

Sistema de Apoio à Decisão para horários de enfermeiros em Unidades de Cuidados Continuados

José Coelho Rodrigues

*Dissertação submetida para satisfação parcial dos requisitos do grau de
Mestre em Engenharia Electrotécnica e de Computadores*
MIEEC

Orientada por:

Prof. José Fernando Oliveira

Prof^a. Maria Antónia Carravilla

(Presidente do Júri)



Universidade do Porto
Faculdade de Engenharia
FEUP

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e de Computadores
Rua Roberto Frias, sn, 4200-465, Porto, Portugal

31 de Março de 2008

Resumo

A elaboração de horários para os enfermeiros de uma Unidade de Cuidados Continuados (UCC) é um problema de escalonamento complexo. É preciso cobrir necessidades diárias, sem interrupções, respeitando restrições monetárias, legais, de preferências e de regras da instituição em que se aplica. O bem-estar dos enfermeiros colaboradores da instituição é essencial para a boa qualidade do serviço que fornecem, e encontrar o equilíbrio entre todas as restrições que constituem o problema é uma tarefa difícil. Assim, é muito importante a existência de um Sistema de Apoio à Decisão (SAD).

O problema abordado é o de escalonamento de enfermeiros numa UCC. Neste tipo de instituições o contacto entre o doente e os enfermeiros que lhe dão assistência é muito valorizado, para permitir o desenvolvimento de uma relação de confiança entre enfermeiros e pacientes. Assim, além de se resolver um problema de *Nurse Rostering* vulgar, trabalhou-se um problema valorizando a manutenção do contacto entre os enfermeiros e os doentes. O objectivo é o de cobrir todas as necessidades diárias, por turno e grau de competência dos enfermeiros, minimizando o desrespeito pelas preferências de alocações dos enfermeiros, de contacto entre enfermeiros, dos enfermeiros com os doentes, assim como das restantes regras institucionais e legais.

Desenvolveu-se um *Algoritmo Genético*, com o qual se testaram vários cenários possíveis, criados com base em dados recolhidos numa UCC, e alcançaram-se resultados muito interessantes, quer em termos de qualidade da solução obtida, quer em termos de tempo de pesquisa. Foram também obtidos resultados óptimos a partir de um modelo em programação linear desenvolvido para o problema. Estes últimos permitiram obter um melhor conhecimento do funcionamento e da influência das diferentes restrições no processo de pesquisa, nomeadamente na solução obtida.

Abstract

Making timetables for nurses who work in an institution that provides continuous health care services is a hard task to be done. The daily demands must be covered and the budget and legal constraints must be satisfied. The well-being of the nurses working in the institution is crucial for the quality of the services they provide. Finding the balance between all the constraints of the problem to achieve a good work environment is an extremely hard task, for which the existence of a Decision Support System is of high importance.

The problem dealt with on this study is the scheduling of nurses in an institution like the above described. In that kind of institutions, the contact between nurses and their patients is an important value to allow the development of a good and trustworthy relation between both (nurse and his patients). Therefore we do not only solve a typical *Nurse Rostering* problem, but we also add this new constraint that values the contact between nurses and their patients. The aim is to cover all the daily demands, for each shift of the day and each nurse grade, minimizing the disrespect of nurse preferences for shift and days allocations, of contact between nurses and their patients and all the other soft constraints associated with the problem.

To solve the problem we created a Genetic Algorithm. It was tested with some instances created based on the knowledge and data obtained from one institution of continuous health care services. Some interesting results were achieved concerning solution quality and time taken to conclude each search. Some optimal results were also obtained to some instances, with a mathematical model also developed to this problem. They were important to study the dynamics of the constraints with the search process and the solution obtained, before creating the Genetic Algorithm.

Agradecimentos

Desde início que estabeleci o que pretendia com este trabalho: adquirir um aprofundado grau de conhecimento do problema trabalhado, esforçando-me por maximizar o meu desempenho, assumindo como objectivo principal a satisfação pessoal, apenas atingida depois de usar e desenvolver todas minhas capacidades.

Considero esse objectivo alcançado e devo-o ao auxílio e apoio que me foi prestado. Desejo por isso exprimir a minha gratidão aos meus orientadores, Professor José António Oliveira e Professora Maria Antónia Carravilla pela dedicação e entusiasmo que desde o primeiro contacto expressaram e que foi fulcral para me manter motivado. Agradeço também os ensinamentos e orientações em matérias organizacionais e metódicas que este trabalho exigiu.

Agradeço ao Professor António Miguel Gomes a dedicação demonstrada em auxílios decisivos. Ao Engenheiro Joel Pacheco, à Doutora Teresa Costa e ao Professor João Claro pela amizade e pelo apoio em momentos oportunos.

Um agradecimento especial à minha família: ao meu pai, pelas sugestões para a redacção e por todo o apoio e incentivo que me deu ao longo destes meses; à minha mãe, pelo interesse demonstrado, pela experiência transmitida e por todo o apoio sempre expresso (ainda que em momentos menos fáceis da vida); ao meu irmão, pela compreensão e pela curiosidade em entender o estudo.

Conteúdo

1	Introdução	1
2	Problema	5
2.1	Introdução	5
2.2	Problema do Projecto Chronos	7
2.2.1	<i>Carlton Life</i>	7
2.2.2	Funcionamento das <i>Unidades de Negócio</i>	8
2.2.3	Requisitos	9
2.3	Problema de <i>Nurse Rostering</i>	13
2.3.1	Do Problema do Projecto Chronos ao Problema para a Dissertação	13
2.3.2	O Problema	14
2.3.3	Pressupostos Considerados	18
3	Revisão Bibliográfica	21
3.1	Problemas de Escalonamento em Investigação Operacional . .	21
3.2	Investigação Operacional Aplicada à Área da Saúde	22
3.3	<i>Nurse Rostering</i>	23
4	Modelo Proposto	41
4.1	Modelo de <i>Nurse Rostering</i>	41
4.1.1	Índices	41
4.1.2	Parâmetros	41
4.1.3	Variáveis	43
4.1.4	Modelização	44
4.2	Implementação	49
4.2.1	Criação de Cenário e Leitura da Solução	49
4.2.2	Criação de Instâncias do Modelo	51
4.3	Testes aos Modelos Matemáticos	58
4.3.1	Instâncias Adaptadas de [24]	58
4.3.2	Cenários Criados	62

4.3.3	Apresentação e Análise dos Resultados Relativos ao Modelo Grande	76
4.3.4	Observações aos Resultados Obtidos	103
5	Introdução aos <i>Algoritmos Genéticos</i>	107
5.1	Introdução	107
5.2	<i>Algoritmo Genético</i> – Uma Visão Inicial	108
5.3	Anatomia dos <i>Algoritmos Genéticos</i>	110
5.3.1	Função Avaliadora	110
5.3.2	População	110
5.3.3	Reprodução	113
5.4	Técnicas de Codificação	113
5.5	Técnicas de Selecção dos Pais	115
5.5.1	<i>Roulette Wheel Parent Selection</i>	116
5.5.2	<i>Rank Parent Selection</i>	117
5.5.3	<i>Tournament</i>	119
5.6	<i>Crossover</i>	119
5.6.1	<i>One-Point Crossover</i>	119
5.6.2	<i>Two-Point Crossover</i>	119
5.6.3	<i>Uniform Crossover</i>	120
5.6.4	<i>Crossover Aritmético</i>	121
5.6.5	Observações	121
5.7	Mutação	124
5.7.1	Exemplo de Aplicação	125
5.8	Observações	125
5.8.1	Interpolação	125
5.8.2	Ajustes e Afinação	126
6	Desenvolvimento do <i>Algoritmo Genético</i>	129
6.1	Introdução	129
6.2	Implementação do <i>Algoritmo Genético</i>	131
6.2.1	Codificação das Soluções	131
6.2.2	Criação da População Inicial	132
6.2.3	Técnica de Selecção de Cromossomas	138
6.2.4	Técnicas de Reprodução	139
6.2.5	Sobrevivência	140
6.2.6	O <i>Algoritmo Genético</i> Inicial	141
6.3	“Soft Constraints”	149
6.3.1	Fins-de-Semana	150
6.3.2	Mínimos de Folgas entre Alocações Consecutivas	155
6.3.3	Máximo de Alocações no Horizonte Temporal	159

6.3.4	Máximo e Mínimo de Alocações Consecutivas	160
6.3.5	Máximo e Mínimo de Dias Livres Consecutivos	162
6.3.6	Equilíbrio entre Escalonamentos Individuais	164
6.3.7	Preferências de Alocações	166
6.3.8	Contactos	168
6.4	O <i>Algoritmo Genético</i> Final	174
6.4.1	Funcionamento	174
7	Conclusão	183
A	Glossário	189
B	Ficheiro de Criação dos Cenários de Teste	191
C	Resultados Obtidos com o “Modelo Grande”	201
C.1	Escalonamentos para o “Cenário 1”	202
C.2	Escalonamentos para o “Cenário 4”	213
C.3	Escalonamentos para Cenários com Diferentes Necessidades . .	217
C.4	Dias de Trabalho	220
D	Anexos à Implementação do <i>Algoritmo Genético</i>	221
D.1	Novas Folhas no Ficheiro de Criação dos Cenários de Teste . .	222
D.2	Um Escalonamento para o <i>Algoritmo Genético</i>	227
D.3	Populações Iniciais	229
D.4	Penalizações Parciais de Fins-de-Semana	232

Capítulo 1

Introdução

Este trabalho centrou-se na definição de um problema de *Nurse Rostering* e na sua resolução, recorrendo à elaboração e implementação, primeiro, de um modelo matemático, que permitiu consolidar o conhecimento do funcionamento do problema, e, posteriormente, de uma heurística, um *Algoritmo Genético*.

O problema identificado partiu de um projecto anterior, o Projecto Chronos, desenvolvido em parceria com uma empresa que está a iniciar a sua actividade na prestação de Cuidados Continuados. Nesse projecto foram recolhidos os requisitos que permitiram definir um sistema de apoio à elaboração de horários para os colaboradores da empresa. Dos requisitos recolhidos, que se assemelham aos requisitos de um problema de *Nurse Rostering*, evidenciou-se a preocupação, por parte da empresa, em garantir um contacto humanizado entre os seus colaboradores e clientes (que serão pessoas que, na sua maioria, residirão na instituição por um longo período de tempo). Isto é, que cada cliente contacte, tanto quanto possível, com os mesmos colaboradores da empresa, de modo a poder estabelecer uma relação de confiança.

Servindo o trabalho aqui apresentado como caso de estudo para elaboração de uma dissertação de mestrado, interessou que o tratamento do problema em causa não se restringisse ao tempo de resposta da empresa cliente, decidindo-se assim abordar um problema que servisse instituições idênticas à empresa de Cuidados Continuados, mas que não fosse tão particularizado a ela. Deste modo, baseado na literatura visando trabalhos realizados na área de *Nurse Rostering*, definiu-se um problema de *Nurse Rostering* como os demais, mas incluindo a restrição de contacto entre enfermeiros e doentes. Tratando-se, portanto, de um problema de escalonamento de enfermeiros, apropriado a instituições com enfermarias onde os doentes são residentes ou residem temporariamente no local em que lhes são prestados os cuidados.

O problema de *Nurse Rostering* é de um problema de escalonamento complexo, que tem merecido atenção de vários praticantes da Investigação Operacional, já desde há vários anos. Um sistema de apoio à elaboração de horários numa enfermaria é muito importante, por se tratar de uma tarefa árdua, que consome muito tempo a quem a realiza, mesmo que tenha experiência destacável, como acontece habitualmente aos enfermeiros a quem ela é confiada. Uma enfermaria precisa de cobrir requisitos sem interrupções diárias, com restrições monetárias, legais, de preferências dos enfermeiros e de regras de práticas da instituição em questão. O bem-estar dos enfermeiros é essencial para a boa qualidade do serviço que fornecem. Encontrar o equilíbrio entre todas as restrições que constituem o problema é, certamente, uma tarefa muito difícil.

Num problema de *Nurse Rostering* o horizonte temporal do escalonamento pretendido varia. Procura-se garantir que todas as necessidades de cobertura sejam satisfeitas, minimizando o incumprimento das restantes restrições do problema. Os enfermeiros são normalmente divididos em diferentes graus de competências e podem também ser considerados alguns enfermeiros em part-time ou contratados para prestar serviços isolados e pontuais. Os dias são divididos em turnos ou blocos de trabalho. A dimensão da enfermaria habitualmente é de 20 a 30 enfermeiros, não sendo comum referir o número de doentes, uma vez que se trata de enfermarias hospitalares, em que muitos dos pacientes não estão acamados (não residem na enfermaria), e não de enfermarias de instituições como as abordadas neste trabalho.

O problema definido consta de uma confluência entre o caso mais geral, com as características identificadas durante a revisão bibliográfica, e o caso do Projecto Chronos. Trata-se de um problema de escalonamento com um horizonte temporal mensal, com três turnos diários (manhã, tarde e noite), com necessidades semanais (especificação deste tipo de instituições, identificada na empresa de Cuidados Continuados) que terão de ser cobertas a qualquer custo e que se definem por turno e por grau de competência dos enfermeiros, e com três graus de competências de enfermeiros (chefe, "regular" e estagiário).

Na primeira abordagem realizada ao problema recorreu-se a um modelo matemático, em programação inteira mista (MIP). Este modelo foi testado em duas versões, uma reduzida às restrições obrigatórias ("hard constraints"), para a qual a função objectivo apenas contempla custos de alocações, e outra em que se consideram tanto as "hard" como as "soft constraints" do problema.

Para testar o modelo recorreu-se a instâncias criadas com dados provenientes do Projecto Chronos e da literatura, e a instâncias adaptadas de

uma biblioteca de problemas de Investigação Operacional online, com acesso a cenários criados para várias situações tipicamente abordadas em Investigação Operacional.

Desses testes aos diversos cenários, que incluíram variações dos parâmetros do problema, consolidou-se o conhecimento sobre o funcionamento do problema. Esse conhecimento permitiu uma melhor e mais fundamentada elaboração da abordagem heurística, assim como uma melhor concepção e análise dos testes que lhe foram realizados.

A heurística utilizada para resolução rápida, com obtenção de uma boa solução, foi um *Algoritmo Genético*. Antes de o implementar, houve necessidade de conhecer o funcionamento deste tipo de heurísticas. Foi então realizado um estudo aprofundado sobre o modo de funcionamento de um *Algoritmo Genético* e sobre as precauções a ter em consideração durante a sua concepção. Daí resultou um documento que introduz brevemente os *Algoritmos Genéticos*, incluído nesta dissertação.

Finalmente, foi desenvolvido o *Algoritmo Genético* adaptado ao problema, implementado e testado, recorrendo a uma das instâncias previamente definidas, bem como a uma sua versão alargada, por razões devidamente fundamentadas.

A dissertação está dividida em sete capítulos.

Após esta introdução, no segundo capítulo, é exposto o problema. Para sua contextualização, é também apresentado o problema identificado no Projecto Chronos e estabelecida a ligação entre esse projecto e o trabalho aqui desenvolvido.

Segue-se o capítulo que regista uma revisão bibliográfica de estudos na área de *Nurse Rostering*. Este capítulo inicia-se com breves introduções aos problemas de escalonamento abordados em Investigação Operacional e aos problemas relacionados com cuidados de saúde de que a Investigação Operacional se ocupa. São, seguidamente, descritos os trabalhos que mais relevância tiveram para a familiarização do autor com o problema de *Nurse Rostering* e para a elaboração de todo o estudo realizado.

O início do quarto capítulo apresenta duas versões do modelo matemático proposto para resolver o problema, como dois modelos separados. Em seguida é explicado como se criam as instâncias de teste e como são utilizadas para criar modelos matemáticos adaptados que as resolvam. Por fim, são apresentados os vários cenários e variações de cenários para teste dos modelos, e analisados os seus resultados, com vista à compreensão do funcionamento das diversas restrições e da influência de suas variações no escalonamento obtido.

No quinto capítulo é introduzido o funcionamento dos *Algoritmos Genéticos*. Nele está reunida toda a informação necessária para a criação de um *Algoritmo Genético*, orientada para problemas de *Nurse Rostering*, com recurso a ilustrações e exemplificações.

No penúltimo capítulo enuncia-se todo o processo de criação, implementação e teste do *Algoritmo Genético* que resolve o problema aqui abordado. A descrição dessas tarefas está faseada, tal como o foi durante a sua realização. Os testes realizados são analisados e recomenda-se um conjunto de parâmetros que definem o melhor funcionamento do *Algoritmo Genético*.

Por fim, uma conclusão analisa todo o trabalho realizado. Seguem-se capítulos anexos com informação adicional, nomeadamente dados que não foi necessário expor ao longo da dissertação, mas que podem ser de interesse de consulta para o leitor.

Capítulo 2

Problema

2.1 Introdução

O problema abordado nesta dissertação surgiu na sequência de um projecto para a elaboração de um “Sistema de Apoio à Elaboração de Horários em *Unidades de Cuidados Continuados*”, encomendado pela empresa *Carlton Life*. Ambicionava-se projectar um sistema que permitisse calcular de forma rápida e eficaz uma boa distribuição das tarefas a prestar aos *Clientes*, pelos *Cuidadores*¹ da empresa. Este sistema deveria respeitar as regras de trabalho da instituição, os contratos de trabalho celebrados com os seus colaboradores e os valores da empresa.

Ao longo do Projecto Chronos, projecto que deu início à elaboração desse sistema, pretendeu-se recolher todos os requisitos que permitissem definir o problema de decisão, propor e testar um modelo matemático que resolvesse o problema identificado. Concluído esse projecto e a familiarização com o problema, aspirava ao desenvolvimento um método que se revelasse mais célere do que o modelo matemático na resolução do problema e pudesse ser implementado num sistema, para ser utilizado na *Carlton Life*.

Os requisitos recolhidos foram agrupados num documento, apresentado na primeira secção deste capítulo, que se inicia com uma breve descrição dos modelos organizativo e de funcionamento da *Carlton Life*, de modo a contextualizar o sistema pretendido. Em seguida são descritos, pormenorizadamente, os requisitos recolhidos. Os requisitos apresentados são os da primeira fase desse trabalho. Mais tarde, devido à proximidade de entrada em funcionamento da empresa e consequentes ajustes na sua organização, sofreram alterações que serviram de ponto de partida para o trabalho exposto

¹Este e outros termos importantes na descrição do problema da *Carlton Life* estão definidos no apêndice A.

2.1. INTRODUÇÃO

nesta dissertação. Essas alterações e o modo como se alcançou o problema de *Nurse Rostering* estão pormenorizados na última parte da primeira secção.

Na segunda secção deste capítulo caracteriza-se o problema de *Nurse Rostering* abordado na dissertação.

2.2 Problema do Projecto Chronos

A primeira parte desta secção introduz brevemente o negócio da *Carlton Life*. A organização genérica das *Unidades de Negócio* é apresentada na segunda parte. Na última descreve-se as necessidades e restrições associadas ao problema identificado na *Carlton Life*.

2.2.1 *Carlton Life*

A *Carlton Life* é uma holding que opera na área de negócio dos cuidados continuados e que é constituída por várias *Unidades de Negócio*, onde são prestados os serviços de Cuidados Continuados.

Áreas de Actuação em *Cuidados Continuados*

As *Unidades de Negócio* prestarão serviços de apoio à perda de capacidades dos *Clientes*. Os Cuidados Continuados podem ser dos seguintes tipos, por ordem crescente da intensidade da intervenção:

- **Convalescença** de perda temporária de capacidades.
 - permanência até 90 dias;
- **Reabilitação Funcional** (recuperação de acidentes, fracturas, ...)
 - permanência por mais de 90 dias;
- **Manutenção Funcional**
 - permanência muito prolongada;
 - os *Clientes* poderão ficar alojados vitaliciamente na instituição;
- **Cuidados Paliativos**
 - cuidados a prestar a um *Cliente* em fase terminal.

Produtos Comercializados

A *Carlton Life* disponibiliza diversos *Produtos* a *Clientes* com necessidades específicas, resultantes da perda permanente ou temporária de capacidade motora, cognitiva ou intelectual.

2.2.2 Modelo de Funcionamento das *Unidades de Negócio* da *Carlton Life*

O modelo de funcionamento que serviu de base ao projecto das *Unidades de Cuidados Continuados* da *Carlton Life* foi criado e tem vindo a ser testado e adaptado há alguns anos em unidades do mesmo tipo nos Estados Unidos da América.

O modelo define a organização física da *Unidade de Negócio* e também o número de *Cuidadores* necessários em cada período de trabalho, consoante o tipo e o nível de dependência dos *Clientes*.

Organização de uma *Unidade de Negócio* segundo o modelo dos EUA

Uma *Unidade de Negócio* de uma empresa deste tipo é constituída por:

- *Unidades Residenciais*;
- *Unidades de Dia*;
- *Unidades de Apoio*;
- *Unidades de Diagnóstico Geriátrico*.

As *Unidades Residenciais* têm uma organização complexa e muito específica, que se ilustra na Figura 2.1 e se apresenta a seguir:

- Cada *Unidade Residencial* é dividida em dois *Núcleos* que, por sua vez, são divididos em três *Sub-Núcleos*;
- Cada *Sub-Núcleo* tem capacidade para alojar 7 *Clientes*.
 - Este número de *Clientes* permite um melhor funcionamento e uma melhor organização dos serviços prestados.
 - Os *Clientes* são agrupados em *Sub-Núcleos* consoante o seu grau de dependência de terceiros (ou *Produto* contratado), garantindo-se dessa forma a homogeneidade dos grupos formados.
- Em cada um dos *Sub-Núcleos*, existem também locais comuns:
 - Restaurante(s);
 - Locais de convívio;
 - Locais para actividades;
 - ...

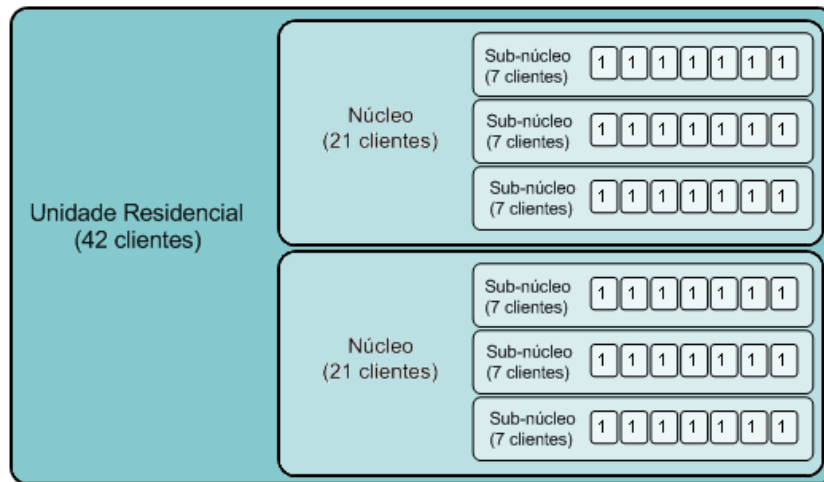


Figura 2.1: Organização de uma *Unidade Residencial* segundo o modelo dos EUA.

2.2.3 Requisitos

Cuidadores

Nesta secção estão registadas todas as informações, relacionadas com os *Cuidadores* da *Carlton Life* e importantes para a elaboração do sistema.

Refere-se também outros dados de interesse relacionados com os *Cuidadores*, pois, mesmo que não tenham implicação directa no sistema, são importantes para a sua concepção e para o enquadramento do problema.

Informação associada aos *Cuidadores* A *Carlton Life* terá *Cuidadores*, que se poderão dividir em três grandes grupos:

- Chefias;
- *Técnicos Especializados*;
- *Técnicos de Apoio Pessoal*.

Os *Cuidadores* cuja afectação será tratada pelo sistema resultante do Projecto Chronos serão os *Técnicos Especializados* e os *Técnicos de Apoio Pessoal*, relativamente aos quais será conhecida a seguinte informação:

- conjunto de competências de cada um (eventualmente com algum grau de proficiência para cada uma delas);
- contrato de trabalho (horário, folgas, etc.);

- histórico das actividades realizadas (por exemplo, *Clientes* com os quais tenham trabalhado).

As competências dos *Cuidadores* permitirão definir a sua afectação:

- a equipamentos;
- a *Sub-Núcleos*;
- a *Núcleos*;
- a *Unidades Residenciais*;
- ou mesmo a *Unidades de Negócio*.

As competências de cada *Cuidador* terão influência na elaboração de horários, já que cada *Cuidador* só poderá ser colocado num *Sub-Núcleo*, num *Núcleo*, numa *Unidade Residencial* ou numa *Unidade de Negócio* que necessite das competências que ele possui.

O número de *Cuidadores* de cada um dos tipos será um dado do problema.

Técnicos de Apoio Pessoal Os *Técnicos de Apoio Pessoal* serão os *Cuidadores* com mais restrições e, por representarem metade do total dos *Cuidadores* da *Carlton Life*, é da maior relevância o cuidado na sua alocação, pois poderá implicar ganhos consideráveis no negócio. Estes *Cuidadores* terão menos mobilidade que os demais e em cada turno prestarão serviços no mesmo *Sub-Núcleo* ou *Núcleo* ou *Unidade Residencial*.

O cálculo da quantidade de *Cuidadores* necessários deverá ser feito para cada *Unidade Residencial* e não para cada *Núcleo* ou *Sub-Núcleo*. No entanto a afectação poderá ir ao detalhe do *Núcleo* ou *Sub-Núcleo*.

Técnicos Especializados Espera-se que os *Técnicos Especializados* tenham mobilidade total ou quase total. Assim, espera-se que:

- tenham mobilidade entre *Unidades Residenciais* durante o mesmo turno;
- por vezes possam ter mobilidade entre *Unidades de Negócio* — nomeadamente no caso de técnicos muito especializados.

Turnos Os dias serão divididos em turnos de 8 horas cada:

- Turno da manhã – das 8:00 às 16:00;
- Turno da tarde – das 16:00 às 24:00;
- Turno da noite – das 24:00 às 8:00.

Não existirá rotatividade entre turnos.

Exemplo: As necessidades de *Técnicos de Apoio Pessoal* numa *Unidade Residencial*, com vários cenários de dependência dos *Clientes*, estão ilustradas na Figura 2.2. Atentando ao cenário de *Clientes* com grau de dependência muito elevado podemos ver na Figura 2.2 que serão necessários por dia 9 *Técnicos de Apoio Pessoal*.

Considerando os dias de não trabalho de cada um desses *Cuidadores* como sendo 52×2 , correspondentes a fins de semana, acrescidos de 22 dias de férias, e ainda uma média de 13 feriados por ano, para colmatar a necessidade de nove *Técnicos de Apoio Pessoal* disponíveis diariamente, será necessário contratar $9 \times \frac{365}{365-139} = 14,54 \approx 15$ desses *Cuidadores*.

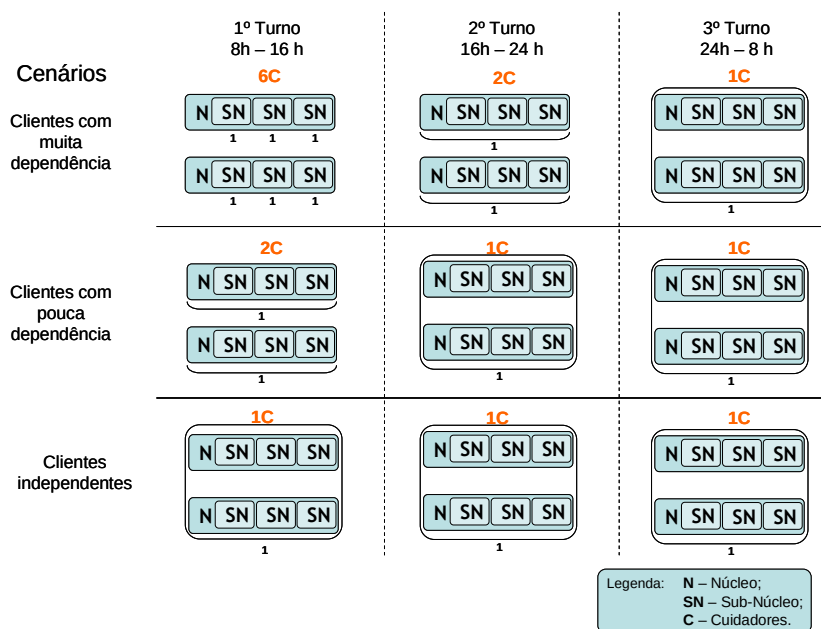


Figura 2.2: Exemplo das necessidades de *Técnicos de Apoio Pessoal*, numa *Unidade Residencial*, durante um dia, para diferentes cenários.

Segundo o modelo dos EUA, já anteriormente referido, o cálculo do número de *Técnicos de Apoio Pessoal* de que uma *Unidade Residencial* necessitará será obtido pela seguinte expressão:

- $f \times 1,61 = 9 \times 1,61 = 14,49 \implies 15$ *Técnicos de Apoio Pessoal* na *Unidade Residencial* apresentada na Figura, em oposição aos nove esperados à partida.

Onde f é o número mínimo de *Técnicos de Apoio Pessoal* para satisfazer as necessidades da *Unidade Residencial*.

Não é do intuito do projecto a consideração de *Cuidadores de Backup*.

Prioridade de Colocação dos *Técnicos de Apoio Pessoal*

- Prioritariamente o *Técnico de Apoio Pessoal* deverá ser colocado no mesmo *Sub-Núcleo* que tem vindo a ocupar.
- Se não for possível, deverá ser colocado no mesmo *Núcleo*.
- Se também não for possível, deverá ser colocado na mesma *Unidade Residencial*.

Leis Laborais e Contratos As leis laborais relevantes para este sistema foram fornecidas posteriormente à conclusão do Projecto Chronos. De qualquer forma, nesse projecto, foi feita uma referência à matéria em que se esperava que essas leis incidissem.

Para cada *Cuidador*, a semana de trabalho tem cinco ou seis dias. As folgas semanais devem coincidir com fins-de-semana com uma certa periodicidade. A empresa funcionará segundo uma filosofia de igualdade e responsabilização dos *Cuidadores*, o que lhes dará mais liberdade para trocas de horários.

Referiu-se também que a *Carlton Life* não descartará a hipótese de contratar *Cuidadores* à hora para prestações pontuais de serviços. Este tipo de contratação poderia ser sugerida pelo sistema desenvolvido neste projecto.

2.3 Problema de *Nurse Rostering*

2.3.1 Do Problema do Projecto Chronos ao Problema para a Dissertação

Depois de terminado o Projecto Chronos e apresentadas as propostas, a empresa *Carlton Life*, já com a sua estrutura organizacional mais próxima da fase de abertura, sugeriu algumas modificações, que se revelaram necessárias para adaptação à nova organização da empresa. Essas pequenas modificações conduziram a uma inevitável divisão do modelo proposto em vários, um para cada tipo de *Cuidador* (*Técnicos de Apoio Pessoal*, *Técnicos Terapeutas*, *Técnicos de Enfermagem* e *Técnicos de Medicina*). Tinha-se agora quatro modelos para estudar separadamente.

Para o desenvolvimento do trabalho aqui relatado, decidiu-se focar apenas um desses problemas, de modo a ter a possibilidade de aprofundar e investigar detalhadamente o melhor modo de resolução a propor. Como trabalho de investigação, interessou também fazer do problema escolhido um problema que se adaptasse a outros casos idênticos, de outras instituições como a *Carlton Life*, e tratá-lo de uma forma conveniente e inserida na área de investigação mais apropriada.

Assim, após uma consulta inicial a alguma literatura sobre problemas de escalonamento, nomeadamente ligados à saúde, a área que se revelou mais interessante e idêntica ao problema identificado foi a de *Nurse Rostering*. Esta é uma área muito explorada, mas que ainda pode evoluir muito. Escolheu-se, então, abordar um problema geral de *Nurse Rostering*, construído com base na bibliografia existente e adaptado aos requisitos específicos da *Carlton Life*, que poderão também ser motivo de preocupação por parte de outras instituições idênticas.

Pesquisada a literatura sobre *Nurse Rostering* e identificado um problema que, apoiado nas necessidades da *Carlton Life*, pudesse ser considerado um problema geral de *Nurse Rostering*, o que distingue este dos demais é a valorização do contacto entre os enfermeiros e os doentes, de forma a que os doentes sejam acompanhados pelo mesmo conjunto de enfermeiros. Vendo sempre os mesmos *Cuidadores*, os doentes estabelecem uma relação com eles e sentem que o seu acompanhamento é personalizado e cuidado, acedendo ao sentimento de segurança e satisfação com o serviço prestado.

2.3.2 O Problema

O problema de “*Nurse Rostering*” foi abordado na literatura pela primeira vez na década de 1970, tendo-se tornado o trabalho de M. Warner uma referência para os estudos seguintes. Normalmente os problemas abordados e descritos em artigos científicos são idênticos em vários pontos. Diferenciam-se porém em certos pormenores de aplicação, específicos para a instituição para a qual se destina o trabalho.

Os problemas deste tipo tratam da elaboração de um escalonamento de tarefas para enfermeiros, periódico e sujeito a uma grande variedade de restrições, tais como:

- necessidades de enfermeiros por cada turno (ou dia, ou tarefa, ou ainda padrão de turnos);
- leis laborais;
- políticas da instituição com os seus enfermeiros;
- preferências dos enfermeiros;
- outros requisitos específicos da instituição ...

Tratam enfermarias com 20 a 30 enfermeiros, contemplam 3 turnos diários (que podem ser de igual duração ou de durações distintas) e 3 ou 4 graus de enfermeiros (por exemplo, chefe + “regular” + estagiário). Os planeamentos são normalmente projectados para um período de 4 semanas (como em [1]–[5]), embora também haja casos em que apenas se fazem planeamentos semanais ou quinzenais (como em [10] e [11]).

Baseado nos problemas descritos nos artigos referenciados e tentando ser o mais geral possível na elaboração do problema a considerar ao longo deste trabalho, o problema criado caracteriza-se, fundamentalmente, por ter três turnos por dia (manhã, tarde e noite), ter um período de planeamento de 28 dias (4 semanas) e contemplar três graus de competências dos enfermeiros:

- sénior: grau 1;
- “normal”: grau 2;
- estagiário: grau 3.

Os turnos podem ter diferentes durações uns dos outros, o que não afectará o funcionamento do sistema aqui desenvolvido, uma vez que a duração ou o início e final dos turnos não serão contemplados, apenas a quantidade de turnos por dia.

Os nomes dados a cada um dos graus também podem ser diferentes dos sugeridos. Os nomes do exemplo em cima pretendem destacar que os enfermeiros de grau 1 são os enfermeiros de grau mais elevado e os enfermeiros de grau 3 os de grau menos elevado.

Durante toda esta descrição do trabalho os turnos serão referidos por “manhã”, “tarde” e “noite” e os graus dos enfermeiros por “grau 1”, “grau 2” e “grau 3”, do mais elevado para o mais baixo na hierarquia de competências.

Objectivos

Tal como na maioria dos trabalhos realizados nesta área, os objectivos do problema resumem-se a:

- Cobrir totalmente as necessidades de todos os graus de enfermeiros em cada turno, no período de planeamento;
- Minimizar as penalizações por não satisfação de “soft constraints” (incumprimento de preferências, alocações de enfermeiros de categoria superior à necessária, incumprimento de regras da instituição, ...);
- Maximizar o equilíbrio de distribuição de cargas horárias entre os enfermeiros, durante todo o horizonte temporal;
- Minimizar custos financeiros.

Pode dizer-se que os objectivos destes problemas, e deste em particular, se centram, genericamente, na minimização de todo o tipo de custos associados à elaboração dos horários dos enfermeiros (primeiro e último pontos anteriores), garantindo a maximização da satisfação dos enfermeiros com os seus horários.

Restrições do Problema

Este tipo de problema está sujeito a um grande número de restrições. Não só as restrições obrigatórias (“hard constraints”), como também muitas restrições de regras laborais e da instituição, preferências, acordos, ... (abordadas como “soft constraints”).

Hard Constraints Estas são as restrições que deverão ser cumpridas. Tal como os objectivos do problema, este tipo de restrições não varia muito de caso para caso, focando-se essencialmente em:

- Cobrir as necessidades por grau de enfermeiro e turno;
- Alocar cada enfermeiro uma vez no máximo por cada turno;
- Garantir um intervalo mínimo entre duas alocações consecutivas do mesmo enfermeiro.

Soft Constraints Este tipo de restrições varia mais de abordagem para abordagem do que as “hard constraints”. Varia com a instituição, com as pessoas e a cultura de trabalho do local, e varia também com a sensibilidade dos autores do trabalho.

Estas restrições podem ser agrupadas por temas, como esquematizado, na forma de um *mind map*, na Figura 2.3. Nesta figura estão também comentadas algumas das restrições e assinaladas com uma cruz as restrições que não foram contempladas neste trabalho. As restrições que são abordadas nos modelos de programação linear estão assinaladas a verde e, as que foram adicionadas no *Algoritmo Genético*, assinaladas com uma estrela ao seu lado.

As restrições relacionadas com fins-de-semana que serão abordadas ao longo deste trabalho são:

- controlar do número máximo de fins-de-semana alocados a cada enfermeiro ao longo do horizonte temporal;
- atribuir os fins-de-semana sempre completos, isto é, sempre que um enfermeiro seja alocado a um fim-de-semana, pretende-se que seja alocado a sexta-feira, sábado e domingo;
- alocar um enfermeiro sempre ao mesmo tipo de turno nos vários dias de um fim-de-semana.

Existem algumas restrições de contacto entre os enfermeiros, por necessidades de tutoria a enfermeiros de menor grau, por parte dos enfermeiros de maior grau de competências, ou por haver incompatibilidades de trabalho entre dois enfermeiros, que os impossibilitem de trabalhar juntos. Restrições contempladas somente pelo *Algoritmo Genético* concebido.

As preferências dos enfermeiros relativas a alocações e dias de trabalho, assim como as suas disponibilidades e indisponibilidades de trabalhar nos

2.3. PROBLEMA DE NURSE ROSTERING

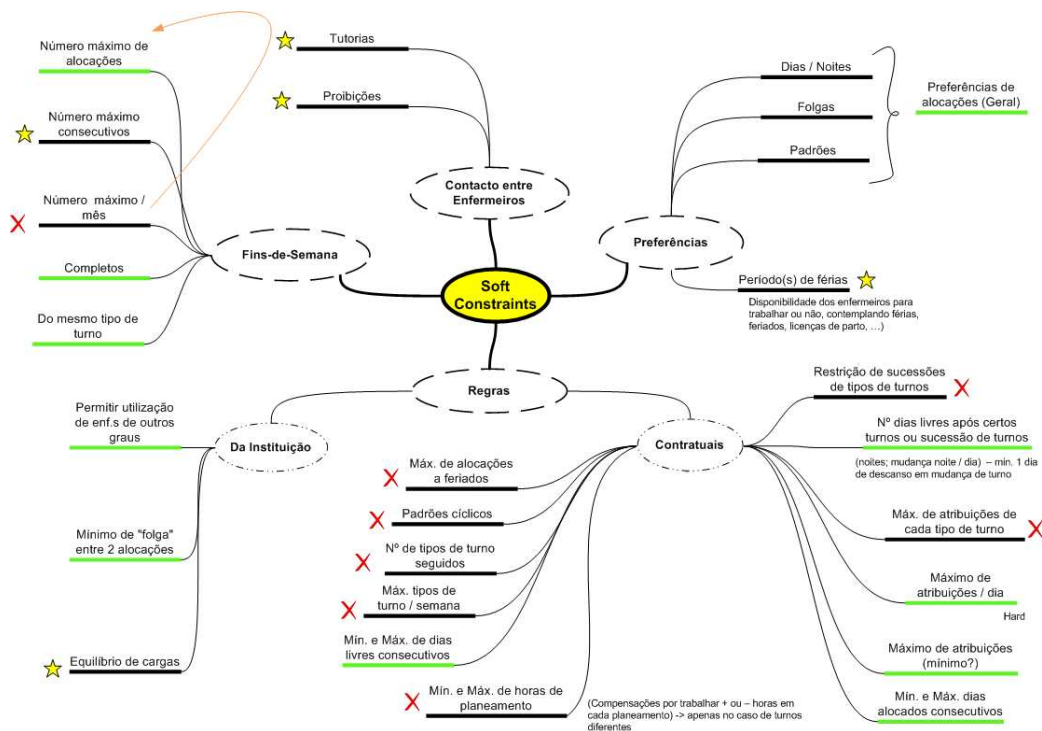


Figura 2.3: *Mind Map* representativo das “*Soft Constraints*” do problema de *Nurse Rostering*.

vários dias do horizonte temporal são também consideradas. As indisponibilidades poderão dever-se a férias, licenças de ausências, entre outras.

Restam as regras, tanto contratuais, como da instituição empregadora. As primeiras englobam máximo de alocações permitidas no horizonte temporal, máximo e mínimo de alocações consecutivas permitidas, assim como máximo e mínimo de dias livres consecutivos permitidos. Por vezes são consideradas mais restrições (identificadas na Figura 2.3). O segundo tipo de regras contempla a permissão de utilização de enfermeiros de graus superiores em funções de menor exigência de competência (funções de menores graus), caso tal opção seja compensatória para a qualidade e reduza os custos do escalonamento final, o mínimo de folgas entre duas alocações consecutivas e equilíbrio entre os escalonamentos individuais atribuídos aos enfermeiros de um mesmo grau de competências.

2.3.3 Pressupostos Considerados

Durante a concepção do problema surgiu a necessidade de decidir acerca de alguns pormenores de definição do problema e da forma de o abordar.

Financeiros

Importou definir como calcular a porção salarial dos custos de alocação dos enfermeiros. Poderiam ser concebidas várias formas de o fazer. Decidiu-se haver custos salariais para a alocação de enfermeiros a cada turno, por grau a que pertence. Não se consideram pagamentos de horas extraordinárias, porque, além de haver outras formas de compensar este tipo de serviço, não se pretende definir número fixo de horas de trabalho semanal. Por fim, todos os enfermeiros trabalham em “full-time”, não havendo também questões relacionadas com pagamentos e alocações a enfermeiros com contrato em “part-time”.

Cobertura de Necessidades

Algumas abordagens consideram a existência de um conjunto de enfermeiros de “back-up” para apoio na cobertura de “falhas” de enfermeiros que apareçam nos horários. Aqui assume-se que existem sempre enfermeiros suficientes para cobrir as necessidades, logo não se consideram situações de “back-up”.

Acontece também haver distinção entre necessidades mínimas e preferenciais a serem cobertas (como em [11]). Neste trabalho, tal não acontecerá, todas as necessidades terão que ser cobertas. A cobertura será feita por turnos e não contemplará padrões que possam separar turnos diurnos e nocturnos ou obrigar enfermeiros a terem escalonamentos pré-definidos.

Os intervalos mínimos entre turnos alocados serão tratados como “hard constraints”.

Tipos de Enfermeiros

No trabalho desenvolvido em [25] os *Cuidadores* (aqui apenas enfermeiros), além de serem separados por graus, são também, dentro de cada grau, diferenciados por competências que tenham e até por níveis de competências. É também atribuído, pelo empregador, um nível de competência desejável para cada necessidade de alocação. Neste trabalho isso não acontece, os enfermeiros são unicamente diferenciados pelos graus de competência que têm.

2.3. PROBLEMA DE NURSE ROSTERING

Quando muito surgirão, no desenvolvimento do *Algoritmo Genético*, grupos de trabalho e preferências de contacto entre os enfermeiros (Figura 2.3).

Capítulo 3

Revisão Bibliográfica

3.1 Problemas de Escalonamento em Investigação Operacional

Há vários tipos de problemas de escalonamento com que a Investigação Operacional se ocupa, nomeadamente: Escalonamento de Produção (responde, em termos gerais, a quantos e quais os produtos a fabricar ao longo do tempo); Escalonamento de Pessoal (determina horários ou distribuições de turnos para trabalhadores, procurando ir de encontro às restrições impostas ao escalonamento, nomeadamente as laborais); “Timetabling” (trata da determinação de horários que exijam combinações de disponibilidades para acções simultâneas de várias pessoas, com determinados recursos – exemplos: escalonamentos de aulas em escolas; escalonamentos de alunos e salas para exames). Todos estes problemas são complexos e têm características particulares, quer os três tipos de problemas identificados, quer cada problema de um determinado tipo, tornando qualquer resolução difícil de generalizar.

Na área de escalonamento de pessoal, em que se insere o problema aqui tratado, podem distinguir-se as áreas de: sistemas de transportes, “call centers”, sistemas em cuidados de saúde (nomeadamente o de *Nurse Rostering*), sistemas de serviços de emergência e de protecção (por exemplo, para polícia, bombeiros e ambulâncias), sistemas de serviços governamentais (especialmente quando são privatizados), gestão de recepção e entregas (designadamente empresas de “handling” em aeroportos), serviços financeiros (em bancos, empresas que prestam serviços de auditoria, . . .), serviços turísticos (pessoal de hotéis, por exemplo), sistemas para serviços de retalho e sistemas em ambientes industriais.

A área de transportes é uma vasta área económica, na qual a aplicação de ferramentas de escalonamento é essencial para o funcionamento do complexo

3.2. INVESTIGAÇÃO OPERACIONAL APLICADA À ÁREA DA SAÚDE

sistema que envolve. Os problemas que aqui se inserem caracterizam-se por existirem rotas e um número de deslocações previamente definido. Por vezes, surge a necessidade de criar deslocações extra (irregulares) para satisfazer as necessidades dos clientes, seja em transportadoras aéreas, rodoviárias ou ferroviárias. Em geral, as tarefas são distribuídas por blocos, de acordo com o lugar e a hora de início e fim das viagens. Um bloco numa transportadora aérea costuma ser um voo entre dois locais, enquanto para uma transportadora rodoviária ou ferroviária pode ser um percurso entre duas paragens no meio de uma viagem. Estes problemas de escalonamento de pessoal, pelo impacto económico que têm, são os mais abordados na literatura.

Em “call centers”, ao contrário do escalonamento de pessoal em transportadoras, os trabalhadores não se deslocam entre diferentes localizações, o que, por um lado, facilita a resolução do problema. Mas, por outro lado, a questão complica-se por as tarefas não serem conhecidas “à priori”. O que se sabe é que há sempre necessidades de pessoal em todo o horizonte temporal. As necessidades variam de dia para dia e de semana para semana. Os turnos devem variar de duração e de hora de início, para se obterem boas soluções. Haverá frequentemente a necessidade de cobrir sobrecargas de trabalho.

