

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

LUCAS HENRIQUE RUSSO DO NASCIMENTO

**RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS EM
HOSPITAL UTILIZANDO *SIMULATED ANNEALING***

BAURU - SP
Agosto/2022

Lucas Henrique Russo do Nascimento

**RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS EM
HOSPITAL UTILIZANDO *SIMULATED ANNEALING***

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, campus Bauru.

Orientadora: Prof^a Dr^a.Andréa Carla Gonçalves Vianna

BAURU
Agosto/2022

Nascimento, Lucas Henrique Russo do.
Resolução do Problema de Escalonamento de
Enfermeiros em hospital utilizando *Simulated
Annealing* / Lucas Henrique Russo do Nascimento,
2022
39 f. : il.

Orientadora: Andréa Carla Gonçalves Vianna

Trabalho de conclusão de curso (Bacharelado -
Ciência da Computação) - Universidade Estadual
Paulista (Unesp). Faculdade de Ciências, Bauru,
2022

1. Otimização matemática. 2. Metaheuristica. 3.
Pesquisa Operacional. I. Universidade Estadual
Paulista. Faculdade de Ciências. I. Título.

Lucas Henrique Russo do Nascimento

**RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS EM
HOSPITAL UTILIZANDO *SIMULATED ANNEALING***

Trabalho de Conclusão de Curso do curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof^a. Dr^a. Andréa Carla Gonçalves Viana

Orientadora

Universidade Estadual Paulista "Júlio

de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

Prof^a. Dr^a. Simone das Graças Domingues Prado

Universidade Estadual Paulista "Júlio

de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

Prof^a. Dr^a. Adriana Cristina Cherri Nicola

Universidade Estadual Paulista "Júlio

de Mesquita Filho"

Faculdade de Ciências

Departamento de Matemática

Dedico este trabalho aos meus pais, que muito me apoiaram nessa jornada de estudo e crescimento.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Paulo e Tatiana, por me darem apoio e incentivo nos momentos difíceis para que eu não desistisse dos meus sonhos, e ao meu irmão, Renan.

Agradeço à minha orientadora, Profa. Dra. Andrea Carla Gonçalves Vianna, por toda a ajuda oferecida, por me guiar no processo de aprendizado e por toda a paciência em me passar seus conhecimentos durante o curso.

Agradeço também à Naumteria e a todos os amigos que fiz lá. Obrigado por tornarem todos os meus dias melhores. Obrigado por terem mudado minha vida inteira pra melhor.

Agradecimento especial às minhas amigas, Giovana Orru, Larissa Duarte, Tainá Bernardes, Gabriela Braga e Yumi Suzuki, e aos meus amigos Henrique Leão, João Paulo Marques, Matheus Vidal, Renan Amorim, Guilherme Ferreira e Guilherme Magalhães. Todos foram peças importantes na minha formação pessoal.

Por fim, agradeço a todos os docentes e funcionários da UNESP que, direta ou indiretamente, contribuíram com o meu desenvolvimento.

RESUMO

O Problema de Escalonamento de Enfermeiros se baseia em definir em quais dias e turnos cada enfermeiro trabalhará respeitando a demanda do hospital. Geralmente, nas instituições hospitalares, essa escala é definida manualmente por uma pessoa responsável a cada mês. Como a quantidade de combinações possíveis é muito grande, encontrar uma escala adequada pode levar muito tempo, além de demandar muita experiência da pessoa encarregada. Este trabalho propõe a resolução do problema utilizando a metaheurística *Simulated Annealing* para determinar uma solução do Problema de Escalonamento de Enfermeiros. Os resultados são apresentados de forma a avaliar a qualidade da solução obtida. Foram feitos diversos testes computacionais comprovando a viabilidade da metaheurística proposta.

Palavras-chave: Problema de Escalonamento de Enfermeiros. Metaheurísticas. *Simulated Annealing*. Otimização Matemática.

ABSTRACT

Nurse Scheduling Problem is based on defining which days and shifts each nurse will work, respecting hospital demand. Generally, in hospital institutions, this scale is manually set by a responsible person every month. As the number of possible combinations is very large, finding a suitable scale can take a long time, as well as demanding a lot of experience from the person in charge. This paper proposes solving the problem using a Simulated Annealing metaheuristic to determine the solution of the Nurses Scheduling Problem. The results are presented in order to evaluate the quality of the solution obtained. Several computational tests were performed proving the feasibility of the proposed metaheuristic.

Key-Words: Nurse Scheduling Problem. Metaheuristic. Simulated Annealing. Mathematical Optimization.

LISTA DE ABREVIATURAS E SIGLAS

PEP Problema de Escalonamento de Pessoal

PEE Problema de Escalonamento de Enfermeiros

PEM Problema de Escalonamento de Médicos

VNS *Variable Neighborhood Search*

LISTA DE ILUSTRAÇÕES

Quadro 1: Variáveis de entrada	17
Quadro 2: Variáveis de decisão	17
Fluxograma 1: Algoritmo da metaheurística <i>Simulated Annealing</i>	20
Quadro 3: Resultado da alocação com instância reduzida	24
Gráfico 1: Variação da quantidade de violações da solução final para instância reduzida ..	28
Gráfico 2: Variação da quantidade de violações da solução final para instância aumentada ..	28
Quadro 4: Demandas mínima e máxima separadas por instância	32

LISTA DE TABELAS

Tabela 1: Comparação de execução com diferentes soluções iniciais para instância reduzida	25
Tabela 2: Comparação de execução com diferentes soluções iniciais para instância aumentada	25
Tabela 3: Variação percentual no espaço de busca para instância reduzida	26
Tabela 4: Variação percentual no espaço de busca para instância aumentada	27
Tabela 5: Quantidade de violações por restrição nas soluções inicial e final, R1 e R2 com peso maior .	29
Tabela 6: Quantidade de violações por restrição nas soluções inicial e final, R3 com peso maior	30
Tabela 7: Quantidade de violações por restrição nas soluções inicial e final, R6 com peso maior	31
Tabela 8: Quantidade de violações por instância, separadas por restrição	32
Tabela 9: Quantidade de violações por instância, separadas por restrição, R1 e R2 com peso maior ..	33

SUMÁRIO

1 INTRODUÇÃO	12
2 REVISÃO BIBLIOGRÁFICA	14
2.1 Problema de Escalonamento de Pessoal (PEP)	14
2.1.1 Problema de Escalonamento de Médicos (PEM)	15
2.1.1 Problema de Escalonamento de Enfermeiros (PEE)	16
3 PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS	17
3.1 Definição do problema	17
3.2 Restrições	17
3.3 Modelagem matemática	18
4 MÉTODOS DE RESOLUÇÃO	21
4.1 Solver CPLEX	21
4.2 Metaheurística <i>Simulated Annealing</i>	21
4.2.1 Implementação computacional.....	22
5 GERADOR DE INSTÂNCIAS	24
5.1 Limite superior da demanda mínima	25
6 TESTES COMPUTACIONAIS	26
6.1 Testes no CPLEX	26
6.2 Testes com a Metaheurística	26
6.2.1 Comparativo entre solução inicial aleatória e solução inicial melhorada	26
6.2.2 Porcentagem de variação do espaço de busca	28
6.2.3 Variação da penalização	31
6.2.4 Testes gerais com instâncias genéricas variadas	33
7 CONCLUSÕES	36
REFERÊNCIAS	37

1. INTRODUÇÃO

Com o avanço da tecnologia e a necessidade de se prestar serviços de maneira mais rápida e eficiente, é essencial que a atribuição de tarefas seja feita da melhor maneira possível. Ao administrar uma indústria, por exemplo, um gestor deve fazer a melhor escala dos seus funcionários para que todas as atividades sejam concluídas e os produtos sejam entregues no prazo correto, evitando atrasos e aumentando o lucro final. Este problema é tratado na literatura como Problema de Alocação ou Problema de Escalonamento.

