relativos ao *solver* CBC foram executados em uma máquina com um processador *Intel Core i7-4790 3.60 GHz*, com 8 GB de memória RAM e um sistema operacional *Windows 7 Home Premium* de 64 *bits*. Foi usada a versão 2.8.8 do *solver*, em sua configuração padrão.

Para a execução do método proposto neste trabalho, foi utilizada uma máquina virtual *Linux Ubuntu 14.04 LTS* de 64 *bits*, usando até 4 GB de memória RAM, rodando em cima da configuração descrita anteriormente. Para a implementação do algoritmo foi usada a linguagem de programação *Java*, versão 7, e o ambiente de desenvolvimento *Eclipse IDE for Java Developers*, versão 4.7.3 (*Oxygen.3*).

[1] <http://www.inf.ufrgs.br/~tiwickert/download/2017/physician/>

5.4 Configuração dos Experimentos

Com o intuito de estudar o desempenho do método implementado, foram usadas as instâncias e demais considerações da Seção 5.2. Isto é, todos os métodos a serem comparados foram aplicados às mesmas trinta instâncias, sendo dez com 50 médicos, dez com 100 médicos e dez com 150 médicos a serem escalonados.

Essa diferença de tamanho entre as instâncias é levada em conta tanto para a apresentação dos resultados e comparações, como para a definição do tempo máximo de execução dos métodos. Assim sendo, a Tabela 5.2 mostra a configuração de limites de tempo usada nos testes, considerando os diferentes métodos e grupos de instâncias.

Tabela 5.2: Limites de tempo de execução de cada método (em segundos), considerando o número de médicos nas instâncias.

Instâncias		Tempo Limite (s)			
Tamanho	Número de médicos	CPLEX	CBC	Heurística F&O	Heurística LAHC
Pequenas	50	1.200	1.200	600	600
Médias	100	2.400	2.400	1.200	1.200
Grandes	150	3.600	3.600	1.800	1.800

Cada linha da tabela está relacionada a um tamanho de instância. Os limites relativos ao *solver* CPLEX e à heurística *Fix-and-optimize* (F&O) foram definidos por Wickert, Neto and Buriol (2018), e constam na terceira e quinta colunas da tabela. Para o CPLEX, foram definidos os limites de 20, 40 e 60 minutos para as instâncias com 50, 100 e 150 médicos, respectivamente. Essa configuração foi adotada para o *solver* CBC, a fim de permitir uma comparação mais justa entre os dois *solvers*. Já para a heurística proposta (F&O), foi definido que o método executasse por, no máximo, 10, 20 e 30 minutos, para as instâncias pequenas, médias e grandes. Essa mesma configuração foi usada para a execução do método proposto neste trabalho, a fim de possibilitar a comparação entre ambas as abordagens heurísticas.

5.5 Ajuste de Parâmetros

Com relação ao método de busca proposto, tomaram-se como parâmetros ajustáveis o tamanho da lista circular, base do LAHC, e a probabilidade de cada um dos tamanhos possíveis de bloco ser escolhido ao gerar-se um vizinho da solução atual, em cada iteração da heurística (considerando a estrutura de vizinhança definida na Seção 4.4.1).

O tamanho da lista a ser usado nos experimentos realizados foi escolhido de forma experimental, observando-se o custo das soluções geradas pelo método dentro de um limite de tempo. Durante esses experimentos, foi detectado um comportamento padrão para convergência do algoritmo de busca, o qual já era esperado. No geral, o tempo consumido para se aproximar de um ótimo local depende diretamente do tamanho da lista do LAHC, além, é claro, do tamanho da instância. Quanto maior a lista, mais tempo a heurística leva para convergir, porém as soluções encontradas após a convergência tendem a ser melhores. Por conta disso, o valor desse parâmetro foi determinado levando-se em conta o tamanho das instâncias e o tempo limite de execução considerado.

Após os testes, foram escolhidos os valores 200, 150 e 100 para as instâncias pequenas, médias e grandes, respectivamente. Os valores baixos com relação aos usados por Burke and Bykov (2012) podem ser explicados pelo tempo limite de execução apertado, pelo baixo número de iterações executadas e pela alta complexidade do problema, em especial das instâncias maiores, o que dificulta a convergência do método quando tamanhos grandes de lista são usados.

