



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**Computação evolucionária aplicada ao
Problema de Geração de Escala de
Enfermeiros**

Aline Marina Lopes

**TRABALHO DE
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:

Prof. Fernando Bernardes de Oliveira

COORIENTAÇÃO:

Prof. George Henrique Godim da Fonseca

**Fevereiro, 2018
João Monlevade–MG**

Aline Marina Lopes

**Computação evolucionária aplicada ao
Problema de Geração de Escala de Enfermeiros**

Orientador: Prof. Fernando Bernardes de Oliveira

Coorientador: Prof. George Henrique Godim da Fonseca

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Fevereiro de 2018

A Ficha Catalográfica é elaborada exclusivamente pela Biblioteca. Substitua esta página pelo documento gerado na versão final da sua monografia.

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

Computação evolucionária aplicada ao Problema de Geração de Escala de Enfermeiros

Aline Marina Lopes

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:

Prof. Fernando Bernardes de Oliveira
Doutor
DECSI – UFOP

Prof. George Henrique Godim da Fonseca
Doutor
DECSI – UFOP

Nome Completo do Convidado1
Titulação
Examinador
DECSI – UFOP

Nome Completo do Convidado2
Titulação
Examinador
DECSI – UFOP

Nome Completo do Convidado3
Titulação
Examinador
DECSI – UFOP

João Monlevade, 20 de fevereiro de 2018

ATA DE DEFESA

No dia 20 do mês de Fevereiro de 2018, às 17:00 horas, na sala C304 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo(a) aluno(a) **Aline Marina Lopes**, sendo a Comissão Examinadora constituída pelos professores: Prof. Fernando Bernardes de Oliveira, Prof. George Henrique Godim da Fonseca, Nome Completo do Convidado1, Nome Completo do Convidado2, Nome Completo do Convidado3. O(a) candidato(a) apresentou a monografia intitulada: “**Computação evolucionária aplicada ao Problema de Geração de Escala de Enfermeiros**”. A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota _____ (_____), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo(a) graduando(a).

Prof. Fernando Bernardes de Oliveira
Doutor
DECSI – UFOP

Prof. George Henrique Godim da Fonseca
Doutor
DECSI – UFOP

Nome Completo do Convidado1
Titulação
Examinador(a)
DECSI – UFOP

Nome Completo do Convidado2
Titulação
Examinador(a)
DECSI – UFOP

Nome Completo do Convidado3
Titulação
Examinador(a)
DECSI – UFOP

Aline Marina Lopes

João Monlevade, 20 de fevereiro de 2018

TERMO DE RESPONSABILIDADE

Eu, **Aline Marina Lopes** declaro que o texto do trabalho de conclusão de curso intitulado “*Computação evolucionária aplicada ao Problema de Geração de Escala de Enfermeiros*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 20 de fevereiro de 2018

Aline Marina Lopes

Este trabalho é dedicado à Deus, que me deu a oportunidade, a força, a fé, a perseverança e a esperança para concluir mais um sonho.

Agradecimentos

Agradeço primeiramente a Trindade Santa, que me permitiu alcançar mais um sonho, erguendo-me em momentos que pensei que não teria força. Agradeço aos enormes anjos que Deus me enviou, ao qual os chamo de pai(Aníbio) e mãe(Lúcia), que sempre vibraram com as minhas conquistas, mas principalmente me incentivaram, ajudaram, empurraram e me ampararam nos momentos mais difíceis. Vocês são com toda a certeza os melhores pais do mundo.

Agradeço também ao meu super namorado Cleber, que escutou todas as minhas comemorações, angústias e choros desses 4 anos, por ter compreendido minha ausência e acreditado na minha capacidade. Ao meus irmãos maravilhosos Rafael e Poia(Amanda) por terem me feito rir quando eu achei que não fosse capaz de ao menos secar minhas lágrimas e por fingirem que me acham inteligente. Meu Deus vocês são incríveis!

Aos meus amores Jucelia, Kamylla e Gleice, a qual chamo de família, por serem a melhor família para se estar quando se está longe, obrigada por cada momento de risada, desespero, luta contra "kalangos" gigantes e baratas mutantes e principalmente pelo companheirismo, que permitiu nunca me sentir totalmente sozinha, pois uma parte da minha família estava nos quartos ao lado. E falando em teto, Edivânia, Marcelo e tia Aparecida, obrigado por terem me ajudado no momento mais assustador, a reviravolta na minha realidade, vocês a tornaram bem mais leve de se adaptar.

Aos meus amigos da 'salinha', vocês foram os melhores seres que conheci nessa faculdade, apesar de serem os mais estranhos são os mais companheiros que eu poderia ter a honra de conhecer, Deus providenciou tudo de forma tão perfeita que ele colocou cada um de vocês bem pertinho de mim, sem vocês eu não estaria formando agora, afirmo com toda a certeza. Aos meus amigos de BH, MonleCity e Visão Jr, saibam que todos vocês tem a capacidade de mudar o mundo para melhor, porque vocês mudaram o meu mundo para melhor.

Ah e um enorme obrigado para a melhor família do universo, a minha, por terem rezado tanto por mim, saibam que a oração de vocês me deu força para seguir em frente em todos os momentos em que pensei que iria desabar, e sei que não desabei porque vocês estavam me sustentando com vossas orações.

Meus orientadores papais, Fernando e George, obrigada por serem tão alto astral e companheiros, não poderia ter escolhido profissionais e pessoas melhores para me ajudarem a finalizar esse passo m chave de ouro.

OBRIGADO à todos vocês por fazerem parte da minha vida!

“Não pretendo dizer que já alcancei(esta meta) e que cheguei a perfeição. Não. Mas eu me empenho em conquistá-la, uma vez que também eu fui conquistado por Jesus Cristo.”

— A Bíblia (Fp. 3:12)

Resumo

A geração de escalas para enfermeiros nas instituições de saúde é uma tarefa complexa e de difícil solução devido a existência de diversas restrições que necessitam ser atendidas. Na literatura o escalonamento de enfermeiros é uma variação do *timetabling*, que é um problema identificada como *Scheduling Nurse Problem*, que configura a geração de horários para os enfermeiros em uma grade, em um período finito de tempo. O escalonamento está sujeito a restrições como recursos humanos, tempo, status de trabalho, ou seja, se o funcionário está de folga ou precisa cumprir horário, regras do hospital, determinação da legislação e preferências dos funcionários. Para solucionar o problema e automatizar o processo de construção da melhor grade,

Palavras-chaves: latex. abntex. editoração de texto.



Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Cromossomo	24
Figura 2 – Tipos de Turno	28
Figura 3 – Exemplo de Escala	29
Figura 4 – Algoritmo	36
Figura 5 – Representação da Solução	38
Figura 6 – Late Acceptance Hill Climbing	42
Figura 7 – Parâmetros Definidos	46
Figura 8 – Parâmetros Finais	47
Figura 9 – Instâncias selecionadas	48
Figura 10 – Média e Desvio Padrão do Limite de Custo	50
Figura 11 – Média e Desvio Padrão do Limite de Custo	50

Lista de tabelas

Tabela 1 – GAP(%) Bilgin et al. (2010)/Trabalho	48
Tabela 2 – GAP(%) IRNC/Trabalho	49

Lista de abreviaturas e siglas

CE	Computação Evolucionária
CGA	<i>Cooperative Genetic Algorithms</i>
CODES	<i>Combinatorial Optimisation and Decision Support</i>
COSI	Colegiado de Sistemas de Informação
DECSI	Departamento de Computação e Sistemas
DP	Desvio Padrão
EE	Estratégias Evolutivas
GLPK	<i>GNU Linear Programming Kit</i>
HC	<i>Hill Climbing</i>
ICEA	Instituto de Ciências Exatas e Aplicadas
ILS	<i>Iterated Local Search</i>
INRC	<i>International Nurse Rostering Competition</i>
JM	João Monlevade
LA	<i>Late Acceptance</i>
LI	<i>Resultados Obtidos Pelo Literatura</i>
LAHC	<i>Late Acceptance Hill Climbing</i>
LT	<i>Resultados Obtidos Pelo Trabalho</i>
NSP	<i>Nurse Scheduling Problem</i>
SI	Sistemas de Informação
SINTEF	<i>Stiftelsen for industriell og teknisk forskning</i>
UFOP	Universidade Federal de Ouro Preto
UPA	Unidades de Pronto Atendimento
PEE	Problema de Escalonamento de Enfermeiros
PSF	Programa de Saúde da Família

Lista de símbolos

λ Lambda

μ Mi

Sumário

1	INTRODUÇÃO	17
1.1	O problema de pesquisa	18
1.2	Objetivos	18
1.2.1	Objetivo Geral	18
1.2.2	Objetivo Específico	19
1.3	Metodologia	19
1.4	Organização do trabalho	19
2	PROBLEMA	21
2.1	Problema	21
2.2	O Problema de Escalonamento de Enfermeiros	22
2.3	Revisão bibliográfica	24
2.4	Descrição do Problema Abordado	27
2.4.1	<i>Internacional Nurse Rostering Competition</i>	27
2.4.2	Restrições	29
2.4.3	Função Objetiva	32
2.5	Considerações Finais	33
3	ALGORITMO	35
3.1	Visão geral do algoritmo	35
3.2	Fluxograma do algoritmo	35
3.3	Leitura de instâncias	36
3.4	Estratégia Evolutiva	37
3.4.1	Operador de Mutação	41
3.5	Late Acceptance Hill Climbing	42
3.6	Considerações Gerais	43
4	RESULTADOS	45
4.1	Parâmetros e estrutura de testes	45
4.2	Análise dos Resultados Computacionais	47
5	CONCLUSÃO	51
	REFERÊNCIAS	53

APÊNDICES

56

ANEXOS

57

1 Introdução

Este trabalho propõe o estudo e aplicação de técnicas de computação evolucionária para solucionar o problema de *timetabling* em um determinado âmbito. O problema de *Timetabling* dedica-se à alocação de horários para distintos contextos, como horários de aula em instituições, escalonamento de transporte público, escalonamento de funcionários na área da saúde, entre outras. O contexto específico que adota este trabalho é o de *Nurse Scheduling Problem*.

O problema de *Timetabling*, no cenário de escalonamento de enfermeiros, vem ganhando atenção nos últimos anos, visto que os hospitais precisam lidar com a tarefa de escalonar enfermeiros para um determinado período. Em geral, os hospitais adotam o regime de funcionamento de 24 horas diárias, e a realização da tarefa de escalonamento requer o uso de um recurso muito importante, o tempo, visto que este problema necessita relacionar e respeitar diversas restrições envolvidas que podem ser provenientes da instituição hospitalar, legislação vigente e/ou dos funcionários.

Petrovic e Berghe (2008) destacam que o problema de escalonamento se tornou muito interessante no campo da pesquisa operacional e inteligência artificial, principalmente a partir da década de 1990. Ressaltam ainda que os modelos desenvolvidos nos trabalhos não são evoluídos para uma modo mais completo, na qual dá-se continuidade aos trabalhos anteriores. Fato que dificulta a implementação de um modelo único para resolução de qualquer problema, mesmo que sejam semelhantes. E o aumento da ocorrência de trabalhos cada vez mais particulares, com restrições e função objetivo mais específicas, torna a comparação com trabalhos anteriores quase impossível.

Bôaventura, Pinto e Yamanaka (2013) apresentam diversas técnicas que podem ser abordadas na resolução desse problema. As principais técnicas utilizadas são Algoritmo Genético (JAN; YAMAMOTO; OHUCHI, 2000), *Greedy Randomized Adaptive Search Procedure* (GRASP) (GOODMAN; DOWSLAND; THOMPSON, 2009), Busca Tabu (DOWSLAND, 1998), Programação Inteira (SANTOS et al., 2012) e (VALOUXIS et al., 2012), dentre outras.