O Problema de Alocação pode ser aplicado nos mais diversos setores da sociedade como forma de aprimorar algum processo de uma empresa, uma fábrica, ou até mesmo um serviço público, como hospitais e escolas.

Em Silva (2008) foi utilizado um modelo heurístico para a resolução do Problema de Alocação de Navios em Berços de uma Estação Portuária. Esta resolução mostrou ser de grande utilidade uma vez que pode agilizar e aumentar a importação e exportação de produtos no mercado internacional, por exemplo.

Subramanian *et al.* (2011) apresenta um estudo sobre a atribuição de aulas a salas em uma instituição universitária. Usualmente esta atribuição é feita de maneira manual, porém este trabalho apresenta uma solução para automatizar esta tarefa, além de trazer uma solução mais eficiente, que atinja todas as restrições apresentadas e ainda com um menor custo à instituição.

No trabalho de mestrado desenvolvido por Meister (2020) é apresentado um estudo de caso no Hospital das Clínicas de Porto Alegre, tratando-se do Problema de Alocação de Enfermeiros. Este trabalho propõe um algoritmo heurístico para resolver o problema de escalonamento de médicos na instituição, comparando os resultados com uma solução exata obtida através de um *solver* gratuito.

Como é possível perceber, a abrangência do Problema de Alocação é muito grande, podendo ajudar e otimizar processos dos mais diversos setores da sociedade.

Existem diversos métodos e algoritmos para se resolver o Problema de Alocação. Soares (2011) apresenta alguns algoritmos já bem conhecidos, como é o caso do *Branch and Bound* e do *Simulated Annealing*, dois algoritmos bem consolidados para tratar deste problema. O autor também apresenta uma breve explicação sobre heurísticas e metaheurísticas. Segundo o autor, heurísticas buscam encontrar valores próximos ao ótimo em um tempo menor, mas nem sempre é o melhor resultado possível. Isso acontece nos casos em que os algoritmos exatos possuem um custo computacional elevado. Portanto, um

algoritmo heurístico pode apresentar um custo-benefício maior do que um algoritmo exato em algumas ocasiões.

Ainda no trabalho citado anteriormente, o autor faz uma comparação dos resultados obtidos por cada algoritmo e método utilizado, apresentando as situações em que cada um obteve o melhor desempenho.

O presente trabalho trata do Problema de Escalonamento de Enfermeiros. Inicialmente será feito um estudo sobre o Problema de Alocação, uma modelagem matemática ao escalonamento de enfermeiros em um hospital e a resolução exata da modelagem através de um *solver*. Em seguida, pretende-se propor uma metaheurística para resolver o problema. Ao final, será feita a análise dos resultados obtidos no que diz respeito à otimalidade da solução.

Com a situação enfrentada pela pandemia de COVID-19, é notória a importância de se fazer uma boa alocação de enfermeiros nos hospitais. A relevância deste trabalho consiste no fato de que, com ou sem a pandemia, o atendimento médico hospitalar não deveria ser prejudicado por uma atribuição ineficiente da escala dos funcionários. Por isso, este trabalho se faz necessário a fim de evitar tanto a sobrecarga da quantidade de pacientes nas instituições, quanto a sobrecarga física e mental dos profissionais da saúde.

O Capítulo 2 apresenta a revisão bibliográfica a respeito do Problema de Escalonamento de Pessoal. O Capítulo 3 descreve o problema analisado, assim como as restrições consideradas neste trabalho e a modelagem matemática proposta. O Capítulo 4 apresenta e detalha os métodos de resolução utilizados. O Capítulo 5 detalha o gerador de instâncias utilizado para facilitar e automatizar os testes. O Capítulo 6 apresenta os resultados dos testes computacionais realizados. O Capítulo 7 traz as conclusões e considerações finais a respeito do trabalho realizado.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta a literatura consultada que trata do Problema de Escalonamento de Pessoal.

2.1. Problema de Escalonamento de Pessoal (PEP)

Devesse (2016) descreve o PEP como um “processo de planejamento ou organização de horários de trabalho de funcionários, considerando regulamentos laborais, organizacionais, contratuais ou sindicais”.

Ernst *et al.* (2004) caracterizaram escalonamento de pessoal como: "o processo de construção de horários de trabalho para funcionários, a fim de satisfazer a demanda de bens ou serviços de uma organização".

Ainda no trabalho de Ernst *et al.* (2004) são apresentados módulos que podem compor parte do processo em busca de encontrar a solução do problema de alocação. Alguns deles são:

- **Modelagem da demanda:** estimar a quantidade de pessoal necessária para satisfazer o problema no período definido no problema. Este módulo deve considerar tanto a característica que define o problema (por exemplo, se a demanda é baseada em tarefas a serem realizadas ou em turnos a serem preenchidos), como possíveis incidentes que possam acontecer no decorrer do período a ser estimado
- **Definição das folgas:** determinar como as folgas serão intercaladas no período considerado. Este módulo é mais recorrente em problemas baseados na atribuição de turnos.
- **Definição dos turnos:** determinar quais turnos precisam de alocações, e estimar quantos e quais são os funcionários a serem alocados em cada turno. Deve-se levar em consideração a legislação vigente a respeito de pausas e intervalos de refeição durante o expediente de trabalho.
- **Definição das tarefas:** definir quais tarefas estarão presentes em cada turno. Cada tarefa pode exigir certas habilidades do funcionário que irá executá-la, e isso deve ser levado em consideração neste módulo.
- **Definição do pessoal:** este módulo define quais funcionários poderão ser alocados em quais tarefas de acordo com as suas habilidades.

Ainda que estes módulos pareçam um passo-a-passo, cada problema pode precisar de uma combinação diferente deles. Por exemplo, um problema que não envolve tarefas com habilidades específicas não precisará passar por esse módulo.

2.1.1. Problema de Escalonamento de Médicos (PEM)

O PEM é uma variação do PEP. Trata-se de aplicar o processo de resolução do PEP ao quadro de funcionários de um hospital, respeitando a legislação trabalhista vigente para esta categoria.

De acordo com Rousseau, Pesant e Gendreau (2002), o PEM é uma tarefa complexa de ser resolvida, já que deve-se levar em consideração muitas restrições (frequentemente conflitantes), tais como número de turnos consecutivos, a disponibilidade dos médicos, regras especiais para turnos noturnos e de fins de semana, entre outros.