A definição das probabilidades também foi feita de maneira experimental, porém observando-se as soluções geradas a fim de se visualizar as alterações que poderiam melhorar sua qualidade. Após os testes, foram escolhidos os valores da Tabela 5.3. Os tamanhos possíveis de bloco que não aparecem na tabela nunca são escolhidos.

Tabela 5.3: Frequência do uso de cada tamanho de bloco no processo de geração de vizinhos.

Tamanho de bloco	Probabilidade
1	26,67%
2	6,67%
3	6,67%
4	6,67%
5	6,67%
6	6,67%
7	13,33%
toda a escala	26,67%

A escolha destes tamanhos especificamente pode ser explicada pela forma como as escalas costumam ser produzidas pelo método, gerando sequências de atribuições em dias consecutivos para cada médico, as quais geralmente não ultrapassam uma semana. Também se encontrou eficácia na troca das atribuições em toda a escala, entre os dois médicos escolhidos.

5.6 Resultados

Nesta seção são apresentados os resultados da execução dos diversos métodos testados. Primeiramente, serão mostrados com detalhes os resultados do método proposto neste trabalho, baseado na meta-heurística LAHC, quando aplicado ao Problema de Alocação de Médicos do HCPA. Na sequência, será apresentada uma comparação geral entre todos os métodos considerados para resolver o problema. Todos os custos apresentados foram calculados a partir da função objetivo presente na formulação matemática apresentada na Seção 3.3. Para todas as instâncias, foram usados os pesos da Tabela 5.4 para violações de restrições fracas, sem qualquer distinção para os diferentes médicos. Além disso, para permitir a medição da qualidade de soluções infactíveis, e, portanto, a distinção entre essas soluções, quanto a sua distância da factibilidade, considerou-se o custo de cada violação de restrição forte com valor 10.000.

Tabela 5.4: Pesos para violações de cada um dos tipos de restrições fracas.

Variáveis	Identificador	Restrição	Peso
ω_n^1	S1	Noites consecutivas	15
ω_n^2	S2	Dias consecutivos	30
ω_n^3	S3	Pedidos de folga	10
ω_n^4	S4	Fim de semana completo	30
ω_n^5	S5	Carga horária mínima	20
ω_n^6	S6	Carga horária máxima	20
ω_n^7	S7	Máximo de fins de semana	30

Na Tabela 5.5 são apresentados os resultados do método proposto. Na primeira coluna consta o nome de cada instância, conforme a Seção 5.2. Na segunda coluna consta o valor da melhor solução conhecida (*Best Known Solution*, ou *BKS*) para aquela instância, obtida por Wickert, Neto and Buriol (2018), a partir de uma única execução do *solver* CPLEX, durante (no máximo) 12 horas. Segundo os autores, para as instâncias pequenas, a solução encontrada é a ótima (indicada por asterisco). A terceira e quarta colunas referem-se à solução obtida através da heurística construtiva proposta, a qual serve de solução inicial para a heurística de busca. Essas colunas mostram, respectivamente, a média do valor de solução encontrado e de seu desvio (*gap*) em relação ao *BKS*, após dez execuções do método para cada uma das instâncias da primeira coluna. O *gap* é calculado através da fórmula $gap = 100 \times (sol - BKS)/sol$, onde sol é o custo da solução a ser comparada com a melhor solução conhecida (*BKS*).

As três colunas seguintes mostram o desempenho da aplicação de uma simples busca local à solução retornada pelo método construtivo, também considerando sempre a média de dez execuções. A primeira mostra o custo da solução obtida. A segunda apresenta o desvio desse custo em relação à *BKS*. Já a terceira informa o tempo de execução da busca local, em segundos. Todos esses resultados são interessantes para verificar a relevância de um método de busca mais elaborado.