Em ambiente industrial, o escalonamento dos trabalhadores tem, muitas vezes, que ser ajustado e combinado com o escalonamento da produção dos bens e das máquinas utilizadas, tendo também em atenção toda a logística de funcionamento do local de trabalho; por exemplo, numa fábrica, a garantia de que o produto percorre todo o processo atempadamente.

Estas áreas detalhadas, juntamente com o *Nurse Rostering* em serviços de cuidados de saúde, são as mais estudadas em Investigação Operacional no que respeita a escalonamento de pessoal. Uma boa revisão sobre escalonamento de pessoal pode ser encontrada em [17].

3.2 Investigação Operacional Aplicada à Área da Saúde

A área da saúde apresenta vários problemas de particular interesse para os praticantes de Investigação Operacional. Diferenciam-se várias esferas de intervenção, em que os problemas se distinguem claramente. Já em 1976 foi feito um levantamento bibliográfico de casos de Investigação Operacional em sistemas de cuidados de saúde, em [19]. Nesse levantamento, o autor identificou um número considerável de áreas, enquadrando nelas trabalhos descritos em vários artigos científicos. Na década de 1970 o número de trabalhos em

Investigação Operacional relacionados com cuidados de saúde aumentou significativamente, especialmente para sistemas a serem utilizados em hospitais, mas já tendo em vista a grande privatização desta área nos Estados Unidos da América. Algumas das áreas identificadas nesse levantamento são: planeamento dos serviços de saúde e elaboração de programas de avaliação do seu funcionamento (planeamento de determinados tratamentos, avaliações de desempenhos profissionais e financeiros, . . .); estimativas de procura de serviços de saúde; planeamento da localização de hospitais e serviços hospitalares; simulação e planeamento de serviços de emergência para ambulâncias; planeamento da ocupação dos hospitais (quer ocupação temporária de camas, quer ocupação de quartos por um considerável período de tempo); determinação e previsão do tamanho de certas unidades de tratamento especiais e escalonamento da sua utilização (nomeadamente blocos operatórios); planeamento e escalonamento de pessoal no hospital (incluindo *Nurse Rostering* e aspectos económicos relacionados com as decisões em causa); escalonamento de exames médicos, entre outras áreas, algumas mais gerais.

Além dos problemas de *Nurse Rostering*, muito abordados, actualmente os temas mais estudados em Investigação Operacional, relacionados com a saúde, são: problemas de escalonamento de pacientes, que procuram a melhor sequência de atendimento dos doentes; escalonamento de cirurgias em blocos operatórios, com a particularidade de ser necessário considerar o tempo de preparação do bloco antes de se iniciar a operação e os procedimentos pós operatórios (desinfecção, . . .) no planeamento; escalonamento de visitas de médicos a pacientes externos, que contempla perdas de tempo em transporte; sistematização de consultas médicas, onde a consulta é estruturada e registada num sistema de informação.

3.3 *Nurse Rostering*

O problema abordado neste trabalho é o escalonamento de enfermeiros ao longo de um determinado horizonte temporal, sujeito a numerosas restrições. No capítulo 2 será descrito em pormenor o problema e o modo como surgiu a oportunidade de o trabalhar. De seguida é feito um levantamento bibliográfico de trabalhos desenvolvidos nesta área, que suscitaram interesse por contribuírem para a compreensão do tema e para o desenvolvimento deste projecto.

Começa por referir-se o problema de *Nurse Rostering* de uma forma geral, recorrendo a um estado de arte acerca desta temática, partindo-se, em seguida, para breves descrições dos trabalhos que, por semelhança ao pro-

blema aqui abordado, são interessantes pelas abordagens que aplicam ou pela relevância do ponto de vista científico.

Burke et al. – Estado da arte Em [14], é feita uma revisão do trabalho realizado, por vários autores, na área de *Nurse Rostering*. Este artigo está dividido em quatro secções principais. Na primeira é feita uma introdução ao artigo, ao problema de *Nurse Rostering* e à sua contextualização em problemas de *Employee Scheduling*. É apresentado um glossário de terminologia importante para a compreensão do texto, linguagem técnica respeitante ao problema de *Nurse Rostering*, e exemplificado o que é um problema de *Nurse Rostering*, para que o leitor se possa sentir familiarizado com o tema ao longo da leitura das restantes secções.

Na segunda secção são feitas referências aos diferentes tipos de abordagem do problema de *Nurse Rostering* e de alguns problemas relacionados, com maior destaque para as descrições dos modelos realizadas em cada artigo analisado.

Esta secção divide-se em quatro partes. A primeira centra-se numa revisão bibliográfica focalizada na abordagem das decisões estratégicas inerentes a este tipo de problemas. A segunda ocupa-se das implicações e decisões de sustentabilidade em problemas de *Nurse Rostering*, os autores preocupam-se ainda em distinguir as separações de níveis de decisão feitas nos artigos analisados (as decisões em causa são tipicamente de sustentabilidade, de alocação do pessoal e de restrições monetárias, que podem situar-se em vários níveis de decisão). A terceira parte explica sucintamente os modos administrativos de operar e suas influências no modo de resolução do problema de *Nurse Rostering*. Refere artigos que se centram num só modo de administração e outros que explicitam vários e apresentam soluções e avaliações das mesmas soluções para os diversos modos de administração. A quarta e última parte aborda a problemática de se optar por fazer um escalonamento *cíclico* ou não. São apresentados artigos que usam e defendem a prática de escalonamentos *cíclicos*, outros que apenas avaliam e comentam as vantagens e desvantagens da utilização deste tipo de escalonamentos.

A terceira secção incide sobre os vários tipos de abordagem usados para resolver o problema de *Nurse Rostering*. Nos primeiros parágrafos é realçada a evolução que as abordagens a este tipo de problemas têm sofrido ao longo do tempo.

Esta secção tem cinco partes. Cada uma delas trata de um tipo de abordagem e, em alguns casos, está dividida por temas.

A primeira parte dedica-se à programação matemática. Esta abordagem, ao procurar a solução óptima, torna-se inviável para problemas de grande

dimensão, como são os de *Nurse Rostering*. Apesar disto, serve muitas vezes de base para encontrar soluções, podendo operar-se algumas simplificações para se conseguir resolver problemas com uma complexidade elevada num período de tempo aceitável. Tem uma relevante importância para as abordagens realizadas por outros tipos de programação, como as apresentadas na segunda parte. Referem-se artigos que abordam a problemática de *Nurse Rostering* apresentando métodos de programação inteira e linear.

A segunda parte relaciona-se um pouco com a primeira, tratando artigos que abordam o problema central (de *Nurse Rostering*) através da programação por objectivos / abordagens por critérios múltiplos (*goal programming / multi-criteria approaches*).

A terceira parte refere-se aos artigos que tratam o problema de *Nurse Rostering* através de aplicação de métodos de inteligência artificial. São separados em dois grupos: os artigos que utilizam metodologias de programação por restrições e declarativa (*declarative and constraint programming*), e os que utilizam sistemas baseados em conhecimento e sistemas periciais (*expert systems and knowledge based systems*).

Na quarta parte são referidos os artigos que apresentam heurísticas como método de resolução, com o objectivo de tornar a resolução do problema mais simples e facilmente aplicável a casos práticos.

Por fim, e no seguimento da anterior, a quinta e última parte aborda as soluções baseadas em Metaheurísticas. Os artigos abordados estão agrupados nas seguintes Metaheurísticas: *Simulated Annealing*, *Tabu Search* e *Genetic Algorithms*.

A última secção do artigo apresenta conclusões dos autores. Nesta secção, o problema de *Nurse Rostering* e as abordagens apresentadas são discutidos de uma forma geral. São confrontadas algumas dessas abordagens (em diferentes tabelas) e estabelecem comparações entre as soluções apresentadas e discutidas no artigo: entre as que foram e as que não foram implementadas, entre as que foram implementadas em casos de pequena dimensão e as usadas em casos de grande dimensão.

Na conclusão, são também apresentadas sugestões de áreas e temas a desenvolver em futuras investigações relacionadas com o problema de *Nurse Rostering*.

Este artigo é ainda composto por dois anexos: um que lista uma série de dissertações de doutoramento sobre escalonamentos em hospitais, e outro que classifica os artigos referidos ao longo do texto por restrições e parâmetros considerados.

A leitura deste texto foi relevantes para seleccionar artigos importantes e interessantes, de entre os referidos, para contextualização e auxílio no tratamento do problema em causa no trabalho agora desenvolvido.

Cheng et al. – Bibliographic Survey Os autores redigiram uma “*Bibliographic Survey*”, [9], em que resumem e compilam modelos e metodologias de resolução do problema de *Nurse Rostering* até à data de publicação do artigo.

Descrevem o problema e as várias abordagens feitas à modelização do mesmo de uma forma geral e categorizada, recorrendo a exemplos e referências a artigos que descrevem trabalhos realizados nesta área de actuação. Identificam modelizações elaboradas sob diversos pontos de vista (enfermeiros por dia, enfermeiros por turno, ...), com as respectivas variáveis de decisão associadas, diferentes para cada tipo de modelização. Definem parâmetros típicos nestes problemas: necessidades de cobertura, turnos por escalonamento, custos, ... e possíveis abordagens a esses parâmetros. Listam as restrições mais comuns, tanto referentes a “hard constraints” como a “soft constraints”, associando-as aos pontos de vista anteriormente referidos. Referem também possíveis perspectivas para elaboração de uma função objectivo para o problema, apropriadas a cada tipo de modelização.

Após descrito, em termos gerais, o problema de *Nurse Rostering* e apresentadas as abordagens possíveis, os autores passam a definir os tipos de problema de *Nurse Rostering*. Enumeram três tipos:

1. Problemas de Optimização – recorrendo a programação matemática ou a programação por objectivos para sua resolução. São métodos de resolução muito centrados na optimização da função objectivo; pouco eficazes para problemas com muitas restrições;
2. Problemas de Decisão – resolvendo o problema como um problema de satisfação das restrições (CSP, *Constraint Satisfaction Problem*). Explicam este tipo de problemas e metodologias de resolução possíveis, sendo algumas Heurísticas (Tabu Search, Hill-Climbing, Genetic Algorithm, ...), Programação por Restrições (*Constraint Programming*), pesquisa em árvores sistemáticas, ...;
3. Problemas de Optimização de Restrições – variantes de CSP, mas com optimização de uma função objectivo. Estes são os mais apropriados a casos reais, por pesquisarem soluções que satisfaçam as restrições, em busca de uma boa solução.

Em seguida os autores descrevem metodologias possíveis de resolução do problema. Isto é, descrevem as fases por que um programa de resolução de um problema de *Nurse Rostering* deve passar e, em cada fase, referem-se aos métodos de aplicação que são mais usados. Essas fases são a inicialização da pesquisa, com a definição de um escalonamento inicial; o pré, o

pós-planeamento e as decisões inerentes (como o tipo de escalonamento – cíclico ou não – e a definição dos dados – requisitos mínimo, preferenciais ou obrigatórios); o planeamento propriamente dito, com a definição do método a utilizar, que está relacionado com o tipo de problema de *Nurse Rostering*.

Ao longo de todo o artigo, enquanto explicam de forma geral o problema de *Nurse Rostering*, os autores detalham algumas abordagens feitas em trabalhos descritos em artigos referenciados.

Na parte final do artigo é possível identificar algumas características de dimensão de um problema típico de *Nurse Rostering*:

- 20 enfermeiros por enfermaria;
- 6 tipos de turno por dia;
- 30 tipos de restrições por pessoa;
- 28 dias de horizonte temporal.

Jaumard et al. No trabalho do artigo referenciado em [15] os autores apresentam uma solução para um problema de escalonamento de enfermeiros, desenvolvida em parceria com o *Royal Victoria Hospital of Montréal* no Quebec, Canadá, como parte do seu sistema de informação de gestão do pessoal do hospital.

Dividem a sua abordagem num problema principal, em que estão definidos os objectivos e as restrições (apresentados na Tabela 3.1) respeitantes a toda a configuração dos escalonamentos individuais, e num problema auxiliar que inclui os requisitos específicos para cada enfermeiro.

O primeiro problema é abordado por um programa linear de variáveis binárias que procura configurações de escalonamentos que satisfaçam as necessidades, minimizando os custos salariais e maximizando a qualidade do serviço. Cada coluna da matriz de restrições deste problema corresponde a um escalonamento possível para cada enfermeiro. Devido à dimensão do problema, apenas consideram uma pequena parte do número de colunas (escalonamentos) possíveis de cada vez, sendo algumas outras geradas quando é necessário melhorar a solução. Esta geração de colunas é feita resolvendo um problema de caminho mais curto com restrições recursivas (problema auxiliar).

Ao longo do trabalho os autores usam e alertam para os seguintes dados:

- Os turnos são de durações distintas entre si e cada turno dá cobertura a um determinado conjunto de períodos de necessidades (períodos curtos).

Restrições	· Cada enfermeiro deverá ser alocado a exactamente um escalonamento s. · O número de enfermeiros de todos os conjuntos possíveis de competências $\mathcal{L}$ alocadas aos diferentes conjuntos de turnos $\mathcal{T}$, ou seja, de todos os pares $(\mathcal{L}, \mathcal{T})$, em cada período p de cada dia d, devem respeitar o mínimo, máximo ou exacto valor de necessidades $q_{\mathcal{L}\mathcal{T}}$.
Objectivos	O problema principal minimiza a soma pesada dos custos salariais, pesos de preferências dos enfermeiros e de equilíbrio entre a utilização de enfermeiros mais e menos experientes, sujeito às necessidades exigidas. Na função objectivo os custos salariais têm peso positivo, enquanto as preferências dos enfermeiros e o equilíbrio de utilização entre os mais e os menos experientes têm pesos negativos.

Tabela 3.1: Restrições e objectivos da abordagem usada no artigo [15]

- os custos salariais incluem tanto o salário regular do pessoal permanente, como os custos de horas extra e os custos de pessoal extra (sub-contratações, por exemplo).
- As preferências dos enfermeiros são obtidas por pedidos de dias de folga, preferências de dias ou de noites, ...
- A qualidade do serviço expressa-se pelo nível de equilíbrio entre enfermeiros mais ou menos experientes.

Os autores não consideram um escalonamento total para todo o conjunto de enfermeiros do hospital, mas sim vários escalonamentos, um para cada enfermeiro, que no final, juntos, e se pretendido, resultam no escalonamento geral para todo o hospital. Assim sendo, existem várias soluções (escalonamentos) possíveis para cada enfermeiro. Destes escalonamentos, apenas alguns aparecerão na solução final e cada escalonamento que aí aparecer terá um único enfermeiro a ele alocado.

No problema auxiliar, formulado, para cada enfermeiro, como um problema de caminho mais curto, os caminhos correspondem a colunas da formulação matricial do problema principal. Ou seja, cada caminho corresponde a um escalonamento (coluna) e este problema auxiliar deverá determinar o caminho que melhore a solução actual do problema principal e, ao mesmo tempo, garantir a satisfação dos requisitos contratuais (cargas de trabalho,

rotações dos turnos, fins-de-semana livre, dias livres, ...). Cada caminho a “encurtar” é constituído por vértices e arcos. Cada vértice corresponde a uma alocação, do enfermeiro em causa, a um turno num determinado dia. O primeiro vértice representa a última alocação do enfermeiro até ao momento actual, os restantes vértices são fictícios e representam possíveis alocações futuras. Os arcos ligam os turnos a que os enfermeiros podem ser sequencialmente alocados, isto é, um arco que ligue dois vértices i e j indica que, após ter sido alocado ao turno i , o enfermeiro pode ser alocado ao turno j . As restrições criadas para este problema procuram satisfazer valores máximos e mínimos, em cada arco, que garantam a satisfação dos requisitos contratuais. Os custos dos arcos são calculados com base nas preferências dos enfermeiros, nos custos e na desejabilidade em alocar enfermeiros com estas competências aos turnos correspondentes aos vértices que estão a ser ligados.

Como método de resolução do problema principal, os autores utilizaram o método *branch-and-bound*. Em cada nó da árvore existe um sub-problema do problema principal constituído por um determinado conjunto de colunas, que será resolvido pelo método simplex, de onde se retirarão variáveis duais do problema a serem utilizadas no problema auxiliar. Em cada problema auxiliar utilizar-se-ão as variáveis duais da solução do problema principal como referência e procurar-se-á um novo conjunto de colunas com um custo inferior. Ao encontrar uma solução admissível para este nó, segue-se para o próximo nó de exploração. Para o problema auxiliar, os autores referem alguma dificuldade em encontrar um algoritmo que o resolva, mas propõem o uso de um algoritmo pseudo-polinomial de duas fases, em que na primeira fase são obtidos os caminhos possíveis entre os vértices e seus custos e na segunda fase se procura a solução para o problema de caminho mínimo.

Meisels e Schaerf No trabalho do artigo em [16], os autores não pretendem resolver um problema de *Nurse Rostering* nem mesmo de *Employee Scheduling* real, mas sim encontrar um modelo padrão de resolução para grande parte dos problemas reais enquadrados em problemas classificados como *Employee Timetabling Problems* (ETP). Pretendem também encontrar novos métodos heurísticos de resolução deste tipo de problemas e, finalmente, testar os métodos com problemas reais. O problema que escolheram para testar os métodos desenvolvidos é um problema de *Nurse Rostering* de alocação de empregados a determinadas tarefas em diferentes turnos. Trata-se de um problema simples, em que os turnos são de duração fixa e apenas dos tipos *manhã*, *tarde* ou *noite*, e somente se consideram as restrições chamadas “hard constraints”, não contemplando “soft constraints” que seriam incorporadas na função objectivo. Para este problema de *Nurse Rostering* os autores apre-

sentam um modelo matemático com as restrições apresentadas na Tabela 3.2.

Parâmetros	i empregados – no conjunto $\{1, \dots, m\}$ j tarefas – no conjunto $\{1, \dots, n\}$ k turnos – no conjunto $\{1, \dots, t\}$ h conjuntos de turnos – no conjunto $\{1, \dots, s\}$
Variáveis de Decisão	x_{mnt} – alocação do empregado m à tarefa t no turno n ou não, <i>variável binária</i> .
Restrições	· <i>Necessidades</i>: $\sum_{i=1}^m x_{ijk} = R_{jk} \quad \forall_{j,k}$ · <i>Competências</i>: $x_{ijk} \leq Q_{ik} \quad \forall_{i,j,k}$ · <i>Disponibilidade</i>: $x_{ijk} \leq A_{ij} \quad \forall_{i,j,k}$ · <i>Conflitos de Alocações em Turnos</i>: $\sum_{k=1}^t x_{ij_1k} \sum_{k=1}^t x_{ij_2k} \leq c_{ij_1j_2} \quad \forall_i \forall_{\{j_1, j_2\} \in \{1, \dots, n\}}$ · <i>Carga de Trabalho</i>: $V_{ih} \leq \sum_{j \in G_h} \sum_{k=1}^t x_{ijk} \leq A_{ij} \quad \forall_{i,h}$
Objectivos	– Definir um modelo padrão para resolução de <i>ETPs</i>. – Desenhar algoritmos específicos para resolver este tipo de problemas. – Experimentar estes novos métodos em casos reais. – Para os métodos a testar: · Minimizar o número total de violações de restrições. · Na função objectivo todas as restrições do problema têm o mesmo peso.

Tabela 3.2: Modelização usada no artigo [16]

- A restrição das necessidades destina-se a garantir que, no horizonte temporal considerado, todas as necessidades em todos os turnos são satisfeitas. As necessidades de cada turno estão guardadas numa matriz de dados, $R_{n \times t}$.
- Na restrição de competências é garantido que cada cuidador em cada turno apenas desempenhará tarefas para as quais tem competência de execução. As competências que cada cuidador tem estão guardadas numa matriz $Q_{m \times t}$.

- As disponibilidades dos empregados são respeitadas pela terceira restrição, em que a matriz $A_{m \times n}$ tem guardada a informação de disponibilidade ou indisponibilidade de cada cuidador em cada turno.
- As regras organizacionais de distribuição de turnos (obrigatoriedade de ter um intervalo de um determinado número de turnos entre duas alocações consecutivas para o mesmo empregado, número máximo de turnos permitidos por semana, ...) estão garantidas na quarta restrição, recorrendo a uma matriz $C_{n \times n \times m}$ em que está definido se o empregado i pode trabalhar nos turnos j_1 e j_2 ou se há algum impedimento.
- Na última restrição, são garantidos limites de tarefas a que cada empregado pode ser alocado em cada turno e limites para alocações específicas (não detalhadas pelos autores). Para isto definem conjuntos de tipos de turnos $G_1, \dots, G_s$ e definem duas matrizes de valores inteiros, $V_{m \times s}$, e $W_{m \times s}$ tais que um empregado i terá que ser alocado a pelo menos V_{ik} e no máximo a W_{ik} turnos do grupo G_k .

Não considerando as “soft constraints”, os autores também não definem, com o modelo matemático, uma função objectivo para aplicar ao modelo.

Depois de definido o problema que serve de exemplo para aplicação dos métodos apresentados neste artigo, os autores passam a explicar que métodos propõem para resolver este tipo de problemas. O primeiro método é resolver o problema através de Programação por Restrições (*Constraint Programming*). Inicialmente propõem a construção de uma rede de restrições (*Constraint Network*) adaptada ao problema, em que as variáveis de decisão passam a ser a alocação dos vários empregados a pares (tarefa requerida, turno) (n, t) . Para um turno pode, no caso geral, haver a necessidade de vários empregados executarem a mesma tarefa, sendo nestes casos os pares anteriores desdobrados em conjuntos como o seguinte, $\{(n, t_1), \dots, (n, t_n)\}$. Depois de definida a rede de restrições, referem ter programado o problema de satisfação das mesmas, utilizando a linguagem ECLⁱPS^e.

O segundo método proposto é a utilização de técnicas de pesquisa local (*local search*) pertencentes à família de técnicas denominadas como *Hill-climbing techniques*. Iniciam a pesquisa com um algoritmo guloso que dará uma solução inicial admissível e depois aplicam os métodos de pesquisa local. Os autores descrevem qual o espaço de pesquisa que estão a utilizar e como o definiram e escolheram. Definem também a estrutura de vizinhança usada, como a criaram e a função objectivo utilizada (objectivos descritos na Tabela

3.2).

Por fim, os autores apresentam os resultados obtidos, comparando-os entre si e avaliando-os. Procedem a uma discussão dos métodos de implementação utilizados e referem que também testaram *Tabu search* na pesquisa local e pesquisa local estendida, com resultados muito parecidos aos obtidos com *Hill-climbing*, mas com algumas dificuldades na implementação de certos mecanismos, o que os levou a preferir o segundo.

Este artigo foi muito importante para o desenrolar do trabalho descrito nesta dissertação, porque o modelo aqui apresentado serviu de inspiração para a criação dos modelos matemáticos desenvolvidos para resolver o problema abordado.

Burke et al. Com o intuito de resolver o problema de *Nurse Rostering* para hospitais públicos na Bélgica, desenvolvem um produto denominado PLANE, que é descrito ao longo de vários artigos, cada um focando um aspecto estudado na forma de resolver o problema.

Em [1], os autores descrevem o seu problema de *Nurse Rostering*, a sua abordagem e métodos de resolução, inicialmente testados. O problema vai sendo detalhado ao longo dos restantes artigos. Apresenta-se aqui um resumo do problema geral que o PLANE procura resolver.

O problema abordado nesta série de artigos pretende minimizar os incumprimentos das “soft constraints”, enquanto garante a cobertura do número mínimo ou do número desejável de enfermeiros de cada grau (categoria), em cada tarefa. Neste problema é possível definir várias categorias de enfermeiros: enfermeiro chefe, “regular”, ajudante, estudante, ... O período de planeamento é de um mês, não existem turnos por dia, mas sim tarefas (6 a 15 por dia), em média, haverá 20 enfermeiros por enfermaria e os enfermeiros de categorias superiores poderão substituir enfermeiros de categorias inferiores. O problema tem três “hard constraints” bem definidas:

- Todas as tarefas especificadas nos requisitos de pessoal têm que ser alocadas a, pelo menos, um enfermeiro;
- Um enfermeiro não pode ser alocado a uma mesma tarefa duas vezes no mesmo dia;
- As tarefas só podem ser alocadas às pessoas de categoria igual ou superior à requerida.

É dada a hipótese, ao utilizador do sistema, de escolher entre definir requisitos mínimos ou preferenciais e entre resolver o problema respeitando de forma rígida os requisitos mínimos, os preferenciais ou respeitar os mínimos,

adicionando, no final, tantos enfermeiros quanto possível, sempre que essa adição não aumente o total de violações das “soft constraints” (nomeadamente preferências). As “soft constraints” estão agrupadas em restrições do hospital (intervalo mínimo entre duas alocações consecutivas, por exemplo), restrições contratuais (número máximo de atribuições permitidas ao longo do horizonte temporal, limites mínimos e máximos de alocações consecutivas permitidas, restrições relacionadas com fins-de-semana, ...) e restrições de acordos entre os enfermeiros e o enfermeiro chefe (dias de folga, preferências de alocações, ...). Todas as restrições estão descritas e explicadas em [6]. Ainda em [1] os autores explicam que as violações, em cada solução possível e avaliada, a “soft constraints” são pesadas na função objectivo e que esses pesos são atribuídos pelo utilizador.

A abordagem de resolução a este problema é iniciada preferencialmente com recurso a uma solução aleatória, por permitir tão bons ou até melhores resultados finais do que outros tipos de geração de soluções iniciais referidos: aproveitamento do escalonamento anterior (para o último mês) ou escalonamento actual (para o caso de alterações necessárias, de última hora). Os métodos que neste artigo são utilizados para resolver o problema são métodos heurísticos que recorrem a Tabu Search. O primeiro método que apresentam é um Tabu Search simples – em que se fazem trocas de pessoas alocadas a tarefas, recorrendo a uma lista de trocas permitidas, constituída, para cada tarefa, pelas pessoas que têm competências suficientes para a desempenhar e não estejam alocadas a nenhuma tarefa nesse período – e o segundo o mesmo Tabu Search, mas funcionando com algoritmos de melhoramento de performance da pesquisa:

- TS1 → recorre ao auxílio de um algoritmo que inicialmente reforça o peso das restrições respeitantes à garantia de que os fins-de-semana são totalmente preenchidos relativamente a outras restrições menos perceptivas, como sobrecargas de trabalho ou mudanças de alocações às segundas-feiras (que implicam mudanças em todo o resto da semana). Depois, caso não haja alterações ao escalonamento na fase anterior, o algoritmo escolhe o enfermeiro com pior escalonamento (dos “actualmente” alocados) e altera esse escalonamento, fazendo trocas com outras pessoas que também estão alocadas a essa solução.
- TS2 → recorre a um algoritmo guloso e evasivo (“greedy shuffling”), que estuda todas as trocas possíveis, seguindo a lógica da segunda parte do algoritmo de TS1, lista-as da melhor para a pior e vai realizando-as uma a uma, até não restarem trocas possíveis.

Com estas abordagens os autores pretendem que TS1 permita calcular uma boa solução em pouco tempo, enquanto TS2, demorando mais tempo, atinja um melhor resultado. Estes três métodos são testados e comparados entre si e com um outro método (não descrito pelos autores), que recorre a “steepest descent”.

Em [2], os autores focam o seu estudo mais no problema de, por vezes, os métodos heurísticos terem dificuldade em ultrapassar mínimos locais na pesquisa de uma boa solução. As heurísticas base da pesquisa continuam a ser o Tabu Search (TS) e o Steepest Descent (SD). Partindo desses métodos, criam meta-heurísticas compostas por uma heurística e um ou mais algoritmos de criação de vizinhanças.

Os métodos de criação de vizinhanças estão bem descritos e alguns deles são utilizados no PLANE. Baseiam-se em trocas de alocações ou trocas de partes de escalonamentos, sempre entre dois enfermeiros, ou em “abanões”, que fazem a pesquisa “saltar” para outro local do espaço de soluções, distante do que no momento estava a ser explorado. Os algoritmos para criação de vizinhanças são então divididos em três categorias:

- Simples:
 - em que as soluções numa vizinhança apenas diferem umas das outras numa alocação;
- Relacionadas com “soft constraints”:
 - em que cada algoritmo se centra numa “soft constraint” específica e tenta melhorá-la, provocando os tais “saltos” (diversificação) no espaço de soluções em exploração (existem vários algoritmos que possíveis para desempenhar este papel);
- Trocas de grandes secções de escalonamentos pessoais (existem vários algoritmos que possíveis para desempenhar este papel).

Estes algoritmos são combinados: o primeiro com um qualquer do segundo tipo, com um qualquer do terceiro tipo, para que, juntamente com as heurísticas base (TS ou SD), permitam obter melhor resultado, em menor tempo de cálculo. Os algoritmos vão sendo avaliados à medida que a pesquisa avança e os que menos contribuam para o melhoramento da solução vão sendo menos utilizados, podendo mesmo ser excluídos.

O artigo seguinte, [3], como já aqui foi referido, trata de algoritmos de avaliação das soluções, relativamente às “soft constraints”, quer a solução final, quer soluções intermédias durante a pesquisa efectuada pelo PLANE. A avaliação é efectuada para cada enfermeiro e apenas para as “soft constraints”

que envolvam relações temporais (dias livres consecutivos, alocações consecutivas, ...). Os autores desenvolvem lógicas matemáticas complexas para “converter” restrições reais em números e depois poder analisar as violações ocorridas a cada restrição, e atribuir penalizações. A lógica desenvolvida tem também a característica de facilitar a inserção de novas restrições reais. Este trabalho serve para o PLANE poder dar feedback ao utilizador da qualidade da solução encontrada.

Em [4] os autores abordam o problema através de um algoritmo memético. Definem um algoritmo genético (GA) a ele associado e combinam-no com os métodos heurísticos anteriormente desenvolvidos (TS simples, TS1 e TS2). Detalham pouco o modo como criam o GA, mas fica explícito que o método de selecção dos cromossomas pais é por “Tournament Selection” (para cada selecção, escolhidos dois cromossomas ao acaso, o melhor é escolhido para ser o pai seleccionado). São apresentadas e resumidamente explicadas diversas variantes do algoritmo memético desenvolvido, que foram testadas (por exemplo, sem selecção feita por “Tournament Selection”, mas feita aleatoriamente).

Por último, em [5], é apresentada uma nova abordagem à função objectivo e aos objectivos do problema. Usam como método de teste o TS híbrido (TS1 e TS2), mas dando um tratamento diferente às “soft constraints”. Passam a ser tratadas por uma abordagem multi-critério, através de um algoritmo que lhes atribui pesos. Para cada escalonamento é identificado o “ponto ideal” (com valor igual ao caso em que nenhum critério – “soft constraint” – é violado), normalmente inatingível, e um “ponto anti-ideal” (onde todos os critérios são violados). Cada escalonamento é avaliado pela sua proximidade ao “ponto ideal” no espaço de valores possíveis para a função objectivo, definido com tantas dimensões quantos critérios de avaliação das soluções (nesse espaço todas as soluções possíveis para o problema podem ser localizadas). O objectivo do algoritmo é minimizar a soma das distâncias de todos os escalonamentos de todos os enfermeiros ao “ponto ideal”.

Dowsland Aborda, em [10], um problema de *Nurse Rostering* para hospitais de grande dimensão. O problema é do tipo de atribuição a cada enfermeiro de um único padrão de alocações a turnos, previamente definido. Os planeamentos são semanais, existem 3 graus de enfermeiros e a possibilidade de recorrer a enfermeiros em *part-time*. Quando um enfermeiro trabalha de noite, tem uma redução do número total de turnos alocados durante a semana. As penalizações referentes à qualidade de cada padrão (escalonamento individual) estão aglomeradas num número para cada par enfermeiro – padrão e são medidas relativamente a vários factores (dias de folga pedidos

pelo enfermeiro, histórico de trabalho do enfermeiro, ...).

Depois de descrever o modelo matemático que desenvolveu para resolver o problema, a autora apresenta três alternativas de abordagem por métodos heurísticos, usando Local Search (LS). A primeira é um relaxamento do modelo, passando a restrição de necessidades de cobertura para a função objectivo, com a desvantagem de ter grande dificuldade em encontrar um equilíbrio entre os pesos das necessidades de cobertura (“hard constraints”) e das restantes restrições na função objectivo do problema. A segunda possibilidade contempla uma abordagem em duas fases, em que, na primeira fase, se procuram soluções que satisfaçam as “hard constraints” (necessidades de cobertura) e, na segunda fase, utilizando somente as soluções encontradas na fase anterior, se procura a solução que melhor satisfaça as restantes restrições do problema. Por fim, aproveitando a opção anterior, a terceira alternativa é a de, depois de corridas as duas fases da opção anterior, usar oscilação estratégica (SO – Strategic Oscillation) para sair das fronteiras do espaço em exploração, onde as “hard constraints” são satisfeitas, caminhando para outra área do espaço total de soluções do problema, onde tal satisfação também acontece, mas ainda está por explorar. A abordagem implementada e testada foi a última, opção justificada pela autora. Essa abordagem é detalhada, com destaque para os métodos utilizados para evitar que o processo de pesquisa ficasse “preso” a um mínimo local, recorrendo a TS (na terceira fase). Por fim, são analisados testes computacionais, identificados problemas, propostas soluções para os mesmos e referidos pontos de interesse para continuação do trabalho.

Em [11] o problema analisado assemelha-se ao anterior, embora esteja mais detalhado, tanto na descrição, como na forma de o abordar. É um trabalho feito com a intenção de desenvolver um produto para utilização em hospitais no Reino Unido, chamado CARE (Computer Aided Rostering Environment).

Trata-se de um problema com os objectivos de cobrir todas as necessidades de enfermeiros, atribuindo, a cada enfermeiro, o número admissível de turnos, maximizar a satisfação de preferências e pedidos individuais dos enfermeiros, respeitar considerações financeiras e respeitar regras e directivas de práticas de trabalho da instituição. Tem entre 20 a 30 enfermeiros, três turnos diários (divididos em dois de manhã e um de noite), os padrões a serem alocados aos enfermeiros são semanais, existem três graus de enfermeiros e as necessidades são definidas por turno e grau de enfermeiro, sendo expressas de forma cumulativa (necessidades de grau 1 – enfermeiros mais experientes –, necessidades de grau 1 e grau 2 e necessidades de grau 1, grau 2 e grau 3). Os padrões são elaborados com mais turnos diurnos do que nocturnos, existindo cargas típicas de combinações entre turnos diurnos e nocturnos num mesmo

padrão.

A pesquisa pelo melhor horário divide-se, novamente, em três fases, mas diferentes das anteriores:

1. Verificação da admissibilidade da solução com um modelo Knapsack (modelo da mochila), em que se determina o número de enfermeiros que serão alocados de dia e de noite e se verifica se existem enfermeiros suficientes;
2. Alocação dos enfermeiros em padrões diurnos e/ou noturnos com o algoritmo de TS e SO desenvolvido em [10];
3. Alocação dos enfermeiros a trabalhar de dia em turnos de manhã ou de tarde, consoante as suas preferências, recorrendo a um modelo de fluxo de redes (Network Flow Model).

Os autores descrevem e analisam testes computacionais à abordagem desenvolvida, recorrendo a 51 instâncias reais e comparando os resultados obtidos com os resultados, para as mesmas instâncias, da mesma abordagem, com, na segunda fase, um modelo de programação inteira (IP), em vez do TS e SO.

No final do artigo, referem um teste de robustez elaborado, no qual:

- passam as necessidades de cobertura dos turnos diurnos para os valores preferenciais, tentando respeitá-las tanto quanto possível;
- permitem as combinações anteriormente referidas entre turnos diurnos e noturnos nos padrões de escalonamentos;
- procuram garantir que não há mais do que 1 enfermeiro de grau 1 alocado a cada turno noturno;
- introduzem uma estratégia de cobertura por equipas.

Este último ponto assemelha-se à nova restrição que se pretende introduzir no trabalho descrito por esta dissertação. No artigo, os autores referem a intenção de criar equipas de enfermeiros e doentes, visando garantir a presença de, pelo menos, um enfermeiro de cada equipa em cada turno. Procuram satisfazer esta restrição fazendo ajustes no final de cada uma das duas últimas fases. Comentam que as soluções pioram, embora apenas ligeiramente.

No trabalho descrito em [12], os autores desenvolvem um algoritmo genético (GA) com o intuito de testar o seu funcionamento na segunda fase da abordagem apresentada em [11], por substituição do TS e do SO. Os autores assemelham o problema a um problema de cobertura de conjuntos, por se

tratar de um problema de definição de padrões de turnos (escalonamentos individuais) a atribuir aos enfermeiros. Cada cromossoma do GA é composto por números inteiros, que identificam os padrões a atribuir a cada enfermeiro. Cada posição do cromossoma corresponde a um enfermeiro. As restrições do problema (necessidades de cobertura e “soft constraints”) estão implementadas na função avaliadora dos cromossomas.

Como existem três graus de enfermeiros e as necessidades estão expressas de forma cumulativa, os autores desenvolveram um conjunto de regras hierárquicas para correr o GA em partes distintas. Em cada parte otimiza-se o problema para cada grau ou conjunto de graus, começando com otimizações individualizadas dos vários graus (nível mais baixo da pirâmide hierárquica (como a da Figura 3.1), considerando depois combinações dos graus dois a dois (nível médio da pirâmide hierárquica) e por fim todos os graus juntos (nível mais elevado da pirâmide hierárquica). Em cada nível de optimização, também se utilizam cromossomas trabalhados em níveis anteriores (mais baixos) como possíveis *cromossomas* pais. No final, apresentam e comentam alguns resultados experimentais.

Moz e Vaz Pato Desenvolveram um sistema para fazer replaneamento dos horários dos enfermeiros para um hospital público de Lisboa, quando surgirem ausências inesperadas de enfermeiro(s) por um ou mais dias, não havendo enfermeiros de *backup*.

Em [7] apresentam o modelo com que abordam o problema, assim como o problema em si e uma heurística construtiva de resolução. Continuam em [8], descrevem uma optimização da heurística construtiva anteriormente desenvolvida com recurso a um GA que também criaram.

As características do problema são de haver três turnos por dia, de 8 horas cada turno, um horizonte temporal de 28 dias, em que as soluções são compostas por um escalonamento para cada enfermeiro. As tarefas possíveis de serem alocadas são de estar alocado a um dos três turnos; não estar alocado a nenhum turno, mas estar disponível; ter o dia livre (“folga”) ou estar ausente. O problema está sujeito a condições (regras e preferências), as “soft constraints”, listadas em [7].

A modelização do problema é feita recorrendo à definição de “Integer Multicommodity Flow Problem”: um problema de fluxos de utilidades (commodities), valores inteiros, entre vários nós. A rede inicia no dia ($d-2$), sendo d o dia em que se verifica a ausência inesperada, com um nó de partida para cada enfermeiro (commodity), entre os dias $d-1$ e 28 existem, para cada dia 4 nós, em que cada nó representa um turno de um dia ou uma folga e cada

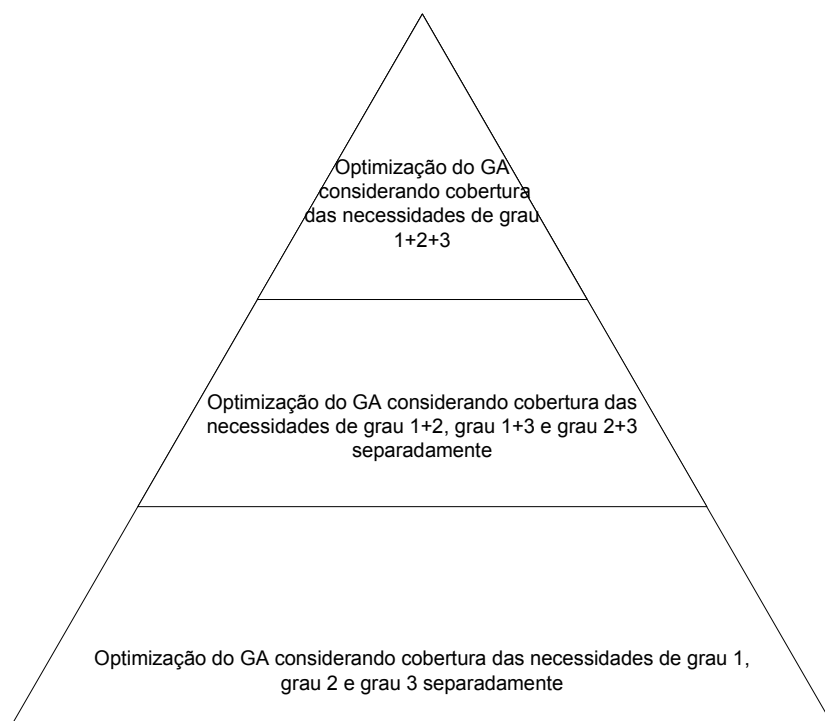


Figura 3.1: Pirâmide hierárquica que esquematiza a evolução do GA desenvolvido em [12].

arco representa as alocações dos enfermeiros, seus custos e fluxos mínimos (restrições), e termina no dia 29, com tantos nós quantos enfermeiros, mais uma vez. Pelo nó de partida, quantidade de fluxo entre os vários nós entre $d-1$ e 28 e pelo nó de saída, é possível determinar as alocações dos vários enfermeiros.

Para resolver o problema, as autoras apresentam duas metodologias. A primeira recorre a “Integer Linear Programming” (programação linear inteira – ILP), sendo o modelo bem descrito, com os seus parâmetros e variáveis bem explicados. A segunda é uma heurística baseada em regras de alocação ordenadas hierarquicamente. Essa heurística é iniciada por uma ordenação aleatória das tarefas a serem cumpridas em todo o horizonte temporal, desde o dia da ausência, sendo depois aplicadas cinco regras de alocação por ordem hierárquica. Os resultados da heurística são comparados com os do modelo ILP.

Em [8], as autoras utilizam a heurística construtiva anteriormente desenvolvida e uma sua variante, com o acrescento de ordenar aleatoriamente os enfermeiros (em vez de os listar por index associado, como anteriormente),

além das tarefas, aumentando a diversidade do espaço pesquisado. Em seguida descrevem um GA que desenvolveram, com representações das tarefas por números naturais, dois tipos distintos de *crossover*, dois tipos de mutação e utilizando elitismo. No final, ainda apresentam uma hibridização das duas heurísticas desenvolvidas (construtiva e GA), concluindo com uma análise dos resultados alcançados pelos diferentes métodos heurísticos.

Capítulo 4

Modelo Proposto

4.1 Modelo de *Nurse Rostering*

O modelo proposto para resolução do problema e aqui apresentado foi criado com base nos modelos considerados e expostos nos trabalhos referidos na bibliografia. Foi também desenvolvido respeitando as considerações anteriormente expressas. Por fim, foi inspirado no modelo matemático desenvolvido no trabalho de fim de curso do autor, em [25], inserido no Projecto Chronos.

4.1.1 Índices

$e \Rightarrow$ enfermeiro e com $e \in \{1, \dots, E\}$

Irá existir um índice por cada enfermeiro existente.

$t \Rightarrow$ Turno s com $s \in \{1, 2, 3\}$

O turno 1 corresponde sempre a uma manhã, o turno 2 a uma tarde e o turno 3 a uma noite.

$d \Rightarrow$ Dia d com $d \in \{1, \dots, D\}$

$g \Rightarrow$ grau de enfermeiros t com $t \in \{1, 2, 3\}$

Grau 1 é o grau mais elevado e grau 3 é o mais baixo.

4.1.2 Parâmetros

G_{eg} Matriz de graus de enfermeiros, em que

$$G_{eg} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ tiver competências do grau } g \\ 0, & \text{se não.} \end{cases}$$

$\underline{R_{tdg}}$ Matriz de necessidades, com $R_{tdg} \in \mathbb{N}$, em que

R_{tdg} = Número de enfermeiros de grau g necessários para o turno t do dia d .

$\underline{S_d}$ Vector que indica os sábados, em que

$$S_d = \begin{cases} 1, & \text{se o dia } d \text{ é um sábado} \\ 0, & \text{se não.} \end{cases}$$

$\underline{D_d}$ Vector que indica os domingos, em que

$$D_d = \begin{cases} 1, & \text{se o dia } d \text{ é um domingo} \\ 0, & \text{se não.} \end{cases}$$

$\underline{FM}$ Número máximo de fins-de-semana que cada enfermeiro poderá trabalhar durante o período de planeamento.

Usualmente, e num período de planeamento de 28 dias (4 semanas) como o considerado, este parâmetro toma o valor 2. Como será demonstrado, nas instâncias criadas para este problema, esse valor é demasiado restritivo, optando-se por utilizar o valor 3 nos testes realizados.

$\underline{Maxa_e}$ Número máximo de atribuições permitidas para o enfermeiro e no período de planeamento.

$\underline{Mxc_e}$ Número máximo de atribuições consecutivas possíveis para o enfermeiro e no período de planeamento.

$\underline{Mnc_e}$ Número mínimo de atribuições consecutivas possíveis para o enfermeiro e no período de planeamento.

Mlc_e Número máximo de dias livres consecutivos para o enfermeiro e no período de planeamento.

L_{etd} Matriz de preferências, em que

$$L_{etd} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ pretende trabalhar no turno } t \text{ do dia } d \\ 0, & \text{se não.} \end{cases}$$

Lista de penalizações: Existem as seguintes penalizações para que o modelo contemple os custos dos enfermeiros e as suas preferências:

c_g – custo salarial associado à alocação de um enfermeiro do grau g , a um qualquer turno;

p – penalização associada ao desrespeito de uma preferência de um qualquer enfermeiro;

a_{12} – custo acrescido por um enfermeiro de grau 1 trabalhar em tarefas de grau 2, associada à variável a_{12etd} ;

a_2 – custo acrescido por um enfermeiro de grau 2 trabalhar em tarefas de grau 3, associada à variável a_{2etd} ;

a_{13} – custo acrescido por um enfermeiro de grau 1 trabalhar em tarefas de grau 3, associada à variável a_{13etd} ;

$$a_{12} \leq a_2 < a_{13}$$

4.1.3 Variáveis

Variáveis de Decisão

$$x_{etd} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ está alocado ao turno } t \text{ no dia } d, \\ 0, & \text{se não.} \end{cases}$$

Variáveis auxiliares

Associadas à penalização por enfermeiros estarem a efectuar um trabalho menos graduado:

$$a_{12etd} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ de grau 1 trabalha como grau 2 no turno } t \\ & \text{do dia } d \\ 0, & \text{se não.} \end{cases}$$

$$a_{2etd} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ de grau 2 trabalha como grau 3 no turno } t \\ & \text{do dia } d \\ 0, & \text{se não.} \end{cases}$$

$$a_{13etd} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ de grau 1 trabalha como grau 3 no turno } t \\ & \text{do dia } d \\ 0, & \text{se não.} \end{cases}$$

Associadas à alocação de fins-de-semana:

$$\delta_{etd} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ está alocado a um turno } t, \text{ num dia } d \\ & \text{pertencente a um fim-de-semana,} \\ 0, & \text{se não.} \end{cases}$$

Associadas à atribuição mínima de dias consecutivos:

$$\alpha_{etd} = \begin{cases} 1, & \text{se o enfermeiro } e \text{ está alocado ao turno } t \text{ no dia } d, \\ 0, & \text{se não.} \end{cases}$$

4.1.4 Modelização

Foram desenvolvidos dois modelos matemáticos em programação inteira mista (MIP) representativos do problema de *Nurse Rostering*. O primeiro modelo (“Modelo Pequeno”) foi concebido para ser o mais simples e pequeno possível, apenas contemplando as “hard constraints”. O segundo modelo (“Modelo Grande”) implementa algumas das “soft constraints” apresentadas, como sendo “hard constraints”, ou seja, obrigatórias. Neste modelo algumas das “soft constraints”, que é previsto contemplar, não foram consideradas:

- Disponibilidade dos enfermeiros;
- Restrições relativas a feriados;
- Contactos entre os enfermeiros (tutorias e impedimentos).