Erhard *et al.* (2018) apresenta uma revisão bibliográfica com inúmeras publicações a respeito do PEM. Os autores analisam o aumento da quantidade de pesquisas feitas acerca deste tópico, e afirmam que isso ocorreu pois nos últimos anos houve uma dificuldade em contratar médicos em diversos hospitais. Eles citam a situação da Alemanha em 2013, onde aproximadamente metade dos hospitais do país tiveram dificuldades em contratar médicos, resultando em uma média de 2,5 vagas ociosas por hospital. Segundo os autores, é de extrema importância que provedores de serviços de saúde encontrem formas de reduzir custos de trabalho sem influenciar negativamente o serviço prestado aos pacientes.

Erhard *et al.* (2018) também criaram uma classificação que divide a bibliografia estudada em três categorias:

- **Problemas de força de trabalho:** aqueles que buscam quantificar o número ideal de funcionários para atingir as demandas
- **Problemas de alocação:** aqueles que buscam encontrar a melhor escala de funcionários para um período pré definido de acordo com o quadro de funcionários disponíveis
- **Problemas de realocação:** aqueles que lidam com eventos adversos não previstos em uma alocação já realizada

O trabalho de Meister (2020) propõe uma heurística *Variable Neighborhood Search* (VNS) aplicado a um estudo de caso do Hospital das Clínicas de Porto Alegre. O algoritmo proposto por este trabalho também é comparado com os resultados exatos obtidos por um

solver gratuito.

2.1.2. Problema de Escalonamento de Enfermeiros (PEE)

Analogamente ao PEM, o PEE também é uma variação do PEP. A diferença entre os dois consiste no fato de que médicos nem sempre têm vínculo empregatício com o hospital em questão, fato que geralmente ocorre com os enfermeiros. Além disso, a quantidade de pessoal a ser tratado em um PEE costuma ser significativamente maior do que em um PEM.

O trabalho de Cheang *et al.* (2003) apresenta três padrões de modelagem para o PEE: *nurse-day view*, *nurse-task view* e *nurse-shift pattern*. As três abordagens são representações bidimensionais do problema.

A abordagem *nurse-day view* relaciona o enfermeiro ao dia a ser trabalhado, podendo se transformar em uma relação tri-dimensional caso os turnos de cada dia sejam considerados na modelagem. A *nurse-task view* relaciona o enfermeiro à sua tarefa a ser realizada no período considerado. Finalmente, a *nurse-shift pattern* relaciona o enfermeiro a um padrão de turno. Esta última abordagem é mais utilizada em casos onde o enfermeiro possui um contrato temporário.

Neste trabalho ainda são apresentadas as restrições mais comuns deste tipo de problema, e são separadas em dois tipos:

- **Restrições de Factibilidade:** restrições que expressam se uma solução é factível ou não. Devem ser respeitadas obrigatoriamente (por exemplo, demanda mínima de enfermeiros em um turno específico).
- **Restrições de Qualidade:** restrições que expressam a qualidade da solução. Quando violadas, associam um custo maior à solução e alteram o resultado final da função objetivo (por exemplo, a preferência de um enfermeiro em trabalhar em algum turno específico).

O trabalho de Burke *et al.* (2008) apresenta a análise de um PEE e sua resolução utilizando a heurística *Variable Neighborhood Search* (VNS). Os resultados são comparados aos resultados obtidos por um algoritmo genético comercial.

3. PROBLEMA DE ESCALONAMENTO DE ENFERMEIROS

Neste capítulo é apresentado a descrição do Problema de Alocação de Enfermeiros, objeto de estudo deste trabalho, e as suas características, no que diz respeito às restrições consideradas.

3.1. Definição do problema

O Problema de Escalonamento de Enfermeiros tratado neste trabalho consiste em determinar a escala de trabalho de vários enfermeiros em um determinado período.

A resolução do problema considera algumas restrições, advindas geralmente de acordos contratuais, legislação trabalhista, regras da instituição hospitalar, preferências pessoais de cada profissional, entre outros.

Os dados do problema são definidos por um conjunto de informações que devem ser fornecidas de forma a definir as restrições a serem aplicadas no escalonamento. Neste trabalho consideramos três turnos diários, sendo manhã, tarde e noite, todos com duração de oito horas, e as seguintes informações:

- O período a ser analisado
- A quantidade mínima e máxima de cada dia/turno
- A lista de enfermeiros do hospital, cada um contendo:
 - Nome
 - Carga horária do período

3.2. Restrições

Para modelar o Problema de Escalonamento de Enfermeiros foram utilizadas as seguintes restrições:

- **R1 - Demanda mínima:** quantidade de enfermeiros mínima por turno de cada dia.
- **R2 - Demanda máxima:** quantidade de enfermeiros máxima por turno de cada dia.
- **R3 - Turnos consecutivos proibidos:** um enfermeiro alocado no turno da noite não pode ser alocado no turno da manhã ou da tarde do dia seguinte.
- **R4 - Carga horária:** a soma do total de horas alocadas para um enfermeiro em todo o período deve ser igual a sua carga horária
- **R5 - Máximo de noites consecutivas:** quando o enfermeiro possui mais de 3 noites alocadas em sequência, a solução é penalizada.

- **R6 - Apenas um turno por dia:** cada enfermeiro só pode trabalhar um turno em cada dia.

As restrições descritas acima podem possuir pesos diferentes na penalização da solução. Este peso pode influenciar na escolha entre duas soluções que possuem violações, porém com pesos diferentes.

O custo de uma solução está associado à sua qualidade. A solução é ruim quando seu custo é alto, definindo-se então o objetivo do trabalho: encontrar a solução com o menor custo.

3.3. Modelagem matemática

Nesta seção é apresentada a modelagem matemática de um problema genérico. A modelagem foi embasada no trabalho de Meister (2020).

O Quadro 1 apresenta os parâmetros de entrada que serão utilizados no modelo. O Quadro 2 apresenta as variáveis de decisão.

Quadro 1: Parâmetros de entrada

Símbolo	Definição
$e \in E$	enfermeiro e pertencente ao conjunto E de enfermeiros do hospital
$d \in D$	dia d pertencente ao conjunto D de dias a serem alocados
$t \in T$	turno t pertencente ao conjunto T de turnos a serem alocados
$dMin_{dt}$	demandá mínima de enfermeiros no dia d no turno t
$dMax_{dt}$	demandá máxima de enfermeiros no dia d no turno t
qH_t	quantidade de horas do turno t
ch_e	carga horária do enfermeiro e

Fonte: elaborado pelo autor.

Quadro 2: Variáveis de decisão

Símbolo	Definição
$x_{edt} \in \{0, 1\}$	1 se o enfermeiro e está alocado no dia d no turno t 0 caso contrário

Fonte: elaborado pelo autor.