Nas últimas quatro colunas da Tabela 5.5, são apresentados os resultados do método de busca proposto neste trabalho, baseado na meta-heurística LAHC, após dez execuções do mesmo para cada uma das trinta instâncias. A primeira coluna mostra a média e o desvio padrão entre os dez custos de solução obtidos. A segunda mostra o desvio percentual (*gap*) da média em relação à *BKS*. A terceira coluna mostra o número médio de iterações executadas pelo método. Já a última coluna apresenta o tempo em segundos de cada execução do método na instância, tal como definido na Seção 5.4. Deve-se notar que, para cada grupo de instâncias, há uma linha especial apresentando a média entre os valores de cada coluna para as dez instâncias do grupo.

Tabela 5.5: Desempenho do método proposto para o Problema de Alocação de Médicos.

		Solução inicial		Solução inicial com busca local			Solução final			
Instância	*BKS*	Custo	*Gap (%)*	Custo	*Gap (%)*	Tempo (s)	Custo	*Gap (%)*	Número de iterações	Tempo (s)
p050_inst_01	30.305*	37.063	18,23	31.981	5,24	11	30.329 ± 29,39	0,08	1.795.015	600
p050_inst_02	30.460*	37.316	18,37	32.253	5,56	12	30.480 ± 24,90	0,07	1.936.971	600
p050_inst_03	30.505*	37.265	18,14	32.180	5,21	9	30.517 ± 24,00	0,04	2.130.980	600
p050_inst_04	30.965*	37.894	18,29	32.746	5,44	9	30.977 ± 14,70	0,04	2.035.470	600
p050_inst_05	30.685*	37.406	17,97	32.105	4,42	9	30.695 ± 20,49	0,03	1.850.144	600
p050_inst_06	31.705*	38.448	17,54	33.496	5,36	10	31.737 ± 21,82	0,10	1.945.784	600
p050_inst_07	30.015*	36.931	18,73	31.630	5,11	11	30.051 ± 29,39	0,12	2.032.394	600
p050_inst_08	30.215*	37.330	19,06	34.169	11,57	10	30.236 ± 23,43	0,07	2.001.998	600
p050_inst_09	31.670*	38.745	18,26	33.490	5,43	10	31.714 ± 51,96	0,14	1.770.605	600
p050_inst_10	30.765*	37.397	17,73	32.595	5,61	10	30.783 ± 27,50	0,06	1.934.284	600
média	30.729	37.580	18,23	32.665	5,89	10	30.752 ± 26,76	0,07	1.943.365	600
p100_inst_01	25.525	44.673	42,86	30.033	15,01	38	25.979 ± 57,88	1,75	3.302.840	1.200
p100_inst_02	27.945	47.138	40,72	32.261	13,38	29	28.479 ± 60,28	1,88	3.124.371	1.200
p100_inst_03	26.300	45.022	41,58	30.956	15,04	27	26.659 ± 44,88	1,35	3.243.465	1.200
p100_inst_04	25.285	44.814	43,58	30.000	15,72	27	25.764 ± 72,71	1,86	3.026.755	1.200
p100_inst_05	25.775	44.880	42,57	30.255	14,81	32	26.144 ± 50,34	1,41	2.896.695	1.200
p100_inst_06	26.920	45.896	41,35	32.101	16,14	26	27.391 ± 108,92	1,72	2.912.637	1.200
p100_inst_07	24.505	43.608	43,81	28.886	15,17	33	25.008 ± 65,20	2,01	3.057.915	1.200
p100_inst_08	26.445	45.484	41,86	31.261	15,41	32	26.867 ± 75,14	1,57	3.151.880	1.200
p100_inst_09	27.130	46.691	41,89	31.260	13,21	39	27.592 ± 59,08	1,67	3.156.441	1.200
p100_inst_10	25.030	44.334	43,54	29.503	15,16	31	25.534 ± 52,53	1,97	3.013.541	1.200
média	26.086	45.254	42,38	30.652	14,90	31	26.542 ± 64,70	1,72	3.088.654	1.200
p150_inst_01	60.030	86.903	30,92	64.032	6,25	104	60.207 ± 87,67	0,29	3.404.783	1.800
p150_inst_02	58.320	84.784	31,21	64.493	9,85	50	58.691 ± 137,67	0,63	3.638.151	1.800
p150_inst_03	58.070	84.794	31,52	63.396	8,40	78	58.539 ± 91,16	0,80	3.203.178	1.800
p150_inst_04	57.595	84.321	31,70	61.300	6,04	86	57.842 ± 92,82	0,43	3.628.287	1.800
p150_inst_05	59.740	85.799	30,37	66.247	9,82	62	59.375 ± 121,59	-0,61	3.358.024	1.800
p150_inst_06	58.720	84.804	30,76	63.034	6,84	87	58.973 ± 88.24	0,43	3.305.515	1.800
p150_inst_07	57.570	84.990	32,26	62.215	7,47	82	58.119 ± 134,54	0,94	3.492.367	1.800
p150_inst_08	56.750	84.227	32,62	61.502	7,73	74	57.378 ± 84,39	1,09	3.123.493	1.800
p150_inst_09	57.580	84.023	31,47	62.278	7,54	69	57.673 ± 96,21	0,16	3.256.407	1.800
p150_inst_10	55.810	83.052	32,80	60.052	7,06	90	55.990 ± 79,83	0,32	3.551.616	1.800
média	58.019	84.770	31,56	62.875	7,70	78	58.279 ± 101,41	0,45	3.396.182	1.800