Outras foram transformadas em “hard constraints”:

- restrições de fins-de-semana;
- limites de atribuições e consecutividades.

Outras, ainda, continuaram representadas no modelo como “soft constraints”, como penalizações na *função objectivo*:

- permissão de enfermeiros fazerem tarefas de graus inferiores, com custos associados;
- preferências dos enfermeiros.

Modelo Pequeno

Restrições

$$\forall_d \forall_t \forall_g \quad \sum_e x_{etd} \cdot G_{eg} \geq R_{tdg} \quad (4.1)$$

$$\forall_e \forall_d \quad \sum_t x_{etd} \leq 1 \quad (4.2)$$

$$x_{etd} \in \{0, 1\} \quad (4.3)$$

(4.1) garante que são cobertas todas as necessidades, para todos os graus de enfermeiros, em todos os turnos de todos os dias.

(4.2) obriga a que cada enfermeiro esteja alocado a um único turno por dia.

(4.3) assegura que a variável de decisão é binária.

Função Objectivo

$$\min \quad \sum_e \sum_g \sum_d \sum_t x_{etd} \cdot G_{eg} \cdot c_g \quad (4.4)$$

A função objectivo deste problema, (4.4), apenas tem em consideração os custos salariais de trabalho de cada enfermeiro que é alocado.

Modelo Grande

Restrições

$$\forall_d \forall_t \quad \sum_e x_{etd} \cdot G_{e1} \geq R_{td1} \quad (4.5)$$

$$\forall_d \forall_t \quad \sum_e (x_{etd} \cdot G_{e2} + G_{e1} \cdot a_{12etd}) \geq R_{td2} \quad (4.6)$$

$$\forall_d \forall_t \quad \sum_e (x_{etd} \cdot G_{e3} + G_{e2} \cdot a_{2etd} + G_{e1} \cdot a_{13etd}) \geq R_{td3} \quad (4.7)$$

$$\forall_e \forall_d \forall_t \quad x_{etd} \cdot G_{e1} - a_{12etd} \geq 0 \quad (4.8)$$

$$\forall_e \forall_d \forall_t \quad x_{etd} \cdot G_{e1} - a_{13etd} \geq 0 \quad (4.9)$$

$$\forall_e \forall_d \forall_t \quad x_{etd} \cdot G_{e2} - a_{2etd} \geq 0 \quad (4.10)$$

$$\forall_d \forall_t \quad \sum_e x_{etd} \geq \sum_g R_{tdg} \quad (4.11)$$

$$\forall_e \forall_d \quad \sum_t x_{etd} \leq 1 \quad (4.12)$$

Restrições relacionadas com fins-de-semana:

$$\forall_e \forall_d \forall_t \quad S_d \cdot x_{etd} - \delta_{etd} = 0 \quad (4.13)$$

$$\forall_e \forall_d \forall_t \quad D_d \cdot x_{etd} - \delta_{etd} = 0 \quad (4.14)$$

$$\forall_e \forall_d \forall_t \quad x_{e,t,d-1} + S_d \cdot x_{etd} + D_{d+1} \cdot x_{e,t,d+1} \geq 3 \cdot \delta_{etd} \quad (4.15)$$

$$\forall_e \forall_d \forall_t \quad x_{e,t,d-2} + S_{d-1} \cdot x_{e,t,d-1} + D_d \cdot x_{etd} \geq 3 \cdot \delta_{etd} \quad (4.16)$$

$$\forall_e \quad \sum_d \sum_t S_d \cdot x_{etd} \leq FM \quad (4.17)$$

Restrições relacionadas com atribuições e consecutividades:

$$\forall_e \quad \sum_d \sum_t x_{etd} \leq Maxa_e \quad (4.18)$$

$$\forall_e \forall_{d \in \{1, \dots, D-Mxc_e\}} \quad \sum_{d_0=d}^{d+Mxc_e+1} \sum_t x_{etd_0} \leq Mxc_e \quad (4.19)$$

$$\forall_e \quad \sum_t x_{et1} - \alpha_{e1} \leq 0 \quad (4.20)$$

$$\forall_e \forall_{d \in \{2, \dots, D\}} \quad \sum_t x_{etd} - \alpha_{ed} - \alpha_{e,d-1} \leq 0 \quad (4.21)$$

$$\forall_e \forall_{d \in \{1, \dots, D-Mnc_e+1\}} \quad \sum_{d_0=d}^{d+Mnc_e-1} \sum_t x_{etd_0} \geq Mnc_e \cdot \alpha_{ed} \quad (4.22)$$

$$\forall_e \forall_{d \in \{1, \dots, D - Mlc_e\}} \sum_{d_0=d}^{d+Mlc_e+1} \sum_t x_{etd_0} \geq 1 \quad (4.23)$$

Domínios das variáveis:

$$\begin{aligned} x_{etd} &\in \{0, 1\} \\ a_{12etd} &\in \{0, 1\}, \quad a_{2etd} \in \{0, 1\}, \quad a_{13etd} \in \{0, 1\} \\ \delta_{etd} &\in \{0, 1\}, \quad \alpha_{ed} \in \{0, 1\} \end{aligned} \quad (4.24)$$

(4.5), (4.6) e (4.7) substituem a restrição (4.1) do “Modelo Pequeno”, possibilitando, em conjunto com (4.8), (4.9) e (4.10), que enfermeiros de grau superior sejam alocados a tarefas de grau inferior, com um determinado custo associado, presente na função objectivo deste modelo.

(4.11) garante que todas as necessidades sejam cobertas em cada turno de cada dia, e que, por um enfermeiro de grau 1 estar a trabalhar num determinado turno em tarefas do seu grau, o modelo não assuma que este mesmo enfermeiro possa desempenhar funções dos graus inferiores no mesmo turno, reduzindo os custos associados.

(4.12) obriga a que cada enfermeiro esteja alocado a apenas um turno por dia.

(4.13) e (4.15) garantem que, se um sábado, num qualquer fim-de-semana, for atribuído ao enfermeiro e , o mesmo turno no domingo e na sexta-feira da mesma semana também lhe serão atribuídos.

(4.14) e (4.16) garantem que, se um domingo, num qualquer fim-de-semana, for atribuído ao enfermeiro e , o mesmo turno no sábado e na sexta-feira da mesma semana também lhe serão atribuídos.

(4.17) garante que não serão alocados mais do que FM fins-de-semana ao enfermeiro e durante o período de planeamento.

(4.18) não permite que sejam atribuídos mais turnos do que $Maxa_e$ ao enfermeiro e .

(4.19) e (4.20), (4.21) e (4.22) não permitem que sejam atribuídos mais ou menos turnos consecutivos ao enfermeiro e do que Mxc_e e Mnc_e , respectivamente.

(4.23) obriga a que não sejam atribuídos mais dias livres ao enfermeiro e do que o permitido por Mlc_e .

(4.24) garante que as variáveis são todas binárias.

Função Objectivo

$$\begin{aligned}
 \min \quad & \sum_e \sum_d \sum_t \sum_g x_{etd} \cdot G_{eg} \cdot c_g + \sum_e \sum_d \sum_t x_{etd} \cdot (1 - L_{etd}) \cdot p + \\
 & + \sum_e \sum_d (a_{12etd} \cdot a_{12} + a_{2etd} \cdot a_2 + a_{13etd} \cdot a_{13})
 \end{aligned} \tag{4.25}$$

A função objectivo deste problema, (4.25), além de ter em consideração os custos salariais de trabalho de cada enfermeiro que é alocado, também contempla custos por atribuições de turnos com tarefas de graus inferiores a cada enfermeiro e penalizações por desrespeito de preferências dos enfermeiros.

4.2 Implementação

Após definidos os modelos matemáticos a utilizar para resolver o problema abordado, é necessário de testar o seu funcionamento. Para isso implementou-se o modelo de modo a que, de forma sistemática e automática, recolhendo os dados que definem cada cenário particular, ele seja escrito em ficheiros, com um formato que permita tratá-lo e resolvê-lo por um programa resolvidor de modelos deste género.

Foi necessário também criar uma interface que permitisse, de forma simples, rápida e intuitiva, criar o cenário a testar, exportar os dados para ficheiros com formatos que facilitem a sua utilização na criação do modelo correspondente, e ler os resultados obtidos, apresentando-os numa forma de leitura fácil.

Com estes dois propósitos e com base nos trabalhos já desenvolvidos em [25], foram criadas duas aplicações de criação de modelos (uma que cria o “Modelo Pequeno” e outra o “Modelo Grande”) e um ficheiro MS Excel, que permite a criação muito simplificada de um cenário de teste (e mesmo de cenários reais), a exportação dos dados e a leitura da solução.

4.2.1 Criação de Cenário e Leitura da Solução

A criação do cenário para o problema específico a resolver deve poder fazer-se de um modo simples e rápido. Assim permite-se que todo o processo de teste e/ou resolução de problemas (incluindo a criação do cenário e a leitura da solução) não consuma muito mais tempo do que o necessário para o cálculo de solução para o problema.

Com estes propósitos foi criado um ficheiro em MS Excel, baseado no ficheiro MS Excel do trabalho [25]. Este ficheiro tem várias folhas de cálculo, como é visível na Figura 4.1.



Figura 4.1: Separadores das folhas de cálculo existentes no ficheiro de MS Excel criado.

Na folha “Indices”, não só se define os índices do problema (n^o de enfermeiros, n^o de dias no horizonte temporal, n^o de turnos por dia e n^o de graus),

4.2. IMPLEMENTAÇÃO

como também existem botões para certas funções, ligados a instruções feitas em VBA (Visual Basic Applications), através dos quais: se cria as matrizes nas restantes folhas, para preenchimento com os dados necessários à definição do problema; se exporta esses dados para ficheiros com o formato anteriormente apresentado; e se lê o resultado, do ficheiro de saída do resolvedor, apresentando-o na folha “Resultado” (esta folha está representada na Figura 4.2).

Índices	QtdÍndices	Quantidade	
Enfermeiros	Enf	20	e
Dias	Dia	28	d
Turnos	Turno	3	t
Graus	Grau	3	g

a modificação deste valor exige uma alteração manual das matrizes com este índice

a modificação deste valor exige uma alteração manual das matrizes com este índice

Passos para a Criação do Problema		
1°	Preencha os índices acima	
2°	Gere Matrizes	Gerar Matrizes
3°	Exporte matrizes	Exportar Matrizes
4°	Leia Dados	Ler Dados

Figura 4.2: Folha “Índices” no ficheiro MS Excel criado.

Depois de se definir os índices do cenário pretendido e criadas as matrizes, atribuir-se-á valores a todas as matrizes e parâmetros do cenário de teste.

Na folha chamada “Parametros” define-se o valor dos parâmetros seguintes: FM , $Maxa$, Mxc , Mnc e Mlc . Nesta folha há também dois quadros em que estão definidos todos os parâmetros a preencher e é apresentado o cálculo de uma estimativa da quantidade de enfermeiros que estarão a trabalhar por dia, independentemente do número máximo de atribuições durante o horizonte temporal para cada enfermeiro.

Em “Custos” atribui-se valores a $c1$, $c2$, $c3$, p , $a12$, $a2$ e $a13$. Nesta folha existe também uma definição de cada custo, para que o utilizador saiba exactamente que valores de custos ou de penalizações está a definir.

Em “FdS” define-se os vectores que indicam os sábados e domingos.

“Necessidades” tem três quadros matriciais, um para cada grau de enfermeiros. Cada quadro tem os três turnos em cada dia para preencher com as necessidades de enfermeiros respectivas.

As folhas “Preferencias_1”, “Preferencias_2” e “Preferencias_3” são iguais e cada uma tem um quadro para preencher com as preferências de cada enfermeiro em cada dia. A primeira folha corresponde às preferências para o turno da manhã, a segunda para o turno da tarde e a última para o turno da noite.

A última folha do ficheiro, “Resultado”, é a folha onde será apresentado o resultado para o cenário projectado. Este resultado é obtido após:

- Exportar as matrizes de dados criadas, com respectivo botão na folha “Indices”;
- Na linha de comandos;
 1. criar o modelo matemático desejado;
 2. correr em seguida o resolvidor com o modelo “.lp” criado no passo anterior;
- depois de estar resolvido e ter uma solução inteira, no ficheiro MS Excel, ler os dados de saída do resolvidor, clicando no botão “Ler Dados”.

Nesta folha é apresentada a distribuição dos enfermeiros pelos dias, assinalando a verde os dias a que cada enfermeiro está alocado. Existe uma tabela com estes resultados para cada turno (a tabela de cima corresponde ao turno da manhã, a do meio ao turno da tarde e a de baixo ao turno da noite). É também apresentado o valor da função objectivo e existe local para registar o “tempo de execução” do resolvidor, a “memória utilizada” e o número de linhas e colunas do problema (restrições e variáveis, respectivamente).

4.2.2 Criação de Instâncias do Modelo

Cada conjunto de dados, que define um cenário (instância do modelo), irá originar a criação de um modelo de representação desse mesmo cenário. Cada modelo criado será diferente de todos os outros modelos que representam os restantes cenários possíveis. Desta forma, e de modo a sistematizar a criação de cada modelo, que poderá ter um elevado número de características distintas, criou-se uma aplicação. Esta aplicação lê os dados e parâmetros

que caracterizam o cenário em questão e cria um ficheiro com o modelo representativo desse cenário, que permitirá a obtenção de uma solução para o problema particular.

A aplicação foi criada em linguagem C e o ficheiro, que é criado com o modelo representativo de cada cenário, é escrito seguindo as regras da linguagem CPLEX LP (em [26]).

Após a criação do modelo na linguagem CPLEX LP, a procura de uma solução inteira para o modelo foi feita recorrendo a resolvedores que lêem este tipo de ficheiros.

A aplicação foi desenvolvida em duas partes:

1. Leitura dos dados que caracterizam o cenário em questão;
2. Criação do modelo representativo do cenário.

Leitura de Dados

Os dados que caracterizam o problema específico a resolver (matrizes ou vectores de dados) são guardados em ficheiros de texto. Existem matrizes com mais de duas dimensões, nestes casos são guardadas em vários ficheiros de texto. Como nunca ultrapassam as três dimensões, respeitando as definições dos modelos matemáticos anteriormente apresentadas, uma das dimensões estabelece o número de ficheiros associados a essas matrizes tridimensionais, passando, cada uma, a estar registada como um conjunto de matrizes bidimensionais.

Ficheiros de Dados Cada ficheiro de dados segue uma lógica simples de escrita, permitindo que a aplicação teste a coerência dos dados lidos com o que espera ler para a criação do modelo matemático. Na primeira linha do ficheiro de texto encontram-se escritas as dimensões da matriz (“nº de linhas” e “nº de colunas”, separados por um espaço). Na linha seguinte está o nome da matriz, onde, caso se trate de uma matriz de 3 dimensões, terá incluída a terceira dimensão (*exemplo*: “R2” para a matriz de necessidades de enfermeiros de grau 2). Nas linhas restantes está escrita a matriz: cada linha do ficheiro corresponde a uma linha da matriz e, em cada linha, as colunas estão separadas por espaços (*exemplo* da disposição de um ficheiro de dados na Figura 4.3).

Existem ficheiros para os seguintes dados:

- índices;

3	28
R1	
1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0	1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0

- custos;
- graus dos enfermeiros;
- valores de máximos e mínimos de atribuições e consecutividades:
um ficheiro para cada um dos seguintes parâmetros: FM, Maxa, Mxc, Mnc e Mlc.
- identificação de sábados e domingos (um ficheiro para cada caso);
- necessidades nos vários dias e turnos (um ficheiro para cada grau);
- preferências dos enfermeiros nos vários dias (um ficheiro para cada tipo de turno).

Leitura e Processamento Para a leitura dos dados é necessário saber quais as dimensões do problema. A primeira leitura a fazer é então a do ficheiro de texto com os índices do problema. Inicialmente é verificado se o tamanho do vector de índices está correcto e, em seguida, são lidos os valores. Caso tal ficheiro não exista ou o tamanho do vector não seja o esperado, a aplicação assume que os valores dos índices são os correspondentes a 20 enfermeiros, 28 dias de horizonte temporal, 3 turnos por dia e 3 graus de enfermeiros.

Depois de sabidas as dimensões do problema, pode passar-se à leitura de todos os ficheiros restantes, com os dados do problema. Essa leitura é feita segundo o Algoritmo 4.1.

Para as matrizes de três dimensões, as matrizes de necessidades e as matrizes de preferências, todo este processo é englobado num ciclo. Nesse ciclo este algoritmo é repetido tantas vezes quantos os ficheiros que compõem a matriz em causa.

4.2. IMPLEMENTAÇÃO

1. *Abre o ficheiro (caso o ficheiro não exista, devolve “erro de abertura” com mensagem no ficheiro de log);*
2. *Lê as dimensões da matriz:*
 - (a) *caso a matriz esteja sub-dimensionada devolve um “erro de leitura” deixando uma mensagem que especifica o erro no ficheiro de log;*
 - (b) *caso a matriz esteja sobre dimensionada, escreve uma mensagem de alerta para tal no ficheiro de log, mas continua a leitura da matriz, desprezando os dados a mais;*
3. *Lê os dados da matriz, escrevendo-os para a memória do computador, no vector de dados correspondente.*

Algoritmo 4.1: Algoritmo das funções de leitura dos ficheiros de dados do problema a resolver.

Este processo de leitura de ficheiros é utilizado nas duas aplicações existentes, quer para a criação do “Modelo Pequeno”, quer para a criação do “Modelo Grande”. Para cada um destes casos, os ficheiros a ler são naturalmente diferentes: enquanto para o “Modelo Grande” são lidos todos os ficheiros, para o “Modelo Pequeno” apenas são lidos os ficheiros que contêm os graus dos enfermeiros, as necessidades e os custos.

Criação do Modelo

Após a leitura dos dados que definem o problema a resolver é necessário criar um ficheiro com o modelo matemático que permita a resolução do problema, recorrendo a um resolvidor de CPLEX. Tal ficheiro deverá cumprir as regras estabelecidas para os ficheiros CPLEX LP (estas regras poderão ser consultadas em [26]).

Apesar de existirem duas aplicações diferentes (uma para o “Modelo Pequeno” e outra para o “Modelo Grande”), o seu funcionamento é idêntico. Assim sendo, apresenta-se aqui o modo de funcionamento geral (algoritmos que descrevem ambos os funcionamentos), sendo depois descritas as particularidades.

O ficheiro CPLEX LP divide-se em três partes:

- Início e definição da função objectivo do problema;
- Definição das várias restrições impostas ao problema;

- Declaração do tipo de variáveis em causa e assinalação do fim do ficheiro.

A função objectivo do problema é escrita segundo o Algoritmo 4.2.

1. *Escreve numa linha isolada “MINIMIZE”;*
2. *Inicia os ciclos de índices dos somatórios existentes na função objectivo. Em cada iteração do ciclo:*
 - (a) *calcula o peso total (soma de todos os custos) associado à variável a escrever (de modo a que não haja repetições da mesma variável ao longo da função – situação não aceite pelo CPLEX);*
 - (b) *se esse peso for nulo:*
 - i. *passa à variável seguinte;*
 - (c) *caso contrário:*
 - i. *adiciona o peso e a variável à função existente, escrevendo tais valores no ficheiro CPLEX LP e garantindo que:*
 - *se ultrapassar o tamanho de linha permitido, escreve numa nova linha;*
 - *se for o primeiro valor a ser escrito, não adiciona, mas inicia a função objectivo (não escreve o sinal “+”).*

Algoritmo 4.2: Algoritmo da função de escrita da função objectivo do problema.

Depois de definida a função objectivo do problema, escrevem-se as restrições. Estas, de uma forma geral, têm o seguinte formato:

$$\forall_A \quad \sum_B C \text{ “relação de ordem” } D$$

“A” e “B” podem ser mais do que um índice e podem ser limitados a um conjunto pertencente ao domínio do índice em causa. Neste último caso, o ciclo para criação do somatório ou das várias restrições (consoante se trate de “B” ou “A”, respectivamente) será limitado a esse mesmo conjunto.

“C” tanto pode ser uma variável como uma variável multiplicada por um valor (normalmente multiplicada por um parâmetro). Ou seja, “C” representa cada uma das parcelas da restrição que contenham variáveis.

“relação de ordem” é, consoante as restrições, $\leq$, $\geq$ ou $=$.

4.2. IMPLEMENTAÇÃO

“D” é um valor. Tanto pode ser um parâmetro ou operação entre parâmetros, como um valor já definido.

As restrições são criadas através de funções que respeitam o Algoritmo 4.3.

1. *Escreve numa linha isolada “SUBJECT TO” (caso se trate da primeira restrição do problema);*
2. *Inicia os ciclos de índices que definem quantas restrições serão escritas e de índices dos somatórios que cada restrição terá. Em cada iteração do ciclo:*
 - (a) *Se for o início da restrição (para os “A” actuais):*
 - i. *escreve o nome da restrição;*
 - ii. *escreve a informação em “C”;*
 - iii. *memoriza o tamanho do que foi escrito na linha actual;*
 - (b) *caso contrário:*
 - i. *Se for a última parcela (“C”) a ser escrita na restrição:*
 - *adiciona o que se vai escrever ao tamanho da linha actual;*
 - *se exceder o tamanho permitido:*
 - muda de linha;*
 - faz “reset” ao contador de tamanho de linha actual;*
 - adiciona o que se vai escrever ao tamanho da linha actual;*
 - *escreve a informação em “C”, a “relação de ordem” e a informação em “D”;*
 - ii. *caso não seja a última parcela da restrição actual:*
 - *adiciona o que se vai escrever ao tamanho da linha actual;*
 - *se exceder o tamanho permitido:*
 - muda de linha;*
 - faz “reset” ao contador de tamanho de linha actual;*
 - adiciona o que se vai escrever ao tamanho da linha actual;*
 - *adiciona, devidamente, à restrição a informação em “C”.*

Algoritmo 4.3: Algoritmo das funções de escrita das restrições do problema.

Por vezes, para poupar texto escrito, poupando também no tamanho do ficheiro CPLEX LP, caso “C” seja composto por uma variável multiplicada por um valor numérico (quer este seja um parâmetro ou uma operação entre parâmetros), antes de escrever no ficheiro essa parcela e adicionar o seu

tamanho ao tamanho da linha actual, testa-se o valor numérico:

- se for diferente de 0, procede-se normalmente;
- se for igual a 0, passa-se de imediato para a parcela seguinte (o “C” seguinte).

No final do ficheiro CPLEX LP são declaradas as variáveis, especificando-se o seu tipo. Nestes modelos todas as variáveis são binárias. Esta declaração é então feita por uma função representada no Algoritmo 4.4.

1. *Escreve numa linha isolada “BINARY”;*
2. *Inicia os ciclos de índices que definem quantas variáveis serão escritas. Em cada iteração do ciclo:*
 - (a) *Escreve a variável, separando-a das restantes com um espaço e garantindo que:*
 - i. *se ultrapassar o tamanho de linha permitido, a escreve numa nova linha.*

Algoritmo 4.4: Algoritmo da função de escrita das declarações das variáveis do problema.

Após todas as declarações, ainda se escreve numa linha separada: “END”.

4.3 Testes aos Modelos Matemáticos

Nesta secção são apresentados os cenários e as instâncias que foram elaborados para testar os modelos apresentados anteriormente. São também apresentados e discutidos os resultados de todos esses testes, comparando-os entre si. Através destes testes pretende-se encontrar qual o modelo mais adequado a ser um bom ponto de partida para a elaboração de um algoritmo heurístico que resolva o problema de *Nurse Rostering* aqui abordado, obtendo uma boa solução num curto intervalo de tempo.

Pretende-se também fazer uma análise de sensibilidades aos modelos matemáticos desenvolvidos.

4.3.1 Instâncias Adaptadas de [24]

Inicialmente, para testar o bom funcionamento do “Modelo Grande” desenvolvido, adaptaram-se instâncias, já testadas e com soluções óptimas apresentadas, de escalonamento de trabalhadores (“Workforce Scheduling”). Estas instâncias encontram-se em [24]. Este sítio da internet contém várias instâncias já testadas para problemas típicos de “Investigação Operacional”.

No conjunto de onde foram retiradas as instâncias aqui usadas estavam apresentadas 20 instâncias diferentes. Foram escolhidas 2. Uma de pequena dimensão, apenas para testar o bom funcionamento do modelo e a proximidade da solução obtida face à apresentada pelos autores das instâncias, visto que, nestes testes, o modelo utilizado (desenvolvido neste trabalho) é certamente diferente do modelo desenvolvido pelos autores das instâncias, os quais tratam de problemas gerais de “Workforce Scheduling” e não particularmente de problemas de *Nurse Rostering*. Outra de dimensão média e idêntica à dimensão dos testes que se projectou fazer ao modelo com os cenários criados.

Para que pudessem ser utilizadas no contexto deste problema, estas instâncias foram adaptadas ao problema de *Nurse Rostering* apresentado. Nem todos os seus dados foram utilizados, foram seleccionados e adaptados apenas os que têm interesse para este trabalho. Por exemplo, as sequências de atribuições de turnos não permitidas não foram proibidas. Outro parâmetro que teve que ser adaptado foi o número de graus: nestas instâncias os empregados não são diferenciados, têm todos as mesmas competências.

Com todas estas modificações e adaptações, as soluções obtidas não foram iguais às apresentadas no documento de soluções para as instâncias, como

seria de esperar. De qualquer forma essas soluções serviram como indicador de existência de uma solução óptima, o que se verificou em todos os casos testados.

A primeira instância está detalhada da seguinte forma:

- Length of the schedule: 7
- Number of Employees: 9
- Number of Shifts: 3
- Temporal Requirements Matrix:
2 2 2 2 2 2 2
2 2 2 3 3 3 2
2 2 2 2 2 2 2
- ShiftName, Start, Length, Name, MinlengthOfBlocks, MaxLengthOfBlocks:
D 360 480 2 7
A 840 480 2 6
N 1320 480 2 4
- Minimum and maximum length of days-off blocks: 2, 4
- Minimum and maximum length of work blocks: 4, 7
- Number of not allowed shift sequences:
NrSequencesOfLength2, NrSequencesOfLength3:
3, 0
- Not allowed shift sequences:
N D
N A
A D

Daqui é possível retirar o número de enfermeiros, 9, continua a haver 3 turnos, o horizonte temporal passa a ser 28 dias (em vez dos 7 sugeridos),

retira-se também os requisitos (replicados ao longo das 4 semanas) e pode-se definir:

- ◇ Máximo de alocações no horizonte temporal ($Maxa$) = 28 (todos por defeito);
- ◇ Máximo de alocações consecutivas (Mxc) = 7 (“Maximum length of work blocks”);
- ◇ Mínimo de alocações consecutivas (Mnc) = 2 (“MinlengthOfBlocks”);
- ◇ Máximo de dias livres consecutivos (Mlc) = 4 (“Maximum length of days-off blocks”);
- ◇ Máximo de alocações a fins-de-semana (FM) = 4 (todos por defeito);

Apenas existe um grau de enfermeiros, os custos foram criados, assim como as preferências.

Todas estas modificações (adaptações da instância em [24] ao “Modelo Grande”) provocaram alterações na solução obtida, comparativamente à solução dada pelos autores das instâncias, mas a solução atingida esteve muito perto da óptima em meia hora. O modelo com estes dados adaptados desta instância foi resolvido com o “glpk”, durante 2000 segundos. A verificação de a solução ser a óptima ou não foi feita através da comparação do número de enfermeiros alocados em cada dia com o número de enfermeiros necessários nesse mesmo dia.

O escalonamento obtido depois de resolvido o modelo, com os dados desta instância, está apresentado na Figura 4.4. De notar que aos domingos, apesar de apenas serem necessários 6 enfermeiros, estão sempre alocados 7, uma vez que aos sábados e às sextas-feiras há necessidade de 7 enfermeiros e o conjunto de restrições obriga um enfermeiro alocado a um sábado a trabalhar na sexta e no domingo desse mesmo fim-de-semana. Há também um dia, a última sexta-feira, em que estão quatro enfermeiros alocado ao turno da tarde, em vez de três (quantidade necessária). Os restantes dias foram preenchidos com o exacto número de enfermeiros necessários por turno, indicando que a solução é satisfatória e, pelo menos, próxima da óptima.

A segunda instância é definida pelo seguinte:

- Length of the schedule: 7
- Number of Employees: 17
- Number of Shifts: 3
- Temporal Requirements Matrix:
 - 5 4 4 4 4 4 3
 - 5 4 4 4 4 4 4
 - 4 3 3 3 4 4 4
- ShiftName, Start, Length, Name, MinlengthOfBlocks, MaxLengthOfBlocks:
 - D 360 480 2 7
 - A 840 480 2 6
 - N 1320 480 2 5
- Minimum and maximum length of days-off blocks: 2, 4
- Minimum and maximum length of work blocks: 4, 7
- Number of not allowed shift sequences:
 - NrSequencesOfLength2, NrSequencesOfLength3:
 - 3, 0
- Not allowed shift sequences:
 - N D
 - N A
 - A D

Após uma adaptação idêntica à anterior e depois de correr o modelo com esta instância em 2000 segundos, tal como tinha acontecido no caso precedente, obteve-se também uma solução. Esta não está tão próxima da óptima como a obtida para o caso anterior. Ainda assim não se encontra muito afastada e é uma boa solução, havendo alguns dias com 1, 2 e, um caso com 3 enfermeiros em excesso (dia 2, com dois enfermeiros a mais no turno da

4.3. TESTES AOS MODELOS MATEMÁTICOS

		dias																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
enfermeiros	1	T	T	T		T	T	T	N	N	N		T	T	T	T	N	T	T					T	T	N	N	N	N
	2	M	M	M	M				M	M	T	M	T	T	T			M	M	T	T	T	T	M		M	M	M	M
	3			T	T	T	T	T	T	T			T	T	T	T			T	T	T	T				T	T		
	4	N	N			N	N	N	N	T	T		T	N	N	N	N		N	T				N	T	N			
	5	M	M			M	M	M	M	M		M	M	M	M	M	M		M	M	M	M	M		M	M	M	M	M
	6	N	N	N		M	M	M	N	N	N	N					M	N	N	N	N	N	N		N	N	N	N	N
	7			M	T					M	M		M	M	M	M	T	M			M	M	M	M	M		T	T	T
	8	T	T	N	N	N	N	N					N	N	N	N	N	N		N	N	N	N	N		T	T	T	T
	9				T	T	T	T				T	T				T	T		T	T	T	T		T	T	T	T	T

Figura 4.4: Resultado do “Modelo Grande”, para a primeira instância adaptada de [24].

tarde e um no turno da noite), em todo o horizonte temporal, como se pode constatar no escalonamento obtido (Figura 4.5).

		dias																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
enfermeiros	1	T	T	N					N	N	T	T					T	T	N	N	N	N	T		N	N	N	N	N
	2	M	T	T		T	T	T	M	M	M		T	T	T	T	M	M					M	M			T	T	T
	3	T	T	T					T	T	T	T	T	T	T		T	T					T	T	T	T			
	4	N	N	N		N	N	N	T	T	T	T				T	T		T	T	T	T	N			T	N	N	N
	5	M	M	M	M	M	M	M		M	M	M	M	M	M	M		M	M	M	M	M	M		M	M			
	6				N	N	N	M		M	M		N	N	N	N	M		M	N	N	N	M	M	M		N	N	N
	7	M	M	M	M				M	M		M	M	M	M	M	M			M	M	M	M		T	T	M	M	M
	8	T	T	T	T	T	T			N	N						N	N	N					N	N	N	T	T	T
	9	T	T		T	T	T	T	T	T							T	T	T					T	T	T	N	T	T
	10	N	T		N	N	N	N	N			N	T	T	T	T	N			N	N	N	N	N	N	N			
	11	T	M		T	M	M	M	M			M	M	M	M	M			M	M	M	M		M	M	M	M	M	M
	12			T	T				T	T			T	T				T	T					T	T	T			
	13	N	N	N		T	T	T				N	N	N	N	N		N	T	T	T	T	T	N			T	T	T
	14	M	M	M	M				M	M			M	M	M	M				M	M	M	M		M	M			
	15	M	N		N	M	M	M	N		N	M		N	N	N	M	N		N	N	N	N	N			M	M	M
	16			M	M	M	M	M	T		M	M		T	T	T	T		M	M	T	T	T	T		M	M	M	M
	17	N	N		N	N	N	N	N		N	N	N	N	N	N	N	N	T				N	T			N	N	N

Figura 4.5: Resultado do “Modelo Grande”, para a segunda instância adaptada de [24].

4.3.2 Cenários Criados

Todos os cenários criados respeitaram os pressupostos apresentados na descrição do problema:

Existem 20 enfermeiros de 3 graus distintos, planeamentos com um horizonte temporal de 4 semanas – 28 dias – e com 3 turnos por dia (manhã, tarde e noite).

Os enfermeiros estão divididos pelos 3 graus da seguinte forma:

- *do enfermeiro nº 1 ao enfermeiro nº 4 são todos de grau 1;*
- *do enfermeiro nº 5 ao enfermeiro nº 10 são todos de grau 2;*
- *do enfermeiro nº 11 ao enfermeiro nº 20 são todos de grau 3.*

Os restantes parâmetros são adaptados para cada cenário que se está a ter em consideração.

Necessidades

Em cada cenário as necessidades de enfermeiros têm um período de uma semana, repetindo pelas quatro semanas do horizonte temporal.

São apresentados os vários cenários para o “Modelo Pequeno” e para o “Modelo Grande” separadamente. Foi escolhida esta forma de apresentação para permitir maior facilidade na leitura dos cenários e testes que lhes foram feitos, possibilitando também, para os cenários do “Modelo Pequeno”, uma simplificação de apresentação e possibilitando o leitor de se alhear de certos parâmetros que são dispensáveis para este modelo, mas importantes para o “Modelo Grande”.

Todos os cenários usados durante esta fase do trabalho foram construídos com base nas adaptações das instâncias anteriormente apresentadas. Houve a preocupação de os construir com um número de enfermeiros idêntico à maior parte dos casos apresentados nos artigos consultados, que descrevem outros trabalhos realizados nesta área. Alguns dos parâmetros foram também adaptados com base nos dados divulgados nesses artigos e em informação obtida durante um anterior trabalho efectuado pelo autor, em [25].

Um sistema que possa resultar de um trabalho como este terá a função de calcular a melhor solução possível para o problema, recorrendo a um computador sem características especiais, de modo a reduzir os custos de utilização. Desta forma facilita-se que tal sistema seja utilizado no maior número de instituições possíveis. Assim sendo, os testes efectuados neste

trabalho foram corridos com o resolvidor “glpk” (um resolvidor de problemas de programação linear e programação inteira mista gratuito e de fácil utilização), num computador pessoal com as seguintes características:

- Processador: *Intel Pentium M 1,60GHz*;
- Memória RAM: 512MB;
- Disco Rígido: 60 GB.

Para o “Modelo Grande”, os testes efectuados foram também corridos num computador com as condições para calcular a solução óptima (extensa Memória RAM – 8 Gb e um processador *Intel®XEON Dual Core 5160* a 3 GHz/1,333MHz/4Mb). Utilizou-se para tal cálculo o software CPLEX da ILOG.

Modelo Pequeno

Para o “Modelo Pequeno” foram criados cinco cenários diferentes. As diferenças entre cenários residem essencialmente na quantidade de enfermeiros necessários em cada dia da semana.

A criação de necessidades para cada tipo de cenário foi baseada nos requisitos do trabalho desenvolvido em [25]. Aí está explícito que as necessidades dos turnos da manhã são maiores do que as necessidades dos turnos da tarde, que por sua vez, são maiores do que as necessidades dos turnos da noite. No caso actualmente em estudo, as necessidades dos turnos de manhãs e tardes são idênticas, mas mais elevadas do que as nocturnas, em que apenas há a necessidade de garantir um funcionamento mínimo, assumindo que a maioria dos pacientes estará em repouso.

Há uma maior necessidade de enfermeiros de graus inferiores, já que estes são os menos dispendiosos e são habilitados a desempenhar uma grande parte das tarefas normalmente existentes, necessitando apenas de supervisão por parte de enfermeiros superiores. Os enfermeiros superiores poderão ser em menor quantidade e estar mais libertos de tarefas que exijam menor experiência, no entanto estarão também condicionados a prestar algum auxílio e supervisão aos enfermeiros de graus inferiores, embora esta tarefa, em média, não exija um acompanhamento constante.

Necessidades dos vários cenários criados para o “Modelo Pequeno”

Cenário 1:

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	0	0
Tarde	1	1	1	1	1	1	0
Noite	0	0	0	0	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	2	1	2	2	1	1
Tarde	2	1	2	1	1	1	1
Noite	0	0	0	1	1	1	0

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	3	2	3	2	2	2	2
Tarde	1	2	1	2	2	1	1
Noite	1	1	1	0	0	0	1

Este cenário tem quase sempre uma necessidade total de 10 enfermeiros/dia (*necessidade total* é a soma de todas as necessidades para todos os graus). Ao sábado apenas há necessidade de um total de 7 enfermeiros e ao domingo de 6. Estas necessidades são iguais para todas as semanas do horizonte temporal.

Cenário 2:

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	0
Tarde	1	1	1	1	1	1	1
Noite	1	1	0	1	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	2	2	2	2	2	1
Tarde	2	1	2	2	1	1	1
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	4	3	5	4	3	3	2
Tarde	3	2	2	2	3	2	1
Noite	1	1	1	0	0	0	1

4.3. TESTES AOS MODELOS MATEMÁTICOS

Este cenário tem no máximo uma necessidade total de 15 enfermeiros/dia, variando entre os 12 de segunda-feira e os 15 de terça. Ao sábado apenas há necessidade de um total de 11 enfermeiros e ao domingo de 8. Estas necessidades são iguais para todas as semanas do horizonte temporal.

Cenário 3:

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	0
Tarde	1	1	1	1	1	1	1
Noite	1	1	0	1	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	2	2	2	2	2	1
Tarde	1	1	2	1	1	1	1
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	4	3	3	4	3	3	2
Tarde	3	2	2	2	3	2	1
Noite	1	1	1	0	0	0	1

Este cenário, muito semelhante ao cenário 2 apenas não atingindo um máximo tão elevado, tem no máximo uma necessidade total de 13 enfermeiros/dia. Ao sábado apenas há necessidade de um total de 11 enfermeiros e ao domingo de 8. Estas necessidades são iguais para todas as semanas do horizonte temporal.

Cenário 4:

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	0
Tarde	1	1	1	1	1	1	1
Noite	0	1	0	1	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	2	1	2	2	2	1
Tarde	1	1	1	1	1	1	1
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	2	2	3	2	3	2	2
Tarde	2	2	2	2	2	1	1
Noite	1	1	1	0	0	0	1

Este cenário tem no máximo uma necessidade total de 11 enfermeiros/dia, com necessidades totais quase sempre iguais a esse valor, apenas não atingindo um valor tão elevado na segunda-feira e no fim-de-semana. Ao sábado apenas há necessidade de um total de 9 enfermeiros e ao domingo de 8. Estas necessidades são iguais para todas as semanas do horizonte temporal.

Resultados Obtidos Os resultados obtidos encontram-se apresentados em tabelas como a seguinte:

	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D	...
Enf. 1	—	—	—	—	—	—	—	...
Enf. 2	—	—	—	—	—	—	—	...
Enf. 3	1	1	1	1	1	—	—	...
Enf. 4	1	1	1	1	1	1	—	...
Enf. 5	—	—	—	—	—	—	—	...
⋮				⋮				

Em cada resultado existe uma tabela para cada turno: uma primeira para o turno da manhã, uma segunda para o turno da tarde e uma última para o turno da noite. Na tabela, quando um enfermeiro tem um “1” num determinado dia, o enfermeiro está alocado a esse dia.

Os resultados estão apresentados nas Figuras 4.6, 4.7, 4.8 e 4.9 (para facilitar a sua leitura, e como descrito na secção 4.2.1, as células correspondentes a alocações dos enfermeiros, além de terem um “1”, estão também preenchidas a verde).

A conclusão apurada destes cenários é que o modelo é muito redutor relativamente ao pretendido, uma vez que quase reduz o problema a um problema de cobertura de conjuntos. Esta semelhança do problema de “*Rostering*” a problema de cobertura de conjuntos deve-se à excessiva simplicidade do modelo. Pode observar-se nas soluções obtidas (nas figuras atrás indicadas) que apenas alguns enfermeiros trabalham (poucos) e que aqueles que trabalham fazem sempre o mesmo tipo de turno (ou no máximo uma sequência de dois tipos de turno) todos os dias (ou conjuntos de dias) em que há necessidade

4.3. TESTES AOS MODELOS MATEMÁTICOS

Turno 1	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Enf 1																									
Enf 2																									
Enf 3	1	1	1	1	1			1	1	1	1	1			1	1	1	1	1			1	1	1	1
Enf 4																									
Enf 5																									
Enf 6																									
Enf 7					1	1					1	1						1	1						1
Enf 8	1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1
Enf 9		1					1		1					1		1					1		1		1
Enf 10																									
Enf 11																									
Enf 12																									
Enf 13																									
Enf 14																									
Enf 15																									
Enf 16	1	1	1					1	1	1					1	1	1					1	1	1	1
Enf 17	1	1	1	1	1			1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1
Enf 18	1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Enf 19						1							1							1					
Enf 20																									

Turno 2	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Enf 1																									
Enf 2																									
Enf 3																									
Enf 4	1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1
Enf 5																									
Enf 6																									
Enf 7																									
Enf 8																									
Enf 9	1		1	1	1	1		1		1	1	1	1		1		1	1	1	1		1		1	1
Enf 10	1	1	1				1	1	1	1				1	1	1	1	1			1	1	1	1	1
Enf 11																									
Enf 12																									
Enf 13																									
Enf 14																									
Enf 15																									
Enf 16																									
Enf 17																									
Enf 18		1						1							1							1			1
Enf 19	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1	1
Enf 20						1	1	1					1	1	1					1	1	1			1

Turno 3	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Enf 1																									
Enf 2																									
Enf 3																									
Enf 4																									
Enf 5																									
Enf 6																									
Enf 7																									
Enf 8																									
Enf 9					1	1	1				1	1	1					1	1	1					1
Enf 10																									
Enf 11																									
Enf 12																									
Enf 13																									
Enf 14																									
Enf 15																									
Enf 16																									
Enf 17																									
Enf 18																									
Enf 19																									
Enf 20	1	1	1	1				1	1	1	1				1	1	1	1			1	1	1	1	1

Figura 4.6: Resultado do “Modelo Pequeno”, para o cenário 1.

para tal. É assim utilizado o menor número possível de enfermeiros, havendo alguns sem qualquer alocação. Há também enfermeiros que não têm qualquer dia de descanso (por exemplo, o enfermeiro 10 na Figura 4.6).