A função objetivo (3.1) minimiza o custo da alocação considerando as restrições propostas.

$$\min \sum_{e \in E} \sum_{d \in D} \sum_{t \in T} x_{edt} \quad (3.1)$$

Sujeito à:

R1 - Demanda mínima

$$\sum_{e \in E} x_{edt} \geq dMin_{dt} \quad \forall d \in D, t \in T \quad (3.2)$$

R2 - Demanda máxima

$$\sum_{e \in E} x_{edt} \leq dMax_{dt} \quad \forall d \in D, t \in T \quad (3.3)$$

R3 - Turnos subsequentes

$$x_{ed3} + \sum_{t=1}^2 x_{e(d+1)t} \leq 1 \quad \forall e \in E, d \in \{1, \dots, |D| - 1\} \quad (3.4)$$

R4 - Carga horária

$$\sum_{e \in E} x_{edt} * qH_t \leq ch_e \quad \forall d \in D, t \in T \quad (3.5)$$

R5 - Três noites consecutivas

$$x_{ed3} + x_{e(d+1)3} + x_{e(d+2)3} + x_{e(d+3)3} \leq 3 \quad \forall e \in E, d \in \{1, \dots, |D| - 3\} \quad (3.6)$$

R6 - Apenas um turno por dia

$$\sum_{t \in T} x_{edt} \leq 1 \quad \forall e \in E, d \in D \quad (3.7)$$

$$x_{edt} \in \{0, 1\} \quad (3.8)$$

Todas as restrições foram detalhadas nas restrições 3.2 a 3.7. As restrições 3.2 e 3.3 garantem que a demanda mínima e máxima de enfermeiros seja atingida para todos os dias e turnos. A restrição 3.4 garante que o enfermeiro que trabalha a noite não trabalha no período da manhã e da tarde no dia seguinte. A restrição 3.5 garante que nenhum enfermeiro extrapole na sua carga horária contratada. A restrição 3.6 garante que nenhum enfermeiro trabalhe mais do que 3 noites em sequência. A restrição 3.7 garante que cada enfermeiro seja alocado apenas em um turno por dia. A restrição 3.8 expõe apenas que a variável de decisão comprehende apenas dois valores, relativos à alocação ou não de cada enfermeiro.

4. MÉTODOS DE RESOLUÇÃO

Neste capítulo são abordados os métodos de resolução utilizados neste trabalho.

4.1. *Solver* CPLEX

No início dos testes foi utilizado o software IBM CPLEX. Este *solver* foi escolhido por possuir uma licença acadêmica gratuita e por possuir grande reconhecimento no que diz respeito à eficiência em problemas de otimização. Para facilitar a modelagem do problema, o CPLEX utiliza a *Optimization Programming Language* (OPL), uma linguagem específica para modelagem matemática.

Para resolver um problema com o CPLEX bastam dois arquivos: um arquivo *.dat* com os dados específicos de uma instância do problema; e um arquivo *.mod* que possuiá a modelagem matemática propriamente dita, contendo a definição das variáveis e das restrições, bem como um pré e pós processamento.

Neste trabalho, o *solver* foi utilizado para validar a modelagem com uma instância simplificada do problema, ou seja, restrições relaxadas, pouca quantidade de enfermeiros, dias e turnos, e as demandas mínima e máxima com grande amplitude.

Após validada a modelagem do problema, foi implementado a metaheurística *Simulated Annealing*, explicada no tópico a seguir.

4.2. Metaheurística *Simulated Annealing*

A metaheurística *Simulated Annealing* é oriunda de um processo da metalurgia. *Annealing* é um processo baseado na termodinâmica, que possui como finalidade aumentar a resistência de materiais como o metal, vidro, entre outros.

O processo consiste em submeter o material em questão a temperaturas extremamente altas, seguido de um resfriamento lento e gradual, de modo a buscar um estado de equilíbrio de moléculas.

Ocorre que, em temperaturas elevadas, as moléculas estão extremamente agitadas, podendo levar o material a um estado ainda mais instável do que já estava (caminhando no sentido contrário do objetivo do método, no início). Conforme a temperatura vai reduzindo, o material tende a recusar mudanças de estado que causem mais instabilidade.

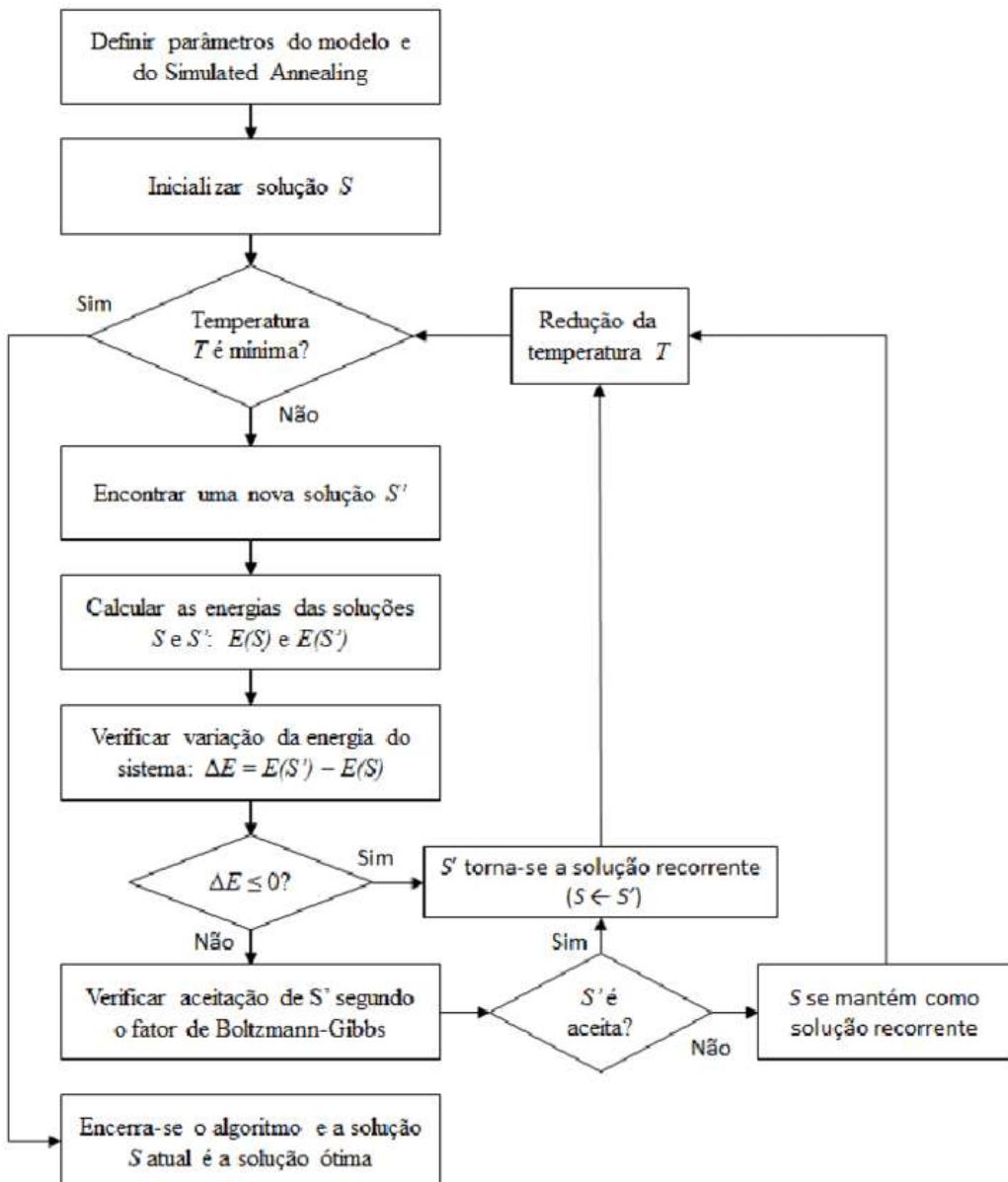
A metaheurística leva o nome *Simulated* à frente pois é traduzida em um ambiente simulado computacionalmente.