O tempo gasto pelo método construtivo não ultrapassou 1 segundo para nenhuma das execuções, e sempre se obteve uma solução factível ao fim de sua execução. Sendo assim, o método de busca apenas ficou encarregado de refinar essa solução.

Já na Tabela 5.6 é apresentada uma comparação entre métodos para resolução do problema. Foram considerados nessa comparação os *solvers* MIP CPLEX e CBC (projeto do COIN-OR), o método baseado na heurística *Fix-and-optimize*, proposto por Wickert, Neto and Buriol (2018), e o método proposto neste trabalho, baseado na meta-heurística *Late Acceptance Hill Climbing*.

Esta tabela segue o formato da Tabela 5.5. Para cada método, são apresentados, para cada uma das instâncias, os valores médios obtidos de custo de solução e de desvio desse custo em relação à *BKS*, entre dez execuções do método na instância. Os melhores valores encontrados para cada instância constam em negrito. Em algumas das execuções dos *solvers* exatos CPLEX e CBC para as instâncias pequenas, a solução ótima foi encontrada e sua otimalidade foi comprovada antes da expiração do tempo máximo de execução. Por conta disso, foi adicionada na tabela uma coluna extra para cada *solver*, mostrando o tempo médio consumido para busca da solução ótima e comprovação de sua otimalidade. Para os métodos heurísticos, o tempo de execução foi sempre o limite estipulado na Seção 5.4.

Tabela 5.6: Comparação entre o desempenho de diversos métodos para solução do PAM.