Assim sendo, este modelo é dispensável para o progresso do trabalho, já que não serve qualquer propósito, nem mesmo o de encontrar rapidamente uma solução inicial admissível e interessante para ser desenvolvida com o *Algoritmo Genético*, sendo mais proveitoso criar soluções iniciais aleatórias, por permitirem maior diversidade entre as soluções possíveis.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Turno 1	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Enf 1																									
Enf 2	1	1		1				1	1		1				1	1		1				1	1		1
Enf 3			1			1	1			1		1	1				1	1		1	1			1	1
Enf 4																									
Enf 5																									
Enf 6			1	1						1	1						1	1						1	1
Enf 7		1	1	1	1	1	1			1	1	1	1			1	1	1	1	1	1	1		1	1
Enf 8	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1
Enf 9		1							1							1							1		
Enf 10																1									
Enf 11																									
Enf 12																									
Enf 13	1		1					1		1					1		1					1			1
Enf 14	1		1					1		1					1		1					1			1
Enf 15	1	1	1	1	1			1	1	1	1	1			1	1	1	1	1			1	1	1	1
Enf 16	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1
Enf 17	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1
Enf 18				1		1	1								1	1	1	1	1	1	1	1	1	1	1
Enf 19															1	1	1	1	1	1	1	1	1	1	1
Enf 20																									

Turno 2	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Enf 1																									
Enf 2																									
Enf 3	1	1		1				1	1		1				1	1		1				1	1		1
Enf 4			1		1	1	1			1		1	1	1			1	1		1	1	1		1	1
Enf 5																									
Enf 6																									
Enf 7																									
Enf 8			1	1						1	1						1	1						1	1
Enf 9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Enf 10	1	1	1	1	1			1	1						1	1						1	1		
Enf 11																									
Enf 12																									
Enf 13																									
Enf 14																									
Enf 15																									
Enf 16																									
Enf 17	1							1							1							1			
Enf 18	1	1	1		1			1	1	1	1				1	1	1	1	1	1	1	1	1	1	1
Enf 19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Enf 20				1	1	1					1	1	1				1	1	1						

Turno 3	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Enf 1																									
Enf 2																									
Enf 3																									
Enf 4	1	1		1				1	1		1				1	1		1				1	1		1
Enf 5																									
Enf 6																									
Enf 7																									
Enf 8																									
Enf 9																									
Enf 10			1	1	1	1	1			1	1	1	1	1			1	1	1	1	1	1		1	1
Enf 11																									
Enf 12																									
Enf 13																									
Enf 14																									
Enf 15																									
Enf 16																									
Enf 17																									
Enf 18																									
Enf 19																									
Enf 20	1	1	1	1				1	1	1	1				1	1	1	1				1	1	1	1

Figura 4.7: Resultado do “Modelo Pequeno”, para o cenário 2.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Turno 1	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Ent 1																									
Ent 2	1	1		1				1	1		1				1	1		1				1	1		1
Ent 3			1			1	1			1		1	1				1		1	1				1	
Ent 4																									
Ent 5																									
Ent 6				1						1							1								1
Ent 7			1	1	1	1	1			1	1	1	1				1	1	1	1				1	1
Ent 8		1		1	1	1	1			1		1	1	1	1		1	1	1	1	1			1	1
Ent 9	1	1						1							1	1		1	1	1	1	1		1	1
Ent 10								1	1							1						1	1		
Ent 11																									
Ent 12																									
Ent 13	1							1								1						1			
Ent 14	1							1								1						1			
Ent 15	1	1	1	1	1			1	1	1	1	1			1	1	1	1	1			1	1	1	1
Ent 16	1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1
Ent 17		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1
Ent 18				1		1	1				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Ent 19						1	1				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Ent 20																									

Turno 2	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Ent 1																									
Ent 2																									
Ent 3	1	1		1		1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Ent 4			1			1	1	1			1	1	1	1	1		1		1	1	1	1		1	1
Ent 5																									
Ent 6																									
Ent 7																									
Ent 8				1						1							1								1
Ent 9			1	1	1	1	1	1			1	1	1	1	1		1	1	1	1	1	1		1	1
Ent 10	1	1						1	1						1	1						1	1		
Ent 11																									
Ent 12																									
Ent 13																									
Ent 14																									
Ent 15																									
Ent 16																									
Ent 17	1							1							1							1			
Ent 18	1	1	1		1			1	1	1		1			1	1	1	1	1	1		1	1	1	1
Ent 19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Ent 20				1	1	1	1				1	1	1	1			1	1	1	1	1				1

Turno 3	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25
Ent 1																									
Ent 2																									
Ent 3																									
Ent 4	1	1		1				1	1		1				1	1		1				1	1		1
Ent 5																									
Ent 6																									
Ent 7																									
Ent 8																									
Ent 9																									
Ent 10			1	1	1	1	1			1	1	1	1	1			1	1	1	1	1			1	1
Ent 11																									
Ent 12																									
Ent 13																									
Ent 14																									
Ent 15																									
Ent 16																									
Ent 17																									
Ent 18																									
Ent 19																									
Ent 20	1	1	1	1				1	1	1	1				1	1	1	1				1	1	1	1

Figura 4.8: Resultado do “Modelo Pequeno”, para o cenário 3.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Turno 1	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25	
Enf 1																										
Enf 2			1		1				1	1		1				1	1		1				1	1		1
Enf 3	1			1		1	1				1		1	1				1		1	1				1	1
Enf 4																										
Enf 5																										
Enf 6					1	1	1				1	1					1	1							1	1
Enf 7					1	1	1				1	1	1	1			1	1	1	1	1				1	1
Enf 8			1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Enf 9	1	1								1						1			1	1	1	1	1		1	1
Enf 10											1						1							1		
Enf 11																										
Enf 12																										
Enf 13									1		1					1		1					1		1	
Enf 14									1		1					1		1					1		1	
Enf 15				1					1	1	1	1	1			1	1	1	1	1	1			1	1	1
Enf 16	1	1	1	1					1	1	1	1	1	1		1	1	1	1	1	1		1	1	1	1
Enf 17	1	1	1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Enf 18					1	1	1	1	1								1	1	1	1	1	1		1	1	1
Enf 19						1	1	1	1	1							1	1	1	1	1	1	1	1	1	1
Enf 20						1																				

Turno 2	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25	
Enf 1																										
Enf 2			1		1				1	1		1				1	1		1				1	1		1
Enf 3				1		1	1	1			1		1	1	1		1		1				1	1		1
Enf 4	1			1		1	1	1			1		1	1	1	1		1		1	1	1		1	1	
Enf 5																										
Enf 6																										
Enf 7																										
Enf 8												1	1				1	1						1	1	
Enf 9				1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Enf 10	1	1							1	1						1	1						1	1	1	
Enf 11																	1	1					1	1	1	
Enf 12																										
Enf 13																										
Enf 14																										
Enf 15																										
Enf 16																										
Enf 17																1							1		1	
Enf 18	1	1	1	1					1	1	1	1		1		1	1	1		1			1	1	1	
Enf 19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Enf 20					1	1	1					1	1	1					1	1	1	1			1	

Turno 3	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18	Dia 19	Dia 20	Dia 21	Dia 22	Dia 23	Dia 24	Dia 25	
Enf 1																										
Enf 2																										
Enf 3																										
Enf 4			1		1				1	1		1				1	1		1				1	1		1
Enf 5																										
Enf 6																										
Enf 7																										
Enf 8																										
Enf 9																										
Enf 10				1	1	1	1	1			1	1	1	1	1			1	1	1	1	1		1	1	
Enf 11																										
Enf 12																										
Enf 13																										
Enf 14																										
Enf 15																										
Enf 16																										
Enf 17																										
Enf 18																										
Enf 19																									</	

Figura 4.9: Resultado do “Modelo Pequeno”, para o cenário 4.

Modelo Grande

O “Modelo Grande” é o modelo mais interessante de testar e analisar por ser o modelo que resolve o problema apresentado da forma mais completa, contemplando um maior número de restrições. Havendo mais restrições no modelo, há também um maior número de testes possíveis de realizar e um maior potencial de as soluções permitirem uma análise mais cuidada.

Deste modo, efectuou-se um elevado número de testes a este modelo, com o propósito de avaliar o seu comportamento face a diferentes variações dos cenários criados.

Interessa então testar o comportamento do modelo desenvolvido face a variações de necessidades, tal como testado com o “Modelo Pequeno”, e a variações dos parâmetros que definem as restrições que completam o “Modelo Grande” relativamente ao “Modelo Pequeno”.

Os testes foram organizados em dois tipos:

- variação das necessidades diárias (cargas diárias);
- variação dos restantes parâmetros:

$$FM, Maxa, Mxc, Mnc \text{ e } Mlc$$

Para o último tipo de testes houve a necessidade de estabelecer valores de referência para os diferentes parâmetros (referidos de aqui em diante por *valores padrão*), definindo um cenário base e permitindo a comparação das variações de cada parâmetro com essa mesma base. A escolha desses valores padrão foi feita com base em referências encontradas nos trabalhos consultados, nos dados já recolhidos durante o trabalho [25] e nos dados das instâncias testadas anteriormente.

Os valores padrão escolhidos foram os seguintes:

- $FM = 3$;
- $Maxa = 26$;
- $Mxc = 5$;
- $Mnc = 2$;
- $Mlc = 4$;

Com estas escolhas:

- ⇒ no máximo, cada enfermeiro poderá trabalhar três fins-de-semana durante o horizonte temporal (de quatro semanas);

- ⇒ no mínimo, cada enfermeiro terá dois dias de folga em todo o período, o que nunca acontecerá, visto que cada enfermeiro apenas poderá trabalhar, no máximo, 5 dias consecutivos e terá que ter pelo menos um fim-de-semana livre, o que obriga a haver mais do que dois dias livres para cada enfermeiro ao longo do horizonte temporal;
- ⇒ no máximo, cada enfermeiro terá 4 dias livres consecutivos, o que indica que, dos enfermeiros considerados, nenhum estará em período de férias durante o horizonte temporal;
- ⇒ no mínimo, cada enfermeiro, quando é alocado a um dia de trabalho, será também alocado no dia seguinte, impedindo que um enfermeiro tenha uma atribuição no meio de uma série de dias livres consecutivos, situação usualmente não desejada.

Foram efectuados vários testes preliminares, que não serão aqui aprofundados, com o intuito de verificar a viabilidade destes valores e o bom funcionamento do modelo. Inicialmente definiram-se conjuntos de necessidades, idênticos aos das necessidades dos cenários apresentados mais à frente. Estes testes foram feitos com uma versão simplificada do modelo, que não contemplava dois parâmetros: Mnc e Mlc . Revelaram que o modelo simplificado permite a obtenção de soluções boas de uma forma rápida, por vezes permite a obtenção da solução óptima em poucos segundos, dependendo da complexidade do cenário em termos de necessidades (entendendo-se por complexidade o grau de liberdade que as necessidades da instância permitem ao conjunto de combinações de alocações possíveis – conjunto de escalonamentos –, ou seja, o grau de restrição imposto pelas necessidades).

Os vários cenários base, criados com o intuito de testar a resposta do modelo à variação das necessidades impostas, têm as seguintes necessidades:

4.3. TESTES AOS MODELOS MATEMÁTICOS

Cenário 1: 10 enfermeiros / dia

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	0
Tarde	1	1	1	1	1	1	1
Noite	0	0	0	0	0	0	1

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	2	1	2	2	2	2
Tarde	2	1	2	1	1	1	1
Noite	0	0	0	1	1	1	0

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	3	2	3	2	2	2	2
Tarde	1	2	1	2	2	2	2
Noite	1	1	1	0	0	0	1

Cenário 2: 11 enfermeiros / dia

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	1
Tarde	1	1	1	1	1	1	1
Noite	0	0	0	0	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	2	2	2	2	2	2
Tarde	1	1	1	1	1	1	1
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	3	3	3	3	2	2	2
Tarde	3	2	2	2	2	2	2
Noite	1	1	0	0	1	1	1

Cenário 3: 12 enfermeiros / dia

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	1
Tarde	1	1	1	1	1	1	1
Noite	0	0	0	0	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	2	2	2	2	2	2	2
Tarde	1	1	2	1	1	1	1
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	3	4	3	4	3	3	3
Tarde	3	2	2	2	2	2	2
Noite	1	1	0	0	1	1	1

Cenário 4: 13 enfermeiros / dia

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	1
Tarde	1	1	1	1	1	1	1
Noite	1	1	0	1	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	2	2	2	2	2	2
Tarde	1	2	2	1	1	2	2
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	4	3	3	4	3	2	2
Tarde	3	2	2	2	3	3	3
Noite	1	1	1	0	1	1	1

Cenário 5: 14 enfermeiros / dia

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	1	1	1
Tarde	1	1	1	1	1	1	1
Noite	0	0	0	0	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	2	2	2	2	2	2	2
Tarde	2	2	2	2	1	2	2
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	4	4	4	4	4	4	4
Tarde	3	3	2	3	3	2	2
Noite	1	1	1	0	1	1	1

Cenário 6-1: 15 enfermeiros / dia

Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	2	2	2
Tarde	1	1	1	1	1	1	1
Noite	1	1	0	1	0	0	0

Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	2	2	2	2	2	2	2
Tarde	2	3	2	2	1	2	2
Noite	0	0	1	1	1	1	1

Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	4	3	5	4	4	4	4
Tarde	3	3	2	3	3	2	2
Noite	1	1	1	0	1	1	1

Cenário 6.2: 15 enfermeiros / dia							
Grau 1	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	1	1	1	1	2	1	1
Tarde	1	1	1	1	1	1	1
Noite	1	1	0	1	0	0	0
Grau 2	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	2	2	2	2	2	3	3
Tarde	2	3	2	2	1	2	2
Noite	0	0	1	1	1	1	1
Grau 3	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	S	D
Manhã	4	3	5	4	4	4	4
Tarde	3	3	2	3	3	2	2
Noite	1	1	1	0	1	1	1

Respeitando e diferenciando os dois tipos de testes a efectuar, agruparam-se em três grupos distintos. Dois para testar a variação dos parâmetros e um para testar a variação das necessidades.

Os testes relacionados com os parâmetros foram feitos em dois grupos pelo seguinte: inicialmente apenas se iria testar a variação dos parâmetros com um cenário (o “**Cenário 1**”), mas certos limites destas variações podem também sofrer alteração com a variação da dimensão das necessidades, uma vez que a complexidade do modelo varia. Assim, decidiu-se conceber e resolver também estes testes com um outro cenário que tivesse necessidades bastante diferentes do primeiro (o “**Cenário 4**” – escolhido por ser dos cenários com necessidades mais altas e com solução para a sua forma base obtida num curto intervalo de tempo no CPLEX). Estes testes foram feitos através da criação de vários cenários iguais ao cenário base, apenas com variação, para cada teste, de um dos parâmetros a testar.

Os cenários usados nos testes, bem como os seus resultados obtidos, estão apresentados nas Tabelas 4.1, 4.3 e 4.10. Estas tabelas incluem um número identificador do teste, o número total de enfermeiros necessários por dia (de todos os graus), as dimensões do modelo (número de linhas e colunas do modelo *MIP* criado), os valores dos parâmetros que podem variar, o tempo de execução do teste, o valor da função objectivo alcançada, caso se tenha obtido solução, e a quantidade de memória utilizada pelo resolvidor durante o teste.

Depois de recolhidos os resultados destes testes, efectuados num computador pessoal (especificado anteriormente), uma vez que não se atingiram as soluções óptimas (ou em poucos casos se conseguiu atingi-las) e não existe forma de saber qual a proximidade das soluções obtidas à solução óptima,

que é um valor importante para uma boa análise dos resultados, decidiu-se correr também os testes (ao “Modelo Grande”) num computador com melhores características e com um programa resolvidor mais poderoso até obter a solução óptima para cada caso. As soluções óptimas estão apresentadas nas Tabelas 4.2, 4.4 e 4.11¹ e são utilizadas para comentar os resultados obtidos no computador pessoal que, como já foi exposto, são os resultados de interesse de estudo neste trabalho.

4.3.3 Apresentação e Análise dos Resultados Relativos ao Modelo Grande

Como referido anteriormente, os testes de variação dos parâmetros foram divididos em dois grupos. No primeiro grupo foram testadas variantes do “**Cenário 1**”, no segundo variantes do “**Cenário 4**”.

As variações foram feitas a cada um dos parâmetros de cada vez, visando a obtenção de limites superior e inferior aceitáveis para cada um. Um limite aceitável é o maior ou menor valor desse parâmetro para o qual se obtém solução num tempo de execução do modelo aceitável, corrido no computador pessoal.

As Tabelas 4.1 e 4.3 apresentam as características e os resultados destes testes, realizados no computador pessoal e com o “glpk”. Na primeira coluna da tabela está o número do teste; na segunda o número total de enfermeiros necessários em cada dia; seguem-se os números de linhas e colunas constituintes do ficheiro LP criado para representar o modelo a resolver – que dão uma noção da dimensão do problema; depois os valores dos vários parâmetros, em que, para cada parâmetro e em cada teste, o valor alterado relativamente ao valor padrão (valor a ser testado) está escrito a negrito; segue-se o tempo de execução do resolvidor (sempre limitado ao valor apresentado); o valor da função objectivo obtida; e por fim, o valor da memória utilizada na resolução do problema. No primeiro teste apresentado em cada uma das tabelas, todos os valores dos parâmetros estão escritos a negrito, por estes serem os valores padrão, ou seja, este é o cenário base.

Em alguns dos testes apresentados não se obteve solução no tempo limite estabelecido. Estes testes são os que não têm valores de FO nem de Memória

¹Nestas tabelas, nos testes limitados a 10 horas de cálculo, assinalados na quarta coluna, o tempo de cálculo apresentado na realidade apresenta valores bastante mais elevados do que os 36 000 segundos correspondentes a esse montante de horas. Isto acontece porque o tempo de cálculo apresentado refere-se ao tempo total de cálculo dos dois processadores, separadamente. Esta medida também não considera as perdas de tempo que acontecem devido à comunicação entre as duas máquinas, por isso poderá ser inferior a $2 \times 36000 = 72000$.

4.3. TESTES AOS MODELOS MATEMÁTICOS

nas tabelas apresentadas.

Nº	Enf./dia	Col.	Linhas	Parâmetros					Tempo	FO	Mem.
				FM	Maxa	Mxc	Mnc	Mlc			
1.1	10	3896	5056	3	26	5	2	4	1800	14034	12.6 Mb
1.2	10	3896	5076	3	26	5	1	4	1800	12312	11.4 Mb
1.3	10	3896	5036	3	26	5	3	4	3600	—	—
1.4	10	3896	5116	3	26	5	2	1	<i>não tem solução inteira</i>		
1.5	10	3896	5096	3	26	5	2	2	1800	15372	16.6 Mb
1.6	10	3896	5076	3	26	5	2	3	1800	14409	12.9 Mb
1.7	10	3896	5036	3	26	5	2	5	1800	13323	13.1 Mb
1.8	10	3896	5016	3	26	5	2	6	1800	12783	16.9 Mb
1.9	10	3896	4996	3	26	5	2	7	1800	12696	14.5 Mb
1.10	10	3896	5156	2	26	5	2	4	<i>não tem solução inteira</i>		
1.11	10	3896	5047	3	17	5	2	4	7200	—	—
1.12	10	3896	5056	3	18	5	2	4	1800	13254	13.2Mb
1.13	10	3896	5056	3	19	5	2	4	1800	14223	12.5 Mb
1.14	10	3896	5056	3	20	5	2	4	1800	13866	14.1 Mb
1.15	10	3896	5056	3	21	5	2	4	1800	13539	14.8 Mb
1.16	10	3896	5056	3	22	5	2	4	1800	12762	12.7 Mb
1.17	10	3896	5056	3	23	5	2	4	1800	12795	13.5 Mb
1.18	10	3896	5056	3	24	5	2	4	1800	14034	13.3 Mb
1.19	10	3896	5056	3	26	3	2	4	<i>não tem solução inteira</i>		
1.20	10	3896	5076	3	26	4	2	4	1800	13584	14.8 Mb

Tabela 4.1: Testes de variações dos parâmetros feitos com as necessidades do “Cenário 1”, corridos com o “glpk”. Registos das suas características e seus resultados.

Todos os valores foram testados até se atingir o limite mínimo – valor para o qual, ou não existe solução inteira, ou não se obteve solução inteira num tempo de execução aceitável. Quanto ao limite máximo, este só foi testado para os parâmetros em que tal teste fazia sentido e para valores aceitáveis.

Com os dados apresentados há dois tipos de análise que podem ser feitas: a variação do valor da função objectivo obtido com a variação de cada parâmetro e a variação da dimensão do problema (linhas e colunas do ficheiro LP) com a variação de cada parâmetro. Para este último tipo de análise é interessante referir que o número de linhas do ficheiro criado corresponde ao número de restrições e o número de colunas corresponde ao número de variáveis do problema particular. O número de variáveis mantém-se constante para todos os testes efectuados, porque os parâmetros que definem esse número também são sempre constantes: o número de enfermeiros, dias do horizonte temporal, turnos por dia e graus de enfermeiros.

Além das análises que são feitas directamente a partir das tabelas apresentadas, também tem interesse compará-las com os resultados óptimos (ou muito próximos do óptimo) obtidos no CPLEX e apresentados nas tabelas 4.2 e 4.4, assim como analisar alguns escalonamentos obtidos.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Nº	FO	Tempo de cálculo	Limitado	
			a 10h	gap (%)
1.1	12312	55.13	Não	0
1.2	12312	6.39	Não	0
1.3	<i>Não encontrou solução</i>		Sim	—
1.4	<i>Não tem solução</i>		Não	—
1.5	13692	67596.89	Sim	0.26
1.6	12336	68333.67	Sim	0.19
1.7	12312	28.73	Não	0
1.8	12312	30.04	Não	0
1.9	12312	12.77	Não	0
1.10	<i>Não tem solução</i>		Não	—
1.11	12372	123.14	Não	0
1.12	12360	81.93	Não	0
1.13	12348	28.30	Não	0
1.14	12348	69918.31	Sim	0.1
1.15	12336	69916.66	Sim	0.1
1.16	12312	63.24	Não	0
1.17	12312	423.04	Não	0
1.18	12312	92.23	Não	0
1.19	<i>Não tem solução</i>		Não	—
1.20	12336	2279.69	Não	0

Tabela 4.2: Resultados dos testes de variação dos parâmetros, com as necessidades do “Cenário 1”, corridos no CPLEX.

Mínimo de alocações consecutivas permitidas – Parâmetro *Mnc*

Relativamente ao parâmetro *Mnc* (testes 1.2 e 1.3, 2.2 e 2.3), o limite máximo foi imediatamente atingido quando testado com o valor 3. Aliás, para este valor, tanto quando corrido com o “glpk”, como quando corrido no CPLEX (limitado a 10 horas de cálculo), e tanto para o “Cenário 1” (teste 1.3), como para o “Cenário 4” (teste 2.3), não foi encontrada qualquer solução, apesar de esta dever existir, já que o CPLEX não indicou a sua inexistência, ainda que não a tenha encontrado.

O limite mínimo é igual a 1 e corresponde a anular esta restrição, já que dizer que no mínimo terá que haver um dia de trabalho “consecutivo”, sempre que haja alguma alocação, é o mesmo que dizer que não existe mínimo de dias consecutivos, podendo haver uma única alocação isolada. Esta situação, em certas circunstâncias, pode ser indesejada. Por exemplo, como acontece em ambos os escalonamentos apresentados na Figura 4.10, pode acontecer um dia alocado a um enfermeiro de forma isolada e a meio de um longo período de dias de descanso (dias 4 e 22 para o enfermeiro 17 no primeiro escalonamento e dias 8 e 18 para o enfermeiro 8 no escalonamento ótimo), ou duas alocações singulares separadas apenas por um dia de descanso e isoladas, no meio de um longo período de descanso (dias 23 e 25 para o enfermeiro 14 no primeiro escalonamento e dias 22 e 24 para o enfermeiro 9 no escalonamento ótimo). Diminuir o valor deste parâmetro para 1 torna também o problema menos complexo de resolver, embora tenha maior dimensão (mais restrições),

4.3. TESTES AOS MODELOS MATEMÁTICOS



Figura 4.10: Escalonamentos obtidos para o teste 1.2. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

por passar a ter mais soluções admissíveis e, portanto, passar a ser mais fácil encontrar uma. O problema é considerado menos complexo uma vez que este rapidamente chegou à solução óptima no CPLEX e também a atingiu no cálculo efectuado com o “glpk”.

Fins-de-semana, Máximo de alocações no horizonte temporal e Máximo de alocações consecutivas permitidas – Parâmetros FM , $Maxa$ e Mxc

Os parâmetros FM , $Maxa$ e Mxc não foram testados para valores acima dos seus valores padrão, porque esses padrões já são muito folgados e valores acima desses passariam a ser demasiado elevados. Além de que os valores não

4.3. TESTES AOS MODELOS MATEMÁTICOS

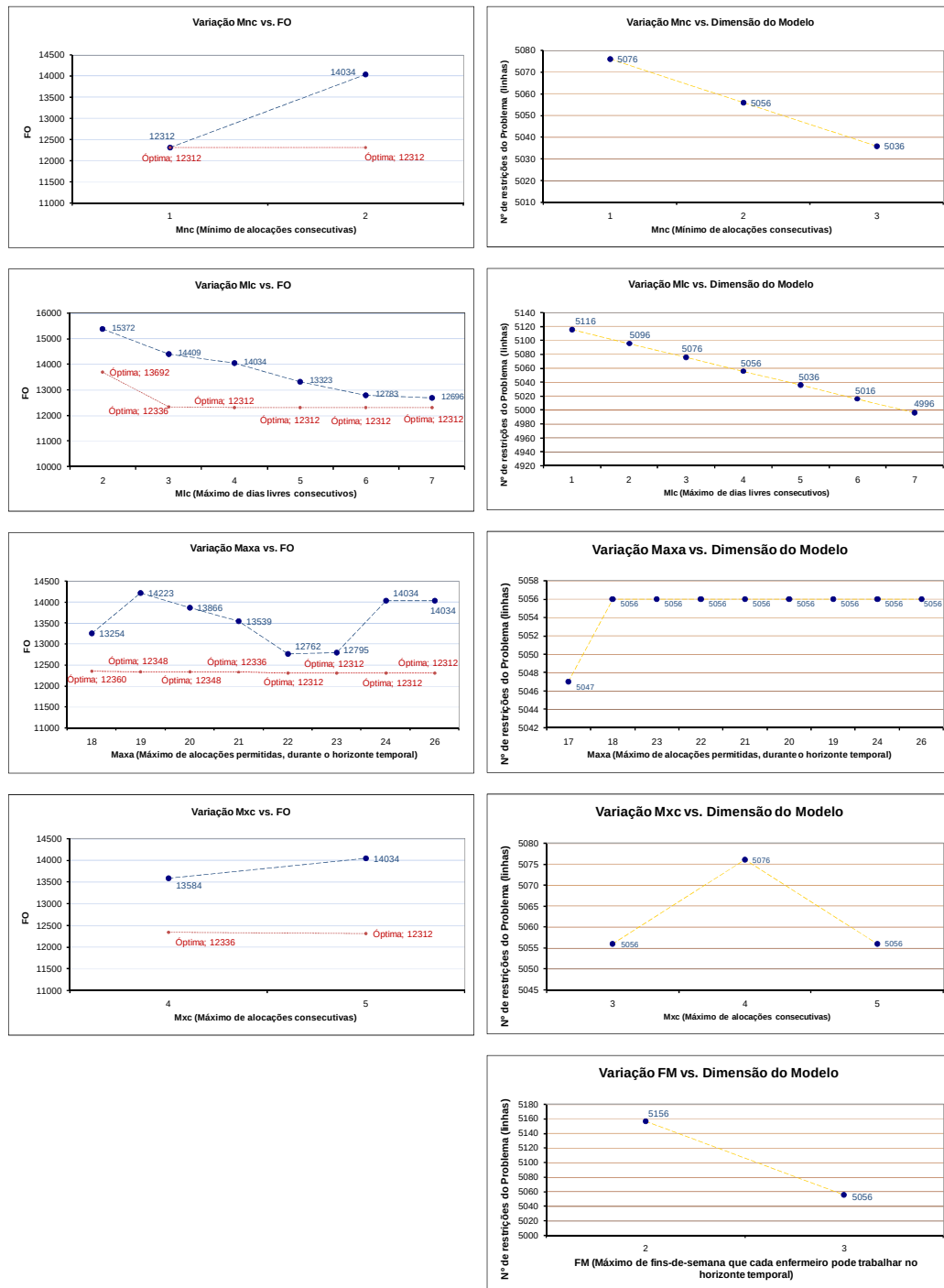


Figura 4.11: Gráficos das variações do valor da Função Objectivo e da dimensão do problema com variações dos parâmetros, relativamente ao “Cenário 1”.

4.3. TESTES AOS MODELOS MATEMÁTICOS

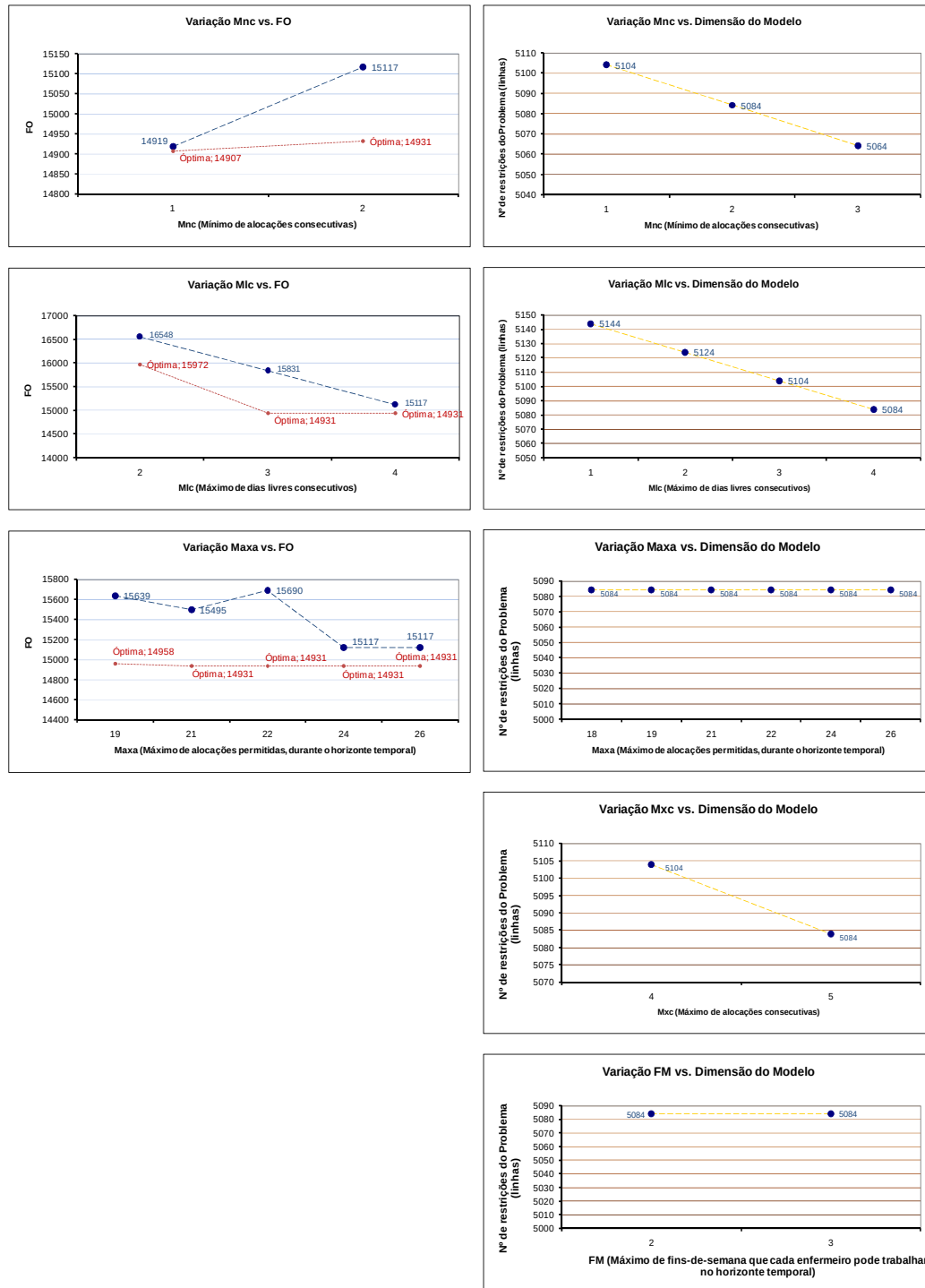


Figura 4.12: Gráficos das variações do valor da Função Objectivo e da dimensão do problema com variações dos parâmetros, relativamente ao “Cenário 4”.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Nº	Enf./dia	Col.	Linhas	Parâmetros					Tempo	FO	Memória
				FM	Maxa	Mxc	Mnc	Mlc			
2.1	13	3896	5084	3	26	5	2	4	1800	15117	18.5 Mb
2.2	13	3896	5104	3	26	5	1	4	1800	14919	27.9 Mb
2.3	13	3896	5064	3	26	5	3	4	1800	—	—
2.4	13	3896	5144	3	26	5	2	1	<i>não tem solução</i>		
2.5	13	3896	5124	3	26	5	2	2	1800	16548	11.1 Mb
2.6	13	3896	5104	3	26	5	2	3	1800	15831	15 Mb
2.7	13	3896	5084	2	26	5	2	4	<i>não tem solução</i>		
2.8	13	3896	5084	3	18	5	2	4	<i>não tem solução</i>		
2.9	13	3896	5084	3	19	5	2	4	1800	15639	13.4 Mb
2.10	13	3896	5084	3	21	5	2	4	1800	15495	13.2 Mb
2.11	13	3896	5084	3	22	5	2	4	1800	15690	13.7 Mb
2.12	13	3896	5084	3	24	5	2	4	1800	15117	15.1 Mb
2.13	13	3896	5104	3	26	4	2	4	1800	—	—

Tabela 4.3: Testes de variações dos parâmetros feitos com as necessidades do “Cenário 4”. Registos das suas características e seus resultados.

Nº	FO	Tempo de execução	Limitado	
			a 10h	gap (%)
2.1	14931	440.22	Não	0.01
2.2	14907	14.20	Não	0
2.3	<i>Não encontrou solução</i>		Sim	—
2.4	<i>Não tem solução</i>		Não	—
2.5	15972	8.83	Não	0
2.6	14931	968.39	Não	0
2.7	<i>Não tem solução</i>		Não	—
2.8	<i>Não tem solução</i>		Não	—
2.9	14958	62452.98	Não	0.01
2.10	14931	1402.14	Não	0
2.11	14931	14598.55	Não	0.01
2.12	14931	9293.81	Não	0.01
2.13	14973	68753.92	Sim	0.33

Tabela 4.4: Resultados dos testes de variação dos parâmetros, com as necessidades do “Cenário 4”, corridos no CPLEX.

testados apenas permitem a obtenção de uma melhor solução, provavelmente obtida mais rápida e facilmente.

FM Para *FM* a análise é simples. Apenas se poderia testar a possibilidade de trabalhar 4 fins-de-semana no horizonte temporal, como neste caso o horizonte temporal tem 4 semanas, isto significaria eliminar esta restrição. O mínimo desta restrição é o seu valor padrão, uma vez que, como visível na Tabela 4.2, basta diminuir tal valor para 2 que o problema deixa de ter solução.

Maxa Quanto ao parâmetro *Maxa*, testes para valores acima de 26 aproximarão esta restrição da extinção (da restrição deixar de restringir o pro-

4.3. TESTES AOS MODELOS MATEMÁTICOS

Enf	2.1		2.2		2.5		2.6		2.9		2.10		2.11		2.12		2.13
	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	opt
1	18	16	17	18	20	18	20	18	19	14	18	15	21	15	18	15	16
2	20	20	21	22	21	23	19	20	18	19	21	20	21	20	20	20	19
3	15	15	14	15	18	17	18	16	18	18	18	15	15	16	15	17	15
4	15	17	16	13	19	18	18	14	19	17	14	18	16	17	15	16	18
5	17	18	22	13	18	18	18	17	19	18	17	15	21	16	17	19	18
6	18	19	18	16	19	20	18	18	19	19	17	17	18	19	18	16	16
7	21	16	16	17	19	18	18	18	19	19	18	18	17	17	21	18	18
8	17	16	18	18	21	20	17	17	16	19	18	18	17	17	17	15	19
9	18	16	15	14	20	19	17	14	19	16	19	18	18	13	18	17	17
10	16	17	13	18	19	18	18	18	19	19	17	18	17	21	16	18	18
11	19	21	18	20	22	20	21	20	19	19	21	21	20	15	19	21	20
12	18	21	19	20	20	20	19	19	19	19	19	20	21	20	18	19	20
13	20	19	22	19	21	21	20	20	19	19	19	21	21	22	20	18	20
14	19	20	19	19	19	19	20	20	19	19	20	19	19	22	19	22	18
15	22	20	21	20	19	22	20	18	17	19	18	20	20	17	22	20	20
16	19	20	19	21	19	20	20	20	19	19	20	20	19	20	19	17	19
17	18	19	21	21	19	20	18	18	19	19	21	20	19	20	18	20	19
18	19	22	18	20	18	19	22	22	19	19	19	19	19	21	19	20	19
19	22	21	21	23	22	22	21	21	19	19	20	18	19	21	22	20	21
20	19	15	20	21	20	20	19	20	19	19	21	18	20	19	19	20	18
Méd	18.5	18.4	18.4	18.4	19.65	19.6	19.05	18.4	18.65	18.4	18.75	18.4	18.9	18.4	18.5	18.4	18.4

Tabela 4.5: Número de dias que cada enfermeiro trabalha, durante todo o horizonte temporal, para todas as soluções obtidas, relativas ao “Cenário 4”.

blema), uma vez que o horizonte temporal é de 28 dias e, como se pode constatar a partir das Tabelas 4.6 e 4.5, em nenhum dos testes efectuados há enfermeiros com mais de 23 alocações durante o horizonte temporal. Isto significa que, apesar de ser possível que o escalonamento se revele diferente à medida que se aproxima o valor de *Maxa* dos 28 dias de horizonte temporal, este limite não influenciará o número total de alocações aos enfermeiros. No entanto, para valores acima de 23 (testes 1.18, 1.1 e 2.12 e 2.1), os valores da função objectivo óptima não são diferentes, mas os escalonamentos são (Figuras 4.13, 4.14, 4.15 e 4.16). A justificação para este facto reside em, com a mudança deste parâmetro, o espaço de soluções também se altera e, por isso, os escalonamentos óptimos serem distintos.

Para este parâmetro, o modelo tem quase sempre a mesma dimensão, apenas assumindo uma menor dimensão quando *Maxa* tem o valor 17.

A procura da sua solução óptima, como se pode observar no gráfico correspondente da Figura 4.11, fica mais afastada da óptima, quer para os valores mais baixos, quer para os valores mais elevados do valor padrão deste parâmetro (os resultados obtidos com o “glpk”, todos limitados a 1800 segundos, ficaram mais próximos da solução óptima nos valores de 22 e 23). Para os valores mais baixos, a justificação prende-se com o facto de esses valores restringirem mais o problema, tornando-o mais complexo e difícil de resolver.

O facto de os valores mais elevados tornarem o problema mais difícil de resolver poderá ser intrigante. Mas, mais uma vez, estes valores permitem que haja um maior espaço de soluções do que os valores que restringem o número máximo de alocações ao longo do horizonte temporal ($Maxa$) a um número próximo do valor máximo que acontece no escalonamento óptimo. Assim, poderá ser mais difícil chegar, no mesmo tempo e com as mesmas condições, à solução óptima com os valores mais elevados, uma vez que existem mais soluções possíveis para explorar (a pesquisa pela solução óptima passa a contemplar a exploração de um maior número de soluções admissíveis, ficando sujeita a mais óptimos locais). Esta dificuldade pode, no entanto, ser superada por haver menos liberdade para as combinações de alocações possíveis, na sequência do aumento do valor médio de necessidades diárias, diminuindo o espaço de soluções, como acontece nos testes relativos ao “Cenário 4” e é visível no gráfico correspondente da Figura 4.12.

Mxc Relativamente a Mxc , trabalhar mais do que 5 dias seguidos raramente acontece e, como se obtém solução para este valor, aumentá-lo iria fazer com que esta restrição deixasse de restringir, relaxando mais o problema. O seu limite inferior para este problema foi o valor 4, uma vez que para $Mxc = 3$ o problema não tem solução (Tabelas 4.1 e 4.2, teste 1.19). Como é visível nos resultados do teste 1.20 da Tabela 4.2, comparativamente ao teste base (com valores padrão), a diminuição do valor deste parâmetro torna a solução pior, ainda que nos resultados obtidos com o “glpk” tal observação não pareça verdadeira (Tabela 4.1, teste 1.20), uma vez que, para $Mxc = 4$, a solução obtida se aproximou mais da solução óptima do que a solução do cenário base (com valores padrão, teste 1.1). No entanto, observando os resultados óptimos, esta afirmação confirma-se, já que, para $Mxc = 4$ a função objectivo aumenta.

Máximo de dias livres consecutivos – Parâmetro Mlc

No parâmetro Mlc , o valor 7 indica que os enfermeiros poderão ter uma semana de férias no horizonte temporal. Como não se pretende considerar períodos de férias neste modelo (indisponibilidades dos enfermeiros), decidiu-se não ultrapassar este limite, até porque o aumento do valor deste parâmetro faz o problema ficar menos restrito, afastando-se das situações reais. Por estas razões no segundo grupo de testes (com o “Cenário 4”) apenas se testa valores inferiores ao valor padrão.

Restringir o máximo de dias livres consecutivos a 1 dia, além de não fazer

4.3. TESTES AOS MODELOS MATEMÁTICOS

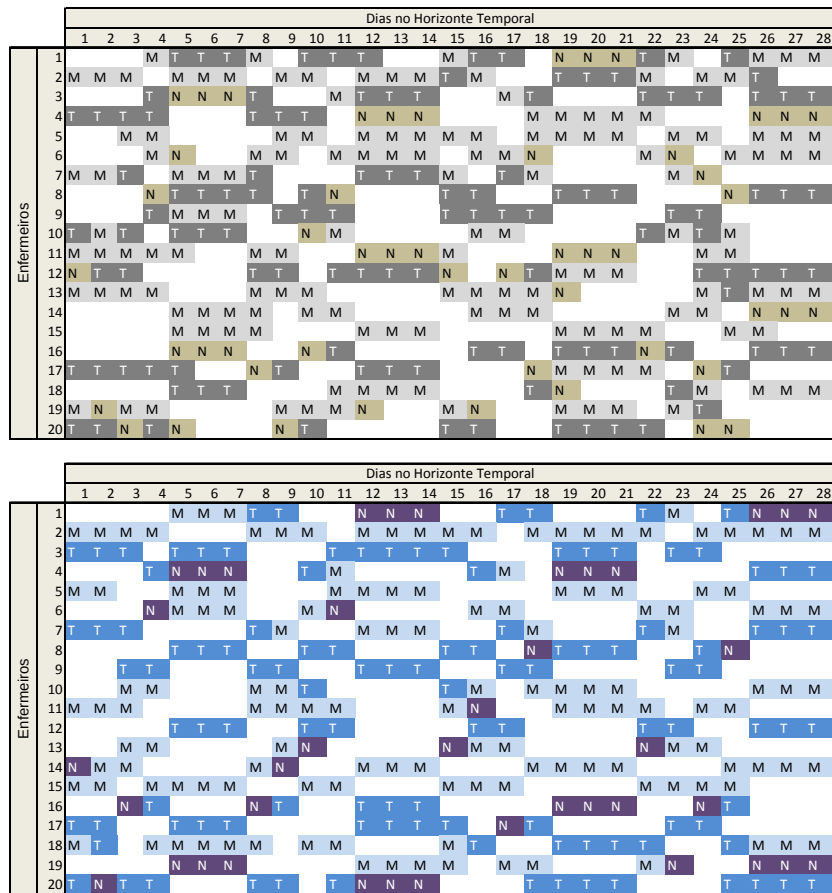


Figura 4.13: Escalonamentos obtidos para o teste 1.1. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

4.3. TESTES AOS MODELOS MATEMÁTICOS

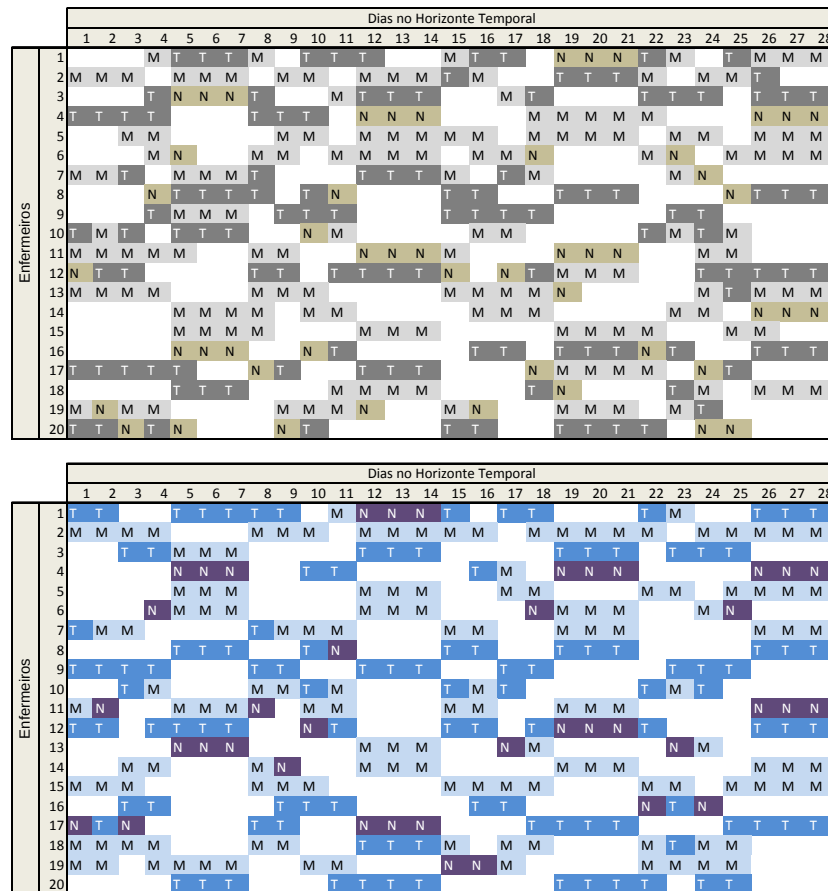


Figura 4.14: Escalonamentos obtidos para o teste 1.18. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

4.3. TESTES AOS MODELOS MATEMÁTICOS

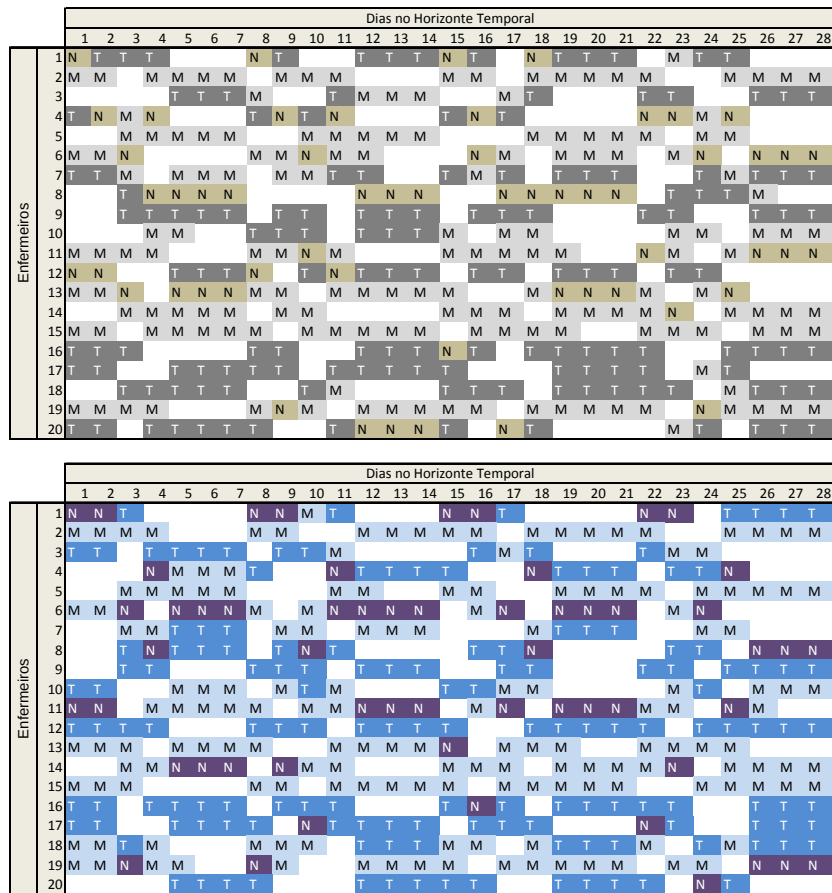


Figura 4.15: Escalonamentos obtidos para o teste 2.1. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

4.3. TESTES AOS MODELOS MATEMÁTICOS



Figura 4.16: Escalonamentos obtidos para o teste 2.12. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

muito sentido, tendo em conta que tal não é praticado, também é impensável neste problema, tornando-o insolúvel.