Analogamente ao processo da metalurgia, a metaheurística tenta encontrar a solução ótima para um problema de otimização. Porém, em temperaturas elevadas, é possível que a metaheurística aceite soluções em que haja piora do resultado.

4.2.1. Implementação computacional

A metaheurística foi implementada na linguagem C# de acordo com o Fluxograma 1:

Fluxograma 1: Algoritmo da metaheurística *Simulated Annealing*



Fonte: Bessegato; Pereira; Toledo , 2016.

Os parâmetros iniciais da metaheurística são:

- temperatura inicial: 100.000
- temperatura final: 0,000001
- taxa de resfriamento: 1%

A cada resfriamento da temperatura, o algoritmo de busca de uma nova solução S' é executado 50 vezes antes de testar novamente se a temperatura T é mínima.

O fator de Boltzmann-Gibbs foi calculado de acordo com a equação 4.1. O fator está representado pela letra β :

$$\beta = \min\{1, e^{-\Delta E / T}\} \quad (4.1)$$

O cálculo do fator de Boltzmann-Gibbs foi utilizado para aceitar ou não uma nova solução de acordo com as seguintes condições:

- se $\beta > \text{número_aleatorio_entre_0_e_1}$ entao S' é aceito
- senão S' não é aceito

Foi necessário fazer uma adaptação na metaheurística para que fosse possível implementar as restrições.

A cada iteração do método, após ser encontrada uma nova solução, seu valor era calculado baseado na função objetivo, porém a variável de decisão era percorrida procurando-se as violações. A cada violação encontrada, o valor da função objetivo era penalizado.

O objetivo dessa penalização é fazer com que o método reconhecesse que aquele movimento no espaço de busca era ruim, fazendo com que o algoritmo fugisse daquela nova solução.

Ao final, busca-se, então, reduzir o número de violações às restrições.

O processo para encontrar uma nova solução era feito de maneira aleatória, ou seja, a cada iteração o algoritmo sorteava uma quantidade de posições da matriz de alocações e invertia o seu valor (1 para 0, 0 para 1).

O processo de definição da solução inicial também era feito de maneira aleatória. Alguns testes foram feitos para verificar a necessidade de uma solução inicial melhorada A justificativa para esta escolha se encontra na seção 6.2.1.

5. GERADOR DE INSTÂNCIAS

Para facilitar e padronizar os testes, foi implementado um gerador de instâncias aleatórias, cujo resultado é um arquivo texto no formato JSON com a instância gerada. A seguir um exemplo deste arquivo:

```
{
  "Period": {
    "Year": 2010,
    "Month": 9,
    "StartDay": 1,
    "EndDay": 30
  },
  "Physician": [
    {
      "Id": 0,
      "Name": "Ana",
      "Hours": 240
    }
  ],
  "Shifts": [
    {
      "Id": 0,
      "Hours": 8,
      "DMin": 1,
      "DMax": 10
    }
  ]
}
```

O arquivo é composto basicamente de três propriedades:

- uma lista com as informações relativas aos enfermeiros (Physicians)
- uma lista relativa a informação dos turnos (Shifts)
- as informações relativas ao período a ser analisado (Period)

A propriedade Physicians é composta por um identificador, um nome, e as horas

relativas à sua carga horária permitida durante o período analisado.

A propriedade Shifts é composta por um identificador, a duração do turno, a demanda mínima e máxima daquele turno. Por padrão, o gerador sempre gera três turnos de oito horas em cada instância para simular a situação real.

A propriedade Period é composta por um data (ano e mês, apenas para simulação), um dia de início e um dia de término.

Dadas todas essas informações, o algoritmo é capaz de organizar e utilizar os dados necessários para resolver o problema proposto.

5.1. Limite superior da demanda mínima

Meister (2020) apresenta um cálculo para o limite superior da demanda mínima, baseando-se na quantidade de médicos disponíveis no hospital.

Adaptando-se a equação proposta ao trabalho aqui sendo realizado, temos a equação 5.1 que define esse limite:

$$LimSupDT = (numEnf / numTurnos) * \alpha \quad (5.1)$$

Na equação 5.1 o limite superior de demanda mínima de enfermeiros por turno é calculado baseando-se no número de enfermeiros disponíveis (*numEnf*) e na quantidade de turnos a serem alocados (*numTurnos*). Na equação, α representa a taxa de aproximação com relação ao limite superior. Quanto mais próximo do limite, maior a chance da solução ser penalizada.

6. TESTES COMPUTACIONAIS

A seguir estão detalhados os testes realizados no decorrer do trabalho.

6.1. Testes no CPLEX

O software CPLEX foi utilizado neste trabalho com a finalidade de validar a modelagem proposta utilizando uma instância relativamente pequena, e outra maior um pouco mais próxima de uma situação real.

A instância reduzida foi composta de 7 enfermeiros (representados pelas letras de A a G) sendo alocados num período de 5 dias. Cada dia possui 3 turnos de 8 horas. Cada enfermeiro possui uma carga horária de 40 horas. Os turnos possuíam uma demanda mínima de 2 e máxima de 5 enfermeiros por turno. No Quadro 3 encontra-se uma solução ótima encontrada pelo *solver*:

Quadro 3: Solução ótima para instância reduzida

	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5
Turno 1	D, E	D, E	D, E	C, D	B, D
Turno 2	A, F	C, F	C, F	B, F	C, G
Turno 3	B, G	A, G	A, G	A, E	E, F

Fonte: Elaborado pelo autor.

6.2. Testes com a Metaheurística

A seguir estão detalhados alguns testes realizados com a metaheurística.

6.2.1. Comparativo entre solução inicial aleatória e solução inicial melhorada

Foi percebido que, em alguns trabalhos da literatura envolvendo metaheurísticas, existia um esforço para iniciar a execução a partir de uma solução inicial melhorada. Decidiu-se então testar se esse esforço era válido também para o caso do *Simulated Annealing*.

Os testes foram feitos para uma instância reduzida (a mesma utilizada nos testes com o CPLEX) e uma instância aumentada (8 a 12 enfermeiros sendo alocados para um período de 30 dias). Nas execuções com solução inicial aleatória, cada execução possuía uma solução

inicial diferente. Já na solução inicial melhorada, todas as execuções partiram da mesma solução, obtida através de uma pequena alteração da solução ótima.

As Tabelas 1 e 2 apresentam os resultados obtidos com estes testes. Todos foram realizados trocando-se apenas uma alocação aleatória do conjunto de alocações a cada iteração do algoritmo, e com o mesmo peso de penalização à cada restrição violada.

Tabela 1: Comparação de execução com diferentes soluções iniciais para instância reduzida

	Solução inicial aleatória		Solução inicial melhorada	
Execução	Quantidade de violações na solução inicial	Quantidade de violações na solução final	Quantidade de violações na solução inicial	Quantidade de violações na solução final
1	40	1	6	1
2	46	0	6	0
3	39	1	6	1
4	37	1	6	0
5	38	2	6	1

Fonte: Elaborado pelo autor.