		CPLEX			COIN-OR (CBC)			F&O		LAHC		F&O e LAHC
Instância	*BKS*	Custo	*Gap (%)*	Tempo (s)	Custo	*Gap (%)*	Tempo (s)	Custo	*Gap (%)*	Custo	*Gap (%)*	Tempo (s)
p050_inst_01	30.305*	**30.305**	0,00	438	30.383	0,26	1.200	30.365	0,20	30.329	0,08	600
p050_inst_02	30.460*	**30.460**	0,00	352	30.665	0,67	1.200	30.500	0,13	30.480	0,07	600
p050_inst_03	30.505*	**30.505**	0,00	428	**30.505**	0,00	714	30.575	0,23	30.517	0,04	600
p050_inst_04	30.965*	**30.965**	0,00	511	**30.965**	0,00	595	30.985	0,06	30.977	0,04	600
p050_inst_05	30.685*	**30.685**	0,00	374	**30.685**	0,00	736	30.695	0,03	30.695	0,03	600
p050_inst_06	31.705*	**31.705**	0,00	415	31.712	0,02	974	31.755	0,16	31.737	0,10	600
p050_inst_07	30.015*	**30.015**	0,00	401	30.077	0,21	954	30.025	0,03	30.051	0,12	600
p050_inst_08	30.215*	**30.215**	0,00	403	30.285	0,23	911	30.275	0,20	30.236	0,07	600
p050_inst_09	31.670*	**31.670**	0,00	447	31.678	0,03	1.091	**31.670**	0,00	31.714	0,14	600
p050_inst_10	30.765*	**30.765**	0,00	408	**30.765**	0,00	769	30.805	0,13	30.783	0,06	600
média	30.729	**30.729**	0,00	418	30.772	0,14	914	30.765	0,12	30.752	0,07	600
p100_inst_01	25.525	26.320	3,02	2.400	852.825	97,01	2.400	**25.845**	1,24	25.979	1,75	1.200
p100_inst_02	27.945	29.940	6,66	2.400	994.515	97,19	2.400	**28.435**	1,72	28.479	1,88	1.200
p100_inst_03	26.300	175.395	85,01	2.400	1.002.990	97,38	2.400	**26.650**	1,31	26.659	1,35	1.200
p100_inst_04	25.285	66.040	61,71	2.400	784.005	96,77	2.400	**25.515**	0,90	25.764	1,86	1.200
p100_inst_05	25.775	28.615	9,92	2.400	1.051.805	97,55	2.400	**26.060**	1,09	26.144	1,41	1.200
p100_inst_06	26.920	28.960	7,04	2.400	794.365	96,61	2.400	**27.130**	0,77	27.391	1,72	1.200
p100_inst_07	24.505	26.075	6,02	2.400	692.650	96,46	2.400	**24.870**	1,47	25.008	2,01	1.200
p100_inst_08	26.445	97.175	72,79	2.400	963.420	97,26	2.400	**26.770**	1,21	26.867	1,57	1.200
p100_inst_09	27.130	48.620	44,20	2.400	904.140	97,00	2.400	**27.560**	1,56	27.592	1,67	1.200
p100_inst_10	25.030	28.185	11,19	2.400	1.002.585	97,50	2.400	**25.430**	1,57	25.534	1,97	1.200
média	26.086	55.533	30,76	2.400	904.330	97,07	2.400	**26.427**	1,29	26.542	1,72	1.200
p150_inst_01	60.030	33.866.360	99,82	3.600	1.707.970	96,49	3.600	61.425	2,27	**60.207**	0,29	1.800
p150_inst_02	58.320	28.101.150	99,79	3.600	1.745.485	96,66	3.600	60.110	2,98	**58.691**	0,63	1.800
p150_inst_03	58.070	5.985.325	99,03	3.600	1.646.850	96,47	3.600	59.625	2,61	**58.539**	0,80	1.800
p150_inst_04	57.595	28.062.400	99,79	3.600	1.556.320	96,30	3.600	59.090	2,53	**57.842**	0,43	1.800
p150_inst_05	59.740	33.867.280	99,82	3.600	1.598.000	96,26	3.600	90.540	34,02	**59.375**	-0,61	1.800
p150_inst_06	58.720	5.563.180	98,94	3.600	1.814.940	96,76	3.600	69.920	16,02	**58.973**	0,43	1.800
p150_inst_07	57.570	12.057.155	99,52	3.600	2.550.140	97,74	3.600	59.255	2,84	**58.119**	0,94	1.800
p150_inst_08	56.750	7.150.135	99,21	3.600	1.586.145	96,42	3.600	59.120	4,01	**57.378**	1,09	1.800
p150_inst_09	57.580	33.868.180	99,83	3.600	1.616.715	96,44	3.600	**57.085**	-0,87	57.673	0,16	1.800
p150_inst_10	55.810	7.480.425	99,25	3.600	1.604.405	96,52	3.600	57.148	2,34	**55.990**	0,32	1.800
média	58.019	19.600.158	99,50	3.600	1.742.697	96,61	3.600	63.332	6,87	**58.279**	0,45	1.800

Observando-se a Tabela 5.6, nota-se a dificuldade dos *solvers* exatos em encontrar soluções factíveis para as instâncias médias e, em especial, para as grandes. Mesmo o CPLEX tendo encontrado soluções factíveis para algumas das instâncias médias, ambos os *solvers* obtiveram desempenho inferior ao dos métodos heurísticos tanto para as instâncias médias, como para as grandes.