A geração de escalas na área da saúde, constitui-se na determinação de uma escala que possa assistir as necessidades do hospital e do funcionário. Esta deve atender de forma suficientemente adequada as determinações impostas no contrato, bem como a demanda mínima exigida. À vista disso, o método proposto objetiva satisfazer o máximo de restrições fracas concebíveis.

Diante disso, o objetivo principal deste trabalho é realizar estudos sobre os métodos de computação evolucionária aplicáveis ao contexto, e adequá-los às características do

problema de escalonamento de enfermeiros. O problema abordado é o de *timetabling*, discutido neste trabalho, no que tange ao escalonamento, jornadas de trabalho adotados, distribuição de turnos, assim como as restrições impostas. Todavia, características próprias foram observadas, como quantidade mínima de profissionais exigidos por turno, padrões indesejados de turnos e folgas, tipos de turno vigentes, particularidades que são definidas nos contratos de trabalho. Estas características serão abordadas na modelagem do problema tratado.

O algoritmo aqui proposto utiliza técnicas de computação evolucionária baseadas em estratégia evolutiva e emprega a meta-heurística *Late Acceptance Hill Climbing*, como estratégia de busca local para a solução. A proposta é desenvolver uma heurística com a aplicação das técnicas citadas e verificar se é possível encontrar soluções heurísticas de boa qualidade para o problema de geração de horários para enfermeiros.

1.1 O problema de pesquisa

Em unidades de saúde que operam durante 24 horas por dia é normal o uso de escalas baseadas em turnos. Quando o número de enfermeiros é elevado, a produção de suas escalas de trabalho se torna uma tarefa pouco trivial, devido a existência de diversas restrições que devem ser atendidas. As restrições podem ser solicitações dos funcionários e/ou regras determinadas pela unidade de saúde e pela legislação vigente. Segundo Poltosi (2007a), a complexidade fundamenta-se nas restrições existentes que devem ser atendidas na geração das escalas, enquadrando-se na classe de problemas de otimização \mathcal{NP} -difícil. Além disso, tal autor aponta também que, a falta de ferramentas computacionais que auxiliem na geração de escalas de trabalho no segmento de enfermagem que gerem resultados condizentes, em um tempo computacional aceitável, induz as unidades de saúde a optarem pela realização da tarefa de forma manual. De acordo com Santos (2006), esse tipo de problema na literatura é definido como “*Nurse Rostering Problem*” ou “*Nurse Scheduling Problem*”.

1.2 Objetivos

~~1.2.1 Objetivo Geral~~

O presente trabalho consiste em desenvolver técnicas baseadas em computação evolucionária e estratégias de busca local para a geração de escalas de horários de enfermeiros considerando as restrições definidas pelas instâncias de teste utilizadas.

~~1.2.2 Objetivo Específico~~

~~Este trabalho possui os seguintes objetivos específicos:~~

1. Modelar e implementar uma estratégia baseada em computação evolucionária para a geração de escala enfermeiros.
2. Estudar mecanismos de busca local para o problema.
3. Validar o algoritmo proposto por meio de experimentos computacionais utilizando instâncias de teste disponíveis na literatura.

1.3 Metodologia

O objeto de pesquisa deste trabalho é o estudo de problemas de otimização, mais especificamente o problema de agendamento de enfermeiros de maneira prática, com a aplicação de técnicas de computação evolucionária por meio da elaboração de um algoritmo capaz de gerar soluções factíveis, realizar o cálculo do custo e definir qual a melhor solução gerada.

Os passos para execução deste trabalho são assim definidos:

1. Revisar a literatura sobre *timetabling* e suas classes, trabalhos correlatos da literatura e demais conceitos, bem como identificar métodos e técnicas utilizadas na solução deste tipo de problema.
2. Estudar métodos para representação do problema identificando abordagens e estruturas de dados específicos.
3. Definir e implementar uma meta-heurística para o problema baseado em métodos da computação evolucionária.
4. Estudar e incorporar mecanismos de busca local aplicados ao tipo do problema.
5. Planejar e realizar experimentos para avaliar a meta-heurística proposta, como performance, soluções obtidas, entre outras.
6. Analisar e discutir os resultados obtidos, além de identificar possíveis melhorias e considerações gerais sobre o processo.

1.4 Organização do trabalho



O restante desta dissertação é organizado como se segue: O [Capítulo 2](#) apresenta o *Nurse Scheduling Problem* e os principais trabalhos que são relevantes ao contexto da dissertação, tal como suas características, abordagens, técnicas empregadas e resultados. Para





uma melhor compreensão sobre o problema tratado o capítulo aborda inclusive sobre as instâncias utilizadas, tal como a justificativa para adoção de instâncias da literatura e suas características. A implementação da heurística é elucidada no [Capítulo 3](#). Com o intuito de validar o algoritmo são realizados experimentos que possibilitam análises e geração de resultados que são encontrados no [Capítulo 4](#). Por fim, uma discussão final sobre a dissertação com intuito de avaliar o trabalho e propor melhorias são explanadas no ??

2 Problema

Neste capítulo, serão apresentados os principais conceitos que abrangem o problema de escalonamento de enfermeiros. O *Timetabling Problem* possui um série de vertentes que abrange diversificadas áreas. Em sua maioria os pesquisadores e estudiosos aplicam otimização combinatória para encontrar uma solução, com isso uma breve explicação sobre o problema é dada na [seção 2.1](#), e dentre as vertentes existentes, encontra-se o conceito de escalonamento de enfermeiros no âmbito hospitalar, que é o abordado neste trabalho. O objetivo deste capítulo é fornecer conhecimento sobre o problema de otimização envolto na programação para enfermeiros, caracterizando sobre suas particularidades e definições, que podem ser vistas na [seção 2.2](#). Um entendimento melhor sobre a contribuição da literatura para novas ideias em direção à resolução do problema, vem com a apresentação de trabalho correlatados na [seção 2.3](#). Com o propósito de contextualizar melhor sobre o problema tratado nesta dissertação a [seção 2.4](#) aborda todas as características, definições cruciais para compreensão do trabalho e das técnicas utilizadas a serem explicadas no [Capítulo 3](#)

2.1 Problema



O problema de otimização a ser tratado neste trabalho é conhecido como *Timetabling*. Este problema dedica-se à alocação de horários para distintos contextos como horários de aula em instituições (KRIPKA; KRIPKA; SILVA, 2011), escalonamento de transporte público (RIBEIRO et al., 2014), área da saúde debatido por (FOURATI; JERBI; KAMMOUN, 2016), entre outros diversos contextos.

Müller e Barták (2002) menciona que o *timetabling* pode ser visto como uma forma de agendamento, onde se aloca recursos respeitando algumas restrições. O problema de *timetabling* apresenta copiosas restrições e grandes números de variáveis, que necessitam ser atendidas ao obter-se uma solução ótima. Essas restrições podem ser restrições fortes ou fracas (GOMES, 2012).

Desse modo, o problema se caracteriza também pela limitação do período de tempo, e os seus recursos não podem ser concebidos para dois ou mais eventos simultaneamente. Logo as variações comuns para *timetabling* consomem a caracterização do problema por condição fixa e restrições que variam.

De acordo com Kuleuven (2010) o problema de *timetabling*, no contexto de escalonamento de enfermeiros vem ganhando atenção nos últimos anos, uma vez que, gradativamente, o surgimento de estratégias podem melhorar o aproveitamento dos recursos disponíveis. O problema é reconhecido como um problema \mathcal{NP} -difícil de otimização

combinatória, que caracteriza a impossibilidade de determinação da solução em tempo polinomial. Nas instituições hospitalares a produção de soluções viáveis e de alta qualidade requer um esforço intenso.

2.2 O Problema de Escalonamento de Enfermeiros

O escalonamento de enfermeiros é um problema complexo que trata a escala de trabalho dos funcionários e afeta diretamente o funcionamento das instituições de saúde. Em unidades de saúde que adotam o regime de funcionamento de 24 horas diárias é normal o uso de escalas baseadas em turnos. Ao elevar o número de enfermeiros, a produção de suas escalas de trabalho torna-se uma tarefa não trivial, devido a necessidade de corresponder com as situações do mundo real e que atenda as diversas restrições provenientes da instituição hospitalar, legislação vigente e/ou as preferências dos funcionários, entre outros.

Burke et al. (2003 apud POLTOSI, 2007b) classifica o Problema de Escalonamento de Enfermeiros (PEE) como sendo um problema de otimização \mathcal{NP} -difícil, devido a complexidade fundamentar-se nas restrições existentes nas escalas. Esse problema é chamado na literatura de "*Nurse Rostering Problem*" ou "*Nurse Scheduling Problem*".

Tal problema expressa motivação econômica, uma vez que busca diminuir os custos com geração de horários de forma manual, como levantado por Poltosi (2007b), bem como o esforço em ajustar às necessidades da instituição e necessidades dos enfermeiros, para conter o desperdício ou excesso de recursos. Existindo também a motivação científica, devido a inexistência de algoritmo capaz de obter em tempo computacional viável, uma solução ótima, afirmado por FURTADO e VARREIRA (2013).

O PEE consiste em gerar escalas de trabalho para enfermeiros atribuindo um conjunto N de enfermeiros à um período D dias, cada enfermeiro tem um turno T atribuído para cada dia D do período. Escalas como já descrito anteriormente, são constituídas por restrições classificadas como fortes e fracas. Essas restrições são definições contidas nos contratos de trabalho, nas preferências dos funcionários, na legislação trabalhista, nas preferências e/ou necessidades levantadas pelo enfermeiro chefe, entre outras coisas. As restrições são responsáveis por definirem a complexidade de geração de solução ótima em tempo computacional viável.

Trabalho por turnos é definido como:

Aquele em que a organização do trabalho é efetuada por equipes em que os trabalhadores ocupam sucessivamente os mesmos postos de trabalho a um determinado ritmo, rotativo, contínuo ou descontínuo. (ECONOMIAS, 2016)

Os turnos variam de acordo com as organizações, estas são livres para determinar qual ou

quais escalas de turnos pretendem adotar. Constantino et al. (2009) define uma jornada como uma sequência de turnos e as tarefas como a necessidade de atribuir a um dia da escala determinado turno. Na enfermagem, de acordo com o site do COREN (2011), a carga horária semanal de trabalho varia de 30 a 40 horas semanais e as jornadas diárias de trabalho variam de 6 horas, 8 horas, 12 horas de trabalho por 36 de descanso e ainda 4 dias de 6 horas e 1 dia de 12 horas.

Causmaecker e Berghe (2004) categoriza os problemas de escala de enfermeiros ou agendamento hospitalar em:

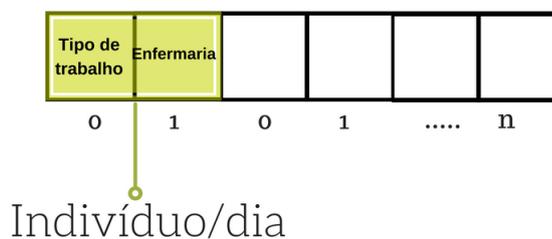
- **Pessoal do Hospital:** a determinação do número de pessoas com as habilidades necessárias, para cumprir com os requisitos existentes. A estrutura organizacional, a forma de recrutamento de pessoal, classes de habilidades do pessoal, preferências dos trabalhadores, necessidades dos pacientes, circunstâncias particulares das unidades de saúde, são alguns dos fatores que tornam a tarefa complexa.
- **Agendamento centralizado:** duas vantagens são a imparcialidade para os funcionários por meio da aplicação consistente, objetiva e imparcial de políticas e oportunidades de contenção de custos por meio do melhor uso dos recursos. Os agendamentos são construídos pelos enfermeiros chefes ou gerentes de unidades. Por essa razão é considerado vantagem, visto que os enfermeiros tem uma atenção mais personalizada, em contrapartida, a enfermeira chefe pode estar a dar tratamento preferencial para certas pessoas, causando desconforto nos demais enfermeiros.
- **Auto agendamento:** é mais demorado do que o agendamento automático, todavia tem a vantagem de os enfermeiros cooperarem e receberem orientação. São realizados geralmente pelos próprios enfermeiros com a coordenação do enfermeiro-chefe. Tal técnica é considerada tão comum, que a automação completa não é recomendada.
- **Agendamento cíclico:** refere-se a organização em que cada pessoa trabalha um ciclo durante várias semanas. Os funcionários tem conhecimento de seu ciclo com bastante antecedência e os mesmos padrões de escala são ~~usadas~~ repetidas vezes. Apesar de possuir benefícios, as desvantagens começam quando este não possui generalidade. Não podendo abordar recursos flexíveis e preferências pessoais sem a realização de grandes mudanças.