Tabela 2: Comparação de execução com diferentes soluções iniciais para instância aumentada

	Solução inicial aleatória		Solução inicial melhorada	
Execução	Quantidade de violações na solução inicial	Quantidade de violações na solução final	Quantidade de violações na solução inicial	Quantidade de violações na solução final
1	369	2	42	3
2	325	2	42	0
3	330	0	42	3
4	346	3	42	0
5	342	4	42	1

Fonte: Elaborado pelo autor.

As informações obtidas por estes testes mostraram que não seria necessário um esforço computacional grande para obter uma solução inicial melhorada. Na prática, a

quantidade de violações da solução final em ambos os casos são facilmente resolvíveis manualmente caso seja necessário.

6.2.2. Porcentagem de variação do espaço de busca

Prosseguindo com os testes, foi realizado uma variação percentual do espaço de busca, ou seja, o algoritmo era executado sequencialmente algumas vezes, variando-se a quantidade (percentual) de trocas de alocações realizadas a cada execução. Todas as execuções tinham sempre uma mesma instância, porém com uma solução inicial diferente obtida aleatoriamente a cada execução.

As tabelas 3 e 4 apresentam os resultados obtidos dos testes realizados.

Tabela 3: Variação percentual no espaço de busca para instância reduzida

Percentual de troca (%)	Valor correspondente à quantidade de alocações trocadas	Violações na solução inicial	Violações na solução final
1	1	30	2
2	2	37	2
3	3	34	1
4	4	46	1
5	5	46	1
6	6	38	2
7	7	55	3
8	8	45	3
9	9	45	3
10	10	37	4
11	12	40	4
12	13	51	5
13	14	38	6
14	15	44	6
15	16	49	6
16	17	53	6
17	18	45	7

Percentual de troca (%)	Valor correspondente à quantidade de alocações trocadas	Violações na solução inicial	Violações na solução final
18	19	43	7
19	20	39	8
20	21	30	6

Fonte: Elaborado pelo autor.

Tabela 4: Variação percentual no espaço de busca para instância aumentada

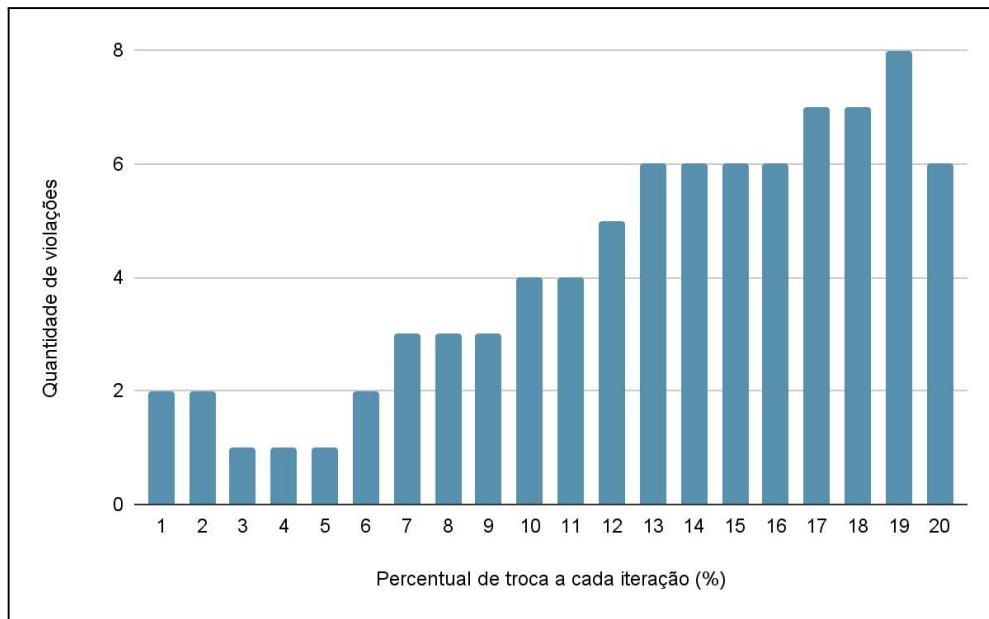
Percentual de troca (%)	Valor correspondente à quantidade de alocações trocadas	Violações na solução inicial	Violações na solução final
0.2	2	349	2
0.4	4	327	2
0.6	5	340	1
0.8	7	338	10
1	9	363	14
1.2	11	316	25
1.4	13	359	25
1.6	14	325	32
1.8	16	355	42
2	18	416	48
2.2	20	304	46
2.4	22	351	57
2.6	23	365	59
2.8	25	300	64
3	27	354	66
3.2	29	336	75
3.4	31	351	76
3.6	32	361	78
3.8	34	359	84
4	36	342	89

Fonte: elaborado pelo autor.

O Gráfico 1 foi gerado a partir dos dados da Tabela 3. O Gráfico 2 foi gerado a partir dos dados da Tabela 4. Ambos mostram a variação da quantidade de violações da solução final de acordo com a respectiva porcentagem de troca.

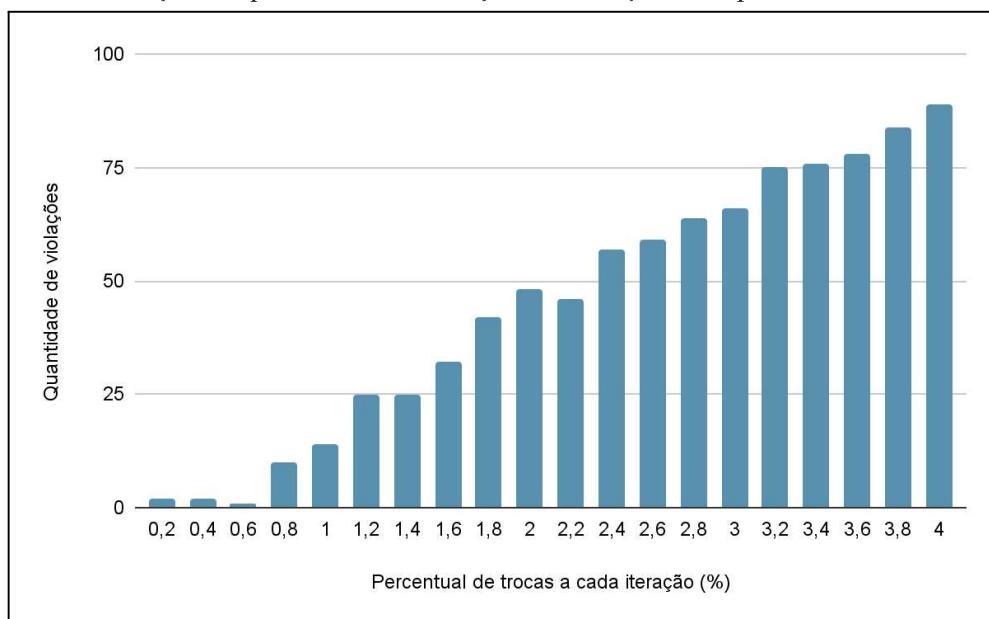
A variação crescente é bem evidente nos dois gráficos, levando a crer que quantidades de variações percentualmente menores levam a resultados melhores.

Gráfico 1: Variação da quantidade de violações da solução final para instância reduzida



Fonte: elaborado pelo autor.