Comparando-se os dois *solvers* entre si, nas instâncias médias apenas o CPLEX conseguiu obter soluções factíveis (subótimas), e nas grandes nenhum dos dois conseguiu. Porém, vale ressaltar que, para as instâncias grandes, onde o limite de tempo é mais apertado, o CBC encontrou soluções muito mais próximas da factibilidade do que o CPLEX. Quanto ao grupo de instâncias pequenas, nota-se que o CPLEX conseguiu encontrar soluções ótimas para essas instâncias, tendo obtido, portanto, o melhor desempenho entre todos os métodos, diferente do CBC, que teve um pouco mais de dificuldade, embora também tenha tido um bom desempenho geral. Nas instâncias em que ambos conseguiram alcançar a otimalidade em todas as dez execuções (*gap* médio igual a zero), o CPLEX executou, na média, durante menos tempo.

Comparando-se os dois métodos heurísticos entre si, ambos obtiveram resultados semelhantes para as instâncias pequenas e médias. Especificamente, nas instâncias pequenas o LAHC obteve, na média geral, valores melhores de solução, enquanto nas médias o F&O encontrou soluções levemente melhores para todas as instâncias. Já nas grandes, o método baseado em LAHC obteve soluções significativamente melhores para todas as instâncias, exceto uma, para a qual o F&O encontrou soluções melhores que a *BKS*, na média das dez execuções. Vale ressaltar que, para outra instância, o LAHC também encontrou soluções melhores que a solução obtida pelo CPLEX executando por 12 horas (tida como *BKS*), resultando, portanto, em um desvio percentual negativo.

Também se viu interessante realizar uma análise das soluções geradas pelo método proposto, através do número médio de violações de cada uma das restrições fracas, considerando dez execuções do método, a fim de se identificar os pontos fortes e fracos do algoritmo. Esses números constam na Tabela 5.7.

Tabela 5.7: Número médio de violações de cada uma das restrições fracas.

Instância	S1	S2	S3	S4	S5	S6	S7
p050_inst_01	143	287	8	119	0	714	55
p050_inst_02	146	312	1	120	0	681	57
p050_inst_03	141	292	1	116	0	729	54
p050_inst_04	143	301	0	114	0	739	53
p050_inst_05	141	303	2	115	0	724	53
p050_inst_06	147	315	2	124	0	730	59
p050_inst_07	141	282	2	116	0	719	54
p050_inst_08	145	282	8	118	0	716	55
p050_inst_09	151	319	4	120	0	726	57
p050_inst_10	148	297	5	122	0	710	58
média	145	299	3	118	0	719	56
p100_inst_01	3	113	8	40	0	1047	11
p100_inst_02	6	123	7	45	0	1142	13
p100_inst_03	3	90	1	49	0	1112	11
p100_inst_04	2	89	3	37	0	1077	7
p100_inst_05	2	121	4	42	0	1052	7
p100_inst_06	2	130	6	48	0	1082	12
p100_inst_07	4	72	3	44	0	1058	13
p100_inst_08	0	97	4	44	0	1117	12
p100_inst_09	5	124	3	48	0	1098	12
p100_inst_10	1	71	6	48	0	1075	14
média	3	103	5	45	0	1086	11
p150_inst_01	123	420	7	154	0	1949	66
p150_inst_02	121	410	9	152	0	1897	64
p150_inst_03	120	408	7	150	0	1897	63
p150_inst_04	120	397	3	156	0	1867	66
p150_inst_05	122	422	5	158	0	1898	68
p150_inst_06	120	417	10	160	0	1872	68
p150_inst_07	121	374	6	158	0	1908	68
p150_inst_08	121	386	15	156	0	1857	66
p150_inst_09	120	409	4	154	0	1852	66
p150_inst_10	120	352	6	156	0	1842	66
média	121	400	7	155	0	1884	66

É possível perceber que o custo das soluções deve-se especialmente a violações da restrição S6, acerca da carga horária máxima de cada médico, por conta da alta demanda de médicos. Já a restrição S5, relativa à carga horária mínima, nunca é violada. Das demais restrições, mostrou-se mais custosa a S2, relativa ao número maximo de dias consecutivos de trabalho para cada médico. As restrições fortes nunca são violadas.

Avaliando-se o método proposto neste trabalho quanto a sua robustez, vale destacar que os valores de desvio padrão encontrados foram, em média, inferiores aos obtidos pelo método proposto por Wickert, Neto and Buriol (2018), para as instâncias médias e grandes, e superiores, para as instâncias pequenas.