A existência de diversos trabalhos na literatura que abordam o problema de escalonamento de enfermeiros, confirma o crescimento de interesse em tal assunto. Observa-se que o estudo deste engloba diferentes objetivos, que podem ser a descoberta de uma nova estratégia para o campo de pesquisa, métodos para resolução de dificuldades encontrados no agendamento de escala de pessoas real em uma organização, inclusive a necessidade de aprimoramento de estratégias já implementadas.

2.3 Revisão bibliográfica

Em [Bôaventura, Pinto e Yamanaka \(2013\)](#) aplicou-se a técnica de algoritmos genéticos na criação de uma grade horária otimizada para enfermeiros optando-se pela representação do cromossomo com valores decimais. No trabalho o indivíduo possui um cromossomo de n genes, em que n é dado por $2 \times \text{quantidade de enfermeiros} \times \text{quantidade de dias no mês}$, como mostrado na [Figura 1](#). O dia do indivíduo no período de trabalho é representado por um par genes, onde na posição par é representado o tipo de trabalho a ser realizado e na posição ímpar a enfermaria onde deverá ser realizado tal trabalho. No cromossomo a representação do tipo de trabalho é definida utilizando valores entre 0 e 6 e a de enfermaria varia de 1 até quantidade de enfermarias cadastradas.

Figura 1 – Cromossomo



Fonte: elaborado pela autora

A função objetivo leva em consideração 11 regras básicas obtidas no Hospital das Clínicas da Universidade de São Paulo (USP) de Ribeirão Preto, que podem ser ativadas ou desativadas pelo usuário. No trabalho em questão as técnicas utilizadas são as de mutação e cruzamento. Na mutação pode-se alterar até 10 genes, mas o padrão é a alteração de somente 1 gene. O cruzamento utilizou técnicas de torneio e roleta, empregados na seleção dos pais e na aplicação posterior de um ponto de corte e reinserção. Para isso, foram utilizadas soluções baseadas em elitismo total ou parcial. O projeto desenvolveu uma interface gráfica que permitiu a interação do usuário com o algoritmo, concedendo ao usuário a opção de manipular as restrições ativando ou desativando cada uma. Os experimentos sugerem que o algoritmo pode evoluir a solução de maneira apropriada, contudo em alguns casos necessita de várias gerações, conseguindo assim em sua maioria dar resultados com solução satisfatória.

[Valouxis et al. \(2012\)](#) em seu trabalho usufruiu das especificações do problema de agendamento de enfermeiros definidos pela Primeira Competição Internacional do Problema de Agendamento de Enfermeiros, que ocorreu em 2010. Os autores constituíram uma das equipes competidoras do evento em questão. As instâncias dos problemas disponibilizadas pela competição, foram divididas pelos autores em subproblemas de

tamanhos que possibilitassem o manejo computacional e posteriormente a resolução dos subproblemas utilizando Programação Matemática Inteira.

 trabalho foi implementado uma estratégia composta por duas fases. A primeira fase realizou a definição da carga de trabalho individual dos enfermeiros para determinado dia da semana, atendendo assim a exigência que define que um funcionário deve ser atribuído a um único turno por dia, e que a demanda mínima de funcionários por dia deve ser atendida. Utilizou-se tr  processos de heurística de busca local com o intuito de melhorar a solução vigente, a solução consistia na realização de movimentos complexos que consideravam vários movimentos simples de horários para os enfermeiros individualmente. A ideia principal é a recombinação de horários parciais para geração de novos horários. Na segunda fase, os turnos diários são atribuídos aos enfermeiros, e todas as combinações de definição do turno são gerados e analisados, considerando que o dia pode ser de trabalho ou de descanso. Também aplicaram técnicas de otimização local para a busca de combinação de modelos parciais para enfermeiros.

O algoritmo solução foi implementado em Java com a biblioteca *GNU Linear Programming Kit (GLPK)*, que foi aplicada na resolução de todas as instâncias apresentadas. Na competição é proibido o uso de produtos comerciais que requerem licenças de tempo de execução. Também utilizaram uma série de padrões de projetos com o objetivo de organizar melhor o código e melhorar a qualidade do mesmo. A competição dispõem de 3 faixas que diferem em tempo máximo permitido para execução e tamanho da instância do problema, elas são chamadas de *sprint*, *medium* e *long*. Além do conjunto de dados conhecidos disponibilizados, também foram lançados conjuntos de dados ocultos ao longo da competição. 

Os resultados foram dados levando em consideração os conjuntos de dados conhecidos e ocultos. A abordagem dos autores rendeu a equipe a colocação entre os melhores resultados em todas as faixas, estatisticamente falando, conseguiram atingir o melhor resultado conhecido em 70% das instâncias disponibilizadas. O pior resultado esteve a apenas 16 unidades do melhor valor. A abordagem dos autores concederam-lhes os melhores resultados em 6 de 10 instâncias *sprint*, 3 de 5 instâncias *medium* e 4 de cada 5 instâncias *long*, definidas todas elas como ocultas. Os autores atribuem a força da abordagem ao particionamento adequado do problema e utilização de formulações matemáticas para resolver otimamente partes dele e a programação eficiente do restante da aplicação.

Em seu trabalho, Borba (2010) utiliza a meta-heurística *Iterated Local Search (ILS)* com busca local gulosa para a solução do problema de equipes de trabalho de saúde pública. O autor observou a existência da diferença no volume de tarefas entre as equipes de trabalho e a divergência nos tipos de tarefas entre as equipes. O objetivo do trabalho dá-se na busca pelo equilíbrio do volume de tarefas entre as equipes e o máximo de igualdade nas tarefas distribuídas.

Os autores desenvolveram dois algoritmos. O algoritmo inicial buscou contabilizar a quantidade de pessoas existentes na área atendida, e a cada 4.000 pessoas uma equipe de trabalho era designada para essa população. Em um segundo momento, esse algoritmo era responsável por distribuir tarefas às equipes. Os resultados sugerem que essa solução inicial proporcionou melhoras superiores a 7% para Unidades de Pronto Atendimento (UPA), e de 18% para Programa de Saúde da Família (PSF). O segundo algoritmo aplica o movimento de busca local com o Método de Descida Local Gulosa, ocasionando a troca de microáreas assistidas. A procura por uma melhor distribuição de tarefas entre as equipes resulta na facilitação da assistência de microáreas. Como resultado alcançou-se melhorias expressivas e de importância relevante, visto que estes foram os primeiros resultados via meta-heurísticas para esse tipo de problema. Em relação às instituições envolvidas obteve-se a redução de custo operacional devido à equalização das tarefas, evitando assim a sobrecarga dos funcionários em ambas realidades, além da melhoria na satisfação dos funcionários por terem suas preferências atingidas e o aumento da satisfação por parte dos pacientes pela maior cobertura dos profissionais em horários de maior demanda no caso do UPA.

O trabalho de [Jan, Yamamoto e Ohuchi \(2000\)](#) busca investigar as dificuldades que ocorrem durante a solução do *Nurse Scheduling Problem* (NSP) utilizando Algoritmos Evolutivos, em particular Algoritmos Genéticos. Assim como este projeto desenvolvido, o trabalho citado leva em consideração os objetivos do hospital, que são os acordos de trabalho, e as preferências individuais dos enfermeiros. Todas os enfermeiros trabalham nos três turnos vigentes e em alguns feriados. O problema é representado por uma matriz $M \times N$, onde M representa o número máximo de dias a serem agendados e N o número máximo de enfermeiros a serem alocados, tal representação é adotada de forma bem similar neste trabalho. São levadas em consideração restrições fortes e fracas, que são as que devem ser atendidas obrigatoriamente e as que serão atendidas caso seja possível, respectivamente.

O cálculo do *fitness* submete todos os enfermeiros individualmente à uma mesma função. Para o objetivo de maximização é realizado o cálculo da média do *fitness* de todos os enfermeiros, e a minimização consiste na variância entre a avaliação de todos os enfermeiros. A escala com o cálculo atingindo média = 0 e variância = 0, representa que o agendamento em questão é a solução desejada para o problema. [Yamamoto, Kawamura e Ohuchi \(1998 apud JAN; YAMAMOTO; OHUCHI, 2000\)](#) propôs um algoritmo para a solução do NSP usando o feromônio. O feromônio é utilizado para a comunicação entre as formigas, e no trabalho em questão cada enfermeira pode melhorar a sua escala com dois tipos de operadores, o *'swap'* e *'slide'*. A informação do feromônio é usado na resolução de conflitos entre enfermeiros, mas infelizmente a adoção dessa prática nem sempre garante a geração de escalas viáveis. Para a diversificação do espaço de solução foram utilizados *crossover* e mutação, contudo a mutação simples não resultou em melhora no desempenho

de busca do *Cooperative Genetic Algorithms*(CGA), visto que a probabilidade de seleção dos turnos diurnos é alta. Devido a tal comportamento optou-se pelo desenvolvimento do operador de fuga, que tem como estratégia a fuga dos mínimos locais, assim este operador seleciona dois enfermeiros aleatoriamente e realiza a troca de turnos em bloco entre elas. Com o objetivo de investigar os problemas que ocorrem durante a solução do NSP usando a abordagem de computação evolucionária, em particular os algoritmos genéticos, o desenvolvimento e adoção do operador de fuga é a melhor escolha para a melhora do CGA, devido a simplicidade e eficiência do mesmo.

Tal como os trabalhos correlatados, a presente dissertação possui uma descrição sobre o problema abordado(seção 2.4). A descrição do problema, assim como suas características, são as mesma do trabalho de Valouxis et al. (2012), devido a competição.



2.4 Descrição do Problema Abordado

Com intenção inicial de utilização de instâncias de dados reais, foram procuradas algumas instituições em Minas Gerais, como o Hospital João XXIII, situado no bairro Santa Efigênia em Belo Horizonte, o Biocor Instituto, instalado no bairro Vila da Serra em Nova Lima e o Hospital Santa Rita, no bairro Jardim Industrial em Contagem. Contudo, após o contato constatou-se que a realidade encontrada é divergente das encontradas em alguns trabalhos na literatura e do objetivo do trabalho. As instituições procuradas tem seu corpo de enfermagem com jornadas de trabalho consideradas fixas e pré-determinadas, em escala de 12 horas de trabalho por 36 de descanso ou escalas onde o dia é dividido em 3 turnos, onde o noturno adota escala de 12 horas de trabalho por 36 de descanso e o matutino e vespertino empregam a escala de 6 horas diárias. As escalas citadas são em relação à setores específicos das instituições. Todas as instituições citadas como procuradas anteriormente têm seus respectivos funcionários em regime fixo de escala de horário.

Visto que as instituições procuradas adotam as escalas fixas de horário, optou-se pela utilização de instâncias da literatura. Os dados de entrada para esse trabalho são disponibilizadas pela *International Nurse Rostering Competition* (INRC).