Gráfico 2: Variação da quantidade de violações da solução final para instância aumentada



Fonte: elaborado pelo autor.

6.2.3. Variação da penalização

Todos os testes até então foram realizados com o mesmo peso para todas as violações das restrições.

Este tópico apresenta os testes feitos com variações dos pesos das penalizações de acordo com a sua respectiva restrição. Os testes foram feitos apenas com instâncias aumentadas. O percentual de alocações trocadas por iteração foi fixado em 1% pois, de acordo com os testes da seção 6.2.2, essa porcentagem é suficiente para obter uma quantidade de violações considerável ao final da execução, facilitando, assim, a visualização dos resultados deste teste.

A cada teste uma das restrições terá peso 10 vezes maior do que as outras. O objetivo do teste é saber se a metaheurística consegue distinguir esses pesos e evitar violar as restrições com maior peso.

Tabela 5: Quantidade de violações por restrição nas soluções inicial e final, R1 e R2 com peso maior

Exe- cução	Quantidade de violações por restrição na solução inicial					Quantidade de violações por restrição na solução final				
	R1 R2	R3	R4	R5	R6	R1 R2	R3	R4	R5	R6
1	16	156	10	31	162	0	19	0	0	15
2	18	138	10	14	146	0	16	0	1	20
3	17	142	10	17	154	0	20	0	0	21
4	14	148	10	26	154	0	20	0	0	17
5	17	149	10	9	147	0	20	0	1	18
6	22	136	10	11	143	0	15	0	0	21
7	19	146	10	21	153	0	19	0	1	12
8	13	162	10	15	161	0	11	0	0	16
9	20	147	10	17	159	0	16	0	1	18
10	20	151	10	23	150	0	20	0	0	15

Fonte: Elaborado pelo autor.

A Tabela 5 mostra a quantidade de violações em cada restrição antes e depois da execução do algoritmo. Neste teste, as restrições R1 e R2 tiveram seu peso multiplicado por 10 em relação às outras restrições (que tinham todas o mesmo peso).

Com os resultados obtidos, é possível perceber que houve um foco, por parte da metaheurística, em evitar violar essas restrições.

Tabela 6: Quantidade de violações por restrição nas soluções inicial e final, R3 com peso maior

Exe- cução	Quantidade de violações por restrição na solução inicial					Quantidade de violações por restrição na solução final				
	R1 R2	R3	R4	R5	R6	R1 R2	R3	R4	R5	R6
1	23	160	10	23	154	13	0	0	0	14
2	19	147	10	14	154	19	0	0	0	7
3	19	141	10	11	136	12	0	0	0	9
4	25	156	10	30	158	10	0	0	1	22
5	15	144	10	16	149	20	1	0	0	11
6	14	142	10	24	146	16	0	0	0	14
7	15	135	10	12	146	16	0	0	0	10
8	21	149	10	21	148	10	0	0	2	14
9	17	138	10	14	147	13	0	0	1	14
10	10	145	10	13	154	11	0	0	1	13

Fonte: Elaborado pelo autor.

A Tabela 6 mostra a quantidade de violações em cada restrição antes e depois da execução do algoritmo. Neste teste, a restrição R3 teve seu peso multiplicado por 10 em relação às outras restrições (que tinham todas o mesmo peso).

Ao final, assim como no teste anterior, também é possível perceber que houve um foco, por parte da metaheurística, em evitar violar essa restrição.

Tabela 7: Quantidade de violações por restrição nas soluções inicial e final, R6 com peso maior

Exe- cução	Quantidade de violações por restrição na solução inicial					Quantidade de violações por restrição na solução final				
	R1 R2	R3	R4	R5	R6	R1 R2	R3	R4	R5	R6
1	18	160	10	29	149	14	13	0	0	0
2	20	135	10	15	147	11	17	0	0	0
3	15	142	10	18	164	12	11	0	1	0
4	20	131	10	19	148	8	13	0	0	0
5	22	151	10	22	156	13	16	0	0	0
6	14	148	10	10	156	12	16	0	0	0
7	9	135	10	5	137	13	11	0	0	0
8	22	1443	10	24	140	12	24	0	0	1
9	13	146	10	26	142	15	11	0	0	0
10	13	129	9	13	143	15	13	0	0	0

Fonte: Elaborado pelo autor.

A Tabela 7 mostra a quantidade de violações em cada restrição antes e depois da execução do algoritmo. Neste teste, a restrição R6 teve seu peso multiplicado por 10 em relação às outras restrições (que tinham todos o mesmo peso).

Ao final, assim como nos testes anteriores, também é possível perceber que houve um foco, por parte da metaheurística, em evitar violar essa restrição.

As restrições R4 e R5 não foram testadas pois nos testes acima já é possível perceber que elas são facilmente satisfeitas

6.2.4. Testes gerais com instâncias genéricas variadas

Para consolidação do método, foram criadas 10 instâncias genéricas, cada uma contendo de 8 a 12 enfermeiros, todos com carga horária de 208 horas (8 horas por dia, prevendo 4 dias de folga), sendo alocados para um período de 30 dias, cada dia com 3 turnos de 8 horas de duração. A demanda mínima foi escolhida aleatoriamente entre 2 ou 3, enquanto que a demanda máxima foi escolhida aleatoriamente entre 6 e 10

Os detalhes das demandas de cada instância estão definidos no Quadro 4. A Tabela 8

mostra os resultados obtidos.

Quadro 4: Demandas mínima e máxima separadas por instância

Instâncias	Quantidade de enfermeiros	Demanda mínima			Demanda máxima		
		Turno 1	Turno 2	Turno 3	Turno 1	Turno 2	Turno 3
Inst_1	11	3	2	3	8	10	8
Inst_2	11	2	2	3	10	10	10
Inst_3	12	3	3	3	7	10	6
Inst_4	12	2	3	3	6	9	6
Inst_5	9	2	3	3	8	8	10
Inst_6	11	2	3	3	10	10	9
Inst_7	11	2	2	3	8	6	10
Inst_8	11	2	3	3	9	10	6
Inst_9	8	3	3	2	10	9	7
Inst_10	8	3	2	2	9	10	8

Fonte: elaborado pelo autor.

Tabela 8: Quantidade de violações por instância, separadas por restrição

Instância	Quantidade de violações por restrição na solução inicial					Quantidade de violações por restrição na solução final				
	R1 R2	R3	R4	R5	R6	R1 R2	R3	R4	R5	R6
Inst_1	5	177	11	25	191	11	0	0	0	0
Inst_2	2	161	11	16	166	7	1	0	0	1
Inst_3	16	164	12	17	181	16	0	0	0	0
Inst_4	17	169	12	20	174	9	1	0	0	1
Inst_5	8	130	9	13	133	25	0	0	0	1
Inst_6	0	156	11	14	156	11	1	0	0	2
Inst_7	14	172	11	32	174	7	1	0	0	1
Inst_8	13	157	11	30	165	12	0	0	0	0
Inst_9	13	114	8	9	118	34	0	0	0	1
Inst_10	3	114	8	9	116	20	0	0	0	0

Fonte: Elaborado pelo autor.