À medida que o valor deste parâmetro cresce (problema mais relaxado), o valor da função objectivo diminui e a solução obtida com o “glpk” vai-se aproximando da óptima. Assim, e analisando os resultados obtidos, deduz-se que a complexidade do problema diminui com o aumento do Mlc (visível, não só pela dimensão apresentada nos gráficos respectivos das Figuras 4.11 e 4.12, mas também pelo tempo em que a solução foi obtida, na Tabela 4.2). Na Tabela 4.4 é visível que a solução óptima encontrada, para o valor deste parâmetro igual a 3, foi mais demorada de encontrar do que para o valor 2. Mas estas duas, como se constata na Tabela 4.2, correspondem aos valores para os quais o problema fica mais demorado de calcular (são portanto os mais complexos), além de o seu valor óptimo decrescer muito entre esses dois valores (do teste 2.5 para o teste 2.6), o que pode dificultar a tarefa quando se trata do teste 2.6.

Estes dois valores resultam também no que aparentam ser os escalonamentos mais equilibrados (Figuras 4.17 e 4.18). Nos testes relativos ao “Cenário 1” com estes dois valores, testes 1.5 e 1.6, nos escalonamentos óptimos obtidos, é possível identificar padrões quinzenais em muitos dos enfermeiros. Padrões em que os dias e turnos alocados nas duas primeiras semanas são iguais aos alocados nas duas últimas. Alguns outros têm também padrões quinzenais de dias alocados, mesmo tendo um ou outro turno diferente num desses dias. Estes padrões apenas são observáveis nos enfermeiros menos graduados (enfermeiros de grau 2 e 3 – enfermeiros 5 a 20). São observados mais para o “Cenário 1” e ocorrem mais vezes para $Mlc = 2$ do que para $Mlc = 3$. Apenas nestes casos os padrões são observados em grande quantidade, porque estes são os casos que, por terem menor necessidade diária total de enfermeiros e por o conjunto de restrições que obriga a que no máximo de cinco em cinco dias (Mxc) haja uma folga, sendo o mínimo de alocações consecutivas (Mnc) 2 dias e o máximo de dias livres consecutivos 2 ou 3, respectivamente, facilitam o aparecimento de tais padrões de escalonamento, não restringindo o problema de forma desequilibrada.

Nestes casos, estes limites (Mxc , Mnc e Mlc) estão muito próximos uns dos outros, o que torna muito fácil preencher uma semana: 5 dias de trabalho + 2 folgas ou 4 dias de trabalho + 3 folgas ou 2 dias de trabalho + 2 folgas + 3 dias de trabalho ... Com várias possibilidades de combinações, como as anteriores, para preencher uma semana completa. Como as necessidades também são iguais para cada semana, torna-se muito simples que esses padrões surjam.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Ainda que nos escalonamentos obtidos com o “glpk” não seja, aparentemente, possível reconhecer tais padrões, no escalonamento do teste 1.5 é possível reconhecer um padrão no enfermeiro 20, que é quebrado na última semana, porque durante as outras 3 semanas este enfermeiro trabalhou sempre aos fins-de-semana. Como no máximo os enfermeiros só podem trabalhar 3 fins-de-semana, na última semana este não poderia repetir o trabalho das semanas anteriores. Também é possível encontrar um padrão para o enfermeiro 8 após a 1ª semana.



Figura 4.17: Escalonamentos obtidos para o teste 1.5. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

Nos testes realizados para o “Cenário 4” (Figuras 4.19 e 4.20) não é possível reconhecer padrões. Neste cenário a falta de padrões, que contraria o que acontece no “Cenário 1”, deve-se certamente ao facto de haver uma

4.3. TESTES AOS MODELOS MATEMÁTICOS

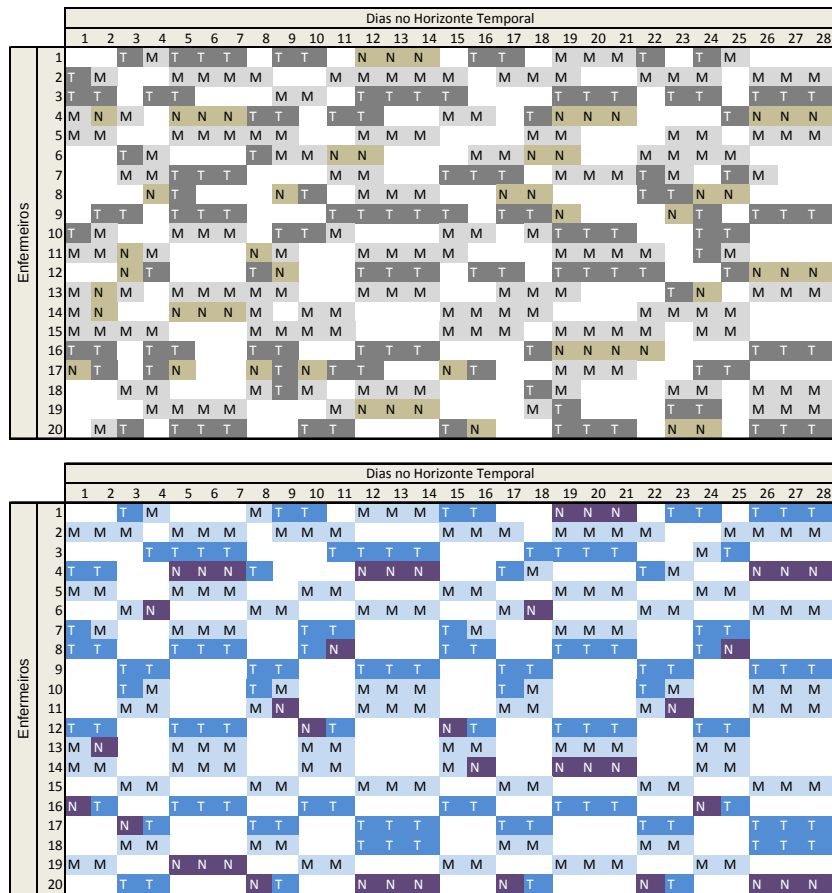


Figura 4.18: Escalonamentos obtidos para o teste 1.6. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

4.3. TESTES AOS MODELOS MATEMÁTICOS

maior necessidade total de enfermeiros, o que não permite tanta liberdade para folgas idênticas durante semanas consecutivas.

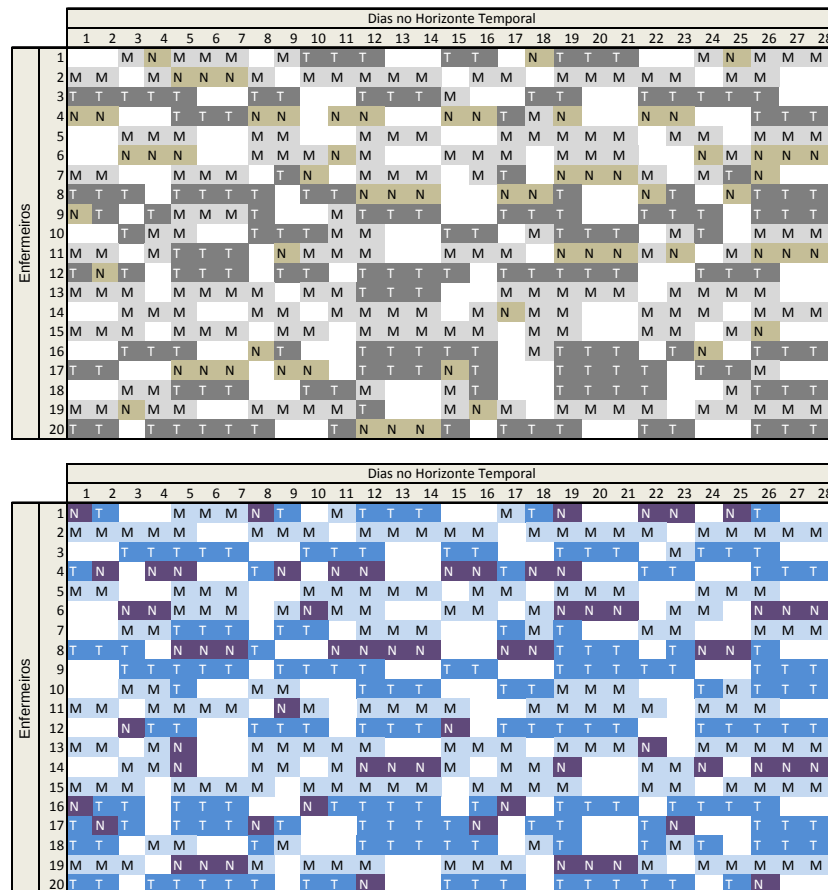


Figura 4.19: Escalonamentos obtidos para o teste 2.5. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

Além dos padrões aqui referidos, é possível também constatar por observação dos escalonamentos obtidos (nomeadamente os óptimos e para os testes 1.5 e 1.6, que têm necessidades mais baixas, facilitando o equilíbrio entre os dias de trabalho e os dias de folga durante todo o horizonte temporal) que existe uma melhor distribuição dos dias de trabalho e de folga entre todos os enfermeiros. Não se verificam discrepâncias tão elevadas como nos restantes escalonamentos entre o enfermeiro com mais turnos alocados e o enfermeiro com menos turnos alocados. Ou seja, não é possível identificar enfermeiros que trabalhem muito mais do que outros. Isto é bem visível nas

4.3. TESTES AOS MODELOS MATEMÁTICOS



Figura 4.20: Escalonamentos obtidos para o teste 2.6. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Tabelas 4.6 e 4.5, especialmente nas soluções ótimas. Por fim, há também um maior equilíbrio entre o número de dias consecutivos alocados e o número de dias de folga seguintes e anteriores a esse conjunto de alocações.

No entanto, todo este equilíbrio de escalonamentos entre os enfermeiros e o equilíbrio do próprio escalonamento de cada enfermeiro podem resultar facilmente num desequilíbrio do número de enfermeiros alocados por dia, especialmente se a solução não for ótima. Isto é observado nas Tabelas 4.7 e 4.8, em que as soluções dos testes 1.5, 1.6, 2.5 e 2.6 são das que atingem mais elevados números de enfermeiros a trabalhar por dia. Estas podem, pois, ser soluções mais interessantes a nível de satisfação dos enfermeiros, mas envolvem custos maiores. Observando os valores:

Teste 1.5, resultados: “glpk” – 20; ótimo – 20;

Teste 1.6, resultados: “glpk” – 18; ótimo – 11;

Teste 2.5, resultados: “glpk” – 20; ótimo – 20;

Teste 2.6, resultados: “glpk” – 17; ótimo – 14;

e tendo em conta que estes escalonamentos (com $Mlc = 2$ e com $Mlc = 3$) são semelhantes em termos de equilíbrio analisado, pode-se concluir que, dos dois, $Mlc = 3$ será o melhor para obter um bom escalonamento, ainda que possa necessitar de mais tempo de execução como acontece para o “Cenário 4”.

Preferências

Enfermeiro	Manhã	Tarde	Noite
1		✓	✓
2	✓	✓	
3		✓	
4		✓	✓
5	✓		
6	✓		✓
7	✓	✓	
8		✓	✓
9		✓	
10	✓	✓	
11	✓		✓
12		✓	✓
13	✓		✓
14	✓		✓
15	✓		
16		✓	✓
17		✓	✓
18	✓	✓	
19	✓		✓
20		✓	✓

Tabela 4.9: Preferências de atribuições de turnos dos Enfermeiros.

As preferências dos enfermeiros, apresentadas na Tabela 4.9, são muito respeitadas (utilizando sempre um valor $p = 12$, ver Anexo B, Figura B.3). Como é observável nos escalonamentos aqui apresentados, estas preferências são poucas vezes violadas nos escalonamentos das soluções ótimas, enquanto nas soluções obtidas pelo “glpk” tais violações acontecem com maior frequência. Ainda assim, estas violações não são muitas e apenas se tornam num número considerável quando alguma(s) dessa(s) violação(ões) engloba um dia de um fim-de-semana e obriga a que haja mais duas violações nos outros dias do fim-de-semana. Isto acontece, por exemplo, nos escalonamentos obtidos para o enfermeiro 1, no teste 1.1 (Figura 4.13) e no escalonamento obtido no “glpk”, para o enfermeiro 1, no teste 1.5 que apanha dois fins-de-semana (Figura 4.17).

É curioso notar que, nos escalonamentos ótimos, são poucos os escalonamentos pessoais (de cada enfermeiro) que têm violações deste tipo e, quando existem, ocorrem em número reduzido, enquanto que nos escalonamentos obtidos pelo “glpk” muitos enfermeiros têm violações deste tipo nos seus escalonamentos e alguns têm-nos em número elevado (como acontece ao enfermeiro 1 no escalonamento do teste 1.5).

Fins-de-semana e Ocupações

É interessante observar que o “Cenário 1”, como é o cenário com menores necessidades totais diárias, permite que haja enfermeiros que não trabalham aos fins-de-semana e apenas trabalhem 10 dias num horizonte temporal de 28, como acontece ao enfermeiro 13 no escalonamento ótimo obtido para o teste 1.1 (Figura 4.13). Este cenário permite que haja enfermeiros com apenas 6 alocações durante todo o horizonte temporal (enfermeiro 9, teste 1.9 – Tabela 4.6), o que nunca acontece no “Cenário 4”, em que os mínimos são de 13 alocações no horizonte temporal (enfermeiro 9, teste 2.11, solução ótima; enfermeiro 4, teste 2.2, solução ótima; enfermeiro 10, teste 2.2, solução “glpk” – Tabela 4.5).

4.3. TESTES AOS MODELOS MATEMÁTICOS

Enf	1.1		1.2		1.5		1.6		1.7		1.8		1.9		1.11		1.12		1.13		1.14		1.15		1.16		1.17		1.18		1.20		
	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	g/pk	opt	
1	20	16	16	14	20	18	18	18	21	12	19	16	18	15	17	18	18	17	19	13	20	16	18	16	15	15	18	17	20	19	19	18	
2	20	22	22	22	21	18	20	20	17	22	20	22	20	22	17	18	18	17	19	18	20	20	21	21	22	20	22	20	22	20	22	18	20
3	17	16	14	17	20	18	18	14	18	19	15	15	17	15	17	17	18	19	17	19	18	17	16	17	16	16	17	16	17	14	17	16	
4	18	14	16	15	20	18	20	16	15	15	15	15	14	16	17	17	14	18	19	17	14	14	15	15	14	14	13	18	13	18	14	14	
5	18	14	12	12	21	16	17	14	15	14	18	17	14	8	14	14	16	16	16	17	15	14	14	16	14	16	14	13	12	18	14	11	
6	17	13	15	15	19	16	15	14	19	12	15	12	15	14	19	16	11	16	15	15	13	15	12	12	17	14	14	17	13	14	15	14	
7	15	15	14	14	19	16	17	14	14	11	16	12	18	10	13	15	15	17	13	16	11	18	16	18	13	15	17	15	15	14	17	14	
8	16	13	13	13	15	16	13	14	11	11	13	14	7	8	12	18	16	18	12	14	14	7	12	16	13	15	12	16	13	13	13	13	
9	13	11	12	10	18	16	18	14	10	8	10	8	9	6	15	10	13	14	14	14	14	9	14	12	14	16	11	13	14	15	11	11	
10	14	14	15	11	16	16	16	14	15	17	13	10	15	15	16	16	12	15	11	15	15	15	11	12	13	15	16	14	12	14	16	16	
11	16	15	17	15	18	16	16	14	18	12	19	16	16	15	15	17	12	15	16	18	12	16	16	20	13	18	16	16	16	18	13	13	
12	20	12	17	15	18	16	17	14	18	18	15	13	14	18	16	16	12	15	16	13	13	14	17	16	10	14	15	20	18	14	15	15	
13	17	10	15	14	17	16	19	14	21	18	16	19	15	14	12	18	16	14	15	14	12	15	14	13	16	12	17	10	17	14	14	14	
14	14	16	12	15	18	16	16	14	13	10	15	18	17	20	15	16	16	18	15	16	16	17	16	13	15	17	17	14	13	14	13	14	
15	13	15	16	14	17	16	17	14	16	15	15	9	14	12	12	13	14	18	16	15	13	14	14	16	14	16	10	13	16	17	12	12	
16	15	12	14	16	17	16	17	14	13	15	14	18	13	18	12	12	14	16	12	14	18	13	13	13	16	15	16	15	10	14	14	14	
17	17	13	10	17	18	16	16	14	15	15	13	14	17	17	16	15	17	17	15	18	13	17	15	13	14	12	13	17	16	17	14	14	
18	14	19	14	14	15	16	15	14	12	20	15	18	17	17	17	18	14	15	12	16	16	18	14	16	12	15	16	14	16	17	12	12	
19	15	14	12	13	18	16	15	14	14	18	10	15	16	16	12	14	13	19	12	16	14	16	15	15	12	15	12	15	15	14	17	12	
20	15	18	16	16	18	16	17	14	10	18	9	14	13	16	12	13	13	15	14	17	15	13	11	12	15	12	15	13	13	17	17	17	
Méd	16,2	14,6	14,6	14,6	18,15	16,4	16,85	14,6	15,35	14,6	15	14,6	15	14,6	14,6	15,6	14,6	16,55	14,6	16,1	14,6	15	14,6	15,05	14,6	15,15	14,6	16,2	14,6	15,75	14,6	14,6	14,6

Tabela 4.6: Número de dias que cada enfermeiro trabalha, durante todo o horizonte temporal, para todas as soluções obtidas, relativas ao “Cenário 1”.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Dia	1.1		1.2		1.5		1.6		1.7		1.8		1.9		1.11		1.12		1.13		1.14		1.15		1.16		1.17		1.18		1.20	
	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt		
1	10	10	10	10	11	10	11	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
2	10	10	10	10	12	10	13	10	10	10	10	10	10	10	10	10	10	10	10	10	12	10	10	10	10	10	10	10	10	10	10	
3	11	10	10	10	11	10	11	10	12	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
4	12	10	10	10	15	10	12	10	10	10	11	10	10	10	10	10	10	10	12	10	12	10	10	10	10	10	10	10	10	10	10	
5	15	11	11	11	20	20	15	11	12	11	14	11	11	11	11	11	15	11	17	11	15	11	11	11	13	11	12	11	15	11	12	
6	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	11	
7	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	11	
8	12	10	10	10	12	10	12	10	10	10	10	10	10	10	10	10	10	10	10	10	11	10	10	10	10	10	10	10	10	10	10	
9	11	10	10	10	12	10	14	10	12	10	10	10	10	10	10	10	12	10	10	10	13	10	10	10	10	10	10	10	10	10	10	
10	12	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	12	10	10	10	10	10	10	10	10	10	10	
11	11	10	10	10	16	10	11	10	11	10	10	10	11	10	10	10	10	10	12	10	10	10	10	10	10	10	10	10	10	10	10	
12	13	11	11	11	20	20	16	11	15	11	14	11	13	11	11	11	15	11	17	11	15	11	13	11	12	11	12	11	13	11	15	
13	11	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	11	
14	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	11	
15	11	10	10	10	10	10	11	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
16	12	10	10	10	12	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
17	10	10	10	10	11	10	10	10	10	10	10	10	12	10	10	10	10	10	12	10	11	10	12	10	10	10	10	10	10	10	10	
18	11	10	10	10	14	10	12	10	10	10	11	10	11	10	10	10	10	10	12	10	12	10	10	10	10	10	10	10	10	10	10	
19	14	11	11	11	20	20	18	11	12	11	12	11	12	11	11	13	11	13	11	13	11	15	11	12	11	13	11	14	11	14	11	
20	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11	12	11	12	11	11	11	11	11	11	11	11	11	11	11	11	11	
21	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11	
22	10	10	10	10	11	10	10	10	10	10	10	10	10	10	10	10	10	10	11	10	10	10	10	10	10	10	10	10	10	10	10	
23	11	10	10	10	12	10	12	10	10	10	10	10	11	10	10	10	10	10	11	10	10	10	10	10	10	10	10	10	10	10	10	
24	14	10	10	10	11	10	16	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
25	11	10	10	10	13	10	11	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
26	13	11	11	11	20	20	12	11	15	11	11	11	11	11	11	13	11	17	11	14	11	11	11	11	11	11	14	11	13	11	13	
27	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	
28	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	
Méd	11.57	10.43	10.43	10.43	12.96	11.71	12.04	10.43	10.96	10.43	10.71	10.43	10.71	10.43	10.43	10.43	11.11	10.43	11.82	10.43	11.50	10.43	10.71	10.43	10.75	10.43	10.82	10.43	11.57	10.43	11.25	10.43

Tabela 4.7: Número de enfermeiros a trabalhar em cada dia do horizonte temporal, para todas as soluções obtidas, relativas ao “Cenário 1”.

4.3. TESTES AOS MODELOS MATEMÁTICOS

Dia	2.1		2.2		2.5		2.6		2.9		2.10		2.11		2.12		2.13	
	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt
1	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
2	13	13	13	13	13	13	13	13	14	13	13	13	13	13	13	13	13	13
3	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
4	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
5	14	14	14	14	20	20	16	14	14	14	16	14	15	14	14	14	14	14
6	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
7	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
8	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
9	13	13	13	13	14	13	13	13	13	13	13	13	13	13	13	13	13	13
10	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
11	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
12	15	14	14	14	20	20	17	14	14	14	15	14	16	14	15	14	14	14
13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
14	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
15	13	13	13	13	13	13	14	13	13	13	13	13	13	13	13	13	13	13
16	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
17	13	13	13	13	13	13	14	13	13	13	14	13	13	13	13	13	13	13
18	13	13	13	13	13	13	13	13	13	13	13	13	14	13	13	13	13	13
19	15	14	14	14	20	20	16	14	15	14	16	14	15	14	15	14	14	14
20	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
21	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
22	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
23	13	13	13	13	13	13	13	13	13	13	13	13	14	13	13	13	13	13
24	13	13	13	13	13	13	13	13	13	13	13	13	14	13	13	13	13	13
25	13	13	13	13	13	13	13	13	13	13	13	13	14	13	13	13	13	13
26	14	14	14	14	20	20	16	14	17	14	15	14	16	14	14	14	14	14
27	13	13	13	13	13	13	14	13	13	13	13	13	13	13	13	13	13	13
28	13	13	13	13	13	13	14	13	13	13	13	13	13	13	13	13	13	13
Méd	13.21	13.14	13.14	13.14	14.04	14.00	13.61	13.14	13.32	13.14	13.39	13.14	13.50	13.14	13.21	13.14	13.14	13.14

Tabela 4.8: Número de enfermeiros a trabalhar em cada dia do horizonte temporal, para todas as soluções obtidas, relativas ao “Cenário 4”.

Testes a Variações das Necessidades

Nº	Enf./dia	Col.	Linhas	Parâmetros					Tempo	FO	Memória
				FM	Maxa	Mxc	Mnc	Mlc			
3.1	10	3896	5056	3	26	5	2	4	1800	14034	12.6 Mb
3.2	11	3896	5068	3	26	5	2	4	1800	12918	15.4 Mb
3.3	12	3896	5068	3	26	5	2	4	1800	14313	14.0 Mb
3.4	13	3896	5084	3	26	5	2	4	1800	15117	18.5 Mb
3.5	14	3896	5072	3	26	5	2	4	1800	16104	13.0 Mb
3.6	15	3896	5084	3	26	5	2	4	1800	—	—
3.7	15	3896	5084	3	26	5	2	4	1800	—	—

Tabela 4.10: Testes de variações das necessidades. Registos das suas características e seus resultados. A diferença entre os testes 3.6 e 3.7 é que 3.6 testa o “Cenário 6_1” e 3.7 testa o “Cenário 6_2”.

Nº	FO	Tempo de execução	Limitado	
			a 10h	gap (%)
3.1	12312	55.13	Não	0
3.2	12516	2320.28	Não	0.01
3.3	13476	1696.62	Não	0.01
3.4	14931	440.22	Não	0.01
3.5	15576	70925.24	Sim	0.54
3.6	17550	70945.02	Sim	1.16
3.7	17580	70897.25	Sim	1.19

Tabela 4.11: Resultados dos testes de variação das necessidades, corridos no CPLEX. A diferença entre os testes 3.6 e 3.7 é que 3.6 testa o “Cenário 6_1” e 3.7 testa o “Cenário 6_2”.

Com o aumento das necessidades a solução óptima vai também aumentando o seu custo, de uma forma quase linear, como se constata no respectivo gráfico da Figura 4.21.

Observando os extremos, verifica-se uma grande diferença entre o escalonamento da solução óptima obtida para o “Cenário 1” (teste 1.1 ou 3.1, Figura 4.13) e o escalonamento das soluções óptimas obtidas para o “Cenário 6_1” e o “Cenário 6_2” (testes 3.6 e 3.7, Figuras 4.22 e 4.23). O preenchimento dos últimos é muito mais elevado do que o do primeiro, o que é natural, uma vez que há necessidade de mais 50% de enfermeiros do que no primeiro caso.

Observando os dois escalonamentos apresentados nas Figuras 4.22 e 4.23, referentes aos testes 3.6 e 3.7, com o “Cenário 6_1” e o “Cenário 6_2”, respectivamente, e observando os valores da função objectivo para estes dois casos, pode concluir-se que não são muito diferentes. Pelas suas funções objectivo, pode mesmo afirmar-se que o custo de ter necessidade de dois enfermeiros de grau 2 ou superior a trabalhar a mais durante todas as sextas-feiras do horizonte temporal, comparativamente a ter menos um enfermeiro de grau 2

4.3. TESTES AOS MODELOS MATEMÁTICOS

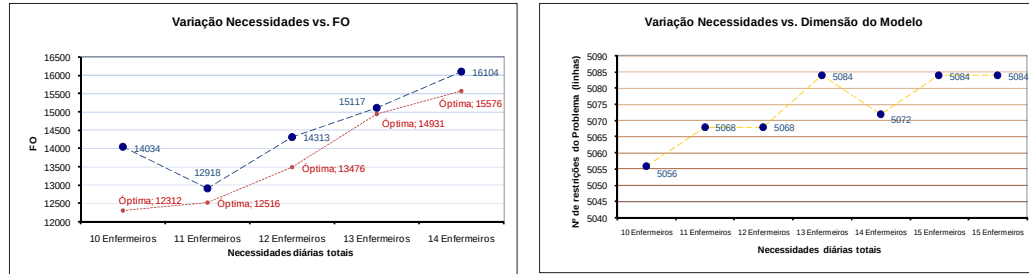


Figura 4.21: Gráficos das variações do valor da Função Objectivo e da dimensão do problema com variações das necessidades.

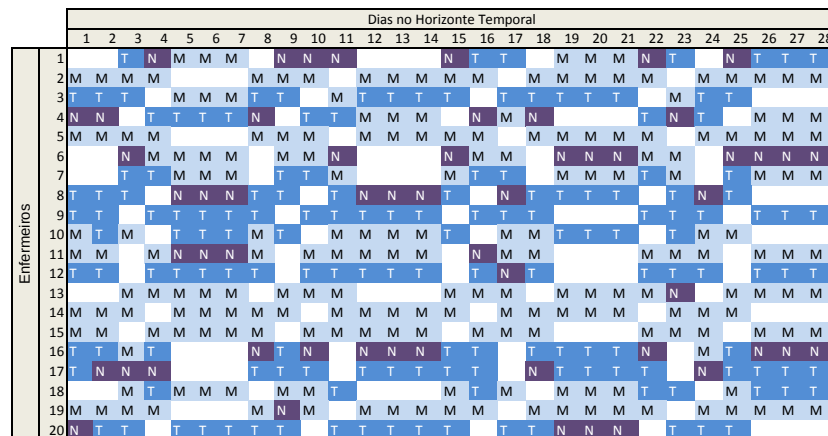


Figura 4.22: Escalonamento óptimo obtido para o teste 3.6 com o CPLEX.

e passando a ter mais um enfermeiro de grau 1 aos fins-de-semana, sem se forçar qualquer dia da semana a mais de 15 enfermeiros alocados, é de 30 (diferença entre os dois valores de função objectivo). Seria de esperar que tal valor fosse mais elevado, mas as soluções obtidas revelaram o interessante facto de o “Cenário 6_2” estar construído de tal forma que permite que apenas se necessite de alocar 15 enfermeiros durante todos os dias do horizonte temporal, incluindo as sextas-feiras. Tal é conseguido da seguinte forma:

- Todos os fins-de-semana de manhã estão alocados 2 enfermeiros de grau 1, 2 de grau 2 e 4 de grau 3:
 - os dois primeiros cobrem as necessidades das sextas-feiras de manhã de enfermeiros de grau 1, as necessidades dos sábados e domingos de manhã de enfermeiros do mesmo grau e ainda um dos enfermeiros de grau 2.

4.3. TESTES AOS MODELOS MATEMÁTICOS

		Dias no Horizonte Temporal																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
Enfermeiros	1		N	N	T	T	T			N	T	N				N	T	M		M	M	M	T	T		N	T	T	T	
	2	M	M	M	M				M	M	M		M	M	M	M		M	M	M	M	M	M		M	M	M	M	M	
	3	T	T	T		M	M	M	T	T		T	M	M	M	T		T	T	T	T		M		T	T				
	4	N	N		T	M	M	M	N		T	M	T	T	T		N	T	N					N	N	T		M	M	M
	5	M	M		M	M	M	M		M	M	M	M	M	M		M	M	M				M	M		M	M	M	M	
	6	M	M	M	M				M	M	M			N	N	N	M	M		N	M	M	M		N	M	M	M	M	M
	7	T	T	M		M	M	M	T	T			M	M	M	M		T	M		T	T	T		T	T	T			
	8			N	N	N	N	N		T	N	N				T	T	N		N	N	N	T	T		N	N	N	N	
	9		T	T	T	T	T		T	T	T					T	T	T		T	T	T	T	T		T	T	T	T	
	10	T	T	T		T	T	T	T	M			T	T	T	T	M		M	T	M	M	M		M	M	M			
	11	M	M		M	M	M	M		M	M	M	M	M	M		M	M					M	M		M	M	M	M	
	12	T	T	T		T	T	T	T				T	T	T	T		N	T	T	T	T		T	T	T		T	T	
	13	M	N	M	M				M	M			M	M	M	M	M		M	M	M	M		M		M	M	M	M	M
	14	M	M		M		N	N	N	M		M	M	M	M	M		M	M				M	N	M		M	M	M	
	15	M	M	M		M	M	M	M	M		M	M	M	M	M		M	M	M	M	M		M	M	M				
	16	N	T		T	T	T	T		T	T	N	N	N		T	T	T					N	T	T		T	T	T	
	17	T	T	T					T	T			T	T	T	N	T		T	N	N	N	T		M	T	N	N	N	
	18			M	T	M	M	M		M	M	T				T	T	M		M	M	M	T	T		M	T	T	T	
	19			M	M	M	M									M	N	M		M	M	M	M	M		M	M	M	M	
	20	T	T	M	T				N	T	N			T	T	T	T	T		T	T	T	T	T		N	T	T	T	T

Figura 4.23: Escalonamento óptimo obtido para o teste 3.7 com o CPLEX.

- meios necessários de grau 2 aos sábados e domingos de manhã;
- os dois enfermeiros de grau 2 cobrem as duas necessidades de grau 2 das manhãs de sextas-feiras e as restantes duas necessidades de grau 2 das manhãs dos sábados e domingos;
- os quatro enfermeiros de grau 3 cobrem as necessidades de 4 enfermeiros do mesmo grau durante as manhãs dos três dias.
- De tarde estão sempre alocados:
 - um enfermeiro de grau 1 a cobrir as necessidades dos três dias do fim-de-semana para esse turno e desse grau;
 - dois enfermeiros de grau 2 que cobrem as duas necessidades de enfermeiros desse grau, durante os sábados e domingos à tarde, cobrem a única necessidade de enfermeiros desse grau, às sextas-feiras de tarde, e uma das três necessidades de enfermeiros de grau 1 ao mesmo dia de tarde;
 - dois enfermeiros de grau 3 que cobrem as restantes duas necessidades das sextas-feiras de tarde e as duas únicas necessidades das tardes de sábados e domingos.
- De noite estão alocados exactamente tantos enfermeiros quantos os necessários durante os três dias e dos respectivos graus: um enfermeiro de grau 2 e um de grau 3.

Aqui se demonstra que o que por vezes parece vir a ter um pior resultado, como era o caso do “Cenário 6_2”, por parecer que iria obrigar a alocar um

4.3. TESTES AOS MODELOS MATEMÁTICOS

enfermeiro a mais às sextas-feiras, pode revelar um resultado surpreendentemente bom, comparativamente ao resultado de um cenário idêntico, com as necessidades aos fins-de-semana (incluindo sextas-feiras) dimensionadas de forma a facilitar o cumprimento do número mínimo de enfermeiros diário durante todos os dias do horizonte temporal. O “Cenário 6_2” também não tem um único dia em que se vejam alocados mais do que 15 enfermeiros, tal como o “Cenário 6_1” (Tabela 4.12).

Dia	3.1		3.2		3.3		3.4		3.5		3.6	3.7
	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	opt	opt
1	10	10	11	11	12	12	13	13	14	14	15	15
2	10	10	11	11	12	12	13	13	14	14	15	15
3	11	10	11	11	12	12	13	13	14	14	15	15
4	12	10	11	11	12	12	13	13	14	14	15	15
5	15	11	11	11	14	12	14	14	15	14	15	15
6	11	11	11	11	12	12	13	13	14	14	15	15
7	11	11	11	11	12	12	13	13	14	14	15	15
8	12	10	11	11	12	12	13	13	14	14	15	15
9	11	10	11	11	13	12	13	13	14	14	15	15
10	12	10	12	11	12	12	13	13	14	14	15	15
11	11	10	11	11	12	12	13	13	14	14	15	15
12	13	11	11	11	14	12	15	14	14	14	15	15
13	11	11	11	11	12	12	13	13	14	14	15	15
14	11	11	11	11	12	12	13	13	14	14	15	15
15	11	10	11	11	12	12	13	13	14	14	15	15
16	12	10	12	11	12	12	13	13	14	14	15	15
17	10	10	11	11	13	12	13	13	14	14	15	15
18	11	10	12	11	12	12	13	13	14	14	15	15
19	14	11	13	11	15	12	15	14	14	14	15	15
20	12	11	11	11	12	12	13	13	14	14	15	15
21	12	11	11	11	12	12	13	13	14	14	15	15
22	10	10	12	11	12	12	13	13	14	14	15	15
23	11	10	11	11	12	12	13	13	14	14	15	15
24	14	10	12	11	12	12	13	13	14	14	15	15
25	11	10	11	11	12	12	13	13	14	14	15	15
26	13	11	11	11	12	12	14	14	15	14	15	15
27	11	11	11	11	12	12	13	13	14	14	15	15
28	11	11	11	11	12	12	13	13	14	14	15	15
	11.57	10.43	11.25	11.00	12.32	12.00	13.21	13.14	14.07	14.00	15.00	15.00

Tabela 4.12: Número de enfermeiros a trabalhar em cada dia do horizonte temporal, para todas as soluções obtidas, relativas aos cenários com diferentes necessidades.

Ainda acerca destes dois escalonamentos, é de notar que para todos os enfermeiros há um mínimo de 4 grupos de folgas ao longo do horizonte temporal, como seria de esperar. O problema aqui é que a maior parte desses grupos apenas tem uma folga e é visível que há muitas alocações consecutivas com apenas uma folga em cada 5 dias de trabalho consecutivo que passam.

Isto significa que estes cenários são demasiado carregados de trabalho para o número de enfermeiros disponíveis e, por isto, também se torna mais difícil encontrar uma solução para o problema (o “glpk” não encontrou qualquer solução para os dois casos).

4.3.4 Observações aos Resultados Obtidos

Com a ajuda da análise dos vários resultados obtidos, nesta secção é feito um comentário direccionado à criação do algoritmo para resolução deste problema. Neste comentário procura-se alertar para os aspectos mais importantes a ter em conta na elaboração de tal algoritmo. As observações e alertas apresentadas são feitas tendo em consideração os resultados do “Modelo Grande”, por serem resultados que mais informação e conhecimento trouxeram e mais aprofundada análise exigiram. A informação apresentada está dividida em partes que facilitam a sua contextualização.

Funções Objectivo

Por vezes, nos testes com o computador pessoal, observa-se que os valores das funções objectivo obtidos para cenários com necessidades diferentes não variam como o esperado. Por exemplo, a solução obtida para o “**Cenário 2**” tem melhor valor de função objectivo do que a solução obtida para o “**Cenário 1**”. A justificação encontrada para esta observação é a de cenários diferentes terem espaços de soluções compostos por soluções admissíveis diferentes, já que uns restringem mais o problema do que outros. É possível observar que o “**Cenário 2**” tem uma grande vantagem relativamente ao “**Cenário 1**”: aos fins-de-semana as necessidades para cada grau são iguais em todos os dias – sexta-feira, sábado e domingo, e, como será explicado mais à frente, isto poderá facilitar muito a procura da solução e a obtenção de uma melhor solução.

Fins-de-semana

Nas soluções obtidas através do computador pessoal, por vezes o número total de enfermeiros alocados por dia é maior do que o número de enfermeiros necessários por dia. Isto pode ocorrer devido à solução obtida não corresponder à solução óptima, o que acontece na maioria dos casos. Mas, comparando

com os valores de enfermeiros a trabalhar por dia nas soluções óptimas (Tabelas 4.7, 4.8 e 4.12), verifica-se que também acontece haver enfermeiros a mais em certos dias. A explicação para este facto é tal observação ocorrer nos fins-de-semana (sexta-feira, sábado e domingo), como é bem visível nas soluções óptimas apresentadas em 4.7, ou à sexta-feira, também visível nas soluções óptimas apresentadas em 4.8. Nos fins-de-semana, sempre que um enfermeiro é alocado a um sábado ou domingo, terá que ser também alocado ao mesmo turno dos restantes dias do fim-de-semana. Se as necessidades ao sábado e ao domingo forem diferentes entre si, vai haver mais enfermeiros alocados durante os três dias do fim-de-semana: tantos quantos os necessários ao sábado mais a diferença entre os necessários ao sábado e os necessários ao domingo, por turno. À sexta-feira haverá tantos enfermeiros alocados como a diferença entre os enfermeiros necessários à sexta-feira e a totalidade dos enfermeiros necessários ao sábado e domingo (cálculo apresentado anteriormente), por isso poderá também acontecer de apenas haver enfermeiros a mais à sexta-feira (caso as necessidades de sábado sejam iguais às de domingo, mas diferentes das de sexta-feira). Estes dois últimos cálculos não são feitos em valores totais, mas sim por cada turno e por cada grau de enfermeiros.

Toda esta análise serve de incentivo a, na continuação do trabalho, pensar bem no modo de atribuições ao fim-de-semana e a forma como se obriga a cumprir os três dias de alocação. Poderá ser interessante procurar, tanto quanto possível, que as necessidades à sexta-feira, sábado e domingo sejam o mais idênticas possível. No entanto, importa também ter em consideração o facto apresentado aquando da comparação entre os “Cenário 6.1” e “Cenário 6.2”. Aí mostrou-se que pode acontecer que as necessidades à sexta-feira, ao sábado e ao domingo não estejam dimensionadas de forma a facilitar, propositadamente, o cumprimento das restrições de fins-de-semana completos e do mesmo tipo de turnos, e, ainda assim, o problema tenha uma solução óptima em que não são alocados mais enfermeiros do que o número de enfermeiros necessários por dia. A chave poderá então estar no estudo das compensações possíveis entre os turnos em conjunto com a distribuição de necessidades. Esta é também questão que começa a entrar num campo de decisão (número de enfermeiros necessários por dia) que não está no âmbito deste estudo e depende de variáveis aqui não consideradas: necessidades dos pacientes da instituição em causa.

Atribuições Nocturnas e Diurnas

As atribuições nocturnas, por ficarem mais dispendiosas devido às leis laborais, podem ter um peso maior no cálculo da função objectivo. Para isso, os custos podem passar a ser por graus e por tipo de turno, com o custo de alocação a turnos nocturnos maior; ou pode ser criado um esquema de contagem das atribuições no horizonte temporal, em que as atribuições nocturnas têm um peso maior do que as atribuições de turnos diurnos, podendo ser criados dois máximos de atribuições no horizonte temporal: um para atribuições de turnos diurnos e outro para atribuições nocturnas (pode também ser criado um terceiro máximo para atribuições mistas).

Nas soluções obtidas (não óptimas), observam-se casos em que um enfermeiro é alocado num turno apenas para cumprir o mínimo estabelecido de 2 turnos atribuídos consecutivos. Nesses casos deverá ser preferencial que o enfermeiro seja alocado a um turno idêntico ao que lhe foi atribuído anteriormente.

Sequências

Acontece muito frequentemente haver sequências de turnos, em dias consecutivos, que podem ter que ser evitadas, por exigirem trabalhar demasiadas horas consecutivas. São essas as sequências:

- Noite/Manhã;
- Manhã/Tarde;
- Tarde/Noite.

Pode também dar-se o caso de este tipo de sequências ser permitido em algumas instituições, onde poderão exigir que após tais sequências haja um número mínimo de turnos de descanso para o enfermeiro repor a capacidade de trabalho. Uma solução possível para um caso geral poderá ser inibir este tipo de sequências, penalizando-as com um custo ou mesmo proibindo-as como “hard constraint”.

Uma forma de evitar estas sequências é impor um intervalo com um certo número mínimo de turnos entre duas atribuições consecutivas em turnos diferentes, que pode variar consoante a sequência de turnos consecutivos. Esta solução pode desrespeitar outras restrições do problema: caso o número

mínimo de dias livres consecutivos seja superior a 1, esta solução pode quebrar essa restrição. Nesse caso pode definir-se uma exceção que permita que haja um dia livre isolado se se tratar de uma mudança de tipo de turno.

É necessário ser definido o número mínimo de dias livres consecutivos de modo a que a restrição do número máximo de dias alocados consecutivos não seja violada, porque, nas soluções obtidas, e por o modelo não contemplar esta restrição, acontece um grande número de atribuições consecutivas com um dia livre no meio delas. Esta situação, normalmente não é desejada porque torna-se facilmente desmotivadora para os enfermeiros.

Poderá ser interessante analisar se será bom para o sistema ou para a motivação dos enfermeiros considerar que, quando ocorrem duas sequências de alocações consecutivas muito próximas uma da outra, estas serem vistas como uma sequência maior e mais abrangente. Isto pode acontecer quando não se define número mínimo de dias livres consecutivos e se pretende que após duas sequências de alocações consecutivas, separadas apenas por um dia livre, se exija um determinado número de dias livres consecutivos mínimos. Isto poderá ser uma alternativa a criar número mínimo de dias livres consecutivos, com exceção para o caso de mudança de turno, referido anteriormente.

Preferências

Como já referido, as preferências dos enfermeiros são em grande parte das vezes respeitadas. Pode observar-se esse cumprimento em casos difíceis de resolver, como os casos dos “Cenário 6_1” e “Cenário 6_2”, em que se observam poucos desrespeitos às preferências, acontecendo mesmo a certos enfermeiros apenas trabalharem num turno (enfermeiros 2, 5, 9, 14 e 15 no escalonamento da Figura 4.22 e enfermeiros 2, 5, 9, 11 e 15 da Figura 4.23).

Grupos

Poderá ser interessante alocar, tanto quanto possível, os enfermeiros por grupos de trabalho. Esses grupos consistem em grupos definidos de enfermeiros que trabalham sempre, ou sempre que possível, juntos. Este pode ser um factor motivador para os profissionais em questão.

Capítulo 5

Introdução aos *Algoritmos Genéticos*

5.1 Introdução

Neste capítulo apresenta-se uma introdução aos *Algoritmos Genéticos*. Apesar de este texto poder ser usado como um guia à aplicação dos *Algoritmos Genéticos* a qualquer problema de Optimização Combinatória, os exemplos dados e as ilustrações dos conceitos referem-se ao problema abordado nesta tese: *Nurse Rostering*.

No que respeita à organização do capítulo, na primeira secção, a estrutura dos *Algoritmos Genéticos* é introduzida de uma forma genérica, enquanto a segunda secção oferece uma visão mais detalhada do funcionamento destes algoritmos. As seguintes secções são dedicadas à descrição dos blocos mais importantes num *Algoritmo Genético*, terminando o capítulo com uma secção onde são feitas algumas observações adicionais sobre a parametrização e afinação do algoritmo.

5.2 *Algoritmo Genético* – Uma Visão Inicial

Os mecanismos essenciais que ligam os *Algoritmos Genéticos* ao problema que se pretende resolver são os seguintes:

- **Codificação** das soluções do problema em cromossomas;
- **Função avaliadora**, que mede a qualidade da solução representada por cada cromossoma, no contexto do problema.

Assente nestes mecanismos, o funcionamento geral de um *Algoritmo Genético* pode ser descrito da seguinte forma:

1. Iniciar uma população de cromossomas (soluções iniciais normalmente obtidas de forma aleatória)
2. Avaliar cada cromossoma da população
3. Criar novos cromossomas “casando” os cromossomas pais:
“Casar” os cromossomas pais significa aplicar-lhes os operadores mutação e *crossover*
4. Apagar membros da população para criar espaço para a inserção dos novos cromossomas (caso se apague apenas alguns dos membros da população parental, deve-se excluir os que têm piores valores de função avaliadora)
5. Avaliar os novos cromossomas e incluí-los na população (pode optar-se por apenas incluir os que têm melhores valores nessa avaliação, seleccionando, por exemplo, os cromossomas com um valor acima de um determinado valor mínimo)
6. Se o tempo definido terminar, parar e devolver, como solução, o melhor cromossoma; se não, voltar para o ponto 3.

Na Figura 5.1 está representado graficamente o funcionamento genérico de um *Algoritmo Genético*, recorrendo a um exemplo com uma população de 100 elementos.