6 CONSIDERAÇÕES FINAIS

Neste trabalho foram propostas a revisão do Problema de Alocação de Médicos do HCPA definido por Wickert, Neto and Buriol (2018), e a confecção de um método heurístico para resolver esse problema, podendo este ser aproveitado pelo hospital. O método proposto é baseado na meta-heurística *Late Acceptance Hill Climbing* e faz uso de um método construtivo e uma estrutura de vizinhança auxiliares, ambos propostos neste trabalho. Na Seção 6.1 são apresentadas conclusões referentes aos resultados exibidos na Seção 5.6. Na Seção 6.2 são apresentadas possibilidades de trabalhos futuros que seguem aquilo que foi desenvolvido neste trabalho.

6.1 Conclusões

Após a avaliação do método proposto sobre um conjunto de instâncias fictícias do problema, observou-se um bom desempenho, levando em conta a simplicidade do algoritmo implementado. Também se nota sua efetividade quando comparado a uma simples busca local. Em média, o método obteve os melhores resultados para as instâncias grandes, dentre todos os métodos testados, e, para os demais tamanhos de instância, ficou atrás apenas de um método (CPLEX, no caso das instâncias pequenas, e *Fix-and-optimize*, no caso das médias).

Tanto o *solver* comercial CPLEX, quanto o método baseado em *Fix-and-optimize*, que faz uso desse *solver*, não são interessantes para uso direto no hospital, por envolverem custos adicionais à instituição. Por conta disso, também foi testado um reconhecido *solver* gratuito, o COIN-OR, a fim de se realizar uma comparação deste com o método proposto, uma vez que ambos poderiam ser adotados pelo hospital. Analisando os resultados experimentais obtidos, é possível visualizar que o método implementado conseguiu retornar soluções viáveis para o hospital para todos os tamanhos de instância, enquanto o *solver* conseguiu apenas para as instâncias pequenas.

Para algumas instâncias pequenas, o método exato encontrou uma solução ótima para o problema e conseguiu comprovar sua otimalidade, dentro do limite de tempo estipulado. Isto pode ser considerado uma das vantagens do método exato com relação à heurística, embora possa exigir mais tempo para determinadas instâncias. Por outro lado, o método heurístico proposto, mesmo executando por menos tempo, encontrou, na média, soluções melhores que o *solver* para as instâncias pequenas. Além disso, exige menos recursos computacionais e pode ser facilmente modificado.

Portanto, conclui-se que, para as instâncias menores, que refletem melhor as condições reais do hospital no momento, ambos os métodos podem ser interessantes para a instituição, e uma possível escolha entre eles dependeria de interesses específicos do hospital, incluindo o tempo máximo destinado a execução do método e o desejo por encontrar, sempre que possível, a solução ótima do problema (comprovadamente).

Caso o problema do hospital seja expandido, ou se houver uma demanda de outra instituição com um problema similar, podendo, em ambos os casos, haver mais médicos para serem considerados no escalonamento, a análise dos resultados feita mostra que o método heurístico tende a possuir um desempenho superior, em comparação com os métodos exatos.

6.2 Trabalhos Futuros

Sem dúvida, a continuidade deste trabalho envolve primordialmente a transição da definição do problema do hospital usada aqui para sua versão atualizada, a fim de se aproximar da realidade da instituição. Essa transição consiste basicamente em:

- Tratar a questão do compartilhamento de médicos entre áreas, isto é, médicos que, na prática, podem trabalhar em mais de uma área no mesmo turno;
- Alterar o tipo de algumas restrições (geralmente passando de fracas para fortes);
- Retirar a restrição que impede que um médico trabalhe em mais de um turno em dias úteis (H3), e adicionar uma restrição semelhante à H5, contendo sucessões proibidas de turnos no mesmo dia (por exemplo, no mesmo dia útil, um médico deve poder trabalhar nos turnos da manhã e da noite, mas não nos turnos da tarde e da noite).
- Inserir mais restrições:
  - Devem ser respeitadas as prioridades de alocação entre os diferentes médicos para cada área e turno de trabalho.