É possível perceber que a metaheurística conseguiu otimizar a solução, porém na maioria dos casos houve um aumento na quantidade de violações nas restrições R1 e R2. Percebe-se que nas instâncias onde a soma das demandas mínimas dos turnos é relativamente próxima à quantidade de enfermeiros disponíveis para alocação, a quantidade de violações nas restrições R1 e R2 foi maior.

Em acréscimo a este teste, outro teste foi realizado, porém dessa vez aumentando o peso das restrições R1 e R2, com o objetivo de forçar a metaheurística a evitar violar essas restrições.

A tabela 9 apresenta os resultados obtidos.

Tabela 9: Quantidade de violações por instância, separadas por restrição, R1 e R2 com peso maior

Instâncias	Quantidade de violações por restrição na solução inicial					Quantidade de violações por restrição na solução final				
	R1 R2	R3	R4	R5	R6	R1 R2	R3	R4	R5	R6
Inst_1	1	160	11	25	171	0	5	0	0	6
Inst_2	1	162	11	23	155	0	1	0	1	1
Inst_3	12	152	12	15	162	0	14	0	0	1
Inst_4	32	191	12	34	200	0	8	0	0	7
Inst_5	6	125	9	15	122	0	24	2	0	20
Inst_6	2	162	11	19	168	0	9	0	0	7
Inst_7	16	176	11	17	179	0	3	0	1	6
Inst_8	10	160	11	20	158	0	15	0	0	1
Inst_9	13	105	8	10	111	1	35	5	0	40
Inst_10	10	111	8	16	114	0	30	2	0	25

Fonte: Elaborado pelo autor.

É possível perceber que, nas instâncias em que haviam mais violações nas restrições R1 e R2 no teste anterior, também apareceram mais violações no restante das restrições neste teste.

7. CONCLUSÕES

A metaheurística foi capaz de resolver o Problema de Escalonamento de Enfermeiros, quase sempre encontrando uma solução ótima ou quase ótima. Nos casos em que o número de violações das restrições eram baixos, essas violações eram facilmente solucionáveis manualmente, mostrando que a metaheurística é eficaz para esse tipo de abordagem.

A respeito da solução inicial melhorada e aleatória pode-se dizer que, em ambos os casos, não há grandes diferenças na qualidade da solução final encontrada, mostrando que não é necessário realizar um esforço computacional a mais na busca por uma solução inicial melhorada. Acredita-se que isso aconteça devido ao início da execução da metaheurística, visto que no início, em temperaturas altas, o algoritmo aceita muitos movimentos de piora da solução. Portanto, a qualidade da solução inicial aparenta ser indiferente neste caso.

Em relação da variação da quantidade de alocações trocadas pode-se dizer que uma menor quantidade de trocas por iteração leva a um resultado final com melhor qualidade, visto que as trocas acabam sendo mais precisas, quase como se o algoritmo estivesse “aparando arestas” da solução a cada iteração.

A respeito dos diferentes pesos para cada violação de restrições pode-se dizer que a metaheurística consegue reconhecer pesos diferentes para restrições diferentes, assim balanceando adequadamente o custo de cada restrição caso elas venham a ter pesos diferentes em uma situação real.

De maneira geral, a metaheurística *Simulated Annealing* mostrou-se eficaz para a resolução do Problema de Escalonamento de Enfermeiros.

Pensando em trabalhos futuros, este trabalho pode ser sucedido de muitas outras abordagens e continuações.

Em um primeiro momento, pode-se considerar fazer um estudo de caso e comparar os resultados aqui obtidos com os resultados obtidos na situação real.

Pode-se também considerar um futuro estudo no qual se proponha uma metaheurística híbrida com o *Simulated Annealing* (uma possibilidade seria a heurística *Variable Neighborhood Search*).

REFERÊNCIAS

- BESSEGATO, L.; PEREIRA, A.; TOLEDO, L. 2016. Planejamento econômico de controle on-line por atributo com intervalo de inspeção variável e classificações repetidas. **EMEPRO**. Disponível em:
https://www.researchgate.net/publication/309357358_Planejamento_economico_de_controle_on-line_por_atributo_com_intervalo_de_inspecao_variavel_e_classificacoes_repetidas. Acesso em: 08 nov 2021.
- BURKE, E.; CURTOIS, T.; POST, D. G.; QU, R.; VELTMAN, D. B. A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. **European Journal of Operational Research**, Elsevier, Amsterdam, v.188, n.2, p.330-341, July 2008.
- CHEANG, B.; LI, H.; LIM, A.; RODRIGUES, B. Nurse rostering problems - a bibliographic survey. **European Journal of Operational Research**, v. 151, n.2, p. 447-460, 2003.
- DEVESSE, V. A. P. A. **Métodos de solução para o problema de escalonamento de médicos**. Orientadora: Maristela Oliveira dos Santos. 2016. Dissertação (Mestrado em Ciências da Computação) - Universidade de São Paulo. Disponível em:
https://teses.usp.br/teses/disponiveis/55/55134/tde-07112016-101859/publico/ValdemarAbraoPedroAnastacioDevesse_revisada.pdf. Acesso em: 1 ago 2022.
- ERNST, A.; JIANG, H.; KRISHNAMOORTHY, M.; SIER, D. Staff scheduling and rostering: A review of applications, methods and models. **European Journal of Operational Research**, v.153,n.2,p.3-27,2004.
- ERHARD, M; SCHOENFELDER, J. FÜGENER, A.; BRUNNER, J. O. State of the art in physician scheduling. **European Journal of Operational Research**, Elsevier, v. 265, n. 1, p. 01–18, 2018.
- MEISTER, T. C. **Um algoritmo heurístico aplicado ao problema de escalonamento de médicos do Hospital de Clínicas de Porto Alegre**. Orientadora: Luciana Buriol. 2020. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Rio Grande do Sul. 2020. Disponível em:
<https://www.lume.ufrgs.br/bitstream/handle/10183/217592/001121923.pdf?sequence=1>. Acesso em: 06 mai. 2021.
- ROUSSEAU, L.-M.; PESANT, G.; GENDREAU, M. A general approach to the physician rostering problem. **Annals of Operations Research**, Kluwer Academic Publishers, v. 115, n. 1-4, p. 193-205, 2002. ISSN 0254-5330.
- SILVA, V. M. D. **Um modelo heurístico para alocação de navios em berços**. Orientador: Antônio Sérgio Coelho, 2008. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Santa Catarina. 2008. Disponível em:
<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/91781/249010.pdf?sequence=1&isAllowed=y>. Acesso em: 06 mai. 2021.

SOARES, H. C. A. **Um estudo sobre o problema de alocação.** 2011. Orientador: Luis Augusto Angelotti Meira. 2011. Dissertação - Universidade Federal de São Paulo, São José dos Campos. 2011. Disponível em: <https://www.ft.unicamp.br/docentes/meira/publicacoes/2011henrique.pdf>. Acesso em: 06 mai. 2021.

SUBRAMANIAN, A; MEDEIROS, J. M. F.; FORMIGA, L. A.; SOUZA, M. J. F. Aplicação da metaheurística busca tabu ao problema de alocação de aulas a salas em uma instituição universitária. **Revista Produção Online**, Florianópolis, v. 11, n. 1, p. 54-75, mar. 2011. ISSN 16761901. Disponível em: <https://www.producaoonline.org.br/rpo/article/view/419/762>. Acesso em: 06 mai 2021.