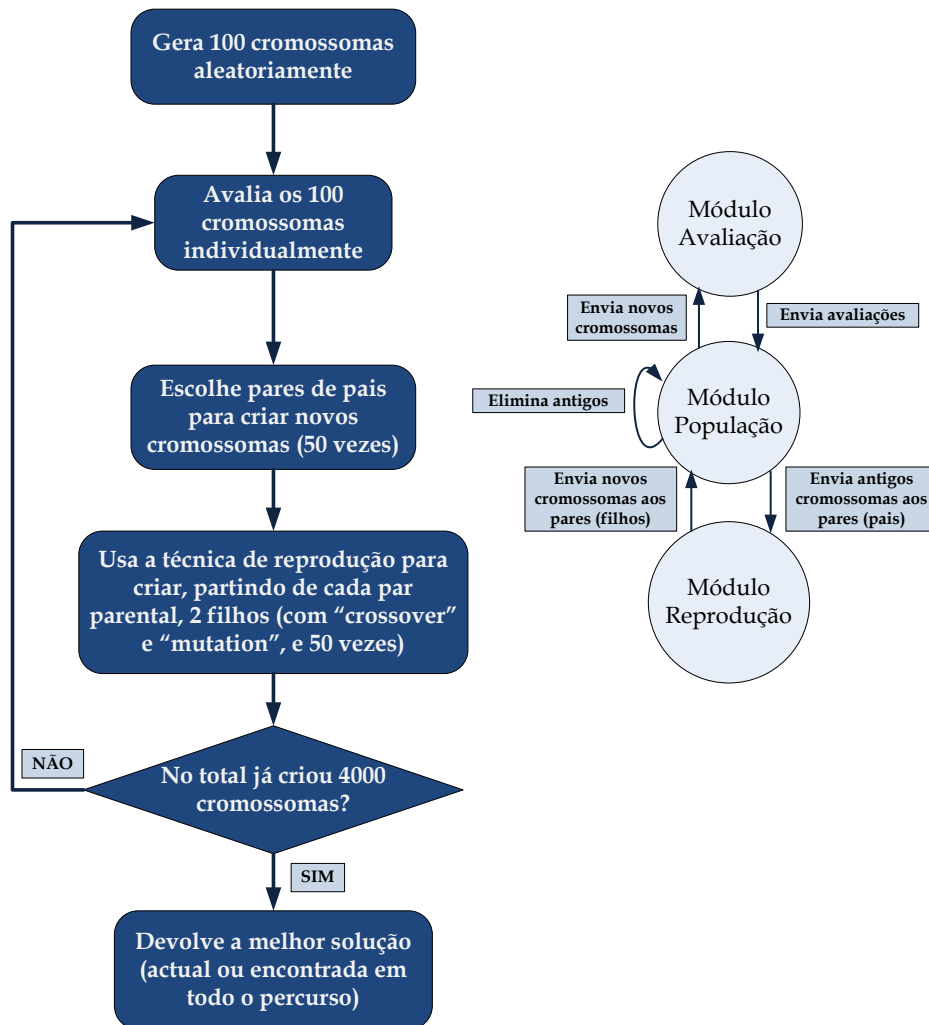


Figura 5.1: Exemplo ilustrativo do funcionamento e organização de um *Algoritmo Genético* (elaborado a partir do texto em [20]).

5.3 Anatomia dos *Algoritmos Genéticos*

Quando se pretende aplicar os *Algoritmos Genéticos* num caso prático, deve-se encontrar uma lógica para a sua organização e implementação. Uma organização adequada a este tipo de algoritmos é a divisão em três módulos:

- Função Avaliadora;
- População;
- Reprodução.

5.3.1 Função Avaliadora

A **Função Avaliadora**, ou **Fitness Measure**, faz a ligação entre o *Algoritmo Genético* e o problema a resolver. O seu funcionamento está ilustrado na Figura 5.2.



Figura 5.2: Funcionamentos da Função Avaliadora.

Esta função serve para medir a qualidade das soluções (cromossomas) existentes em cada população criada durante o processo de pesquisa. A função de avaliação deve estar construída com base nos objectivos e nos propósitos do problema que se está a resolver. Assim, esta terá, normalmente, uma ligação directa à função objectivo do problema, podendo ser igual à mesma.

5.3.2 População

Este módulo tratará de todas as acções e definições relacionadas com os cromossomas constituintes da população em análise. Tem que definir:

- Técnica de Representação ou Técnica de Codificação
Contemplando o tamanho da representação (tamanho dos cromossomas)
- Técnica de Inicialização (usualmente a população inicial é gerada aleatoriamente)

- Técnica de Eliminação
- Técnica de Reprodução (modo de substituição, entre gerações de cromossomas)
- Técnica de Selecção dos Pais (para a criação de novos cromossomas)
- Técnica “Fitness” (“Reproductive Fitness” – normalmente é a função avaliadora, que avalia cada cromossoma)
- Técnicas de Parametrização (por exemplo, parametrização das probabilidades de *crossover* e mutação em cada evolução para a geração seguinte)
- Tamanho da População (uma população de pequena dimensão, pode impedir que haja diversidade suficiente para que o *Algoritmo Genético* faça uma boa exploração do espaço de soluções, limitado-o a uma pequena parte desse espaço; por outro lado, uma população demasiado grande pode abrandar substancialmente a velocidade de progressão do *Algoritmo Genético*)
- Número de tentativas desejadas (populações criadas até ao final do algoritmo)

Inicialização

Ao implementar *Algoritmos Genéticos*, tem que se decidir como se pretende criar a população inicial para a pesquisa a efectuar. Muitas vezes utiliza-se uma população aleatória. Neste caso, são criados tantos cromossomas quantos os necessários para definir uma população completa de forma aleatória. Os cromossomas criados terão que ser todos diferentes e que representar soluções admissíveis para o problema em questão. Para garantir a admissibilidade de todas as soluções, cria-se um método para verificar cada um dos cromossomas, incluindo uma lógica de alteração do cromossoma, caso este não represente uma solução admissível, para o tornar admissível.¹

Quando o *Algoritmo Genético* é um *Algoritmo Genético Híbrido*, ou seja, é usado em conjunto com outro tipo de algoritmos para resolver o problema em questão, a população inicial pode já existir, vinda de outra fase do processo global de optimização. Neste caso, as soluções que aí existem devem ser adaptadas para o formato dos cromossomas (o equivalente a codificá-los), antes de serem usadas pelo *Algoritmo Genético*. Isto acontece, por exemplo, em Burke et al., [4].

¹Uma solução admissível é entendida, neste documento, como sendo uma solução que respeita todas as restrições obrigatórias (“hard constraints”) do problema.

Interessa que a população inicial não seja composta por soluções muito próximas umas das outras no espaço de soluções. Se o *Algoritmo Genético* for iniciado com soluções distantes umas das outras (ou seja, longe das vizinhanças umas das outras), o algoritmo genético poderá ter uma maior abrangência e evitar que fique rapidamente preso a um espaço limitado de soluções, dominado por um ótimo local.

Eliminação

Para evoluir de uma população para a seguinte, é necessária a definição de uma técnica de eliminação de cromossomas da população antiga, que permita a inclusão de novos cromossomas na nova população.

Há duas técnicas de eliminação muito utilizadas nos trabalhos que usam *Algoritmos Genéticos*. Pode-se eliminar toda a população actual e substituí-la integralmente por novos cromossomas criados, ou pode-se eliminar somente uma parte seleccionada da população, fazendo sobreviver alguns cromossomas, os melhores, e criando apenas o número de cromossomas necessários para completar o resto da nova população.

Este segundo modo pode ser implementado recorrendo a uma de duas técnicas:

Elitismo Por vezes, ao passar de uma população para a seguinte, substituindo todos os cromossomas existentes, perdem-se soluções importantes e de boa qualidade, com grande potencial de ajuda na procura de solução. Ao passar para a geração seguinte, arriscamo-nos a que o potencial desses cromossomas de grande qualidade não seja tão bem aproveitado como o desejável. Para colmatar estas falhas, pode optar-se por fazer resistir, de uma população para a seguinte, o melhor, ou alguns dos melhores cromossomas da população de origem. Esta estratégia pode aumentar a eficácia do *Algoritmo Genético*. Por outro lado, permanece também o risco de permitir que uma população seja dominada por um “super indivíduo”.

“Steady State Selection” Usando a técnica de eliminação elitismo, continuar-se-á a perder um grande número de soluções com boa qualidade, ao progredir de geração para geração, uma vez que se fazem sobreviver um número muito reduzido de cromossomas. Se se quiser aproveitar melhor o potencial de grande número das soluções existentes em cada geração, deve-se utilizar o “Steady State Selection”. Esta técnica consiste em deixar sobreviver um grande número de soluções: as melhores. Normalmente, os praticantes desta técnica criam muito poucos cromossomas novos de geração para geração (um ou dois de cada vez), mas pode optar-se por criar mais cromossomas. Esta

técnica introduz, assim, mais um parâmetro: o número de cromossomas que serão criados, que é o mesmo que o número de cromossomas que serão eliminados de uma geração para a seguinte. O “Steady State Selection” torna o algoritmo mais eficaz, comparativamente às outras duas técnicas, quando criado o mesmo número total de cromossomas numa pesquisa. Mas, a evolução da pesquisa de optimização é muito mais lenta, uma vez que precisa de muitas mais gerações para criar o mesmo número de cromossomas.

Nesta técnica acontece facilmente a criação de cromossomas duplicados, ou seja, acontece a criação de novos cromossomas iguais a cromossomas já existentes na população. Isto atrasa ainda mais a evolução da pesquisa. Para evitar este atraso, existe a hipótese de configurar o *Algoritmo Genético* de modo a utilizar o “Steady State Selection Without Duplicates”, apresentado em [20]. Esta variante, aqui resumida, exigirá a definição de testes no módulo de reprodução. Estes testes ocuparão mais tempo de execução durante a pesquisa, mas, segundo Davis, em [20], em problemas reais a maior parte do tempo de execução é utilizada na avaliação das soluções e o tempo adicional que estes testes requerem é insignificante quando comparado ao tempo total de execução.

5.3.3 Reprodução

Neste módulo define-se tudo o que está directamente relacionado com a reprodução dos cromossomas. Este é o módulo a que se recorrerá sempre que se tencione evoluir para a geração de soluções seguinte. Ter-se-á que definir:

- ◇ Técnica de Selecção do Operador
- ◇ Operadores (*Crossover*, Mutação, ...) e seus parâmetros (como probabilidades de utilização)
- ◇ Técnicas de Parametrização (por exemplo, parametrização de interpolação dos pesos dos operadores, que será explicado no final do capítulo)
- ◇ Pesos dos operadores

5.4 Técnicas de Codificação

Existem várias técnicas de codificação. Entre elas destaca-se a “bit string”, proposta por John Holland nas primeiras aplicações destes algoritmos. Nesta técnica, a solução é sempre codificada por um vector de bits:

0 1 1 0 0 1 0 1 1

Esta técnica é usada por Kathryn A. Dowsland na resolução do problema de *Nurse Rostering*, em [12] e [13]. Na abordagem aí desenvolvida, o problema é de escalonamento semanal e são considerados dois tipos de turno: dia ou noite. É de salientar que a resolução desse problema está organizada em três fases: verificação da existência do número de enfermeiros necessários para resolver o problema; alocação dos enfermeiros a dias ou noites; especificação de alocações a dias, como tardes ou manhãs. O *Algoritmo Genético* ocupa-se da segunda fase. Está dividido em duas partes: definição de escalonamentos; atribuição dos escalonamentos criados aos enfermeiros. Na primeira parte, os cromossomas representam escalonamentos que podem ser atribuídos aos enfermeiros. Cada cromossoma tem 14 genes (parcelas): as primeiras sete representam os turnos diurnos e as últimas sete representam os turnos nocturnos de cada dia da semana. O valor contido em cada gene indica se o turno correspondente pertence a esse escalonamento (no caso de ser um “1”) ou não (no caso de ser um “0”).

A técnica usada tem que se adaptar ao problema particular que se pretende resolver com o *Algoritmo Genético*. Assim sendo, podem ser criadas, além do vector de bits, outras formas de codificar a solução.

No problema do Caixeiro Viajante (problema típico de Optimização Combinatória), a codificação deverá ser diferente. Neste caso, o caixeiro terá que passar por um determinado número de cidades, terminando o seu percurso na cidade de onde partiu e fazendo-o sem passar mais do que uma vez pela mesma cidade. Imaginemos que o caixeiro tem que passar por nove cidades. A codificação usualmente associada a este problema é um vector de números inteiros como o seguinte:

2 3 7 9 4 1 5 6 8

Além do problema do Caixeiro Viajante, esta técnica pode também ser usada noutros problemas de Optimização Combinatória em que as soluções sejam normalmente representadas por sequências de números. Margarida Moz e Margarida Vaz Pato usam-na como uma das técnicas de codificação num problema de *Nurse Rerostering*, em [8], em que cada cromossoma representa um escalonamento com as tarefas que cada enfermeiro irá realizar ao longo do horizonte temporal (ordenadas pelos dias do horizonte temporal nos genes do cromossoma).

Outra técnica possível é a de codificar as soluções com outro qualquer tipo de valores, como sejam valores reais, caracteres ou outra simbologia:

1.567 3.613 2.767 6.123 1.946 1.892 5.001 3.912 4.376

A B Y W H D A R Y

(black) (yellow) (yellow) (green) (black) (blue) (yellow)

Margarida Moz e Margarida Vaz Pato, em [8], propõem também uma outra técnica de codificação, baseada em representação com números reais aleatórios, por elas criada. Esta é usada para criar os cromossomas representados com a codificação anteriormente referida (codificação inteira). O processo de codificação é feito seguindo os passos:

1. Inicialmente tem-se um vector em que cada posição corresponde à tarefa com o índice dessa posição: posição 1 do vector $\rightarrow$ tarefa 1; posição 2 do vector $\rightarrow$ tarefa 2; ...;

2. Preenche-se esse vector com valores reais, entre 0 e 1, aleatórios:

$(0.50 \ 0.02 \ 0.10 \ 0.03 \ 0.25 \ 0.09);$

3. O valor que está em cada posição corresponde à tarefa identificada com o número dessa posição do cromossoma: $0.50 \rightarrow$ tarefa 1; $0.02 \rightarrow$ tarefa 2; ...;

4. Ordena-se o vector reposicionando os valores reais por ordem crescente:

$(0.02 \ 0.03 \ 0.09 \ 0.10 \ 0.25 \ 0.50);$

5. Substituindo os valores reais pelos números das tarefas a que correspondem (segundo o passo 3), obtém-se um novo cromossoma para o problema:

$(2, 4, 6, 3, 5, 1).$

5.5 Técnicas de Selecção dos Pais

A selecção de dois cromossomas da população, para criação de dois novos cromossomas para a população seguinte, deve ser feita com base na qualidade dos cromossomas “pais”, medida pela função de “Fitness”. Mas, para garantir a diversidade da população, é também necessário garantir que todos os cromossomas tenham possibilidade de ser escolhidos para pais, de modo a que não se restrinja a nova população às combinação possíveis entre alguns dos melhores cromossomas da população actual (por *crossover* e mutação).

O importante, ao definir o mecanismo de selecção de cromossomas pais

para criação da nova população, será então garantir que, havendo a possibilidade de todos os cromossomas da população existente serem escolhidos, haja maior probabilidade de o serem os cromossomas com melhor “Fitness”. Poderão existir diversas formas de conseguir este propósito, das quais se apresenta seguidamente as mais utilizadas em Optimização Combinatória.

5.5.1 *Roulette Wheel Parent Selection*

Esta técnica pressupõe que os cromossomas existentes na população estão numerados de acordo com uma determinada ordem (que pode ser aleatória). Funciona da seguinte forma, na escolha de cada cromossoma pai:

1. Somar os valores de “Fitness” de todos os cromossomas da população existente $\Rightarrow$ “Total Fitness”;
2. Gerar um número n , aleatoriamente, com valor entre 0 e “Total Fitness”;
3. Devolver o primeiro cromossoma da população cujo valor de “Fitness”, adicionado aos valores de “Fitness” dos cromossomas anteriores, é maior ou igual a n .

Exemplo: Em seguida é apresentado um exemplo do funcionamento da técnica “*Roulette Wheel Parent Selection*”.

Na Tabela 5.1 estão apresentados os valores de “Fitness” de cada cromossoma da população actual, de um problema hipotético. Os cromossomas desta população estão numerados, ordenadamente, como sugerido nessa tabela. Para cada novo cromossoma é também calculado o valor de “Fitness” acumulado (soma do valor de “Fitness” do cromossoma em questão com todos os valores de “Fitness” dos cromossomas anteriores), em que o valor do último cromossoma corresponde ao “Total Fitness”.

Cromossoma	1	2	3	4	5	6	7	8	9	10
Valor de “Fitness”	8	2	17	7	2	12	11	7	3	7
“Fitness” acumulado	8	10	27	34	36	48	59	66	69	76

Tabela 5.1: Valores de “Fitness” dos cromossomas existentes numa população.

Na Tabela 5.2 estão apresentadas sete ocorrências da aplicação desta técnica. Na primeira linha estão os números n , criados aleatoriamente com

5.5. TÉCNICAS DE SELECÇÃO DOS PAIS

Número Aleatório	23	49	76	13	1	27	57
Cromossoma Escolhido	3	7	10	3	1	3	7

Tabela 5.2: Escolha de cromossomas para pais, segundo a técnica *Roulette Wheel Parent Selection*.

uma distribuição uniforme, e, na segunda, o cromossoma que, segundo as regras desta técnica, é escolhido para cromossoma pai.

Como é bem visível, comparando as duas tabelas referidas, quanto maior for o valor de “Fitness” de um cromossoma, maior é a probabilidade de ser escolhido para pai. Na Tabela 5.2, o cromossoma mais escolhido foi o cromossoma 3, que é também o que tem maior valor de “Fitness”.

Este modo de escolha do cromossoma pai pode ser comparado a uma roleta de casino, onde cada “fatia” da roleta tem o tamanho percentual equivalente ao valor de “Fitness” do cromossoma que representa, relativamente ao “Total Fitness”. A Figura 5.3 representa a roleta deste exemplo.

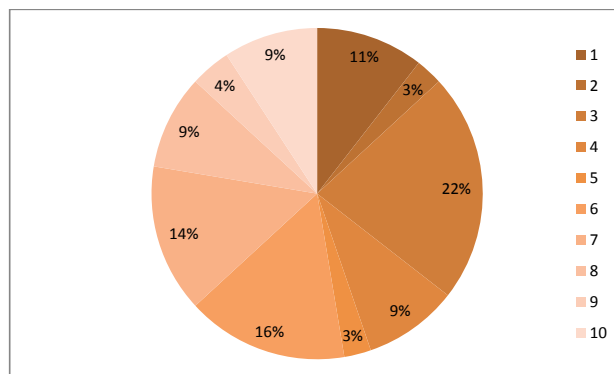


Figura 5.3: Roleta representativa da probabilidade dos cromossomas serem escolhidos, consoante o seu valor de “Fitness”. Em cada fatia está escrita a probabilidade do cromossoma correspondente ser escolhido.

5.5.2 Rank Parent Selection

O modo de selecção *Roulette Wheel Parent Selection* facilita a ocorrência de problemas na selecção dos cromossomas quando os valores de “Fitness” diferem muito uns dos outros, especialmente se houver um cromossoma cujo valor de “Fitness” ocupe 90% da roleta. Em tal situação os restantes cromossomas serão muito poucas vezes escolhidos, o que condiciona o bom desenrolar do processo de pesquisa.

5.5. TÉCNICAS DE SELECÇÃO DOS PAIS

Para evitar esse tipo de inconveniente, o *Rank Parent Selection* procede da seguinte forma:

1. Faz um ranking dos cromossomas consoante os seus valores de “Fitness”, em que na primeira posição está o cromossoma com pior “Fitness” e na última o cromossoma com melhor;
2. Atribui a cada cromossoma um peso igual à posição que ocupa no ranking.

No exemplo anterior, aplicando esta técnica, passa-se a ter a distribuição apresentada na Tabela 5.3 e uma roleta correspondente, ilustrada na Figura 5.4.

Cromossoma	1	2	3	4	5	6	7	8	9	10
Valor de “Fitness”	8	2	17	7	2	12	11	7	3	7
Ranking	7	1	10	4	2	9	8	5	3	6

Tabela 5.3: Ranking dos cromossomas existentes numa população relativo aos seus valores de “Fitness”.

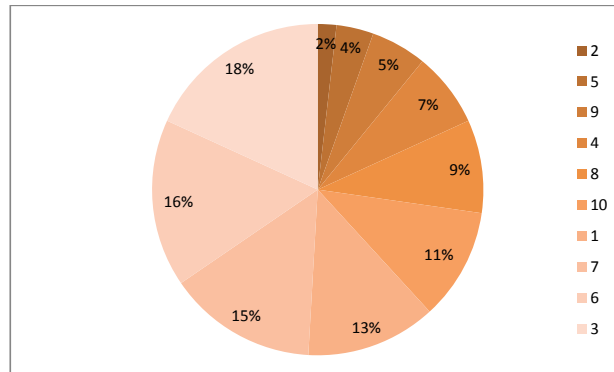


Figura 5.4: Roleta representativa da probabilidade dos cromossomas serem escolhidos, consoante o seu ranking. Em cada fatia está escrita a probabilidade do cromossoma correspondente ser escolhido.

A desvantagem de utilizar esta técnica é que conduz a uma convergência mais lenta, por a diferença entre os melhores e os piores cromossomas não ser tão elevada.

5.5.3 *Tournament*

O *Tournament* é uma técnica muito utilizada em aplicações de *Algoritmos Genéticos*, pela simplicidade de funcionamento e por evitar convergências prematuras, graças à sua aleatoriedade na selecção dos cromossomas, distribuída quase igualmente por todos eles.

A técnica funciona, para a escolha de cada cromossoma “pai”, do seguinte modo:

1. são seleccionados dois cromossomas da população aleatoriamente;
2. entre os dois, o que tiver maior valor de “Fitness” é escolhido.

Assim, diminui o risco de convergir muito rapidamente para um óptimo local, não condicionando a pesquisa ao conjunto de soluções envolventes, por um cromossoma ter uma probabilidade muito alta de ser escolhido (tornando-o um cromossoma dominante), como pode suceder com a técnica *Roulette Wheel Parent Selection*. Nem se corre um risco tão elevado, como com a técnica de *Rank Parent Selection*, de convergir com demasiada lentidão. Todos os cromossomas têm a mesma probabilidade de serem seleccionados para o “torneio”, em que vencerá o que tiver melhor “Fitness”, como descrito anteriormente.

5.6 *Crossover*

Existem várias técnicas de *crossover* dos cromossomas. Aqui estão apresentadas algumas delas, que se poderão adaptar a vários tipos de problema, para os quais, certamente, umas serão mais apropriadas do que outras.

5.6.1 *One-Point Crossover*

O *One-Point Crossover* é uma técnica simples de *crossover*. Para cada *crossover*, exigindo dois pais, escolhe-se aleatoriamente um ponto do cromossoma que servirá de referência. Dos dois pais resultarão dois filhos: um constituído pela cópia do lado esquerdo, relativamente ao ponto de referência, do cromossoma de um dos pais e pela cópia do lado direito do outro pai; o segundo filho, constituído por cópias das outras partes dos pais. Esta técnica está esquematizada na Tabela 5.4.

5.6.2 *Two-Point Crossover*

O *Two-Point Crossover* é idêntico ao *One-Point Crossover*, mas, em vez de um, tem-se dois pontos de referência, escolhidos aleatoriamente. Um dos

5.6. Crossover

Pai 1	0	1	0	0	1		1	1	0	1
Pai 2	1	1	0	1	1		0	0	0	0

Filho 1	0	1	0	0	1		0	0	0	0
Filho 2	1	1	0	1	1		1	1	0	1

Tabela 5.4: Exemplo de *One-Point Crossover*. O filho 1 é constituído pelo lado esquerdo do pai 1 e o lado direito do pai 2. O filho 2 é constituído pelo lado esquerdo do pai 2 e pelo lado direito do pai 1.

filhos será gerado copiando o conteúdo exterior a esses dois pontos de um dos pais e o conteúdo interior a esses dois pontos do outro pai. O outro filho será gerado do modo contrário. Este procedimento está exemplificado na Tabela 5.5, com os mesmos cromossomas pais do exemplo da Tabela 5.4.

Pai 1	0	1		0	0	1	1	1		0	1
Pai 2	1	1		0	1	1	0	0		0	0

Filho 1	0	1		0	1	1	0	0		0	1
Filho 2	1	1		0	0	1	1	1		0	0

Tabela 5.5: Exemplo de *Two-Point Crossover*. O filho 1 é constituído pelas partes externas aos pontos de referência do pai 1 e pela parte interna do pai 2. O filho 2 é constituído pelas partes externas do pai 2 e pela parte interna do pai 1.

Usualmente, esta técnica tem melhor performance do que o *One-Point Crossover*, embora dependa do problema que se está a resolver.

Quer no *One-Point Crossover*, quer no *Two-Point Crossover*, a escolha dos pontos de referência exige a definição de uma probabilidade de escolha para cada ponto.

5.6.3 Uniform Crossover

Esta técnica é a mais complexa, porque exige a criação de um “template”. O “template” é um vector binário, com tantos elementos quantos os elementos dos cromossomas. Na criação dos filhos, para um deles, sempre que no template houver um “1”, o elemento correspondente será igual ao elemento do pai 1, quando houver um “0”, o elemento correspondente será igual ao do pai 2. No segundo filho será ao contrário (Tabela 5.6).

Pai 1	0	1	0	0	1	1	1	0	1
Pai 2	1	1	0	1	1	0	0	0	0
Template	1	1	0	1	0	0	1	1	0
Filho 1	0	1	0	0	1	0	1	0	0
Filho 2	1	1	0	1	1	1	0	0	1

Tabela 5.6: Exemplo de *Uniform Crossover*. O filho 1 é constituído pelos elementos do pai 1, sempre que o template tem um “1”, e pelos elementos do pai 2, sempre que o template tem um “0”. O filho 2 é constituído pela lógica inversa.

São poucas as vezes em que esta técnica tem melhores resultados do que o *Two-Point Crossover*. São necessários casos muito específicos, com determinados parâmetros, para que ela se revele melhor.

5.6.4 Crossover Aritmético

Outra forma de fazer *crossover* é aplicando uma operação aritmética aos cromossomas pais para criar um cromossoma filho. Por exemplo, aplicando um “E” lógico a dois cromossomas pais binários, como na Tabela 5.7.

Pai 1	1	1	0	0	1	0	1	1
Pai 2	1	1	0	1	1	1	1	1
Filho Único	1	1	0	0	1	0	1	1

Tabela 5.7: Exemplo de *Crossover Aritmético*. O filho é constituído pelo resultado da operação lógica “E” entre os seus dois pais.

5.6.5 Observações

Estas técnicas de *crossover* terão que ser adaptadas quando a técnica de codificação não permite que num cromossoma existam genes com valores repetidos. Por exemplo, no caso da codificação proposta para o problema do Caixeiro Viajante, se se aplicar o *One-Point Crossover*, certamente que aparecerão pontos repetidos nos cromossomas resultantes, Tabela 5.8. No entanto, isto não significa que esta técnica não possa ser utilizada, tal como as restantes técnicas de *crossover*. Tem simplesmente de ser adaptada ao modo como os cromossomas estão codificados.

5.6. CROSSOVER

Pai 1	1	5	8	2		4	6	7	3
Pai 2	2	7	1	3		5	4	8	6

Filho 1	1	5	8	2		5	4	8	6
Filho 2	2	7	1	3		4	6	7	3

Tabela 5.8: *One-Point Crossover*, sem alterações, para o problema do Caixeiro Viajante. Nenhum dos filhos passa por todos os pontos e ambos passam por pontos repetidos.

Modificando esta técnica como sugerido em [22], ela passa a ser viável para problemas como o do Caixeiro Viajante. Para este tipo de problemas, cada filho será constituído por uma cópia de um dos pais até ao ponto de *crossover*. A partir desse ponto, percorre-se o segundo pai, da esquerda para a direita, e, sempre que se encontrar um gene que ainda não está representado no filho, acrescenta-se-o à primeira posição livre do filho, obtendo-se os resultados desejados, como demonstrado na Tabela 5.9.

Pai 1	1	5	8	2		4	6	7	3
Pai 2	2	7	1	3		5	4	8	6

Filho 1	1	5	8	2		7	3	4	6
Filho 2	2	7	1	3		5	8	4	6

Tabela 5.9: *One-Point Crossover* adaptado para o problema do Caixeiro Viajante. Nenhum dos filhos passa por pontos repetidos.

Em [8], cuja técnica de codificação já aqui foi referida, são apresentadas duas adaptações idênticas a esta, mas para o *Two-Point Crossover*. Na primeira adaptação, copia-se todos os genes de um dos pais entre os dois pontos de *crossover*, e somente aqueles do outro pai que ainda não existam no filho, fora dos pontos de *crossover*. Para os genes que ficam a faltar usa-se a técnica exemplificada na Tabela 5.10:

*Para a primeira posição do **Filho 1**: o segundo cromossoma pai tem um 8. Este número já existe no cromossoma filho, ocupando o gene com o valor 2 nesse segundo pai. Este valor, 2, também já existe no cromossoma filho e ocupa a posição do valor 6 no pai. O valor 6 não existe no filho, logo este será o valor que preencherá a primeira posição do **Filho 1** ($8 \rightarrow 2 \rightarrow 6$). Esta lógica também é seguida para preencher os restantes valores em falta dos cromossomas filhos.*

Pai 1	3	5		2	10	8	4		6	1	7	9
Pai 2	8	9		6	4	2	3		7	10	5	1
Filho 1'	-	9		2	10	8	4		7	-	5	1
Filho 2'	-	5		6	4	2	3		-	1	7	9
Filho 1	6	9		2	10	8	4		7	3	5	1
Filho 2	10	5		6	4	2	3		8	1	7	9

Tabela 5.10: Primeiro *Two-Point Crossover* adaptado para o problema em [8]. Nenhum dos filhos passa por pontos repetidos. Após copiados os genes que não se repetem, obtêm-se os filhos **Filho 1'** e **Filho 2'**. Mas estes ainda não são cromossomas completos. Para os completar usa-se a técnica descrita e obtém-se os filhos **Filho 1** e **Filho 2**.

Uma segunda adaptação do *Two-Point Crossover* é apresentada no mesmo artigo. Nessa, e tal como na anterior, os genes entre os pontos de *crossover* são copiados de um dos pais. Os restantes são copiados do outro pai, percorrendo-o a partir do segundo ponto de *crossover*, à medida que são encontrados valores que ainda não fazem parte do cromossoma filho, para a primeira posição livre a partir do segundo ponto de *crossover* (Tabela 5.11).

Pai 1	3	5		2	10	8	4		6	1	7	9
Pai 2	8	9		6	4	2	3		7	10	5	1
Filho 1	6	3		2	10	8	4		7	5	1	9
Filho 2	10	8		6	4	2	3		1	7	9	5

Tabela 5.11: Segundo *Two-Point Crossover* adaptado para o problema em [8]. Nenhum dos filhos passa por pontos repetidos. Os filhos **Filho 1** e **Filho 2** são obtidos após aplicação da técnica descrita no texto referente a esta adaptação.

Estas adaptações têm propósitos distintos. A primeira procura garantir que as posições dos genes dos cromossomas pais sobrevivem para a geração seguinte, tanto quanto possível. A segunda procura que seja a ordem dos genes dos pais a sobreviver para a geração seguinte, ou seja, procura minimizar as alterações nas ordens dos genes constituintes dos cromossomas pais, quando se criam os cromossomas filhos.

5.7 Mutação

Além da aplicação do *crossover* a dois cromossomas pais, a criação de novos cromossomas inclui também, tal como na genética biológica, mutação de alguns genes em alguns dos cromossomas criados. Existe uma probabilidade associada à aplicação da mutação a cada cromossoma criado e uma probabilidade de aplicação da mutação a cada gene do cromossoma. Normalmente a primeira é muito pequena, bastante menor do que a probabilidade de aplicação do *crossover*. A segunda poderá também ser muito reduzida para impedir que sejam escolhidos mais do que um ou dois genes do cromossoma escolhido, embora dependa do tipo de aplicação, pois, por vezes, pode exigir a escolha de mais de um cromossoma.

A mutação garante a existência de diversidade na população, de geração para geração. Mas a sua aplicação necessita de ser bem medida, para que não se perca demasiada informação contida em cromossomas de uma geração, quando se cria uma nova geração, tornando a nova geração completamente diferente e independente da anterior, sem manter um historial de ligação, e desrespeitando a intenção base dos *Algoritmos Genéticos*.

A mutação consiste na alteração da informação contida num gene do cromossoma. Isto pode ser feito de várias formas e o modo de aplicação depende da codificação utilizada. Se se tratar de uma codificação binária, uma mutação é uma simples inversão do bit constituinte do gene escolhido. Por exemplo, no cromossoma seguinte:

0 1 1 0 1 1 0 1 1

Aplicando a mutação ao gene destacado, este passará a “0” e o cromossoma transformar-se-á em:

0 1 1 0 0 1 0 1 1

Quando a codificação é inteira e os números não se podem repetir, como no problema do Caixeiro Viajante, a mutação passará a ser aplicada como uma troca entre dois genes. São escolhidos dois genes, segundo a mesma lógica probabilística anterior, e os seus valores são trocados, por exemplo:

2 3 7 9 4 1 5 6 8

Passará a:

2 3 7 5 4 1 9 6 8

Outro tipo de codificações exigirá outras adaptações da técnica de mutação. Por exemplo, numa codificação feita por valores reais, a adaptação mais comum é adicionar ou subtrair um pequeno valor ao gene escolhido:

1.567 3.613 [2.767](#) 6.123 1.946 [1.892](#) 5.001 3.912 4.376

Subtraindo e adicionando, respectivamente, a quantidade 0.010, obter-se-á:

1.567 3.613 [2.757](#) 6.123 1.946 [1.902](#) 5.001 3.912 4.376

5.7.1 Exemplo de Aplicação

Em [8] são apresentados dois tipos de mutação, um para cada tipo de codificação, com números reais aleatórios e com números naturais. Para os números reais aleatórios, as autoras aplicam uma mutação em que escolhem um gene do cromossoma seleccionado e alteram-no ligeiramente, subtraindo ou adicionando uma pequena quantidade. Para a codificação com números naturais, optam pela técnica de escolha de dois genes do cromossoma, com probabilidades adaptadas, e trocam os seus valores.

5.8 Observações

5.8.1 Interpolação

Os operadores base destes algoritmos, mutação e *crossover*, apoiam-se mutuamente. A mutação inclui diversidade na população, enquanto o *crossover* combina as partes boas dos cromossomas existentes. Se a mutação for eliminada, a população evoluirá rapidamente, com certas combinações de algumas partes dos cromossomas (“schemata”), possivelmente importantes, a faltar. No caso extremo, quando um bit numa determinada posição já tiver convergido, o *crossover* não será mais capaz de introduzir diversidade, restringindo-se as combinações sem alterar esse bit, ou seja, restringindo-se à população com um determinado “schemata”². Se o *crossover* for eliminado, a mutação fará uma pesquisa na vizinhança de um certo número de pontos,

²“schemata” é uma combinação de valores fixos em certos genes do cromossoma, que permite formar um conjunto restrito de cromossomas. Esse conjunto é formado por todos os cromossomas compostos por todas as combinações possíveis de valores, sem variar os genes que compõem esse “schemata”. Exemplo de “schemata”: 1***** (cromossomas binários que começam com “1”)

perdendo a capacidade de combinar o melhor desses pontos. Isto conduzirá a uma diminuição da performance do algoritmo.

Assim, uma mistura adequada destes dois operadores conduzirá o algoritmo a melhores resultados. A percentagem adequada de utilização dos dois operadores pode alterar-se no decorrer da pesquisa. Para conseguir fazer com que o *Algoritmo Genético* acompanhe essa alteração na percentagem óptima de utilização de cada operador que se está a utilizar, pode implementar-se esquemas de interpolação dos valores dos pesos dos operadores. Esses esquemas não são aqui tratados, mas uma explicação mais detalhada pode ser encontrada em [20].

Por vezes escolhem-se funções de “Fitness” que podem ter uma melhor performance na diferenciação entre os cromossomas, ao longo da execução do algoritmo, se essas funções forem interpoladas entre valores de normalização definidos. Por exemplo, quando os cromossomas têm valores de “fitness” muito distantes uns dos outros, a função de “Fitness” terá um bom desempenho se for utilizada com uma normalização de 100 (para aproximar mais os “piores” cromossomas dos “melhores”). Mas, quando, numa fase mais adiantada da procura de solução, os cromossomas têm valores mais próximos, interessará utilizar a função com normalização 1, de modo a distinguir melhor os cromossomas mais importantes.

5.8.2 Ajustes e Afinação

Todos os parâmetros dos *Algoritmos Genéticos* têm que ser ajustados à medida que se vai testando o algoritmo, desde as probabilidades de aplicação de *crossover* e mutação na criação de cada cromossoma até às técnicas de selecção de parentes e de eliminação e os seus parâmetros. Não existem boas “receitas” para a criação de um *Algoritmo Genético*, os algoritmos criados têm que ser ajustados para cada caso de aplicação, de forma a se obter a melhor performance possível.

Uma característica crucial dos *Algoritmos Genéticos*, já aqui referida, é a garantia de suficiente diversidade nas populações, de modo a permitir a exploração de várias “boas regiões” de soluções, evitando viciar a pesquisa restringindo-a à vizinhança de um óptimo local. Segundo Pirlot, [21], essa diversidade é garantida com a calibração cuidada dos seguintes parâmetros:

- Dimensão da População;
- Número de cromossomas a substituir de geração para geração;
- Probabilidades de *crossover* e mutação;

- Número de iterações.

Ainda no mesmo artigo é realçada a grande importância que outras escolhas também têm para a obtenção de um bom *Algoritmo Genético*, como:

- Boa codificação das soluções;
- Técnicas de *crossover* e mutação adequadas ao problema.

Capítulo 6

Desenvolvimento do *Algoritmo Genético*

6.1 Introdução

Definido o problema abordado na presente tese de mestrado, construído e testado um modelo linear que resolve uma sua versão simplificada, desenvolveu-se um *Algoritmo Genético* para a sua resolução. Esse algoritmo deve ser capaz de alcançar uma boa solução para o problema num intervalo de tempo aceitável.

Um “*intervalo de tempo aceitável*” será um intervalo que os utilizadores do sistema estejam dispostos a aceitar, tendo em consideração que o algoritmo correrá num computador pessoal. Tendo em conta os tempos de execução de outros algoritmos que resolvem problemas idênticos, descritos no capítulo 3, pode afirmar-se que um intervalo de tempo aceitável para um problema deste tipo será algo inferior a uma hora de execução. No entanto, como se verá mais adiantes, após a introdução da nova restrição para a questão de contactos entre enfermeiros e doentes, a execução do problema tornou-se substancialmente mais lenta, uma vez que essa restrição exigiu a implementação de rotinas que consomem mais tempo de processamento. Além disso, o equilíbrio que se procura neste caso é entre a qualidade da solução e o tempo de execução do algoritmo. Assim, optou-se por aumentar o limite superior de tempo de execução para um intervalo de duas horas. Uma “*boa solução*” será uma solução que terá poucas violações às “soft constraints” impostas ao problema.

O desenvolvimento do *Algoritmo Genético* foi conseguido em duas fases. Na primeira pretendeu-se desenvolver um algoritmo que, respeitando to-

das as “hard constraints”, encontrasse a melhor solução possível num número aceitável de gerações e com um número razoável de indivíduos constituintes de uma geração. Nesta primeira fase, após implementado o algoritmo que resolvesse o problema de *Nurse Rostering* (não considerando as suas “soft constraints”), otimizando os custos associados, afinaram-se os parâmetros do *Algoritmo Genético* para obter a melhor solução possível, num curto intervalo de tempo.

Na segunda fase foram introduzidas as “soft constraints”, uma a uma, e foi estudado de que modo cada uma delas influencia o problema. No final desta fase, foram introduzidas todas as restrições em causa e estudados os melhores valores de pesos a atribuir a cada restrição, por forma a obter a melhor performance possível do *Algoritmo Genético*.

6.2 Implementação do *Algoritmo Genético*

Para projectar e implementar o *Algoritmo Genético* é necessário parametrizar o funcionamento do algoritmo, mais especificamente, a técnica de codificação, a técnica de selecção de pais para reprodução dos cromossomas, a técnica de sobrevivência de cromossomas na criação de uma nova geração, a técnica de obtenção de uma população inicial e as técnicas de reprodução.

Nesta secção é apresentada a primeira fase do desenvolvimento do *Algoritmo Genético*. São descritas e fundamentadas as decisões relativas às técnicas que compõem o *Algoritmo Genético* criado, apresentando-se e comentando-se posteriormente o algoritmo que representa o funcionamento de cada uma dessas partes. Por fim, enuncia-se o conjunto de todas essas técnicas e a sua conjugação, por forma a definir o *Algoritmo Genético* inicial, são descritos e comentados os procedimentos a que se recorreu para conseguir o afinamento dos vários parâmetros do *Algoritmo Genético*.

6.2.1 Codificação das Soluções

Para poder implementar um *Algoritmo Genético* ao problema de *Nurse Rostering* tratado neste trabalho, é necessário, antes de mais, definir uma codificação para as soluções que permita representá-las em cromossomas susceptíveis de serem usados pelo *Algoritmo Genético* na sua pesquisa por uma boa solução.

Relembrando, as soluções são horários para um determinado número de enfermeiros, durante um horizonte temporal de alguns dias. Trata-se, portanto, de uma solução com representação matricial com e linhas e d colunas, sendo e o número de enfermeiros e d o número de dias do horizonte temporal. Cada enfermeiro terá uma atribuição em cada dia: trabalha num dos três turnos possíveis (manhã, tarde ou noite) ou não trabalha. Estas quatro possibilidades de atribuição podem ser descritas por:

- 0 – não trabalha;
- 1 – trabalha no turno da manhã;
- 2 – trabalha no turno da tarde;
- 3 – trabalha no turno da noite.

Um cromossoma pode ser definido como uma representação vectorial da matriz de solução, em que a cada enfermeiro, em cada dia, é atribuído um

dos valores anteriores. Para isso, as linhas da matriz de solução serão colocadas consecutivamente umas em relação às outras numa só linha, pela sua ordem, com $(e \times d)$ colunas. A Figura 6.1 representa esquematicamente um cromossoma, para melhor compreensão do modo de codificação utilizado.

Dia	1	2	3	...	28	1	2	3	...	28	...	1	2	3	...	28
Cromossoma	1	1	2	...	0	0	0	3	...	2	...	2	2	0	...	0
Enfermeiro	1					2					...	e				

Figura 6.1: Codificação das soluções – Cromossoma.

Para facilitar a compreensão das soluções que serão expostas ao longo deste capítulo, os cromossomas serão apresentados como matrizes (de dimensões $e \times d$) e os valores dos genes passarão a ser os seguintes:

- 0 – não trabalha;
- M – trabalha no turno da manhã;
- T – trabalha no turno da tarde;
- N – trabalha no turno da noite.

6.2.2 Criação da População Inicial

A população inicial é criada de forma aleatória. Procura-se criar uma população que contenha indivíduos provenientes de diversos locais do espaço de soluções possíveis. Ou seja, procura-se criar uma população diversificada, que permita explorar o maior número possível de soluções existentes e que evite limitar a pesquisa a uma pequena área do espaço de soluções.

A criação da população inicial é executada em duas partes. Primeiro criam-se os cromossomas com valores aleatórios (entre 0 e 3) que codificam soluções (independentemente de serem admissíveis ou não). Depois, para cada cromossoma criado, verifica-se se satisfaz ou não as necessidades de cobertura, as “hard constraints”. Caso um cromossoma não cubra as necessidades impostas pelo problema, é aplicado um mecanismo de compensação que modifica o cromossoma (procurando alterar o menor número de genes possível) para garantir a satisfação de todas as necessidades. Além deste procedimento, é também verificado e garantido que os enfermeiros apenas estão alocados nos dias em que estão disponíveis, o que torna a disponibilidade dos enfermeiros uma “hard constraint”.

Criação dos Cromossomas

Na primeira parte da criação da população inicial preenche-se os cromossomas que constituem essa população com valores aleatórios (entre 0 e 3). O Algoritmo 6.1 descreve como é feito esse preenchimento.

1. *Inicia um ciclo que percorrerá todos os cromossomas, c , que constituem a população inicial. Em cada iteração do ciclo:*
 - *Inicia um ciclo que percorrerá todos os genes, g , do cromossoma, c . Em cada iteração do ciclo:*
 - (a) *Cria um número aleatório, **aleatório**, entre 0 e 100;*
 - (b) *Se **aleatório** for menor do que 25:*
 - *é atribuído ao gene, g , do cromossoma, c , o valor 0;*
 - (c) *Se não:*
 - i. *Se **aleatório** for menor do que 50:*
 - *é atribuído ao gene, g , do cromossoma, c , o valor 1;*
 - ii. *Se não:*
 - A. *Se **aleatório** for menor do que 75:*
 - *é atribuído ao gene, g , do cromossoma, c , o valor 2;*
 - B. *Se não:*
 - *é atribuído ao gene, g , do cromossoma, c , o valor 3.*

Algoritmo 6.1: Algoritmo da função de criação dos cromossomas da população inicial.

“Hard Constraints”

Existem dois tipos de “hard constraints” a considerar: a disponibilidade dos enfermeiros e as necessidades de cobertura por turno de cada dia. Ambas as restrições dependem de dados que informam acerca da disponibilidade e indisponibilidade dos enfermeiros, e informam do número de enfermeiros, de cada grau, necessários em cada turno diário. Estes dados serão importados para matrizes que se chamarão: Dis_{ed} , para a matriz de disponibilidades dos enfermeiros e, reaproveitando o que foi utilizado na modelização do problema, R_{tdg} , para o número de enfermeiros de grau g necessários no turno t do dia d . A nova matriz Dis_{ed} é uma matriz binária em que:

$$Dis_{ed} = \begin{cases} 0, & \text{se o enfermeiro } e \text{ estiver disponível para trabalhar no dia } d \\ 1, & \text{se não.} \end{cases}$$

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

Antes de se verificar e modificar, se necessário, os cromossomas, procede-se à contagem do número de enfermeiros alocados, por grau, a cada turno diário, segundo o Algoritmo 6.2.

1. *Inicia a matriz de contagem do número de enfermeiros, de cada grau, alocados a cada turno diário, matriz A_{tdg} , com todos os seus valores nulos;*
2. *Inicia um ciclo que percorrerá todos os enfermeiros, e , existentes. Em cada iteração do ciclo:*
 - *Inicia um ciclo que percorrerá todos os dias, d , do horizonte temporal. Em cada iteração do ciclo:*
 - (a) *Inicia um ciclo que percorrerá todos os graus, g . Em cada iteração do ciclo:*
 - (b) *O gene onde está representado o dia d para o enfermeiro e é calculado: $gene = d + n^o \text{ de dias do horizonte temporal} \times e$;*
 - (c) *Se e for de grau g e estiver disponível para trabalhar neste dia d :*
 - *Verifica o valor de $gene$ no cromossoma. Se:*
 - i. *For uma manhã (valor = 1): Adiciona uma unidade a A_{0dg} (Conta mais uma alocação, no dia d , para o grau g , de manhã);*
 - ii. *For uma tarde (valor = 2): Adiciona uma unidade a A_{1dg} (Conta mais uma alocação, no dia d , para o grau g , de tarde);*
 - iii. *For uma noite (valor = 3): Adiciona uma unidade a A_{2dg} (Conta mais uma alocação, no dia d , para o grau g , de noite).*

Algoritmo 6.2: Algoritmo da função de contagem, para um cromossoma, do número de enfermeiros, por grau, alocados a cada turno diário.