  - Devem ser respeitados os limites mínimo e máximo de horas trabalhadas em dias não úteis para cada médico.
  - A distribuição de turnos diurnos e noturnos para cada médico em dias não úteis deve ser feita de forma equilibrada.
  - Cada médico deve trabalhar majoritariamente em uma única área de trabalho (dado um valor de porcentagem mínimo pré-definido).
  - Cada médico deve trabalhar no menor número de áreas distintas ao longo de todo o período.
  - Ao final do escalonamento, a solução deve respeitar o limite máximo geral de horas extras a serem pagas.

Após a transição, o método proposto, tendo a devida adequação, poderia inclusive ser aplicado a instâncias reais do problema, o que possibilitaria uma comparação mais fidedigna de seu desempenho com o do processo manual adotado no hospital. Além disso, outros trabalhos também poderiam seguir o que foi produzido com este, incluindo:

- Proposta de outros métodos construtivos;
- Ajuste mais confiável dos parâmetros usados no método de busca (importante especialmente para a nova versão do problema);
- Uso de outras estruturas de vizinhança;
- Comparação do desempenho do LAHC com o de outras meta-heurísticas.

REFERÊNCIAS

BILGIN, B. et al. A hyper-heuristic combined with a greedy shuffle approach to the nurse rostering competition. In: **Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT'10)**. [S.l.: s.n.], 2010.

BRUNI, R.; DETTI, P. A flexible discrete optimization approach to the physician scheduling problem. **Operations Research for Health Care**, Elsevier, v. 3, n. 4, p. 191–199, 2014.

BRUNNER, J. O.; BARD, J. F.; KOLISCH, R. Flexible shift scheduling of physicians. **Health Care Management Science**, Springer, v. 12, n. 3, p. 285–305, 2009.

BURKE, E. K.; BYKOV, Y. A late acceptance strategy in hill-climbing for exam timetabling problems. In: **PATAT 2008 Conference, Montreal, Canada**. [S.l.: s.n.], 2008.

BURKE, E. K.; BYKOV, Y. The late acceptance hill-climbing heuristic. **Department of Computing Science and Mathematics University of Stirling—Technical Report CSM-192. ISSN**, p. 1460–9673, 2012.

BURKE, E. K. et al. The state of the art of nurse rostering. **Journal of scheduling**, Kluwer Academic Publishers, v. 7, n. 6, p. 441–499, 2004.

BURKE, E. K. et al. A classification of hyper-heuristic approaches. **Handbook of Metaheuristics**, Springer, v. 146, p. 449–468, 2010.

CESCHIA, S. et al. Second international nurse rostering competition (inrc-ii) problem description and rules. 2015.

DANG, N. T. T. et al. Solving the multi-stage nurse rostering problem. In: PATAT. **Proceedings of the 11th International Conference of the Practice and Theory of Automated Timetabling**. [S.l.], 2016. p. 473–475.

HASPESLAGH, S. et al. First international nurse rostering competition 2010 (august 10-13, 2010, belfast, uk). In: **PATAT 2010-Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling, Belfast, Northern-Ireland, UK**. [S.l.: s.n.], 2010.

KHEIRI, A. et al. A sequence-based selection hyper-heuristic. In: PATAT. **Proceedings of the 11th International Conference of the Practice and Theory of Automated Timetabling**. [S.l.], 2016.

MEINDL, B.; TEMPL, M. Analysis of commercial and free and open source solvers for linear optimization problems. 2013.

NONOBE, K. Inrc2010: An approach using a general constraint optimization solver. **The First International Nurse Rostering Competition (INRC 2010)**, 2010.

NONOBE, K.; IBARAKI, T. A tabu search approach to the constraint satisfaction problem as a general problem solver. **European Journal of Operational Research**, Elsevier, v. 106, n. 2–3, p. 599–623, 1998.

ROUSSEAU, L.-M.; PESANT, G.; GENDREAU, M. A general approach to the physician rostering problem. **Department of Computing Science and Mathematics University of Stirling—Technical Report CSM-192. ISSN**, Springer, v. 115, n. 1–4, p. 193–205, 2002.

WICKERT, T. I.; NETO, A. K.; BURIOL, L. S. An integer programming approach for the physician rostering problem. 2018.