2.4.1 *Internacional Nurse Rostering Competition*

A *International Nurse Rostering Competition* busca desenvolver o interesse na área de escala de horários e agendamento. A competição fornece aos pesquisadores modelos de problemas enfrentados que incorporam um aumento do número de restrições do mundo real. A competição tem três objetivos, primeiro gerar novas abordagens para os problemas associados atraindo usuários de todas as áreas de pesquisa. Segundo é fechar o vão entre a pesquisa e a prática na área da pesquisa operacional e o terceiro é estimular o debate dentro da crescente comunidade de pesquisa de escalas de horários e agendamentos. A competição

é organizada e administrada pelo grupo de pesquisa *Combinatorial Optimisation and Decision Support (CODeS)* na *Katholieke Universiteit Leuven* na Bélgica, *Stiftelsen for industriell og teknisk forskning (SINTEF) Group* na Noruega, e na *University of Udine*, na Itália.

O documento de [Haspeslagh et al. \(2010\)](#) fornece informações necessárias para a facilitação da interpretação dos dados de entrada e das regras, apesar da explicação não ser clara o suficiente para o entendimento completo e correto em alguns pontos do documento.

O modelo deve ser capaz de formular soluções para o problema dentro das restrições definidas na competição. O principal objetivo é a geração de um cronograma que atribua tipos de turnos para os enfermeiros dentro de um período de planejamento e capaz de gerar uma solução viável com o menor custo possível.

Os turnos de trabalho são um período predefinido com horário de início e fim que podem se diversificar dependendo da realidade encontrada. Os membros podem se encontrar de folga ou trabalhando nestes turnos. A [Figura 2](#) explicita de forma simplificada os possíveis horários encontrados nas instâncias disponibilizadas.

Figura 2 – Tipos de Turno

	TIPO	INICIO	FIM
L	Late	14:30	22:30
D	Day	08:30	16:30
E	Early	06:30	14:30
N	Night	22:30	06:30
HD	Head nurse	08:30	16:30

Fonte: elaborado pela autor

Os dados de entrada são arquivos de texto que especificam as informações de cada contrato, a quantidade de enfermeiros, tipos de turno entre outras informações. O período de planejamento é de 28 dias, podendo a data de início e fim se diferenciarem de arquivo para arquivo. A matriz, na qual as atribuições de turnos são armazenadas, é chamada de escala. Cada tipo de turno ou folga é atribuído a um enfermeiro em uma determinada data. Encontra-se ilustrada na [Figura 3](#) o significado de atribuição de turnos a funcionários com um exemplo simples.

Observa-se que existem quatro funcionários para um período de planejamento de sete dias, onde cada coluna representa um dia do período de planejamento e os turnos atribuídos são correspondentes aos tipos de alguns turnos disponibilizado na [Figura 2](#). As

Figura 3 – Exemplo de Escala

	DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7
FUNCIONARIO1	E	E	N	-	L	E	E
FUNCIONARIO2	L	D	-	L	N	D	-
FUNCIONARIO3	N	-	D	E	D	-	N
FUNCIONARIO4	-	L	L	N	-	L	L

Fonte: elaborado pela autor

limitações presentes no cenário são chamadas de restrição, que não são nada mais que os contratos e regulamentos de trabalho junto a organização. Esses documentos especificam a condição para trabalho, bem como a existência de horários parciais, turnos noturnos e trabalho aos fins de semana. Este trabalho levou em respeito as restrições impostas pela competição.

2.4.2 Restrições

O problema de escala de horários para enfermeiros é um problema real e que possui restrições impostas que influenciam na construção de um modelo de otimização viável. Existem dois tipos de restrições: **restrições fortes**, que são as restrições que devem ser satisfeitas a todo custo e as **restrições fracas ou suaves**, que são restrições que devem ser satisfeitas caso haja a possibilidade, ou seja, pode violar alguma restrição suave para geração de uma solução satisfatória. As restrições existentes são as definidas pela competição e encontradas na documentação da mesma. O detalhamento do problema, bem como as informações referentes a competição são dados pelo artigo de [Haspeslagh et al. \(2010\)](#). Para que uma solução seja considerada viável, ela não deve violar qualquer uma dessas restrições fortes:

1. Todos os turnos exigidos devem ser atribuídos a um enfermeiro;
2. Um enfermeiro pode trabalhar um turno por dia somente, ou seja, não se pode atribuir a um enfermeiro mais de dois turnos em um único dia;

Como o problema espelha as situações do mundo real, a documentação demanda essas mesmas volumosas situações. As restrições fracas serão atendidas de modo a buscar atender todas, mas se tal cenário não for possível, a violação de algumas ocorrerá provavelmente. Não é desejável a violação destas, pois a quantidade de violações repercutirá no custo final da solução. A lista de restrições fracas são descritas como:

1. Número máximo e mínimo de atribuições para os enfermeiros;
2. Número máximo e mínimo de dias trabalhados consecutivos;
3. Número máximo e mínimo de dias de folga consecutivos;
4. Número máximo de fins de semana trabalhados consecutivamente;
5. Número máximo de fins de semana trabalhados em 4 semanas;
6. Fins de semana completos;
7. Fins de semana com turnos completamente idênticos;
8. Não trabalhar no turno noturno antes de um fim de semana livre;
9. Dois dias de folga depois de um turno noturno;
10. Requisição do enfermeiro para folga ou trabalho em um certo dia: o enfermeiro tem a oportunidade de expressar a sua vontade de trabalhar ou não em uma determinada data;
11. Requisição do enfermeiro para folga ou trabalho em um certo turno em um determinado dia: o enfermeiro tem a oportunidade de expressar a sua vontade de trabalhar ou não em um certo turno em uma determinada data;
12. Alternativa de habilidade: um enfermeiro não pode trabalhar em um turno que ele não tem a habilidade necessária, ou seja, não se pode atribuir ao funcionário um turno noturno se este não tem habilidade para realizar tal tarefa.
13. Padrões indesejados: é uma sequência de atribuições de diferentes tipos de turno para uma enfermeira, como por exemplo, não trabalhar na sexta-feira que antecede um fim de semana livre.

Os padrões indesejados podem ser de dois tipos:

- Padrões indesejados que não envolvam tipos de turno específicos.
- Padrões indesejados que envolvam tipos específicos de turno.

Apesar do documento buscar ser claro em suas informações, existiram explicações que não contribuíram para a interpretação completa dos dados, por conseguinte, as restrições serão detalhadas de acordo com a implementação adotada pelo autor, para que se possa entender como as penalizações foram aplicadas.

Restrições Fortes

$R_1 \leftarrow$ Todos os turnos exigidos devem ser atribuídos a um enfermeiro, ou seja, a demanda

diária deve ser satisfeita obrigatoriamente;

$R_2 \leftarrow$ Um enfermeiro pode trabalhar um único turno por dia.

Restrições Fracas

$Rf_1 \leftarrow$ Número máximo de atribuições para os enfermeiros: a definição da quantidade máxima de dias que devem ser trabalhados por determinado enfermeiro no período de planejamento;

$Rf_2 \leftarrow$ Número mínimo de atribuições para os enfermeiros: a definição da quantidade mínima de dias que devem ser trabalhados por determinado enfermeiro no período de planejamento;

$Rf_3 \leftarrow$ Número máximo de dias trabalhados consecutivos: quantidade máxima de dias trabalhados consecutivos sem atribuição de folgas;

$Rf_4 \leftarrow$ Número mínimo de dias trabalhados consecutivos: quantidade mínima de dias trabalhados consecutivos sem atribuição de folgas;

$Rf_5 \leftarrow$ Número máximo de dias de folga consecutivos;

$Rf_6 \leftarrow$ Número mínimo de dias de folga consecutivos;

$Rf_7 \leftarrow$ Número máximo de fins de semana trabalhados consecutivamente: o fim de semana é considerado completo se em nenhum dia, dentro do período definido como fim de semana, foi atribuído folga ao enfermeiro;

$Rf_8 \leftarrow$ Número máximo de fins de semana trabalhados em 4 semanas: quantidade máxima de fins de semana sem folga em 4 semanas, se no período definido como fim de semana tem ao menos uma folga, o mesmo não é considerado na conta;

$Rf_9 \leftarrow$ Fins de semana completos: a enfermeira tem que trabalhar todos os dias em um final de semana, se existir uma ou mais folgas atribuídas à ela, ocorrerá penalização pela violação da restrição;

$Rf_{10} \leftarrow$ Fins de semana completamente idênticos: o enfermeiro precisa necessariamente trabalhar em turnos idênticos no fim de semana, por exemplo, se ele trabalhar no turno D no sábado, necessariamente necessita trabalhar no domingo no turno D . Se em todos os dias do período informado como fim de semana estiver como folga, também é considerado um final de semana completamente idêntico;

$Rf_{11} \leftarrow$ Não trabalhar no turno noturno antes de um fim de semana livre: é considerado violação se o primeiro dia do fim de semana é uma folga e o dia anterior é um turno noturno;

$Rf_{12} \leftarrow$ Dois dias de folga depois de um turno noturno: o enfermeiro que trabalhou em um turno noturno em um determinado dia, não poderá ter atribuído à ele o turno noturno novamente nos próximos dois dias;

$Rf_{13} \leftarrow$ Requisição do enfermeiro para folga em um certo dia: o enfermeiro tem a oportunidade de expressar a sua vontade de folga em uma determinada data;

$Rf_{14} \leftarrow$ Requisição do enfermeiro para trabalho em um certo dia: o enfermeiro tem a oportunidade de expressar a sua vontade de trabalhar em uma determinada data;

$Rf_{15} \leftarrow$ Requisição do enfermeiro para folga em um certo turno em um determinado dia: o enfermeiro tem a oportunidade de expressar a sua vontade de folga em um certo turno em uma determinada data;

$Rf_{16} \leftarrow$ Requisição do enfermeiro para trabalho em um certo turno em um determinado dia: o enfermeiro tem a oportunidade de expressar a sua vontade de trabalhar em um certo turno em uma determinada data;

$Rf_{17} \leftarrow$ Alternativa de habilidade: Um enfermeiro não pode trabalhar em um turno que ele não tem a habilidade necessária, ou seja, não se pode atribuir ao funcionário um turno noturno se este não tem habilidade para realizar tal tarefa;

$Rf_{18} \leftarrow$ Padrões indesejados: é uma sequência de atribuições ou não, de tipos de turno para uma enfermeira. Os padrões indesejados são os que estão definidos na instância.

Os funcionários são submetidos a avaliação perante as restrições apresentadas. Todas as restrições tem um peso associado, caso a restrição não seja respeitada o peso é aplicado sobre cada violação verificada. O cálculo da função *fitness* é baseado nas penalizações registradas (subseção 2.4.3).

2.4.3 Função Objetiva

O objetivo do problema proposto é encontrar soluções viáveis e/ou factíveis que minimizem o NEP. A heurística é conduzida por uma função de avaliação, chamada de função objetivo, que busca estimar a qualidade dos planejamentos. Um vez que para que uma solução seja considerada factível, ela precise respeitar as restrições fortes, as restrições fracas são as que irão determinar o custo da escala gerada. Nesta seção, é apresentado de forma sucinta e precisa o método desenvolvido para modelar e avaliar as restrições envolvidas no problema. As instâncias do problema são as envolvidas no contrato, bem como o peso da penalização para cada restrição violada.

A forma com que a população foi gerada e é representada, assim como operador de mutação desenvolvido, garantem que as restrições fortes (R_1 e R_2) não são violadas, evitando que a solução se torne infactível e/ou inviável.