Conhecidas as alocações permitidas a cada turno diário, por grau de enfermeiro, prossegue-se para executar os acertos necessários com a intenção de tornar a solução admissível, caso esta ainda não o seja. Esse processo divide-se em duas partes, uma para cada tipo de “hard constraint”. Inicia-se por retirar as alocações que possam estar atribuídas aos enfermeiros durante dias em que estejam indisponíveis, Algoritmo 6.3.

De seguida, passa-se ao acerto das alocações em cada turno diário, para cada grau, de modo a garantir a cobertura de todas as necessidades. Esta verificação e alteração aos genes do cromossoma é feita por processos idênticos, um para cada grau de enfermeiro e cada turno dos dias, e que se encontram

- *Inicia um ciclo que percorrerá todos os enfermeiros, e , existentes. Em cada iteração do ciclo:*
 1. *Inicia um ciclo que percorrerá todos os dias, d , do horizonte temporal. Em cada iteração do ciclo:*
 - *Se e não estiver disponível para trabalhar neste dia d :*
 - (a) *Calcula o gene onde está representado o dia d para o enfermeiro e : $gene = d + n^o \text{ de dias do horizonte temporal} \times e$;*
 - (b) *atribui a esse $gene$ o valor 0 (“não trabalha”).*

Algoritmo 6.3: Algoritmo da função de acertos, para tornar um cromossoma numa solução admissível, relativo ao cumprimento de disponibilidades.

descritos, de forma geral, no Algoritmo 6.4.

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

Inicia um ciclo que percorrerá todos os dias, d , do horizonte temporal. Em cada iteração do ciclo:

- *Para cada grau g , em cada turno t :*
 - *Se $\sum (R_{tdg}$, para graus g e superiores (caso existam)) $>$ $\sum (A_{tdg}$, para graus g e superiores (caso existam)), então:*
 1. *Se $\sum (R_{t'dg}$, para graus g e superiores (caso existam)) $>$ $\sum (A_{t'dg}$, para graus g e superiores (caso existam)), sendo t' um dos outros dois turnos, então:*
 - * *Inicia um ciclo que percorrerá os enfermeiros, e , enquanto as duas condições anteriores se verificarem ou até ter percorrido todos. Em cada iteração do ciclo:*
 - (a) *Se o enfermeiro for de grau g , estiver disponível para trabalhar e estiver alocado a t' :*
 - i. *Troca-se a alocação de t' para t , no gene devido;*
 - ii. $A_{tdg} = A_{tdg} + 1$;
 - iii. $A_{t'dg} = A_{t'dg} - 1$;
 - 2. *Se, ainda assim, $\sum R_{tdg} > \sum A_{tdg}$ e $\sum (R_{t''dg}$, para graus g e superiores) $>$ $\sum (A_{t''dg}$, para graus g e superiores), sendo t'' o último dos três turnos, então:*
 - * *Inicia um ciclo que percorrerá os enfermeiros, e , enquanto as duas condições anteriores se verificarem ou até ter percorrido todos. Em cada iteração do ciclo:*
 - (a) *Se o enfermeiro for de grau g , estiver disponível para trabalhar e estiver alocado a t'' :*
 - i. *Troca-se a alocação de t'' para t , no gene devido;*
 - ii. $A_{tdg} = A_{tdg} + 1$;
 - iii. $A_{t''dg} = A_{t''dg} - 1$;
 - 3. *Se, ainda assim, $\sum R_{tdg} > \sum A_{tdg}$, então:*
 - * *Inicia um ciclo que percorrerá os enfermeiros, e , enquanto as duas condições anteriores se verificarem ou até ter percorrido todos. Em cada iteração do ciclo:*
 - (a) *Se o enfermeiro for de grau g , estiver disponível para trabalhar e não estiver alocado a qualquer turno:*
 - i. *Aloca-se no turno t , no gene devido;*
 - ii. $A_{tdg} = A_{tdg} + 1$;

Algoritmo 6.4: Algoritmo da função de acertos, para tornar um cromossoma numa admissível, relativo ao cumprimento da satisfação das necessidades.

Avaliação dos Cromossomas

Criada a população inicial, é necessário passar à avaliação dos cromossomas, para que se tenha a noção dos melhores e dos piores. O passo é muito importante para o bom funcionamento de todo o *Algoritmo Genético*, já que a boa evolução das populações dependerá da informação que resultar desta avaliação.

Nesta implementação inicial, em que não se consideram as “soft constraints”, a avaliação resume-se ao total dos custos de alocação durante o horizonte temporal. Como no *Algoritmo Genético* é tradicional tratar um problema de maximização e não de minimização, a avaliação de cada cromossoma é feita subtraindo a valor maximizante os custos de alocações. Desta forma, o valor de avaliação de cada cromossoma é calculado pelo Algoritmo 6.5.

- Calcula: *maximizante* = n^o de enfermeiros $\times$ n^o de dias $\times$ custo de alocação de um enfermeiro do grau mais elevado $\times$ *factor multiplicador* (será utilizado quando as “soft constraints” entrarem para o cálculo da Função Avaliadora – nesta primeira implementação vale 1);
- Inicia um ciclo que percorrerá todos os enfermeiros, *e*, existentes. Em cada iteração do ciclo:
 1. Inicia um ciclo que percorrerá todos os dias, *d*, do horizonte temporal. Em cada iteração do ciclo:
 - Inicia um ciclo que percorrerá todos os graus *g*. Em cada iteração:
 - (a) Se *e* é de grau *g* e está alocado a um qualquer turno no dia *d*, então:
 - i. subtrai ao *maximizante* o custo de alocação de um enfermeiro de grau *g*.

Algoritmo 6.5: Algoritmo da função de escolha de um cromossoma para reprodução.

Mais adiante, quando incluídas as “soft constraints”, a Função Avaliadora passa a ter outra componente no seu cálculo. Além do cálculo anterior, subtrai-se o valor total das penalizações atribuídas pelas “soft constraints” (incluindo os pesos que melhoram a performance do *Algoritmo Genético*), o que obriga ao *factor multiplicador* assumir o valor 2.

Quando consultados os valores das avaliações dos cromossomas de uma população inicial, descobre-se que diferem pouco uns dos outros.

Exemplo Considerando um cenário com 20 enfermeiros, 28 dias, as mesmas necessidades que as do “**Cenário 1**” do “Modelo Grande” e custos de alocação com valores iguais aos praticados nos testes aos modelos matemáticos:

- Grau 1 – 60 unidades monetárias (u.m.);
- Grau 2 – 45 u.m.;
- Grau 3 – 30 u.m..

Os valores das avaliações dos cromossomas iniciais (sem considerar as “soft constraints”), em 500 cromossomas criados, variam entre valores pouco superiores a 16000 e valores pouco superiores a 18000. A maior parte dos cromossomas criados tem valores próximos de 17000. Isto significa que os cromossomas diferem, no máximo, em 2000 nos seus valores de avaliação e que, no mínimo, haverá $2000 \div 60 \text{ u.m.} \simeq 33$ alocações de enfermeiros do grau mais elevado. Assim, esta diferença revela-se melhor do que aparenta, analisando somente os valores de avaliação.

A explicação encontrada, para a diferença reduzida entre os valores das Funções Avaliadoras das soluções iniciais, é a diversidade das soluções estar condicionada pela obrigação de cobertura das necessidades e pela disponibilidade dos enfermeiros.

6.2.3 Técnica de Selecção de Cromossomas

A selecção de cromossomas para reprodução é feita recorrendo à técnica de “Roulette Wheel Selection”. Esta técnica foi desenvolvida para problemas de maximização e nela há uma relação directa entre a probabilidade de escolha dos cromossomas e os seus valores de avaliação – quanto maior for o valor da Função Avaliadora de um cromossoma, maior a probabilidade de que este seja escolhido, uma vez que ocupa uma maior fatia da “roleta”.

Como os cromossomas têm valores de avaliação ou de fitness relativamente próximos uns dos outros, como já foi referido, a escolha de cromossomas não será demasiado diferenciada. Isto é, a probabilidade dos cromossomas com maior fitness serem escolhidos não excede em demasia a probabilidade dos que têm menor, aumentando assim a possibilidade de escolher um qualquer cromossoma da população e de serem aproveitadas boas características de cromossomas com menor fitness.

A selecção dos cromossomas é executada como descrito no Algoritmo 6.6. É possível constatar que esse algoritmo é uma aplicação da forma geral da técnica de “Roulette Wheel Selection”.

1. *Calcula valor total de fitness:*

Soma de todos os valores de avaliação dos cromossomas constituintes da população;

2. *Cria um número aleatório entre 0 e o valor total de fitness;*

3. *Percorre os cromossomas da população existente, por ordem. Em cada iteração:*

(a) acumula o valor da sua Função Avaliadora numa variável de fitness local;

(b) Se $\text{fitness local} \geq \text{número aleatório}$, então:

- Escolhe este cromossoma;*
- Termina devolvendo este cromossoma.*

Algoritmo 6.6: Algoritmo da função de cálculo de custos de alocações.

6.2.4 Técnicas de Reprodução

Depois de escolhidos dois cromossomas pais para reprodução, aplicam-se as técnicas de reprodução para criar dois cromossomas filhos que farão parte da população da geração seguinte. Estes são criados recorrendo à possível aplicação de crossover entre os cromossomas pais e de mutação a cada cromossoma filho criado.

No Algoritmo 6.7 encontra-se uma descrição de alto nível da função geradora de novos cromossomas, a função de reprodução.

Como é possível constatar no Algoritmo 6.7, a mutação apenas poderá ser aplicada uma vez a cada cromossoma criado. Sendo a sua probabilidade de aplicação reduzida, como acontece usualmente, a mutação acontecerá poucas vezes. Procura-se assim garantir alguma diversidade nas populações das várias gerações criadas, sem arriscar a eliminação de bons “schemata” existentes, que poderão sobreviver e evoluir pela aplicação do crossover.

São várias as técnicas possíveis para aplicação do crossover. Neste caso, utiliza-se o “One-Point Crossover”, que combina duas partes dos cromossomas pais, separadas por um ponto escolhido aleatoriamente.

Os modos como são aplicados o crossover e a mutação estão detalhados, respectivamente, nos Algoritmos 6.8 e 6.9.

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

1. Cria um número aleatório entre 0 e 100;
2. Se número aleatório criado $\leq$ probabilidade de crossover:
 - (a) aplica crossover entre os dois cromossomas pais, para gerar dois cromossomas filhos;
3. Se não:
 - (a) os dois cromossomas filhos serão cópias exactas dos cromossomas pais;
4. Cria mais um número aleatório entre 0 e 100;
5. Se número aleatório criado $\leq$ probabilidade de mutação:
 - (a) aplica mutação ao primeiro filho criado;
6. Cria outro número aleatório entre 0 e 100;
7. Se número aleatório criado $\leq$ probabilidade de mutação:
 - (a) aplica mutação ao segundo filho criado.

Algoritmo 6.7: Algoritmo da função de reprodução.

6.2.5 Sobrevivência

A melhor solução da pesquisa poderá encontrar-se em qualquer uma das gerações que o *Algoritmo Genético* cria. Para que o cromossoma que contém essa melhor solução não se perca na evolução de uma geração para a seguinte, caso este não seja criado apenas na última geração, surge a necessidade de fazer sobreviver, pelo menos, a melhor solução de cada geração para a geração seguinte. Desta forma, se a melhor solução for encontrada logo na primeira geração criada, ela sobreviverá até à última geração e será apresentada como a solução do problema no final da pesquisa.

A sobrevivência forçada de um reduzido número de cromossomas, que têm as melhores soluções da população de origem, num processo de reprodução, é o elitismo. Nesta implementação dos *Algoritmos Genéticos* é aplicado elitismo aos dois melhores cromossomas de cada população, para garantir, não só que a melhor solução encontrada ao longo de toda a pesquisa seja apresentada no final, tenha ela sido encontrada na primeira ou numa outra qualquer população, mas também que as propriedades desses dois cromossomas possam ser usadas na criação das gerações seguintes.

A aplicação do elitismo é conseguida de um modo simples: depois de cri-

1. Cria um número aleatório entre 0 e *número de genes dos cromossomas*;
Este número indica o ponto de crossover entre os dois cromossomas pais.
2. Gera os dois cromossomas filhos:
 - (a) Filho 1:
 - copia os genes do pai 1, até ao ponto de crossover;
 - copia os genes do pai 2, a partir do ponto de crossover e até ao final do cromossoma;
 - (b) Filho 2:
 - copia os genes do pai 2, até ao ponto de crossover;
 - copia os genes do pai 1, a partir do ponto de crossover e até ao final do cromossoma.

Algoritmo 6.8: Algoritmo da aplicação do crossover.

ados todos os cromossomas da nova população, pelo processo de reprodução, os dois últimos elementos da nova população serão cópias exactas dos dois melhores cromossomas da população de origem. São percorridos todos os cromossomas da população de origem e, sempre que se encontre um cromossoma com melhor valor de avaliação que o anterior, guarda-se a sua referência (posição na matriz de cromossomas), como a localização do melhor cromossoma encontrado até ao momento. No final do ciclo devolve-se a identificação do melhor cromossoma. Em seguida, procede-se da mesma forma para encontrar o segundo melhor cromossoma. Desta vez, procede-se com um cuidado adicional: o de não incluir nesta pesquisa o melhor cromossoma.

6.2.6 O Algoritmo Genético Inicial

Nas descrições anteriores de como funcionam as várias partes do *Algoritmo Genético*, assumiu-se que os dados do problema já se encontravam em matrizes acessíveis pelo programa do *Algoritmo Genético*. Para que isso aconteça, é necessário ler os dados, tal como aconteceu na implementação dos modelos matemáticos. A leitura dos dados é processada exactamente da mesma forma que a utilizada no caso da implementação dos modelos matemáticos. Acresce apenas que, nesta fase, passa a haver mais quatro matrizes para ler: a matriz de mínimo de dias livres consecutivos permitidos para cada enfermeiro, $Mnlc_e$, a matriz de disponibilidades, Dis_{ed} , a matriz

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

1. Cria um número aleatório entre 0 e número de genes dos cromossomas;
Este número indica o gene onde será aplicada a mutação.
2. Cria um número aleatório entre 0 e 100;
 - (a) Se o número criado estiver entre 0 e 25: É atribuído o valor 0 ao gene (não trabalha);
 - (b) Se o número criado estiver entre 26 e 50: É atribuído o valor 1 ao gene (trabalha no turno da manhã);
 - (c) Se o número criado estiver entre 51 e 75: É atribuído o valor 2 ao gene (trabalha no turno da tarde);
 - (d) Se o número criado estiver entre 76 e 100: É atribuído o valor 3 ao gene (trabalha no turno da noite).

Algoritmo 6.9: Algoritmo da aplicação da mutação.

de contactos entre enfermeiros, Cte_{ee} , e a matriz de contactos entre enfermeiros e doentes Ctd_{ep} , em que p é o índice que representa os doentes da enfermaria.

Isto exige também a adição dessas mesmas quatro matrizes de dados, para preenchimento pelo utilizador, no ficheiro em MS Excel (apresentadas nas figuras do Anexo D.1).

Depois de lidos os dados, e guardados nas matrizes apropriadas, é iniciada a pesquisa pela melhor solução, correndo o *Algoritmo Genético*. Todo o funcionamento do *Algoritmo Genético* Inicial está apresentado no Algoritmo 6.10.

Parâmetros

Tendo o *Algoritmo Genético* construído e pronto a funcionar, tal como descrito no Algoritmo 6.10, é chegada a altura de o testar e afinar os seus parâmetros para otimizar o funcionamento.

Os parâmetros que influenciam o *Algoritmo Genético* e que devem ser calibrados para se obter a sua melhor performance são:

- A dimensão da população;
- O número de gerações a criar – critério de paragem da pesquisa;
- A probabilidade de aplicação do crossover no processo de reprodução;

- E a probabilidade de aplicação da mutação no processo de reprodução.

A dimensão da população e o critério de paragem (número de gerações) são parâmetros que exigem maior ponderação na decisão sobre os seus valores. Isto porque, para além de estes dois parâmetros influenciarem o bom desempenho do *Algoritmo Genético*, também influenciam totalmente o tempo de execução do algoritmo, já que quantos mais indivíduos tiver uma população ou mais gerações se pretender gerar ao longo do *Algoritmo Genético*, mais cromossomas se terão que criar, acertar e avaliar. Assim sendo, para estes dois parâmetros será imperativo encontrar valores que, permitindo alcançar uma boa solução para o problema, não condicionem em demasia o tempo de execução do *Algoritmo Genético*, tendo especialmente em consideração que esse tempo aumentará com a inserção das “soft constraints”.

Para as probabilidades de crossover e mutação interessa encontrar os valores que melhorem a performance da pesquisa. A probabilidade de crossover costuma assumir valores elevados, entre 50% e 100%, enquanto a probabilidade de mutação assume valores reduzidos, usualmente entre 0% e 5%.

A calibração destes parâmetros foi efectuada em três fases:

1. Definiram-se parâmetros iniciais, para ter um ponto de partida. Testaram-se diferentes valores para a dimensão da população e para o critério de paragem, até se terem detectado os que permitiram alcançar a melhor solução, num intervalo de tempo de execução curto. Nestes testes foram atribuídos valores considerados razoáveis às probabilidades de crossover e mutação, e todos eles tiveram a mesma população inicial;
2. Com os melhores valores de dimensão da população e de critério de paragem definidos, optimizaram-se os valores das probabilidades de crossover e mutação de forma a atingir o melhor desempenho possível do *Algoritmo Genético*. Mais uma vez, nestes testes, a população inicial foi sempre a mesma;
3. Por fim, usando diferentes populações iniciais para confirmar os valores escolhidos para as probabilidades de crossover e mutação, para cada população inicial, variaram-se as duas probabilidades, tal como na fase anterior.

Para os testes efectuados nesta fase do trabalho, utilizou-se uma instância alargada relativamente à usada ao longo do trabalho. A única diferença relativamente à instância elaborada com o “**Cenário 1**”, apresentada no exemplo da secção 6.2.2, reside no facto de haver 30 enfermeiros em vez de 20. Esta

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

modificação justifica-se pelo facto de apenas se pretender comparar o comportamento do *Algoritmo Genético* com as alterações impostas pelas variações dos seus parâmetros. O alargamento da instância de teste permite uma maior diversidade nos cromossomas iniciais, o que torna mais clara a análise dos resultados finais (No Anexo D.3 encontra-se uma listagem dos valores das funções avaliadoras dos cromossomas que constituem uma população inicial para o cenário alargado).

Dimensão da População e Critério de Paragem Os parâmetros iniciais escolhidos foram de 90% de probabilidade de crossover e 5% de probabilidade de mutação. A dimensão da população foi testada com 500 e 1000 indivíduos e o critério de paragem foi variado entre 1000 e 50000 gerações. Estes dois parâmetros foram testados em simultâneo, usando várias combinações possíveis entre si.

Critério de Paragem	Dimensão da População	
	500	1000
1000	33330	34275
3000	36045	36660
5000	37410	37950
10000	38460	38460
50000	38490	38865

Tabela 6.1: Testes a vários valores de dimensão da população e de critério de paragem do *Algoritmo Genético*.

A Tabela 6.1 apresenta o valor da Função Avaliadora de cada solução obtida para cada uma das combinações possíveis. Desta tabela é possível concluir que os melhores valores são:

- para a dimensão do problema: 500 indivíduos;
- para o critério de paragem: 10000 gerações.

Estes foram os valores escolhidos, porque as soluções melhoram significativamente até ao critério de paragem igual a 10000 gerações, tendo uma alteração pouco significativa das 10000 para as 50000 gerações quando comparada com o aumento da dimensão do programa – que tem que criar um número muito mais elevado de cromossomas (o que influencia directamente o tempo de execução do *Algoritmo Genético*). Para o valor do critério de paragem escolhido, a solução obtida para populações com 500 indivíduos é igual à obtida para 1000 indivíduos, o que justifica a escolha do primeiro valor, já que, mais uma vez, permite reduzir o número de cromossomas criados ao longo da pesquisa, diminuindo o tempo de execução do *Algoritmo Genético*.

Foram testados valores mais reduzidos para a dimensão da população, mas foram excluídos desta análise por se pretender garantir diversidade nas populações durante a pesquisa. Para que isso aconteça interessa ter um número relativamente elevado de cromossomas em cada população. Esta opção confirmou-se ser a mais ajustada depois de testar o *Algoritmo Genético* a funcionar com todas as restrições (“hard” e “soft”), já que as diferenças de soluções obtidas foram significativamente melhores e mais aceitáveis para populações com 500 indivíduos do que para populações com populações menores (ver testes apresentados na secção 6.4).

Probabilidade de Crossover Com os valores da dimensão da população e do critério de paragem calibrados, continuando a assumir como boa a percentagem de 5 para a probabilidade de mutação, testaram-se vários valores para a probabilidade de crossover, entre 70% e 100%, tendo sido obtidos os resultados apresentados na Tabela 6.2.

Probabilidade de Crossover (%)	Função Avaliadora da Solução
70	38490
75	38595
80	38790
85	38490
90	38460
95	38670
100	38505

Tabela 6.2: Testes a vários valores da probabilidade de crossover do *Algoritmo Genético*.

Observando os valores obtidos para estes testes depressa se conclui que o melhor valor para a probabilidade de crossover é 80%.

Probabilidade de Mutação Por fim, fizeram-se variar os valores para a probabilidade de mutação, de 0,1% a 15%. Os resultados destes testes conduziram à escolha de 6% para a probabilidade de mutação. O valor é alto para este parâmetro, comparativamente àquele que costuma assumir, mas os testes com outras populações iniciais ajudarão a certificar se ele será realmente o melhor. Os resultados dos testes (valores da Função Avaliadora das soluções obtidas) estão apresentados na Tabela 6.3.

Testes com mais Populações Iniciais Os últimos testes realizados nesta fase do trabalho, sem considerar as “soft constraints”, serviram para confirmar os valores escolhidos para a probabilidade de crossover e de mutação. Foram efectuados para os mesmos valores testados anteriormente, mas partindo de populações iniciais diferentes.

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

Probabilidade de Mutação (%)	Função Avaliadora da Solução
0,1	37560
0,5	38070
0,9	38445
1	38595
2	38805
3	38625
4	38595
5	38790
6	38850
7	38805
8	38655
9	38550
10	38475
11	38580
15	38565

Tabela 6.3: Testes a vários valores da probabilidade de mutação do *Algoritmo Genético*.

A apresentação dos resultados obtidos contempla a comparação entre as soluções encontradas para os valores de probabilidade seleccionados, por terem tido os melhores, e os restantes valores testados.

Ainda se realizou mais um teste, com uma quarta população inicial diferente das restantes, para comprovar que o valor de probabilidade de mutação de 6% é melhor do que o de 4%. Obtida assim, esta confirmação, não restam dúvidas relativamente à escolha do valor de 6% para esta probabilidade.

População	Valor escolhido para a Probabilidade de Crossover	Probabilidades que conduziram a soluções:		
		Piores (%)	Iguais (%)	Melhores (%)
1	80%	70; 75;	—	—
		85; 90;		
		95; 100.		
2	80%	70; 75;	—	—
		85; 90;		
		95; 100.		
3	80%	70; 75;	—	—
		85; 90;		
		95; 100.		

Tabela 6.4: Testes a vários valores da probabilidade de crossover do *Algoritmo Genético*, com diferentes populações iniciais.

Elitismo

Apesar de não ser um parâmetro, o funcionamento do elitismo também foi testado com a intenção de afinar o seu funcionamento, depois de calibrados todos os parâmetros do *Algoritmo Genético*. Interessou estudar se seria

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

População	Valor escolhido para a Probabilidade de Mutação	Probabilidades que conduziram a soluções:		
		Piores (%)	Iguais (%)	Melhores (%)
1	6%	0,1; 0,5; 0,9; 1; 2; 3; 4; 5; 7; 8; 9; 10; 11; 15.	—	—
2	6%	0,1; 0,5; 0,9; 2; 3; 5; 7; 8; 9; 11; 15.	1; 10	4
3	6%	0,1; 0,5; 0,9; 1; 2; 3; 4; 5; 7; 8; 9; 10; 11; 15.	—	—

Tabela 6.5: Testes a vários valores da probabilidade de mutação do *Algoritmo Genético*, com diferentes populações iniciais.

mais proveitoso, em termos de melhorar a performance do *Algoritmo Genético*, fazer sobreviver os dois melhores cromossomas de uma geração para a seguinte nas duas posições finais dos cromossomas da nova população, ou em posições intermédias. Isto tem importância porque, no processo de selecção de cromossomas para a criação de cromossomas filhos para uma nova população, a posição dos cromossomas determina quando serão escolhidos, já que isso depende do fitness acumulado de cada um deles (o qual varia quando se altera a posição).

Como se pode constatar na Tabela 6.6, a posição dos cromossomas sobreviventes influencia o comportamento da pesquisa e a solução final obtida. Mas o seu posicionamento num local intermédio da população não influenciou positivamente a pesquisa, antes pelo contrário, indicando que a suposição anterior não estava correcta. Deste modo, decidiu manter-se a abordagem anterior, com os cromossomas sobreviventes a serem guardados em posições finais.

População	Elitismo para Posições Intermédias	Elitismo para Posições Finais
1	38655	38850
2	38520	38715
3	38595	38820
4	38505	38655

Tabela 6.6: Valores das Funções Avaliadoras para testes da melhor posição para guardar os sobreviventes de uma população para a seguinte.

6.2. IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

1. Lê os dados do problema e guarda-os em matrizes, vectores e variáveis;
2. Calcula o tamanho de cada cromossoma (n° enfermeiros $\times$ n° dias) e aloca espaço em memória para `numero_cromossomas` e `numero_cromossomas` filhos, cada um com o tamanho calculado. Estes últimos serão usados na criação de novas gerações;
3. Cria os cromossomas da população inicial;
4. Inicia um ciclo que percorrerá todos os cromossomas da população actual. Em cada iteração, ou seja, para cada cromossoma:
 - (a) Faz a contagem dos enfermeiros alocados, segundo o Algoritmo 6.2;
 - (b) Faz os acertos relativos à disponibilidade dos enfermeiros, descrito no Algoritmo 6.3, e os acertos relativos à cobertura das necessidades, Algoritmo 6.4.
5. Inicia outro ciclo que percorrerá todos os cromossomas da população actual. Em cada iteração, ou seja, para cada cromossoma:
 - (a) Calcula os custos das alocações, Algoritmo 6.5, e avalia o cromossoma, como descrito em 6.2.2.
6. Inicia um ciclo que terminará quando se atingir tantas gerações criadas como o indicado pelo `critério_de_paragem`. Em cada iteração, correspondente a cada geração criada:
 - (a) Enquanto não criar (`numero_cromossomas-2`):
 - i. Selecciona dois cromossomas pais, recorrendo ao Algoritmo 6.6;
 - ii. Aplica o Algoritmo 6.7 a esses dois cromossomas pais para criar dois cromossomas filhos;
 - (b) Selecciona os dois melhores cromossomas da população actual, respeitando o processo descrito em 6.2.5, e memoriza-os nos dois cromossomas filhos que restam;
 - (c) Substitui os cromossomas da população actual pelos cromossomas filhos criados e sobreviventes;
 - (d) Repete pontos 4 e 5;
7. Devolve a melhor solução encontrada, seleccionada por um processo idêntico ao elitismo, em que apenas se selecciona um cromossoma, o melhor.

Algoritmo 6.10: Algoritmo inicial, apenas considerando as “hard constraints”.

6.3 “Soft Constraints”

Depois de testado e afinado o funcionamento base do *Algoritmo Genético* para este problema, resta acrescentar as diferentes “soft constraints” que particularizam o problema relativamente aos demais problemas de escalonamento.

As “soft constraints” foram adicionadas uma a uma ao problema, de forma a compreender a sua influência no funcionamento do *Algoritmo Genético*. Enquanto se testou a influência de cada uma delas, verificou-se também de que forma, ao acrescentar um peso ao seu valor de penalização, que o aumentasse e tornasse a restrição em causa mais influente no processo de pesquisa da melhor solução, se influenciaria o comportamento do *Algoritmo Genético*, considerando os custos de alocações e as penalizações associadas à restrição.

Depois de testadas todas as restrições individualmente, testou-se o *Algoritmo Genético* com todas as “soft constraints” em funcionamento. Nesta última fase, os pesos das restrições foram calibrados para os valores que melhor desempenho conferiram ao *Algoritmo Genético* em pleno funcionamento, partindo dos valores anteriormente encontrados para cada restrição individualmente e calibrando-os até se encontrar o melhor desempenho. Pretendia-se também normalizar os valores das restrições, como ponto de partida para esta última pesquisa, mas, como será explicado, tal normalização é impossível de conseguir.

Os testes efectuados nesta última parte do trabalho foram realizados com um critério de paragem de 20000 gerações, em vez de apenas 10000, como escolhido na primeira fase da implementação do *Algoritmo Genético*. A mudança deste parâmetro foi testada, durante o trabalho, com a restrição relativa a fins-de-semana (a primeira a ser testada), quando se obteve resultados com elevadas penalizações finais e, experimentando aumentar o número de gerações para o dobro (20000), se conseguiu melhorá-las. Este novo valor de critério de paragem tem especial significado no funcionamento do *Algoritmo Genético* com todas as “soft constraints”, uma vez que permite uma pesquisa de maior pormenor. O facto é importante, já que o melhoramento conjunto de todas as restrições se torna significativamente mais complexo do que a satisfação de cada uma isoladamente.

Depois de constatada esta melhoria para a primeira restrição, nos testes efectuados às restantes restrições o critério de paragem passou a ser utilizado com 20000 gerações.

6.3.1 Fins-de-Semana

Todas as restrições relacionadas com fins-de-semana foram incluídas numa “soft constraint” única. Pretende-se, relativamente a fins-de-semana, que cada enfermeiro, de preferência:

1. quando for alocado a um dia do fim-de-semana (sábado ou domingo), seja também alocado ao outro dia do fim-de-semana e à sexta-feira, ficando assim com o fim-de-semana completo (sexta-feira, sábado e domingo);
2. estando alocado a um fim-de-semana, tenha sempre o mesmo tipo de turnos alocados durante os dias de fim-de-semana;
3. não esteja alocado a mais do que “FM” fins-de-semana durante o horizonte temporal;
4. tenha o mínimo de fins-de-semana alocados possível, ao longo do horizonte temporal.

Na Figura 6.2 está esquematicamente apresentada a função que verifica os primeiro e segundo pontos anteriores.

No Algoritmo 6.11 está descrita a função que garante o terceiro ponto, e na Figura 6.3 esquematizada a função elaborada para o quarto ponto.

1. *Percorre, com um ciclo, todos os enfermeiros e todos os dias. Em cada iteração:*
 - *Se o dia actual for um sábado:*
 - (a) *Se este fim-de-semana estiver alocado para este enfermeiro, mesmo que seja um só dos dias (sábado ou domingo):*
 - i. *Conta mais um fim-de-semana alocado;*
2. *Para cada enfermeiro, se o número total de fins-de-semana alocados for maior do que “FM”:*
 - (a) *Acrescenta a diferença entre o número total de fins-de-semana alocados e “FM” à penalização deste tipo ($fstotal$).*

Algoritmo 6.11: Algoritmo da função de penalização de totais máximos de fins-de-semana.

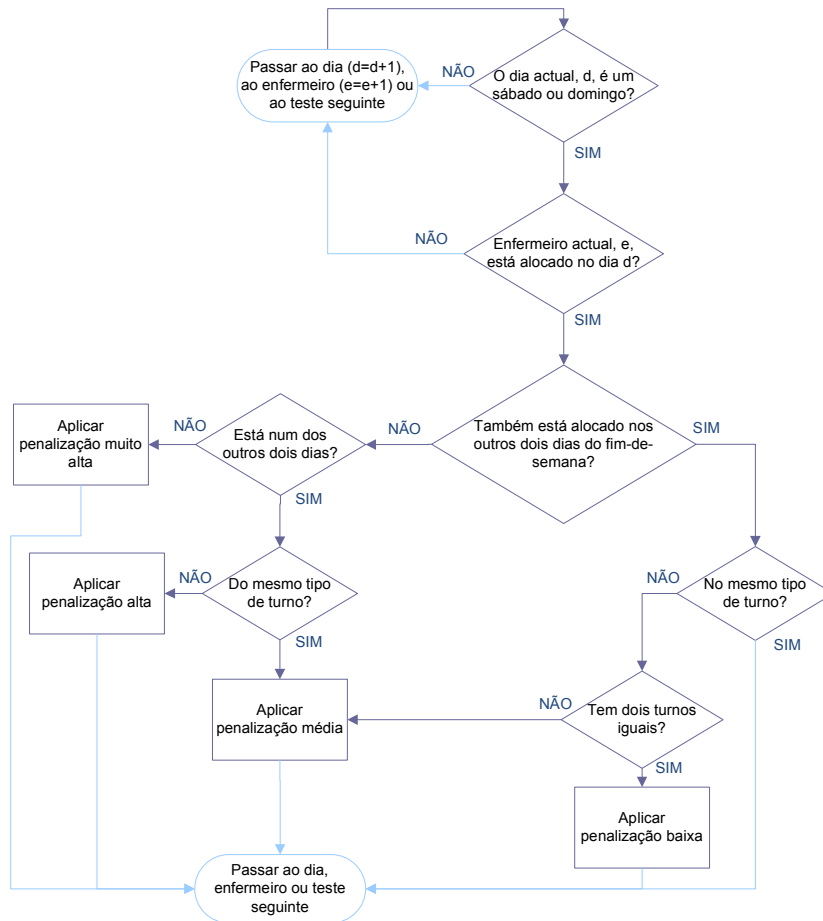


Figura 6.2: Esquema de funcionamento da função de verificação de fins-de-semana completos.

Restrição de fins-de-semana no *Algoritmo Genético*

Resultam três valores de penalizações relacionadas com fins-de-semana, mas pretende-se traduzir a qualidade dos cromossomas relativamente a fins-de-semana apenas num único valor. Para isso, não é suficiente somar as três componentes, já que têm valores de partida muito diferentes. É necessário encontrar uma normalização que calibre os três pesos na penalização atribuída ao cromossoma, de modo a que o *Algoritmo Genético*, ao longo do seu desenvolvimento, possa identificar claramente os melhoramentos inerentes a alterações nos cromossomas que os aproximem do cumprimento de cada uma das parcelas desta penalização.

As três parcelas desta penalização têm valores médios na população ini-

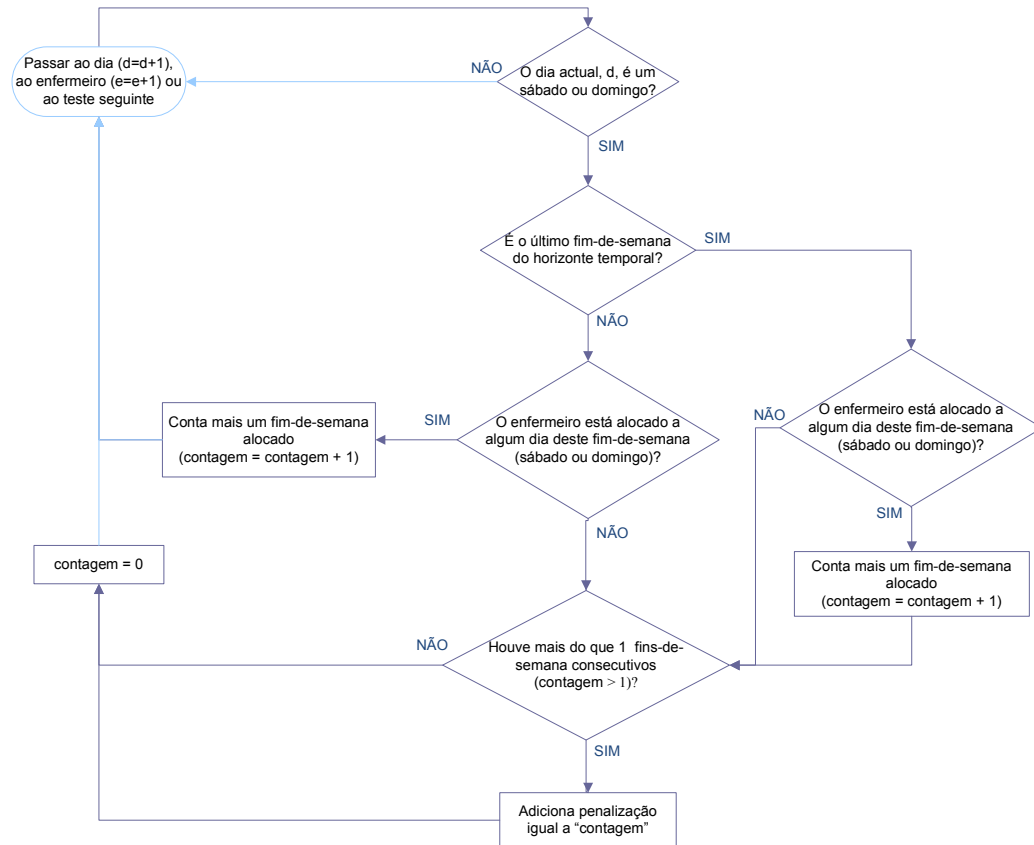


Figura 6.3: Esquema de funcionamento da função de verificação de fins-de-semana consecutivos.

cial diferentes e evoluções diversas ao longo da pesquisa pela melhor solução. A sua segunda característica impossibilita o processo de normalização por aplicação de um factor a cada parcela, caso se queira que as parcelas se mantenham normalizadas ao longo de toda a pesquisa. Mas o que aqui interessa é ter um ponto de partida em que todas as parcelas assumam a mesma importância no valor da penalização, de modo a que o *Algoritmo Genético* parta de valores que atribuam o mesmo peso às três componentes. Não se espera que tenham a mesma evolução, mas, calibrando o peso atribuído à penalização total de fins-de-semana na Função Avaliadora, espera-se que o *Algoritmo Genético* evolua para o melhor valor das três parcelas.

Para a instância elaborada com o “**Cenário 1**”, com um máximo de 3 fins-de-semana ao longo do horizonte temporal (“FM”), os cromossomas das populações iniciais tinham os seguintes valores parcelares de penalização

(um exemplo de valores parcelares de penalização em todos os cromossomas constituintes de uma população inicial estão apresentados no Anexo ??):

- Para a verificação de fins-de-semana completos (parcela identificada por **fscompletos**) e do mesmo tipo de turno, a penalização aplicada rondava o valor de 150, identificou-se o valor máximo de 182 e o mínimo de 124;
- Para a verificação do número total de fins-de-semana alocados ao longo do horizonte temporal (parcela identificada por **fstotais**), a penalização rondou o valor de 14, foram identificados o valor máximo de 17 e o mínimo de 7;
- Para a verificação de fins-de-semana consecutivos (parcela identificada por **fsconsecutivos**), as penalizações variam entre 75 e 55, e rondam o valor 68.

Para que estas três parcelas influenciassem de forma idêntica a penalização total de fins-de-semana, decidiu-se calcular a penalização total através da seguinte fórmula:

$$fscompletos + 10 \times fstotal + 2 \times fsconsecutivos$$

A fase seguinte do estudo desta restrição visou verificar a sua influência no *Algoritmo Genético*, ou seja, nas Funções Avaliadoras dos cromossomas, e estudar qual o melhor peso a atribuir a esta penalização, para alcançar o valor mais baixo. O estudo foi realizado com uma análise aos valores médios de penalizações da última geração criada em cada pesquisa e aos valores de custos de alocação e de penalização da solução encontrada, para cada peso atribuído à restrição. Os pesos atribuídos foram valores entre 1 e aquele que não melhorou de forma significativa a solução encontrada.

Nesta fase os testes foram efectuados quer para a instância alargada, quer para a instância elaborada com o “**Cenário 1**”. Destes testes, os que foram realizados com a instância alargada revelaram exprimir melhor as diferenças entre os vários pesos atribuídos. Acontece assim, porque, permitindo a instância alargada uma maior liberdade, as diferenças de evolução do *Algoritmo Genético* para os vários pesos tornam-se mais nítidas, propriedade que é objecto de estudo destes testes. Por esta razão, para todas as restantes restrições, quando testadas isoladamente, utilizou-se a instância alargada.

Os resultados obtidos para a instância com o “**Cenário 1**” estão apresentados na Tabela 6.7. Os resultados obtidos para a instância alargada estão na Tabela 6.8.

Dos resultados pode concluir-se que o melhor valor de peso para esta restrição, sendo a única “soft constraint” em funcionamento, é 30. Foi escolhido por ser, dos valores de penalização mais reduzidos nas soluções obtidas, o que não obriga a grandes aumentos nos custos de alocações, comparativamente aos resultados base (com peso de valor 1) e aos outros resultados relativos aos restantes pesos. Esta constatação é mais nítida nos testes realizados com a instância alargada, para a qual, apesar de as penalizações finais se reduzirem com peso de valor 40, os custos de alocações aumentam. Para ambos os valores as penalizações são reduzidas, tendo em conta que se estão a avaliar quatro factores divididos em três tipos de penalizações parciais, que essas três penalizações parciais têm pesos diferentes no cálculo da penalização total e que as pesquisas começam em valores perto de 10 vezes maiores de penalização.

Estes testes servem apenas para ter uma ideia do valor que melhor se adequará ao peso atribuído a esta restrição, quando funcionar em conjunto com as restantes “soft constraints”. Depois da análise anterior, constata-se que o peso a atribuir a esta restrição no *Algoritmo Genético* final, que deverá permitir o melhor desempenho possível, quer com o funcionamento em pleno (considerando todas as “soft constraints”), quer funcionando apenas com esta “soft constraint”, será de valor 30 ou 40. Garantem-se assim bons resultados para esta restrição, nomeadamente quando funcionar sem as restantes “soft constraints”, quando permite reduzidos valores de penalização, como aqui demonstrado.

Permitem uma noção da boa evolução que este peso proporciona, os valores médios de penalizações iniciais com o valor médio de 421, para a instância com o “**Cenário 1**”, valores médios de 635 para as populações 1 e 2, e 638 para a população 3, com a instância alargada.

O maximizante utilizado para a Função Avaliadora, na instância com o “**Cenário 1**”, foi multiplicado por 2, tomando o valor:

$$20 \text{ enfermeiros} \times 28 \text{ dias} \times 60 \times 2 = 67200$$

No cálculo anterior, o valor 60 diz respeito ao custo de alocação de um enfermeiro de grau mais elevado. O maximizante é duplicado para que não seja excedido com a adição da penalização à Função Avaliadora (FA). Sem duplicação, o maximizante teria o valor 33600. O maximizante para a instância alargada teve o valor:

$$30 \text{ enfermeiros} \times 28 \text{ dias} \times 60 = 50400$$

Peso	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1		
1	203	55590 : 170
20	115	55485 : 97
30	103	55395 : 94
40	88	55410 : 72
50	111	55395 : 103
60	80	55185 : 71

Tabela 6.7: Valores obtidos para os testes de influência da penalização de fins-de-semana no *Algoritmo Genético*, com vários pesos atribuídos, para a instância com o “**Cenário 1**” (critério de paragem: 20000 gerações).

Na Tabela 6.8 destaca-se o aperfeiçoamento alcançado com a mudança de critério de paragem, de 10000 gerações para 20000, nomeadamente no primeiro peso (valor 1) da primeira população, em que a penalização relativa a fins-de-semana decresceu, na solução obtida, quase um valor de 50. Outro facto relevante verifica-se para o peso 30 da mesma população, em que, para além de o valor da penalização ter diminuído, o valor dos custos reduziu-se significativamente, aproximando-se do seu valor para o peso de valor 1.

Confrontando as duas tabelas apresentadas, pode também verificar-se que se torna mais fácil analisar os resultados e a influência dos pesos atribuídos à restrição com a instância alargada, onde as diferenças entre os resultados para os vários pesos são mais significativas.

6.3.2 Mínimos de Folgas entre Alocações Consecutivas

Com a codificação utilizada no *Algoritmo Genético* implementado, não existe qualquer possibilidade de alocar um enfermeiro a mais do que um turno num dia, o que evita períodos de trabalho muito longos, com a possibilidade de ocorrerem sem um posterior período de descanso adequado. Mas, ainda assim, podem ocorrer determinadas sequências em dias consecutivos indesejáveis a muitos enfermeiros e instituições, são elas as apresentadas nos pontos seguintes:

- N/M $\Rightarrow$ alocação a um turno nocturno e, no dia seguinte, a um turno de manhã;
- T/M $\Rightarrow$ alocação a um turno de tarde e, no dia seguinte, a um turno de manhã;
- N/T $\Rightarrow$ alocação a uma noite e a uma tarde do dia seguinte;

6.3. “SOFT CONSTRAINTS”

Peso	Critério Paragem	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1			
1	10000	257	38820 : 244
20	10000	134	38265 : 107
30	10000	87	38010 : 76
1	20000	203	38820 : 197
20	20000	100	38310 : 83
30	20000	80	38340 : 67
40	20000	72	37305 : 60
POPULAÇÃO 2			
1	10000	225	38550 : 225
20	10000	113	38265 : 100
30	10000	107	37980 : 92
1	20000	234	38655 : 216
20	20000	99	38640 : 89
30	20000	93	38175 : 81
40	20000	74	37995 : 72
POPULAÇÃO 3			
1	10000	218	38730 : 215
1	20000	206	38730 : 206
20	20000	104	38730 : 100
30	20000	86	38655 : 84
40	20000	70	38535 : 69

Tabela 6.8: Valores obtidos para os testes de influência da penalização de fins-de-semana no *Algoritmo Genético*, com vários pesos atribuídos e de critérios de paragem, para a instância alargada.

- N/T/M $\Rightarrow$ alocações iguais às do ponto anterior, seguidas de uma alocação a uma manhã no terceiro dia consecutivo.

No primeiro caso o enfermeiro trabalhará dois turnos consecutivamente. Mas, após esses dois turnos, terá um repouso de, pelo menos, outros dois turnos, e antes da sequência de turnos, também teve pelo menos dois turnos de repouso, o que faz desta situação, que aparenta ser a pior, a menos má de todas as possíveis e apresentadas. Nos restantes casos, o enfermeiro será alocado a uma série de turnos (dois ou três) apenas separados por um turno de descanso entre as alocações. Estes casos são considerados mais fatigantes, especialmente o último, em que a situação fatigante é extendida ao longo de três dias consecutivos.