O cálculo da função objetiva leva em consideração todo o horizonte de planejamento, ou seja, todas as restrições fracas ($Rf_{1...18}$) são verificadas nos funcionários individualmente para todo o horizonte de agendamento definido. Ao identificar o número de violações V dentro da solução, é aplicado o peso W correspondente, o produto entre esses fatores é o valor da penalização a ser aplicada (autoEquação 2.1). O algoritmo para avaliação é considerado simples.

$$Rf_i = V_i * W_i \quad (2.1)$$

O cálculo do custo (Equação 2.2) é dado pelo somatório de todas as penalizações

calculadas para cada restrição violada. De modo a afirmar a qualidade do algoritmo de avaliação, foi feito o uso do "Avaliador" disponibilizado pela competição. O "Avaliador" é um .jar que ao submeter as informações da escala de acordo com o modelo de arquivo solicitado, o próprio retorna o valor da função *fitness* da solução, tal como onde se encontram as violações. O resultado gerado pelo "Avaliador" da competição foi comparado com o "Avaliador" do trabalho. Como o esperado, a qualidade da avaliação empregada neste trabalho foi afirmada.

$$FO = \sum_{i=1}^{18} Rf_i \quad (2.2)$$

2.5 Considerações Finais

Timetabling Problem é a base para o entendimento do problema a ser estudado no trabalho e ajuda a compreender o interesse ligado ao contexto de alocação de horário para recursos por parte dos pesquisadores. Como explanado, o problema possui diversas áreas que podem e são exploradas por diferentes motivos, como de pesquisa por parte dos estudiosos, aplicação da teoria na prática ou mesmo o encontro de soluções para facilitar a execução da tarefa.

O contexto do problema de agendamento de enfermeiros é de fácil entendimento, visto que é uma situação costumeira para a população que faz uso de tais recursos. É visto que as restrições são numerosas e variadas, podendo resultar, em condições manuais, a impossibilidade de atingir uma solução viável, afinal um outro fator é levado em consideração, o tempo. A revisão bibliográfica descreveu sobre diversas possibilidades de contextos a serem abordados e as infinitas oportunidades de aplicação de estratégias para atingir algum resultado. Foram citados trabalhos que aplicaram Busca Tabu, algoritmos genéticos, matemática inteira, heurísticas de busca local, mas com predomínio da opção pela utilização de estratégias híbridas, que contribuiu para que atingissem resultados satisfatórios.

A competição teve grandes pesquisadores já envolvidos com problemas similares. A riqueza de dados sobre o ecossistema do problema fez com que se tornasse viável a utilização das instâncias disponibilizadas e das restrições definidas, proporcionando assim um ambiente de fácil esclarecimento sobre a forma como a avaliação se deu. Outro fator importante para a escolha dos dados, foi a disponibilização dos menores custos encontrados para cada instância fornecida, podendo assim permitir uma possível comparação com os custos atingidos por este trabalho.

Outro ponto determinante ao resolver adotar uma metodologia para um problema de otimização combinatória qualquer, é a formulação de um modelo de dados para o problema a ser tratado. A utilização de várias heurísticas indica a obtenção da melhor

solução possível ou mesmo ótima. É mais que desejável que o algoritmo seja capaz de gerar uma solução inicial de boa qualidade, para que assim possa atingir resultados considerados satisfatórios.

3 Algoritmo

O algoritmo externaliza de modo prático o estudo realizado até então de forma teórica, visto que as estratégias e heurísticas foram escolhidas e devem ser implementadas para solução do problema. O capítulo intenciona explicar sobre como a implementação afluiu, para que assim facilite a compreensão da lógica implementada e continuação do trabalho posteriormente, uma vez que a estrutura foi desmiuçada para clara compreensão da lógica.

O capítulo estrutura-se de forma que a compreensão do implementado seja o mais claro possível. A [seção 3.2](#) tem por finalidade esclarecer o fluxo de funcionamento do algoritmo. Após a elucidada geral, o meio adotado para a leitura das instâncias é clarificada na [seção 3.3](#). A manipulação dos dados após o armazenamento dá-se início ao processo de manipulação de dados com a Estratégia Evolutiva na [seção 3.4](#), a aplicação do operador de mutação [subseção 3.4.1](#) e seu refinamento com a blusa local elucidado na [seção 3.5](#).

3.1 Visão geral do algoritmo

Para desenvolvimento de um método eficaz para a realidade do problema tratado, é necessário avaliar os processos presentes como o processamento dos dados da instância e suas restrições, o armazenamento das informações e por fim a aplicação das meta-heurísticas escolhidas.

A implementação do algoritmo deu-se por meio da linguagem Java. A escolha da linguagem ocorreu devido a facilidade e experiência da autora com o Java, além da facilidade para a modelagem do problema. O algoritmo utiliza dois tipos de processos, a fim de proporcionar a produção de cronogramas viáveis. Não é utilizado nenhum programa ou biblioteca que auxilie na resolução do problema.

3.2 Fluxograma do algoritmo

O fluxo do algoritmo pode ser visto na [Figura 4](#). Uma explicação sucinta do fluxograma vê-se necessária, para permitir um melhor entendimento e visão sobre a estrutura utilizada. As instâncias disponibilizadas pela competição estão estruturadas em um arquivo, que será processado e lido nesse primeiro momento. Os dados precisam ser armazenados em um objeto que permita a manipulação das variáveis processadas. O algoritmo utiliza os dados do objeto para poder criar uma população inicial e submeter a mesma a heurística aplicada. O processo geral do algoritmo será explicado de forma

separada nas seções seguintes.

Figura 4 – Algoritmo



Fonte: elaborado pela autor

3.3 Leitura de instâncias

A competição disponibiliza instâncias em dois formatos, que são *.txt* e um *.xml*, o algoritmo implementado é capaz de realizar a leitura das instâncias que estejam no formato de texto *.txt*. O [línea 11](#) detalha por meio do pseudocódigo como a instância é lida pelo algoritmo.

As instâncias que foram testadas pertencem a duas faixas da competição, que são *early* e *medium*. O endereço das instâncias bem como as descrições correspondentes, são informados no início do código, onde individualmente ocupam uma determinada posição no *array* de endereços. Os parâmetros são definidos na classe de configuração, tal classe tem o intuito de centralizar todos os parâmetros do algoritmo. Como a instância está definida na classe de configuração, é realizado a abertura do documento para leitura dos dados por meio deste parâmetro. Composta pela definição de suas características, a instância possui informações como o período de planejamento desejado, os dados dos contratos, informações sobre os funcionários, padrões indesejados entre outros informes. As características são definidas como tipos, que ao realizar a leitura dos dados, são destinados a tipagem correta, possibilitando uma melhor organização do código e acesso as informações quando necessárias.

Os tipos identificados são passados como parâmetros para a classe *Problem*. A classe *Problem* é responsável por conter todas as informações da instância do problema, possibilitando que toda manipulação de dados ao longo do algoritmo seja realizada por meio desta classe. Após a estruturação dos dados ser concluída, se inicia a estratégia evolucionária, que é explicada na [seção 3.4](#).

Algorithm 1 LEITURA DA INSTÂNCIA

Data:**Result:**

```

1 begin
2   arquivo ← nome                                ▷ Recebe o nome da instância
   abrirArquivo(arquivo)
   l ← linhaAtual()                               ▷ Lê a linha atual
   tipo1..n ← inicializa()                       ▷ Inicializa os tipos existentes na instância
   for l ≠ null do
3     op ← verifica(l)                            ▷ linha ≠ final do documento
   op ← verifica(l)                               ▷ Verifica as informações da linha
   switch op do
4     case descricaoDoTipo do
5       tipo ← dados                              ▷ Salva características do tipo
6     end
7   end
8   l ← proximaLinha()                            ▷ Lê a próxima linha
9 end
10 problem ← inicializa(tipo1..n)
   fecharArquivo(arquivo)
   estrategiaEvolutiva ← inicializa(problem)
   estrategiaEvolutiva.executa()
11 end

```

Fonte: Algoritmo desenvolvido pela autor

3.4 Estratégia Evolutiva

Estratégias Evolutivas (EE) (ZUBEN, 2000) foram propostas inicialmente com o objetivo de solucionar problemas de otimização de parâmetro, e justamente por empregarem somente operadores de mutação, obtiveram grandes contribuições em relação a síntese e análise destes operadores. O trabalho além de empregar EE, também emprega o *Late Acceptance Hill Climbing* (LAHC) como estratégia de Busca Local, mas isso será particularizado na [seção 3.5](#).

O [línea 31](#) apresenta o pseudocódigo detalhado da estratégia evolutiva utilizada. É dado início a um *time* para que se possa realizar o controle da quantidade de tempo

necessário para concluir a execução do método. A população inicial é construída pela classe responsável por manipular os dados que foram lidos e armazenados na classe *Problem*, como explicado na [seção 3.3](#). A manipulação dos dados consiste em usar as variáveis e parâmetros necessários para construção da solução inicial. Gerada de forma aleatória, a solução ao ser gerada garante que as restrições fortes existentes não serão violadas. A representação do agendamento é dado por uma matriz $F \times P$, sendo que F é o número de funcionários informados pela instância e P o horizonte de planejamento informado, ou seja, a quantidade de dias que compõem o período de agendamento imposto, como na [Figura 5](#).

Ao gerar a população(μ), o primeiro indivíduo tem sua função objetiva calculada, bem como os indivíduos restantes. Todos os indivíduos são armazenados em um *array* que define a população. Ao concluir o processo, a população é ordenada pelo valor da sua função objetiva. Como o objetivo é minimizar o custo, o indivíduo com menor custo será atribuído a melhor solução.

Figura 5 – Representação da Solução

	DIA 1	DIA 2	...	DIA N
FUNCIONARIO1	E	-	...	D
FUNCIONÁRIO2	L	D	...	L
...
FUNCIONÁRIO N	-	N	...	N

Fonte: elaborado pela autor

O critério de parada é a quantidade de gerações ou o limite de tempo definido para execução. Cada indivíduo da população gera λ descendentes, por essa característica ele é considerado o indivíduo pai e transferirá para seus filhos todas as suas características. O descendente será submetido a um operador de mutação, que tem sua implementação explanada na [subseção 3.4.1](#). O indivíduo filho resultante da mutação tem sua função objetivo calculada([subseção 2.4.3](#)) e com o intuito de melhorar a solução, o elemento é sujeito à busca local *Late Acceptance Hill Climbing*, que é elucidada na [seção 3.5](#), e por fim o indivíduo resultante é adicionado em um vetor de descendentes.

Com a finalização da geração da prole é efetuado um *merge* entre os pais, vetor da população e os filhos, o vetor de descendentes. Ocorre a ordenação do vetor resultante, para que possam ocupar as melhores posições os indivíduos com melhor função objetivo. Um corte é realizado para limitar o tamanho do vetor de candidatos ao tamanho da população

estipulada(μ).

Com a atualização da população é necessário verificar se o indivíduo melhor classificado na população tem um resultado de função objetivo inferior ao que é considerada a melhor solução, visto que a função aplicada é a de minimização, se sim, a melhor solução será setada com o indivíduo de menor custo da nova população.

Algorithm 2 ESTRATÉGIA EVOLUCIONÁRIA**Data:** $\rho, \eta, \mu, \lambda, \sigma, v$

```

12     ▷  $\rho$  =problema,  $\eta$  =gerações,  $\mu$  =população,  $\lambda$  =descendentes,  $\sigma$  =funcionários,
     $v$  =período de planejamento
13 begin
14    $cont \leftarrow \emptyset, contGeracao \leftarrow \emptyset$ 
     $timeAtual \leftarrow \emptyset$ 
    while  $cont < \mu$  do
15     |
    |      $solucao \leftarrow \text{contruirSolucao}(\rho)$                                 ▷ Gerar população inicial
    |      $indivíduo \leftarrow \text{inicializa}(solucao, \rho)$                         ▷ Construir matriz de horário
    |      $indivíduo \leftarrow \text{funcaoObjetivo}(solucao)$                         ▷ Avalia indivíduo criado
    |      $populacao.adiciona(indivíduo)$                                        ▷ Adicionar indivíduo na população
    |      $cont ++$ 
16   end
17    $ordena(populacao)$ 
     $melhorSolucao \leftarrow populacao.get(\emptyset)$                                ▷ Recupera o melhor indivíduo
    while  $contGeracao < \eta$  do
18     |
    |      $contPai \leftarrow \emptyset$ 
    |     while  $contPai < \mu$  do
19       |
    |       |
    |       |      $contDescendente \leftarrow \emptyset$ 
    |       |     while  $contDescendente < \lambda$  do
20         |
    |         |
    |         |      $indivíduoFilho \leftarrow populacao[contPai]$ 
    |         |      $mutacao \leftarrow \text{inicializa}(\rho)$                                ▷ Submeter à Mutação
    |         |      $indivíduoFilho \leftarrow \text{mutacao}(indivíduoFilho.solucao())$ 
    |         |      $indivíduoFilho \leftarrow \text{funcaoObjetivo}(indivíduoFilho.solucao)$    ▷ Avaliar
    |         |     descendente
    |         |      $lahc \leftarrow \text{inicializa}(indivíduoFilho.solucao, tamanhoLAHC, \rho)$ ▷ Submeter
    |         |     à Buscal Local
    |         |      $indivíduoLAHC \leftarrow \text{lahc}((indivíduoFilho.solucao()))$ 
    |         |      $indivíduoLAHC \leftarrow \text{funcaoObjetivo}(indivíduoLAHC.solucao)$    ▷ Avalia
    |         |     indivíduo
    |         |      $novaPopulacao.adiciona(indivíduoLAHC)$  ▷ Adicionar indivíduoLAHC
    |         |     na nova população
    |         |      $contDescendente ++$ 
21       |     end
22       |      $contPai ++$ 
    |       |      $contDescendente \leftarrow \emptyset$ 
23     |     end
24      $populacao.addAll \leftarrow novaPopulacao$                                ▷ Realiza merge entre as populações
    |      $ordena(populacao)$                                                        ▷ Avalia população
    |      $populacao.corta(\mu)$ ;
    |     if  $melhorSolucao > populacao.get(\emptyset)$  then
25     | |  $melhorSolucao \leftarrow populacao.get(\emptyset)$                        ▷ Atualiza melhor solução
26     | end
27      $contGeracoes ++$ 
    |     if  $tempoAtual \geq 600000$  then
28     | |
    |     |     ▷ Verifica o limite de tempo break
29     | end
30   end
31 end

```

Fonte: Algoritmo desenvolvido pela autor

3.4.1 Operador de Mutação

Com o objetivo de gerar diversidade e melhora na população, o trabalho empregou o operador de mutação. O operador de mutação (ZUBEN, 2000) altera um ou mais genes de um cromossomo. O indivíduo é submetido a uma probabilidade de ocorrência que chamada de *taxa de mutação*, isto é, ele terá seu gene modificado somente se ele estiver dentro da *taxa de mutação*, contudo, neste contexto foi definido que todos os indivíduos sofrerão mutação.

No caso do problema e representação adotada, a mutação é aplicada sobre os funcionários para aumentar a variabilidade da população. Ao observar a [línea 37](#), é possível notar que a princípio um funcionário é inicializado e um dia aleatório dentro do horizonte de planejamento é sorteado. Isso ocorre para se possa garantir uma das restrições fortes, que é assegurar a demanda mínima exigida de enfermeiros para determinado dia. Exemplificando: se você garante que fará alterações em um determinado dia e que essas alterações se darão com um *swap*, a demanda mínima necessária definida não será comprometida, visto que quantidade de pessoas por turno não é violada. Em seguida são selecionados dois funcionários aleatoriamente por meio da função *sortear()*. Como o dia e funcionários foram determinados, a mutação ocorre por meio da troca/*swap* desses genes, o gene é uma posição na matriz de representação da solução. Explicitando melhor, o turno do *funcionário 1* se tornará o turno do *funcionário 2* e vice e versa. Isso garante que a restrição forte não seja quebrada e a solução não se torne inviável. Tal fluxo acontecerá até que o critério de parada seja atingido, que na operação de mutação é a quantidade de iterações igual a metade do total de funcionários existentes.

Algorithm 3 MUTAÇÃO

Data: *indivíduo*

Result: *indivíduo*

```

32 begin
33   func ← inicializa()
      d ← Random()           ▷ d dia aleatório no horizonte de planejamento
      for cada f ∈ Funcionarios/2 do
34     func[2] ← Sortear(Funcionarios.lenght) ▷ //Retorna dois funcionários aleatórios
        aux ← indivíduo.solucao[d][func[0]]
        indivíduo.solucao[d][func[0]] ← indivíduo.solucao[d][func[1]]
        indivíduo.solucao[d][func[1]] ← aux
35   end
36 end
37 indivíduo

```

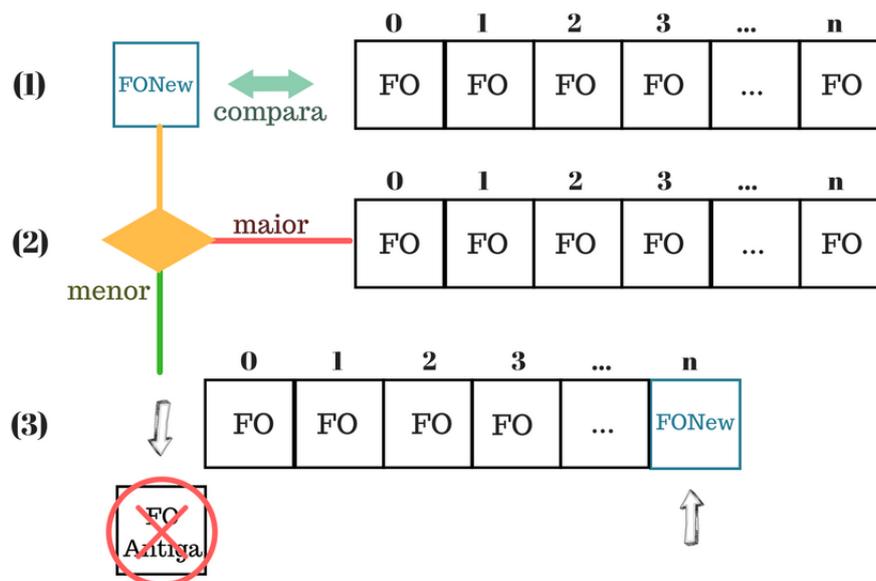
Fonte: Algoritmo desenvolvido pela autor

3.5 Late Acceptance Hill Climbing

A *Late Acceptance* (LA) [Burke e Bykov \(2017\)](#) tem uma ideia bastante simples, o parâmetro de controle na condição é retirado do histórico de soluções. A LA pode ser vista como uma extensão do *Hill Climbing* (HC), com a diferença que esse último a solução candidata é comparada com a solução atual.

O *Late Acceptance Hill Climbing* (LAHC) é a combinação dessas duas heurísticas, nesta heurística o candidato é comparado com uma solução que foi gerada a várias iterações anteriores. Com a ressalva da condição explicitada anteriormente, o algoritmo é estabelecido com uma solução inicial, ou seja, todas as posições do *array* são inicializadas com o menor custo do indivíduo submetido a mutação. Sucessivamente ele aceita ou rejeita a nova solução de maneira iterativa até atingir a condição de parada, bem como no método HC.

Figura 6 – Late Acceptance Hill Climbing



Fonte: elaborado pela autor

O algoritmo verifica se o indivíduo será submetido ao LAHC, observe na [linha 50](#), caso seja, ele inicia um vetor e preenche todas as posições do citado com o valor da função objetiva do indivíduo recebido como parâmetro. O critério de parada é o tempo, e por consequente, até que tal critério seja atingido, o indivíduo temporário recebe o clone do indivíduo recebido por parâmetro e o submete ao operador de mutação, que posteriormente tem seu custo calculado. A regra de aceitação é mantida em uma lista de tamanho fixo que mantém os valores anteriores da função objetiva. Se a função objetivo do indivíduo

resultante for melhor do que o custo armazenado na primeira posição da lista, o candidato é aceito e a lista é atualizada adicionando o novo resultado no final da lista e removendo o elemento que se encontra na primeira posição. Observe que, caso o valor da função objetivo calculada seja rejeitado a lista permanece intacta, pois não ocorre atualização de valores, a [Figura 6](#) ilustra o conceito. A melhor solução tem seus valores atualizados.

Algorithm 4 LATE ACCEPTANCE HILL CLIMBING

Data: *indivíduo, l, ρ*

```

38                                     ▷ l = tamanho do vetor LAHC, ρ = problem
Result: indivíduo
39 begin
40   foBest ← funcaoObjetivo(indivíduo)
      n ← Random()
      if n ≤ txBuscaLocal then
41     vetorLAHC ← inicializa()
      preencheLAHC(l, foBest)
      tempoAtual ← 0
      while timeAtual ≤ timeLimite do
42     indivíduoTemp ← indivíduo.clone()
      indivíduoTemp ← mutacao(indivíduoTemp)
      foNew ← funcaoObjetivo(indivíduoTemp)
      if vetorLAHC[0] > foNew then
43     | vetorLAHC[0].remove()
      | vetorLAHC.adiciona(foNew)
44     end
45     if foNew < foBest then
46     | indivíduo ← indivíduoTemp.clone()
      | foBest ← foNew
47     end
48   end
49 end
50 end

```

Fonte: Algoritmo modificado pela autor

3.6 Considerações Gerais

O Problema de Escalonamento de Enfermeiros foi solucionado com diversas estratégias, meta-heurísticas e técnicas diferentes ao longo do tempo. Pesquisadores sempre estão a procura de alguma solução inovadora que possa trazer resultados capazes de quebrar barreiras chamadas custo mínimo. Um problema corriqueiro mundialmente e de fácil compreensão, mas com soluções que não se limitam a utilização de uma só técnica na maioria das vezes. O trabalho adota também mais de uma técnica na busca por soluções.

O algoritmo estruturou-se de forma a aplicar a Computação Evolucionária atrelada a uma heurística de refinamento, afim de desenvolver uma solução para o problema.

Por conseguinte o trabalho busca descrever a implementação de modo claro, para que auxilie na compreensão da estratégia e a identificação de melhoras em posteriores trabalhos. Com o fluxograma é possível compreender de antemão como ele se organiza e funciona. A explanação da leitura de dados do problema possibilita entender o processo de fluxo de informação que acompanha todo o algoritmo. A Estratégia Evolutiva, o operador de mutação e o *Late Acceptance Hill Climbing* são descritas de forma detalhada pelo pseudocódigo e esclarecimento do mesmo.

Após a conclusão do algoritmo o processo de teste para validação do mesmo tem a necessidade de ser efetuado. Os testes propriamente ditos vem para avaliar os resultados com base em parâmetros que definem a execução. Os parâmetros para serem definidas precisam passar por um processo de parametrização, onde identifica-se os parâmetros capazes de revelar o melhor cenário para realização dos testes com base na heurística desenvolvida. Com a finalização deste os testes podem ser iniciados e os resultados gerados avaliados.

4 Resultados

Esta fase de prosseguimento do trabalho trata da validação da implementação do algoritmo. Sendo a proposta do trabalho a implementação de um algoritmo que emprega Computação evolucionária aplicada ao Problema de Geração de Escala de Enfermeiros, é necessário verificar os resultados obtidos com a heurística desenvolvida.

Com esses experimentos pretende-se verificar se o algoritmo é capaz de cumprir com o objetivo de gerar soluções viáveis e examinar o quão perto ou longe esteve de atingir o custo mínimo encontrado pela literatura para as instâncias utilizadas.

O algoritmo proposto foi implementado utilizando a linguagem Java e os experimentos foram realizados em uma máquina com 4 Processadores Intel(R) Core(TM) i5-2400 cpu@3.10GHz com *cache size* 6144kb, memória 8058672(Total) e sistema operacional Ubuntu Server. As instâncias utilizadas foram obtidas no site da Nurse Rostering Problem Competition [Kuleuven \(2010\)](#). O site disponibiliza várias instâncias que possuem características particulares, pois cada faixa tem dados e tamanho diferentes. As instâncias se encaixam em diferentes categorias, que são elas: *toy*, disponibilizada para iniciantes na compreensão e implementação da estrutura das instâncias e trabalha com o horizonte de planejamento de 7 dias, a *sprint* que tem sua execução limitada a 10 segundos, a *medium* com o limite de 10 minutos e a *long*, que contempla um tempo limite de execução de 10 horas, as últimas 3 incorporam um horizonte mensal de agendamento.

Para explorar a capacidade do algoritmo, vários testes com diferentes parâmetros são realizados. As instâncias utilizadas foram as do conjunto *sprint* e *medium* e *long*. O processo de parametrização assim como a estrutura de teste adotada no trabalho é detalhado na [seção 4.1](#). Após a definição de parâmetros os testes são executados e avaliados na [seção 4.2](#).

4.1 Parâmetros e estrutura de testes

A fim de avaliar a heurística implementada, foram realizados experimentos computacionais com o intuito de relatar os resultados obtidos no trabalho, permitindo que os resultados possam ser comparados com outros experimentos.

Na perspectiva, a experimentação tem por objetivo testar diferentes configurações para geração de amostragem para erros experimentais. Logo, adotou-se os seguintes princípios: a aplicação da definição de valores dos parâmetros na heurística desenvolvida; a aleatorização dos valores definidos, com o propósito de gerar estivas de execução válidas; e a replicação das configurações definidas.

A definição dos valores dos parâmetros foi definido de forma arbitrária. A [Figura 7](#) exibe os valores dos parâmetros. A aleatorização dos experimentos busca eliminar eventuais problemas de seleção e regressão estatística, e neste trabalho foram aleatorizados a execução das instâncias juntamente com a configuração dos parâmetros.

Figura 7 – Parâmetros Definidos

		PARÂMETROS	
NÚMERO DE GERAÇÕES		100	1000
μ		20	50
λ		5	10
TAMANHO DO VETOR LAHC		15	25
TAXA DE BUSCA LOCAL		0.3	0.7
TIMEOUT LAHC		100	1000

Fonte: elaborado pela autora

As instâncias submetidas ao processo de parametrização foram 4, na qual buscou-se representar cada uma das faixas adotadas, que são: *sprint*, *sprint_hidden*, *medium* e *long*. Com a aleatorização, foi possível obter a minimização dos ruídos da tendência dos resultados. A replicação deu-se necessária, afinal é preciso verificar se o desempenho é similar em várias replicações, podendo avaliar a variabilidade do processo e a configuração que resultou no melhor custo.

A quantidade de experimentos de parametrização realizados é representada pela expressão $\kappa * (\sum_i^4 2^6)$. Sendo $\kappa = 10$ a quantidade de replicações a serem aplicadas sobre as 2^6 possibilidades de configuração para cada uma das 4 instâncias selecionadas, temos o total de 2.560 experimentos iniciais realizados para este trabalho. A cada nova replicação a ordem de execução dos experimentos sofre alteração. A prática da aleatorização foi adotada buscando a utilização de recursos computacionais diversificados, levando em consideração que a máquina pode estar usando uma quantidade maior ou menor de recursos em determinados momentos.

Na finalização do processo obteve-se o valor dos parâmetros que constituíam uma configuração com os melhores resultados para as diversas instâncias. A [Figura 8](#) descreve os valores adotados para aplicação da experimentação final da heurística, com base no processo parametrização.

Em sua totalidade foram selecionadas nove instâncias que fazem parte das faixas

Figura 8 – Parâmetros Finais

	PARÂMETROS
NÚMERO DE GERAÇÕES	1000
μ	20
λ	5
TAMANHO DO VETOR LAHC	25
TAXA DE BUSCA LOCAL	0.3
TIMEOUT LAHC	100

Fonte: elaborado pela autora

de *early*, *medium* e *long*, estas são apresentadas na Figura 9. Com apoio nestas instâncias, os resultados são dados na próxima seção.

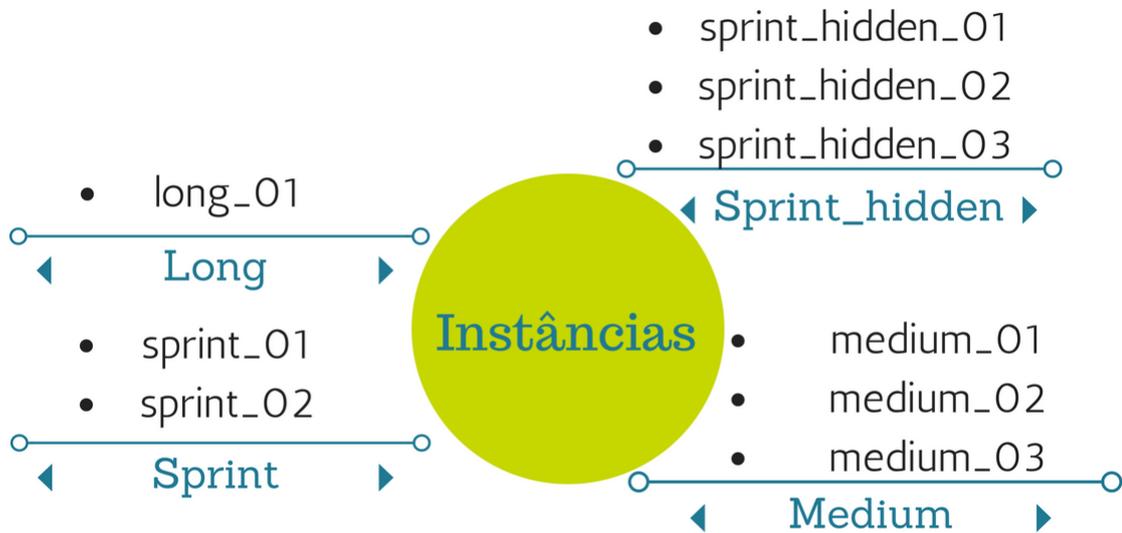
4.2 Análise dos Resultados Computacionais

Os experimentos finais levam aspectos iguais ao dos empregados nos experimentos de parametrização. A quantidade de execuções realizadas é representada pela expressão $\kappa * (\sum_i^9 1)$. Sendo $\kappa = 30$ a quantidade de replicações a serem aplicadas sobre a configuração determinada para cada uma das 9 instâncias selecionadas, temos o total de 270 experimentos realizados para este trabalho. A cada nova replicação a ordem de execução dos experimentos sofre alteração, bem como ocorreu na etapa de parametrização.

Ressalta-se que todas as faixas da competição tiveram instâncias que às representaram. A faixa não inclusa nos testes é a *toy*, pois ela objetiva a iniciação da compreensão da estrutura e funcionamento das instâncias. O tempo de execução da faixa *long* foi limitado à um tempo de execução de 30 minutos, para que assim permitisse que todas as instâncias tivessem uma limitação de tempo de execução igual. Declara-se que o *timeout* não é a única condição de parada de execução, tendo que a quantidade de gerações também está ativa. Devido a escolha de utilização de instâncias disponibilizadas pela competição, optou-se em realizar a comparação dos resultados com os encontrados por um equipe participante.

Bilgin et al. (2010) é uma das equipes competidoras e aplicou uma hiper-heurística de última geração combinada com uma heurística de embaralhamento guloso muito antigo. Tal trabalho é usado para comparação devido a algumas estratégias de estruturação e implementação que se assemelham a empregada nesta dissertação. A escolha da representação da solução inicial dá-se através de uma matriz e é construída aleatoriamente

Figura 9 – Instâncias selecionadas



Fonte: elaborado pela autora

Tabela 1 – GAP(%) Bilgin et al. (2010)/Trabalho

Instância	Bilgin et al. (2010))	Trabalho	GAP(%)
Sprint_01	57	91	37,36
Sprint_02	59	111	46,85
Medium_01	242	402	39,80
Medium_02	241	397	39,29
Medium_03	238	389	38,82
Long_01	197	609	67,65

Fonte: elaborado pela autora

de modo que garanta que a solução seja viável. Sua viabilidade é mantida ao longo da execução, bem como neste trabalho, por meio de movimentos de atribuição e mudança realizadas dentro da mesma coluna. Além das escolhas aleatorizadas de enfermeiros e dias que sofrerão alteração. Ambas as abordagens não aceitam custos piorados após cada iteração.

Após a execução dos testes computacionais para o problema de escalonamento de enfermeiros, obteve-se o resultado da execução do algoritmo. A Tabela 1 e Tabela 2 apresentam a melhor solução encontrada para cada instância. Na Tabela 1 as colunas

Tabela 2 – GAP(%) IRNC/Trabalho

Instância	IRNC	Trabalho	GAP(%)
Sprint_hidden_01	32	115	72,17
Sprint_hidden_02	32	107	70,09
Sprint_hidden_03	62	150	58,67

Fonte: elaborado pela autora

apresentam os melhores resultados obtidos com a abordagem de Bilgin et al. (2010) (LI) e com a abordagem desenvolvida neste trabalho (LT). A Tabela 2 é estrutura da mesma forma, com exceção que os valores da segunda coluna referem-se ao custo mínimo da literatura. Em ambas as tabelas a última coluna reporta o GAP(%) entre os limites de LI e LT obtidos através da Equação 4.1:

$$\frac{LT - LI}{LT} \times 100 \quad (4.1)$$

A abordagem conseguiu garantir que todas as soluções fossem consideradas factíveis. O cálculo do GAP, que é o desvio de percentual dos valores, representa em porcentagem o quão pior a solução proposta é em relação ao atendimento das restrições fracas encontrada no trabalho de Bilgin et al. (2010), que conseguiu atingir o melhor resultado em algumas instâncias, mesmo quando este não foi possível os valores não estiveram muito distantes.

O GAP demonstra que a solução desenvolvida não foi eficiente ao buscar atingir resultados próximos aos do trabalho citado. Estes estiveram em uma média GAP de 40% revelando a desproporção da solução proposta em relação a LI no atendimento das restrições fracas. O cálculo da média informada contemplou somente as instâncias *Sprint_01*, *Sprint_2*, *Medium_01*, *Medium_02*, *Medium_03*, pois as *Sprint_hidden_01*, *Sprint_hidden_02*, *Sprint_hidden_03* não tem seus valores expressos no documento. As instâncias citadas como não contempladas tiveram o GAP calculado levando em consideração os resultados disponibilizados pela IRNC e estão expressas na Tabela 2. A instância *Long_01* não foi levada em deferência, em razão da mesma possuir uma particularidade resultante do critério de parada *timeout*.

Com o intuito de medir a dispersão de valores resultantes da abordagem em relação a média do valor final das replicações, é apresentado na Figura 10 e Figura 11 a média dos valores e o Desvio Padrão(DP) de cada problema. A média é determinada pela linha vermelha e o Desvio Padrão pela linha azul. Os resultados em cada replicação tiveram valores de função objetivo que não se distanciavam muito entre si. A exceção se deu na instância *Sprint_01* no qual os valores finais das replicações ocorreram em um intervalo maior que o encontrado nos outros problemas. A média e o DP externam que em todas as instâncias o menor custo não estava incluso dentro do intervalo do DP.

Figura 10 – Média e Desvio Padrão do Limite de Custo

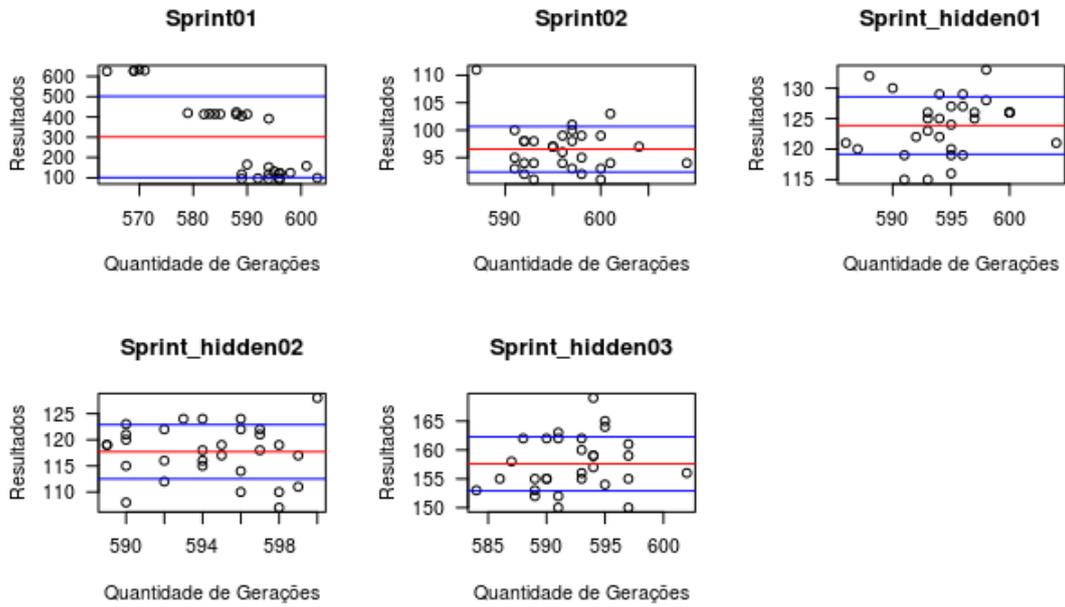
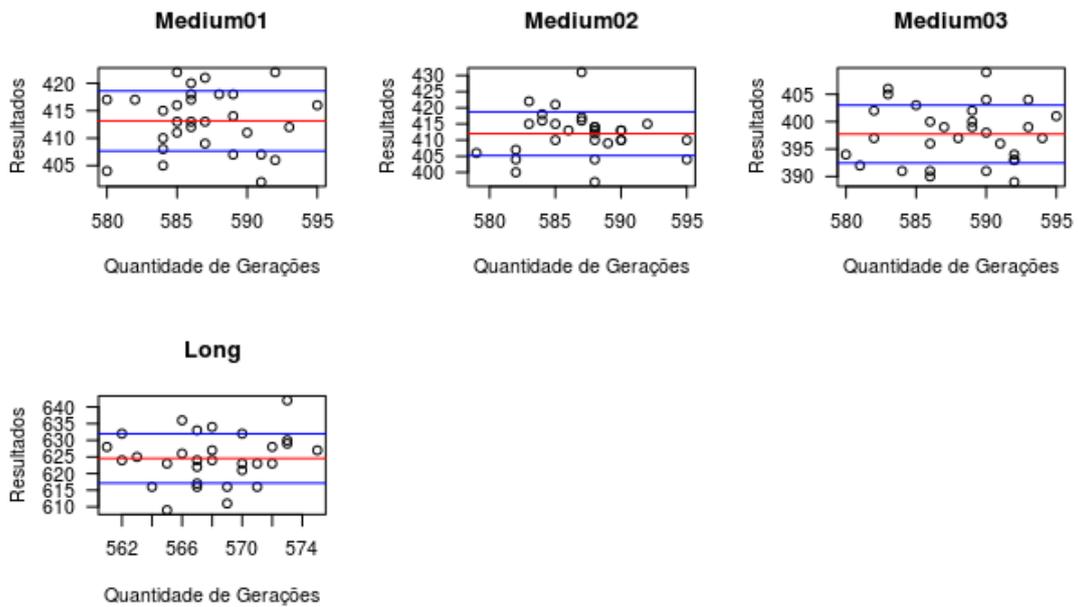


Figura 11 – Média e Desvio Padrão do Limite de Custo



5 Conclusões e Trabalhos Futuros

Este trabalho apresentou Técnicas de Computação Evolucionária para o Problema de Escalonamento de Enfermeiros. Ainda que neste cenário existam vários trabalhos e resultados publicados, em que cada um se distingue por suas características individuais e diversas formas de aplicação de heurísticas, acredita-se que para uma avaliação de eficácia das heurísticas empregadas, a utilização de uma instância com valores de seus custos mínimos disponibilizados proporciona uma comparação e avaliação mais sensível. A abordagem da dissertação objetiva fechar o vão entre a pesquisa e a prática, logo a busca por uma melhor e mais profunda compreensão da área de otimização dá-se através da aplicação da pesquisa teórica. Com isso a proposta que utiliza Técnicas de Computação Evolucionária baseadas em Estratégia Evolutiva e emprega a meta-heurística *Late Acceptance Hill Climbing* como estratégia de refinamento para a solução, busca avaliar se uma abordagem simplificada é capaz de atingir resultados satisfatórios.

A proposta construída foi comparada com a proposta adotada por uma das equipes competidoras da IRNC, esta é constituída por grandes pesquisadores envolvidos na área de otimização a vários anos. A heurística proposta construída sobre a formulação apresentada, garante o cálculo correto da solução, uma vez que foi realizada a verificação do cálculo da função objetivo com o uso do "Avaliador". O "Avaliador" da competição permitiu a detecção de erros que posteriormente foram corrigidos garantindo a avaliação. Os experimentos realizados revelam que nenhuma solução foi considerada ineficaz devido a estrutura de construção e manipulação da solução adotada, mas os limites encontrados não atingiram a otimalidade e não se posicionaram próximos a estes limites. A escolha pela limitação de critério de paradas iguais para todos as instâncias permitiu observar que a *Long_01* obteve um resultado pouco satisfatório, uma vez que o *timeout* impede, como imaginado inicialmente, que o mesmo atenda as restrições fracas uma vez que a quantidade de funcionários é relativamente mais alta.

Apesar da abordagem empregada ser híbrida, esta demonstrou ainda ser tecnicamente fraca na solução dos problemas apresentados devido a escolha de uma heurística híbrida simplificada, e a alternativa de não utilização de softwares de terceiros para auxiliar na resolução.

Como trabalhos futuros, pretende-se:

- armazenar as informações de custo dos funcionários individualmente, para reduzir custo de processamento, uma vez que o cálculo do custo só será realizado no indivíduo que sofreu modificação;

- propor a execução paralela de tarefas;
- estudar a incorporação de outras abordagens ao modelo apresentado, com o propósito de gerar resultados mais próximos da otimalidade.

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. Citado 4 vezes nas páginas 12, 47, 48 e 49.
- BÔAVENTURA, R. S.; PINTO, B. Q.; YAMANAKA, K. Utilizando técnicas de algoritmo genético para resolução do problema de geração de grade horária para enfermarias. Conferência IADIS Ibero-Americana Computação Aplicada, 2013. Citado 2 vezes nas páginas 17 e 24.
- BORBA, N. A. d. S. *Uma Solução do Problema de Programação de Equipes de Saúde Pública Via Metaheurísticas*. Tese (Doutorado) — Centro Educacional de Educação Tecnológica de Minas Gerais, 2010. Citado na página 25.
- BURKE, E. et al. Variable neighborhood search for nurse rostering problems. In: *Metaheuristics: computer decision-making*. [S.l.]: Springer, 2003. p. 153–172. Citado na página 22.
- BURKE, E. K.; BYKOV, Y. The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, Elsevier, v. 258, n. 1, p. 70–78, 2017. Citado na página 42.
- CAUSMAECKER, P. D.; BERGHE, G. V. Novel meta-heuristic approaches to nurse rostering problems in belgian hospitals. 2004. Citado na página 23.
- CONSTANTINO, A. A. et al. Um algoritmo heurístico para otimização do problema de escalonamento de enfermeiros. 2009. Citado na página 23.
- COREN. *Coren*. 2011. Disponível em: <<http://www.coren-df.gov.br/site/no-0052011/>>. Citado na página 23.
- DOWSLAND, K. A. Nurse scheduling with tabu search and strategic oscillation. *European journal of operational research*, Elsevier, v. 106, n. 2-3, p. 393–407, 1998. Citado na página 17.
- ECONOMIAS. *Economias*. 2016. Disponível em: <<https://www.economias.pt/trabalho-por-turnos/>>. Citado na página 22.
- FOURATI, Z.; JERBI, B.; KAMMOUN, H. Planning and modeling nurse timetabling at an intensive care unit in a tunisian university hospital. 2016. Citado na página 21.
- FURTADO, J. C.; VARREIRA, L. D. A. Otimização de escala de horários de equipes de enfermagem através da simulação discreta em hospital do vale do rio pardo. *Anais do Salão de Ensino e de Extensão*, p. 377, 2013. Citado na página 22.
- GOMES, R. A. M. Técnicas de programação inteira para o problema de escalonamento de enfermeiras. Programa de Pós-Graduação em Ciência da Computação. Departamento de Ciência da Computação, Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto., 2012. Citado na página 21.

- GOODMAN, M. D.; DOWSLAND, K. A.; THOMPSON, J. M. A grasp-knapsack hybrid for a nurse-scheduling problem. *Journal of Heuristics*, Springer, v. 15, n. 4, p. 351–379, 2009. Citado na página 17.
- 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. Citado 2 vezes nas páginas 28 e 29.
- JAN, A.; YAMAMOTO, M.; OHUCHI, A. Evolutionary algorithms for nurse scheduling problem. In: IEEE. *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. [S.l.], 2000. v. 1, p. 196–203. Citado 2 vezes nas páginas 17 e 26.
- KRIPKA, R.; KRIPKA, M.; SILVA, M. d. Formulação para o problema de alocação de salas de aula com minimização de deslocamentos." *XLIII Simpósio Brasileiro de Pesquisa Operacional*", Ubatuba, SP. *Anais do XLIII SBPO*, 2011. Citado na página 21.
- KULEUVEN. *Nurse Rostering Competition*. 2010. Disponível em: <<https://www.kuleuven-kulak.be/nrpcompetition>>. Citado 2 vezes nas páginas 21 e 45.
- MÜLLER, T.; BARTÁK, R. Interactive timetabling: Concepts, techniques, and practical results. In: *PATAT*. [S.l.: s.n.], 2002. p. 58–72. Citado na página 21.
- PETROVIC, S.; BERGHE, G. V. Comparison of algorithms for nurse rostering problems. In: *Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling*. [S.l.: s.n.], 2008. p. 1–18. Citado na página 17.
- POLTOSI, M. R. Elaboração de escalas de trabalho de técnicos de enfermagem com busca tabu e algoritmos genéticos. 2007. Citado na página 18.
- POLTOSI, M. R. Elaboração de escalas de trabalho de técnicos de enfermagem com busca tabu e algoritmos genéticos. Universidade do Vale do Rio do Sinos, 2007. Citado na página 22.
- RIBEIRO, V. G. et al. Simulated annealing e sistema multiagente na abordagem de timetabling aplicada ao escalonamento de veículos de transporte público. *Revista de Sistemas e Computação-RSC*, v. 4, n. 1, 2014. Citado na página 21.
- SANTOS, H. G. et al. Heurísticas e programação inteira para o problema das enfermeiras. 2012. Citado na página 17.
- SANTOS, I. C. *Problema do Escalonamento de Enfermeiros*. [S.l.], 2006. Citado na página 18.
- VALOUXIS, C. et al. A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research*, Elsevier, v. 219, n. 2, p. 425–433, 2012. Citado 3 vezes nas páginas 17, 24 e 27.
- YAMAMOTO, M.; KAWAMURA, H.; OHUCHI, A. Collective approach to optimization problems. In: *Proceedings of ITC-CSCC*. [S.l.: s.n.], 1998. v. 98, p. 1479–1482. Citado na página 26.

ZUBEN, F. J. V. Computação evolutiva: uma abordagem pragmática. *Tutorial: Notas de Aula da disciplina IA707, Faculdade de Engenharia Elétrica e de Computação-Universidade Estadual de Campinas*, 2000. Citado 2 vezes nas páginas 37 e 41.

Apêndices

Anexos