Na Figura 6.4, está esquematizada a função que evita este tipo de sequências, penalizando a Função Avaliadora sempre que alguma das sequências se verifique. As penalizações aplicadas têm diferentes valores, dependendo do tipo de sequência que se concretize. As sequências consideradas piores sofrem maiores penalizações do que as menos más.

Restrição de mínimos de folgas no *Algoritmo Genético*

O funcionamento desta restrição no *Algoritmo Genético* foi estudado da mesma forma que a utilizada para a restrição relativa a fins-de-semana. Compararam-se os resultados obtidos com vários pesos atribuídos a esta restrição e escolheu-se o que conduziu a menores valores de penalização.

O melhor peso encontrado para esta restrição foi o de valor 10, aproximando o resultado obtido da anulação desta restrição. Mesmo tendo conseguido essa anulação com o peso de valor 20, a solução, em termos gerais (custos de alocação e valor de penalização), é melhor para um peso de valor 10. Os valores de penalização médios das populações iniciais são de 422 para as três populações diferentes, o que demonstra que esta restrição é muito melhorada, mesmo com um peso unitário.

Os resultados estão apresentados na Tabela 6.9. Os testes foram efectuados com a instância alargada e os parâmetros do *Algoritmo Genético* definidos na fase anterior da sua implementação, com excepção do critério de paragem, que teve o valor 20000, redimensionado para esse valor nos testes da restrição anterior.

Peso	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1		
1	32	38565 : 17
10	16	38505 : 3
20	14	38475 : 2
POPULAÇÃO 2		
1	33	38580 : 23
10	26	38670 : 9
20	11	38310 : 0
POPULAÇÃO 3		
1	32	38670 : 31
10	15	38820 : 3
20	18	38715 : 7

Tabela 6.9: Valores obtidos para os testes de influência da penalização de mínimos de folgas no *Algoritmo Genético*, com vários pesos atribuídos, para a instância alargada.

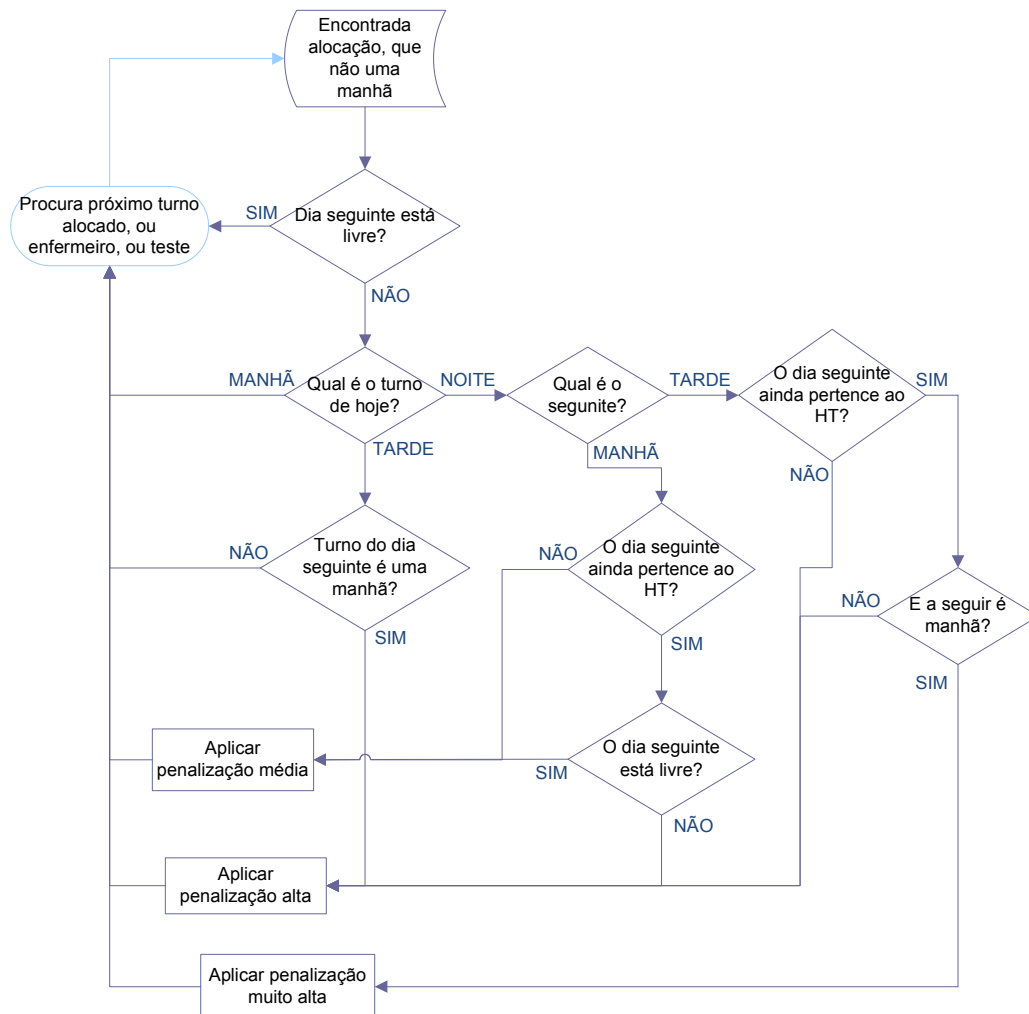


Figura 6.4: Esquema de funcionamento da função que evita as sequências identificadas como fatigantes e indesejadas.

6.3.3 Máximo de Alocações no Horizonte Temporal

Esta “soft constraint” é de simples implementação, sendo apenas necessário verificar, para cada enfermeiro, se o seu total de turnos de trabalho alocados, ao longo do horizonte temporal, não excede o máximo permitido. A penalização será tanto maior, quanto maior for o número que excede o máximo de alocações permitido, tomando o mesmo valor que esse excesso. Este funcionamento está descrito no Algoritmo 6.12.

1. *Percorre, com um ciclo todos os enfermeiros. Em cada iteração:*
 - (a) *Percorre, com um ciclo todos os dias do horizonte temporal. Em cada iteração:*
 - i. *Se o enfermeiro estiver alocado a um qualquer turno de trabalho:*
 - A. *Conta mais uma alocação na variável de contagem para este enfermeiro, Tot_e ;*
2. *Percorre, com um ciclo todos os enfermeiros. Em cada iteração:*
 - (a) *Se $Tot_e > \text{Número máximo permitido (Maxa)}$:*
 - *Aplica penalização igual a $Tot_e - Maxa$.*

Algoritmo 6.12: Algoritmo da função que penaliza excessos do número de alocações durante o horizonte temporal, relativamente ao máximo permitido.

Restrição de número máximo de alocações no *Algoritmo Genético*

Esta é talvez a restrição que mais facilmente será cumprida, uma vez que, normalmente, não é muito restritiva. Depende do número de enfermeiros disponíveis e da quantidade de enfermeiros necessários em cada turno de cada dia. Mas, assumindo que o número de enfermeiros contratados é suficiente para garantir o bom funcionamento da instituição (problema que não diz respeito ao nível de decisão deste tipo de sistemas), este maximizante número de alocações no horizonte temporal servirá unicamente para garantir que não haverá sobrecargas desnecessárias.

Ainda assim, para ter uma melhor percepção da possível influência desta restrição no funcionamento do *Algoritmo Genético*, testou-se-a com um valor excessivamente reduzido, forçando a ocorrência de algumas violações à restrição. O valor utilizado foi de um máximo de 16 alocações ao longo do horizonte temporal. Não se identificaram muitas violações à restrição, mas o

número médio de penalização nas populações iniciais foi de 129, o que significa que, analisando os resultados dos testes, a restrição é significativamente melhorada, mesmo tendo um peso unitário a si associado. Analisando os resultados pode também concluir-se que o peso de valor 20 será mais vantajoso, já que garante, para todos os casos, valores baixos de penalização, mantendo bons valores de custos de alocações. Os resultados podem ser consultado na Tabela 6.10.

Peso	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1		
1	7	38820 : 2
10	11	38715 : 8
20	5	38610 : 1
POPULAÇÃO 2		
1	17	38580 : 13
10	14	38520 : 10
20	9	38535 : 5
POPULAÇÃO 3		
1	3	38505 : 1
10	9	38805 : 4
20	3	38745 : 1

Tabela 6.10: Valores obtidos para os testes de influência da penalização de máximo de alocações permitidas, ao longo do horizonte temporal no *Algoritmo Genético*, com vários pesos atribuídos, para a instância alargada.

6.3.4 Máximo e Mínimo de Alocações Consecutivas

Por forma a ter algum controlo sobre a consecutividade dos turnos alocados, podem definir-se máximos e mínimos de alocações consecutivas permitidas para cada enfermeiro. Nos testes efectuados a esta restrição todos os enfermeiros têm o mesmo máximo e mínimo de alocações consecutivas, mas a forma de implementação do programa permite diferenciação, caso seja desejada.

Esta “soft constraint” é praticada no *Algoritmo Genético* impondo penalizações tão grandes quantos os dias em excesso ou defeito para garantir o cumprimento do máximo e mínimo permitidos. A lógica de funcionamento está representada na Figura 6.5.

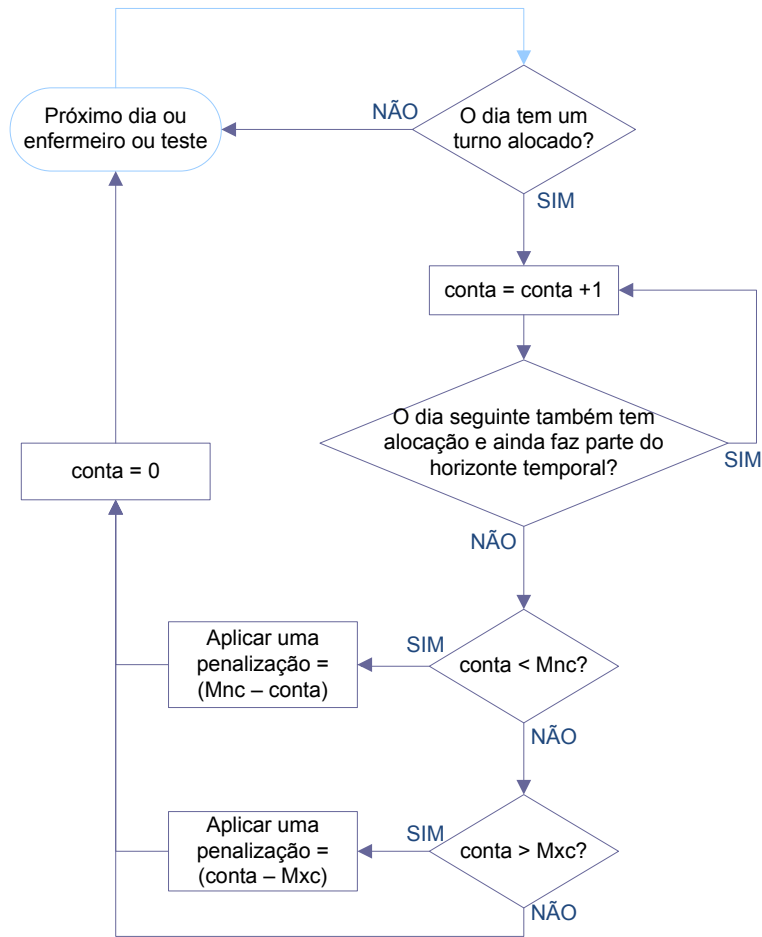


Figura 6.5: Esquema de funcionamento da função que impõe o cumprimento dos máximo e mínimo de dias alocados consecutivos.

Restrição de máximo e mínimo de alocações consecutivas no *Algoritmo Genético*

Os valores máximo e mínimo utilizados para teste desta restrição foram de, respectivamente, 5 e 2 dias de trabalho consecutivos. Mais uma vez, foi utilizada a instância alargada, com os parâmetros do *Algoritmo Genético* iguais aos projectados inicialmente, com excepção do critério de paragem, que foi de 20000 gerações, tal como em todas as restantes “soft constraints”.

Analisando os resultados obtidos para testes a vários pesos atribuídos a esta restrição, apresentados na Tabela 6.11, pode concluir-se que o melhor valor será o 40 ou o 30. Isto porque, partindo de populações com penalizações médias relativamente elevadas (152 ou 153 dias de excesso ou defeito), estes

valores de peso permitem alcançar valores muito reduzidos de penalização: pouco mais de uma dezena para o valor 30, o que não provoca redução significativa dos custos de alocação; e 4 ou 5 violações da restrição para o peso 40, apesar de, para este caso, haver mais custos associados.

Peso	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1		
1	48	38550 : 40
10	40	38700 : 29
20	20	38475 : 13
30	19	38430 : 13
40	12	37830 : 5
POPULAÇÃO 2		
1	71	38640 : 57
10	48	38550 : 37
20	36	38565 : 26
30	15	38505 : 11
40	7	37590 : 4
POPULAÇÃO 3		
1	52	38760 : 44
10	32	38715 : 25
20	26	38580 : 20
30	13	38430 : 11
40	5	37115 : 4

Tabela 6.11: Valores obtidos para os testes de influência da penalização de máximo e mínimo de alocações consecutivas permitidas, no *Algoritmo Genético*, com vários pesos atribuídos, para a instância alargada.

6.3.5 Máximo e Mínimo de Dias Livres Consecutivos

Esta restrição segue a lógica contrária à anterior. Em vez de se pretender ter algum controlo sobre o número de dias consecutivos alocados, pretende-se ter o mesmo tipo de controlo sobre o número de dias livres consecutivos. Faz-se assim os possíveis por condicionar o número de dias livres consecutivos, definindo os seus máximos e mínimos. Mais uma vez, esses valores podem ser diferentes para os vários enfermeiros, embora, para os testes aqui efectuados, eles tenham tido todos os mesmos valores.

Neste caso, o esquema de funcionamento é muito semelhante ao anterior, com a única diferença de ser orientado para dias livres, em vez de ser orientado para dias de trabalho. Esse esquema está apresentado na Figura 6.6.

tendo em conta que as populações iniciais têm valores médios de penalizações de 153 e 109, se conclui que bons valores de peso para esta restrição são o 40 e o 50, não obrigando a um significativo aumento dos custos de alocação e permitindo obter reduzidos valores de penalização na solução encontrada.

Peso	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1		
1	95	38850 : 94
10	67	38550 : 62
20	41	37885 : 37
30	29	38250 : 22
40	16	37905 : 11
50	11	37440 : 5
POPULAÇÃO 2		
1	104	38610 : 101
20	37	38685 : 33
40	14	37980 : 9
50	10	38010 : 4
POPULAÇÃO 3		
1	101	38655 : 97
20	41	38580 : 37
40	19	38370 : 15
50	12	38040 : 8

Tabela 6.12: Valores obtidos para os testes de influência da penalização de máximo e mínimo de dias livres consecutivos permitidos, no *Algoritmo Genético*, com vários pesos atribuídos, para a instância alargada.

6.3.6 Equilíbrio entre Escalonamentos Individuais

É muito importante para o bom funcionamento da instituição que os horários atribuídos aos enfermeiros sejam o mais possível equilibrados entre si. Assim, a probabilidade de se sentirem injustiçados será menor do que no caso de não haver esta preocupação, por parte do decisor, ou por parte do sistema que auxilia a elaboração dos horários.

O equilíbrio entre os horários dos enfermeiros deve ser conseguido entre aqueles que pertencem ao mesmo grau, já que as cargas de trabalho entre os diferentes graus podem ser distintas.

Para a verificação desta restrição, conta-se, inicialmente, o número de alocações de cada enfermeiro ao longo do horizonte temporal. Para isso recorre-se, mais uma vez, à matriz de contagem do número de alocações totais, por enfermeiro, Tot_e , já utilizada na função que garante a restrição do máximo de total de alocações no horizonte temporal. Após o cálculo

dos totais de alocações para cada enfermeiro, fazem-se médias de alocações, uma para cada grau de enfermeiros. No final, verifica-se, para cada grau, se cada enfermeiro tem mais ou menos alocações do que a média dos seus pares, e aplica-se uma penalização igual ao valor em excesso ou defeito de alocações. Estes processos, após a contagem de alocações ao longo do horizonte temporal, feita do mesmo modo que está demonstrado no Algoritmo 6.12, encontram-se esquematizados na Figura 6.7.

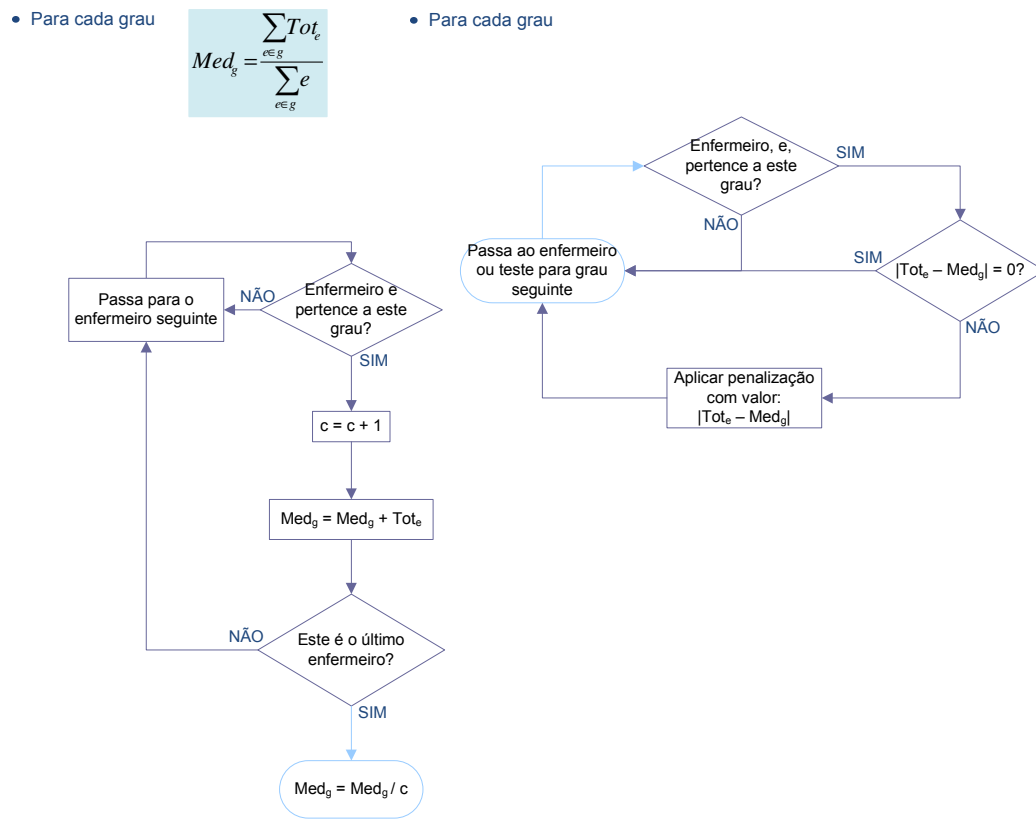


Figura 6.7: Esquema de funcionamento dos módulos da função que procura garantir o equilíbrio entre os escalonamentos individuais (horários) dos vários enfermeiros de cada grau.

Restrição de equilíbrio no *Algoritmo Genético*

Seguindo a lógica de teste da restrição que tem vindo a ser utilizada, esta “soft constraint” foi testada com vários valores de pesos atribuídos. A partir dos resultados desses testes construiu-se a Tabela 6.13. Analisando

os dados aí fornecidos, o melhor peso para esta restrição em funcionamento isolado é o de valor 40. Com este valor obtêm-se penalizações muito baixas nas soluções alcançadas pelo *Algoritmo Genético*, apesar de tais soluções exigirem um grande aumento dos custo de alocações. O que significa que, para diminuir a violação desta restrição, o *Algoritmo Genético* aumenta o número de alocações por forma a aproximar o número médio de alocações dos enfermeiros dos valores mais elevados de número de alocações totais. Daqui se conclui que, num funcionamento do *Algoritmo Genético* com toda as “soft constraints”, esta não poderá ter uma importância muito elevada na calibração de pesos, já que a procura da sua satisfação conduzirá a um aumento significativo dos custos de alocações. Ainda assim, deverá ter alguma importância, uma vez que, com pesos de valor baixo (1 e 10), o número de violações pouco se altera relativamente à população inicial (começa em valores próximos de 90 violações e decresce apenas para valores pouco inferiores – 74, 73, chegando mesmo a ser superior – 89 para a terceira população de partida).

Peso	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1		
1	74	38670 : 69
10	73	38640 : 57
20	26	37725 : 15
30	13	36270 : 6
40	10	35475 : 4
50	14	33795 : 8
POPULAÇÃO 2		
1	74	38730 : 65
20	34	37395 : 23
30	27	36540 : 17
40	10	35880 : 5
POPULAÇÃO 3		
1	89	38730 : 85
30	23	35940 : 19
40	17	36645 : 13
50	16	34200 : 12

Tabela 6.13: Valores obtidos para os testes de influência da penalização de equilíbrio entre escalonamentos, para enfermeiros do mesmo grau, no *Algoritmo Genético*, com vários pesos atribuídos, para a instância alargada.

6.3.7 Preferências de Alocações

As preferências de alocações de cada enfermeiro, quer a dias de trabalho, quer a turnos dentro de cada um desses dias, também são aqui consideradas. Procura-se satisfazer estas preferências aplicando uma penalização sempre

que o enfermeiro trabalhe num dia que não pretendia, ou uma penalização, menos elevada, sempre que um enfermeiro, apesar de trabalhar num dia desejado, não trabalhe num turno de sua preferência.

O esquema de funcionamento da função que penaliza as violações de preferências em alocações está esquematizado na Figura 6.8. A penalização muito alta tem o valor 4 e a alta o valor 3.

- Para cada enfermeiro

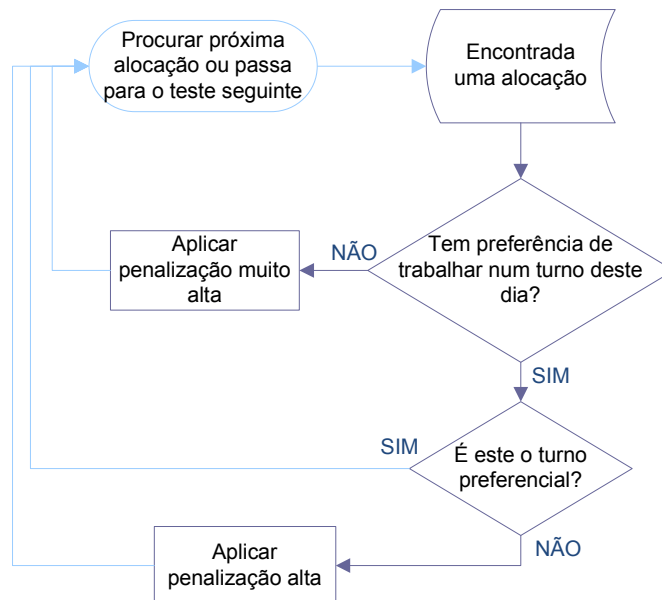


Figura 6.8: Esquema de funcionamento da função que procura garantir o cumprimento de preferências de alocações dos enfermeiros.

Restrição de preferências no *Algoritmo Genético*

Apesar de se pretender que, tal como as restantes, seja o menos violada possível, para esta restrição, tal como para a anterior (de equilíbrio entre os escalonamentos), um pequeno número de violações não será, em geral, muito problemático, já que, quando se pretende agradar a todos, haverá sempre uns poucos que não serão totalmente satisfeitos.

Assim, analisando os resultados obtidos para os testes com vários pesos atribuídos à restrição, apresentados na Tabela 6.14, conclui-se que o peso de

valor 30 conduz a soluções de grande satisfação global. Este peso permite obter soluções que apenas têm penalizações com valores de 6 e 18, para as populações de partida usadas. No primeiro caso, o valor 6 indica que há duas violações do tipo de um enfermeiro trabalhar num turno que não prefere, apesar de ser um dia em que pretende trabalhar. No segundo, o valor 18 indica que, no pior caso (em que todas as violações são como a anterior, já que aplicam o menor valor de penalizações), há 6 violações de preferências. Estes números de violações, num universo de 30 enfermeiros e 28 dias de horizonte temporal, são muito satisfatórios. Os valores de penalizações médias das populações iniciais são de 683, 688 e 685 para cada uma das populações 1, 2 e 3, respectivamente. De notar que, além da significativa melhoria no valor das penalizações para qualquer um dos pesos, a modificação dos pesos e a melhoria da satisfação desta restrição não obrigam a aumentos significativos dos valores dos custos de alocações.

Peso	Média de Penalizações na última população	Solução Obtida (Maximizante – Custos) : Penalização
POPULAÇÃO 1		
1	120	38730 : 105
10	66	38625 : 51
20	36	38445 : 27
30	16	38625 : 6
POPULAÇÃO 2		
1	122	38610 : 114
20	49	38505 : 24
30	39	38745 : 18
POPULAÇÃO 3		
1	130	38730 : 129
20	32	38850 : 27
30	20	38625 : 18

Tabela 6.14: Valores obtidos para os testes de influência da penalização de preferências de alocações dos enfermeiros, no *Algoritmo Genético*, com vários pesos atribuídos, para a instância alargada.

6.3.8 Contactos

Por fim, existem as restrições relativas aos contactos, quer entre enfermeiros que trabalham no mesmo dia e no mesmo turno, quer entre doentes e enfermeiros.

A primeira restrição, comum em trabalhos que abordam este problema, permite evitar que certos enfermeiros trabalhem juntos ou então procurar que outros trabalhem no mesmo turno. Por vezes esta restrição é importante, quer para garantir o bom ambiente na instituição, evitando contacto entre

enfermeiros que não conseguem trabalhar juntos, por qualquer razão, quer para garantir que determinadas tutorias são, tanto quanto possível, cumpridas, permitindo que um enfermeiro de menor grau tenha acompanhamento sempre do mesmo superior.

Em determinadas instituições, como as de Cuidados Continuados, o contacto entre o doente e os técnicos de saúde que lhe dão assistência é muito valorizado, por permitir o desenvolvimento de uma relação de confiança entre os mesmos. A segunda restrição procura manter constante o contacto entre os enfermeiros e os doentes, permitindo o estabelecimento e a durabilidade dessas relações.

Para a implementação destas duas restrições, foi necessário acrescentar mais duas matrizes de dados ao problema (já aqui mencionadas – matrizes Cte_{ee} e Ctd_{ep}). A primeira tem os dados das preferências de contactos entre os enfermeiros, a segunda os dados dos enfermeiros que cada doente costuma ter como assistente, referidos de aqui em diante por enfermeiros conhecidos do doente.

A matriz Cte_{ee} é uma matriz triangular superior, já que sabendo se o enfermeiro 1 deve contactar, não deve contactar ou é indiferente que contacte com o enfermeiro 2, o mesmo vale para o enfermeiro 2 relativamente ao enfermeiro 1, acontecendo o mesmo para as relações entre os demais enfermeiros. Esta poderá ser chamada a matriz de contactos “proibidos” e “obrigatórios” entre os enfermeiros e tomará os seguintes valores:

- 1 – caso o contacto entre os dois enfermeiros seja “proibido”;
- 0 – caso o contacto entre os dois enfermeiros seja indiferente;
- -1 – caso o contacto entre os dois enfermeiros seja “obrigatório”.

A matriz Ctd_{ep} será preenchida com “zeros” nas células que correspondam a enfermeiros que o doente respectivo não conhece, e “uns” nas células que correspondam aos enfermeiros que o doente conhece.

Contactos entre Enfermeiros

O contacto entre enfermeiros é implementado da forma que está descrita no Algoritmo 6.13. Neste caso, quando dois enfermeiros que devem trabalhar, preferencialmente separados, estão alocados ao mesmo turno, é aplicada uma penalização; quando um de dois enfermeiros que devem trabalhar preferencialmente separados trabalha num turno em que o segundo enfermeiro

não trabalha, é aplicada uma pequena compensação; quando dois enfermeiros que trabalham preferencialmente juntos (por razões de tutorias, por exemplo) trabalham no mesmo turno, é aplicada uma compensação mais elevada que a anterior.

1. *Percorre, com um ciclo, todos os enfermeiros, $e1$, até ao penúltimo. Em cada iteração:*
 - *Percorre, com um ciclo, todos os enfermeiros, $e2$, desde $e1$ até ao último. Em cada iteração:*
 - *Percorre, com um ciclo, todos os dias do horizonte temporal. Em cada iteração:*
 - * *Se $e1$ estiver alocado a um qualquer turno no dia actual:*
 - (a) *Caso os dois enfermeiros trabalhem preferencialmente separados:*
 - i. *Se $e2$ também estiver alocado ao mesmo turno: Aplica uma penalização muito alta;*
 - ii. *Se não: Aplica uma compensação baixa;*
 - (b) *Caso os dois enfermeiros trabalhem preferencialmente juntos:*
 - i. *Se $e2$ também estiver alocado ao mesmo turno: Aplica uma compensação alta.*

Algoritmo 6.13: Abordagem de funcionamento da função que procura garantir o cumprimento das preferências de contactos entre enfermeiros.

Contactos Enfermeiros – Doentes

Foram concebidas duas formas de procurar que os doentes alojados na instituição possam encontrar, em cada dia, pelo menos um enfermeiro do seu conhecimento, que lhes preste assistência.

Poderia ter sido desenvolvida uma lógica da mesma procura, por turno de cada dia. Mas, se já com a procura de cumprimento da restrição por dia o *Algoritmo Genético* fica substancialmente mais lento, se se considerasse por turno, além de pouco proveitoso, o *Algoritmo Genético* ficaria exageradamente lento. Qualifica-se esta possibilidade de pouco proveitosa, já que, por exemplo, no turno nocturno, em que a assistência poderá acontecer com menor frequência, uma vez que os doentes estarão a repousar, esta garantia

de contacto pode ser dispensada. Deste modo, decidiu-se procurar que cada doente tenha o máximo número de enfermeiros conhecidos a trabalhar por dia e não por turno.

1. *Percorre, com um ciclo todos os doentes. Em cada iteração:*
 - (a) *Inicia o contador do número de enfermeiros conhecidos, para este doente (Contador_e), com o valor 0;*
2. *Percorre, com um ciclo todos os enfermeiros e todos os dias do horizonte temporal. Em cada iteração:*
 - (a) *Se o enfermeiro está alocado a este dia:*
 - *Percorre, com um ciclo todos os doentes. Em cada iteração:*
 - i. *Se o doente conhecer o enfermeiro alocado: Conta mais um enfermeiro conhecido e alocado a um dia ($\text{Contador}_e = \text{Contador}_e + 1$);*
3. *Percorre, com um ciclo todos os doentes. Em cada iteração:*
 - (a) *Se $\text{Contador}_e > 0$: Aplica compensação igual ao valor de Contador_e ;*
 - (b) *Se não: Aplica penalização muito alta.*

Algoritmo 6.14: Algoritmo de funcionamento da segunda abordagem que procura garantir que o contacto entre os enfermeiros e os doentes permanece, tanto quanto possível, sempre entre as mesmas pessoas.

A primeira abordagem a esta restrição é feita contando o número de enfermeiros, que cada doente conhece, a trabalhar em cada dia. Caso, num dia, um doente não conheça ninguém, é aplicada uma penalização de valor pré-definido. Caso o doente conheça, será aplicada uma compensação de valor igual ao número de enfermeiros a trabalhar nesse dia que o doente conhece. A Figura 6.9 esquematiza esta situação.

A segunda abordagem, que permite um funcionamento mais célere do *Algoritmo Genético*, conta o número de enfermeiros que cada doente conhece e que trabalham em todos os dias do horizonte temporal. Para cada doente, se ele não conhecer nenhum dos enfermeiros alocados (caso que, em princípio, não ocorrerá por significar que, em todo o horizonte temporal, o doente não conhece nenhum dos enfermeiros alocados, mas fica aqui salvaguardado), será aplicada uma penalização, caso contrário, será aplicada uma compensação com o mesmo valor que o número de enfermeiros conhecidos pelo doente.

Esta abordagem centra-se, portanto, nas compensações atribuídas, que serão tanto maiores quantos mais enfermeiros os doentes conhecerem nos vários dias do horizonte temporal, e está apresentada no Algoritmo 6.14. Esta foi a abordagem utilizada nos testes finais ao algoritmo. No final serão também apresentados testes com a primeira abordagem a esta restrição.

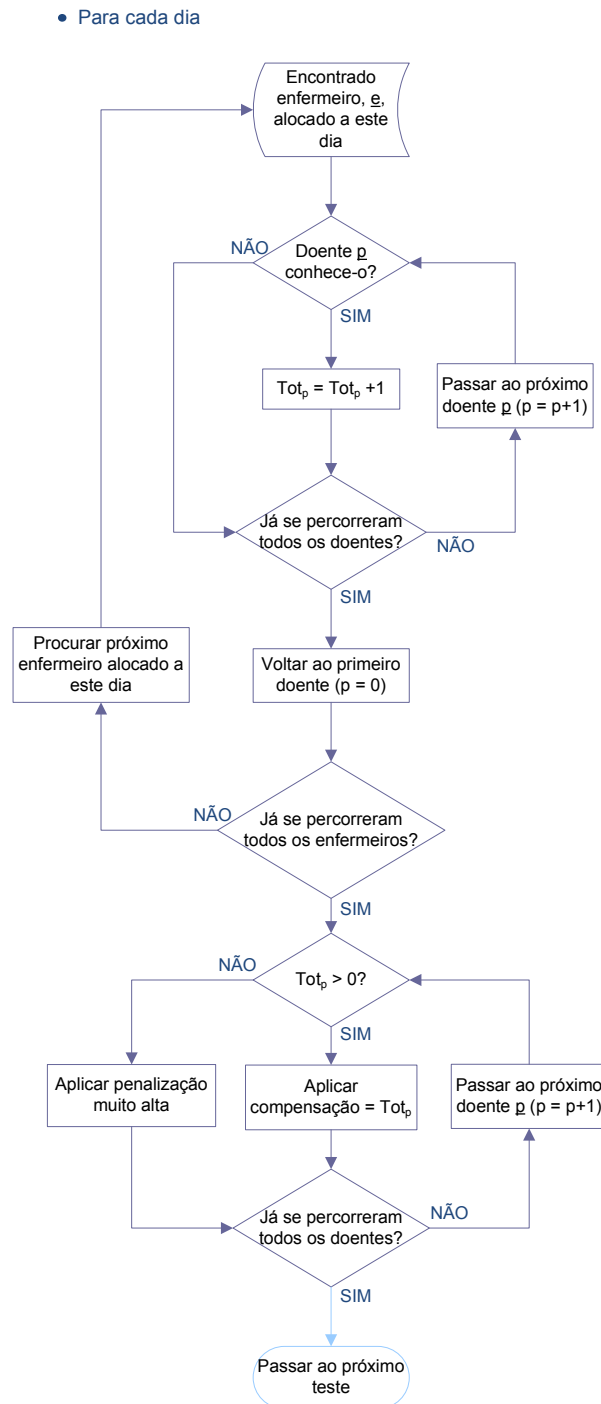


Figura 6.9: Esquema da primeira abordagem de funcionamento da função que procura garantir que os contactos entre enfermeiros e doentes são o mais constantes possíveis.

6.4 O Algoritmo Genético Final

O funcionamento do *Algoritmo Genético* com todas as restrições é conseguido acrescentando ao Algoritmo 6.10, nas alturas das avaliações, um módulo que corre as funções de “soft constraints” para cada cromossoma e subtrai, à Função Avaliadora anteriormente apresentada, o valor resultante de todas as penalizações aplicadas ao cromossoma.

Ao utilizador é dada a hipótese de correr o *Algoritmo Genético* com todas as “soft constraints” do problema, ou apenas com algumas. Surge, antes de o *Algoritmo Genético* começar a pesquisa de uma boa solução, no início do programa, a hipótese de, para cada “soft constraint” dos nove módulos apresentados¹, o utilizador escolher quais serão contempladas para a pesquisa e quais não serão consideradas.

A função que percorre todas as restrições anteriores e calcula o valor total de penalização a atribuir a cada cromossoma funciona segundo o Algoritmo 6.15. Esta função é chamada na altura em que para cada cromossoma é feita a sua avaliação, calculando também os custos de alocação associados e, finalmente, calculando a Função Avaliadora, pela subtracção destas duas parcelas (custos de alocações e valor de penalização total pelas “soft constraints”) ao maximizante, aqui duplicado relativamente ao maximizante usado inicialmente.

6.4.1 Funcionamento

Nesta última fase do trabalho, para uma boa integração das “soft constraints”, foi necessário calibrar os seus pesos relativos, por forma a obter resultados melhores e mais equilibrados.

Inicialmente, foi pensado calibrar os pesos de forma a normalizar os pesos de todas as restrições. Mas, depois de testadas as restrições uma a uma, verificou-se que tal pretensão é difícil de conseguir, uma vez que as penalizações das diferentes restrições têm evoluções diferentes, como se pode constatar pelas análises anteriores. No entanto, pode normalizar-se os valores iniciais das penalizações, mesmo sabendo que as suas evoluções serão diferentes e que, com o evoluir da pesquisa, os valores deixarão de estar nor-

¹fins-de-semana; mínimos de folgas entre alocações; máximos de alocações no horizonte temporal; máximos e mínimos de alocações consecutivas; máximos e mínimos de dias livres consecutivos; equilíbrios entre os horários; preferências de alocações dos enfermeiros; preferências de contactos entre enfermeiros e preferências de contactos entre enfermeiros e doentes.

1. *Inicia todos os valores das penalizações associadas a cada um dos nove módulos de “soft constraints” com valores nulos;*
2. *Para cada um dos módulos faz:*
 - (a) *Se o utilizador quiser que esta “soft constraint”, ou conjunto de “soft constraints” (caso do módulo relativo a fins-de-semana), seja incluída na pesquisa:*
 - i. *Chama a função que executa as verificações necessárias para essa restrição;*
 - ii. *Guarda o valor de penalização atribuída ao cromossoma em causa;*
3. *Soma todos os valores de penalizações, multiplicados pelos respectivos pesos atribuídos às restrições a que correspondem;*
4. *Devolve o valor resultante dessa soma.*

Algoritmo 6.15: Algoritmo de funcionamento da função de chamada das “soft constraints” para cálculo da penalização total.

malizados.

Definem-se assim dois conjuntos de partida para os testes de calibração dos pesos das “soft constraints”. Um corresponde à normalização dos valores médios de penalizações iniciais dos vários módulos de “soft constraints” e outro é constituído por valores considerados melhores para cada restrição, descobertos nas análises anteriores.

A normalização dos valores médios iniciais das penalizações foi feita considerando valores próximos dos mais elevados valores médios de penalização inicial (aproximados a valores divisíveis por 10, para mais fácil normalização² – *Valor Aproximado* na Tabela 6.15), para cada restrição. Desses valores mais elevados, escolheram-se os mais altos para referência. As referências foram multiplicadas por 10 e os restantes pelos valores devidos para se aproximarem dos valores de referência. O processo está registado na Tabela 6.15.

Para o outro conjunto de partida, em que se pretendia experimentar

²Tendo em conta que se está a trabalhar com valores médios, haverá valores mais elevados do que esses nas penalizações das populações iniciais. Para simplificar o processo de atribuição de pesos, uma vez que apenas se pretende obter um conjunto de valores de partida para depois serem aprimorados, arredondam-se os valores das penalizações para valores próximos que facilitem a normalização (a determinação do conjunto de partida).

6.4. O ALGORITMO GENÉTICO FINAL

Módulo Relativo a:	Média Mais Alta de Penalizações Iniciais	Valor Aproximado (penalizações)	Peso Atribuído
Fins-de-semana	638	700	10
Mínimo folgas	422	500	15
Máximo de alocações no horizonte temporal	129	150	50
Max./min. alocações consecutivas	153	170	40
Max. /min. dias livres consecutivos	152	170	40
Equilíbrio	88	100	70
Preferências alocações	688	700	10

Tabela 6.15: Normalização dos pesos atribuídos às “soft constraints”.

o funcionamento do *Algoritmo Genético* com todos os pesos anteriormente identificados como sendo melhores para as “soft constraints”, optou-se pelos seguintes valores: penalizações da restrição de *fins-de-semana* – valor 30; penalizações da restrição de *mínimos de folgas* – valor 10; penalizações da restrição de *máximos de alocações no horizonte temporal* – valor 20; penalizações das restrições de *máximos e mínimos de alocações consecutivas*, de *máximos e mínimos de dias livres consecutivos* e de *equilíbrio* entre os horários – valor 40; por fim, penalizações da restrição de *preferências de alocações* – valor 30.

Definidos os dois conjuntos de partida, ainda antes de se testar algumas variações aos conjuntos de valores, na busca de um conjunto de pesos melhor do que algum dos já definidos, e antes também de introduzir as duas restrições relativas a contactos, testaram-se estes dois conjuntos de restrições (o de *normalização* e o de *melhores pesos individuais*) com o intuito de descobrir se valeria a pena considerar os dois como pontos de partida, ou se haveria uma diferença de resultados tão significativa que permitisse excluir um deles desde início. Os resultados desses testes estão registados na Tabela 6.16. Todos os testes realizados nesta última parte do trabalho utilizaram a instância elaborada com o “**Cenário 1**”.

Comparando os resultados, apesar de o conjunto composto pelos melhores pesos individuais não equilibrar tão bem os horários dos vários enfermeiros e ter mais algumas (poucas) violações às restrições de alocações consecutivas e de mínimo de folgas, a vantagem da sua escolha é muito clara por dimi-

6.4. O ALGORITMO GENÉTICO FINAL

Conjunto de Pesos	Penalizações finais para a restrição de:			
	Fins-de-semana	Folgas	Máx. Alocações	Alocações Consec.
Normalizados	208	50	0	15
Melhores Individuais	129	62	0	19
	Penalizações finais de:			(Maximizante – Custos)
	Dias Livres	Equilíbrio	Preferências	
Normalizados	24	24	282	53940
Melhores Individuais	24	39	147	54030

Tabela 6.16: Testes aos dois conjuntos de pesos iniciais (conjunto de pesos normalizados e conjunto dos melhores pesos individuais).

nuir significativamente as penalizações relativas a fins-de-semana e relativas a preferências dos enfermeiros (estas quase reduzidas para metade), assim como permite uma ligeira redução dos custos de alocações. Na verdade, as pequenas diferenças não ditaram a escolha de nenhum dos conjuntos, por poder tratar-se de diferenças pontuais, que com outras populações iniciais são susceptíveis de se alterarem. O que ditou a escolha, que recaiu para o conjunto dos melhores pesos individuais como conjunto de partida para a continuação de melhoria do conjunto de pesos, foram essas grandes diferenças entre as penalizações de fins-de-semana e de preferências de alocações, que, ao contrário de todas as restantes diferenças, são muito significativas.

A fase seguinte do aperfeiçoamento do *Algoritmo Genético* serviu para divergir a quantidade de conjuntos de pesos dados às “soft constraints”, para depois convergir num conjunto equilibrado, que permita obter, partindo de diferentes populações, boas soluções.

A primeira parte desta segunda e última fase de calibração do *Algoritmo Genético* no seu funcionamento pleno partiu então do conjunto de melhores pesos para as restrições consideradas individualmente. Os restantes conjuntos foram criados variando alguns dos pesos atribuídos inicialmente a algumas restrições, com a intenção de as melhorar, por serem as que detiveram piores valores nos testes anteriores. O que se verifica é que, variando um peso para uma restrição, essa acção tem repercussões nas restantes, dificultando a tarefa de calibração. Os conjuntos de restrições utilizados para os testes estão expostos na Tabela 6.17 e os seus resultados na Tabela 6.18. Nestas tabelas, cada conjunto é identificado por dois caracteres, compostos por “Tn”, em que n é o número do conjunto, as restrições estão referenciadas por siglas: restrições de fins-de-semana – *f.d.s*; restrições de mínimos de folgas – *folgas*; restrições de máximo de alocações no horizonte temporal – *maxa*; restrições de alocações consecutivas – *maxc*; restrições de dias livres consecutivos – *mlc*; restrições de equilíbrios de horários – *equil.*; restrições

6.4. O ALGORITMO GENÉTICO FINAL

de preferências de alocações – *pref.*; restrições de contactos entre enfermeiros – *cont_e* e restrições de contactos entre enfermeiros e doentes – *cont_d*. Os “custos” referem-se à diferença entre o maximizante e os custos de alocação, “F.A.” são os valores de Função Avaliadora e “S.C.” os totais de penalizações das “soft constraints”.

Os pesos relativos às restrições de contactos entre enfermeiros e de contactos entre enfermeiros e doentes são sempre calibrados com valor 1, por terem valores muito elevados e serem sempre valores de compensação (negativos). O que significa que, mesmo não lhes atribuindo peso, eles têm um impacto significativo na solução obtida, como é possível constatar comparando os resultados obtidos para o conjunto de pesos iniciais, incluindo estas restrições e não as incluindo (na Tabela 6.16). Esse impacto traduz-se numa melhoria quase geral nos restantes medidores de resultado (só os custos aumentam e a penalização relativa a dias livres consecutivos aumenta ligeiramente).

Id	f.d.s.	folgas	maxa	mxo	mlc	equil.	pref.	cont_e	cont_d
T1	30	10	20	40	40	40	30	1	1
T2	30	20	10	40	40	40	30	1	1
T3	40	20	10	40	40	40	40	1	1
T4	50	10	10	40	40	40	30	1	1
T5	50	20	10	40	40	40	40	1	1
T6	60	20	10	50	50	40	30	1	1
T7	50	10	10	50	50	40	30	1	1
T8	50	10	10	50	40	40	30	1	1

Tabela 6.17: Conjuntos de pesos para as restrições.

Id	F.A.	custos	f.d.s.	folgas	maxa	mxo	mlc	equil.	pref.	cont_e	cont_d	S.C.
T1	49756	53730	120	59	0	15	31	38	102	-952	-5684	3974
T2	48033	54555	117	43	0	14	27	38	159	-420	-5358	6522
T3	46951	54810	118	53	1	27	28	42	105	-784	-5227	7859
T4	47889	54510	83	86	0	16	29	46	138	-784	-5385	6621
T5	45746	54180	107	44	0	21	31	41	117	-672	-5524	8434
T6	44616	53730	109	64	0	23	28	32	126	-644	-5672	9114
T7	44866	54000	103	64	0	23	28	49	180	-896	-5670	9134
T8	45186	54030	111	68	1	17	33	47	165	-784	-5612	8844

Tabela 6.18: Resultados dos testes aos vários conjuntos de pesos para as restrições.

Dos oito conjuntos de pesos testados, escolheram-se os três cujos resultados estão assinalados a negrito. Estes revelam-se os mais equilibrados, em termos de valores de penalizações de todas as “soft constraints”.

O conjunto *T1*, apesar de ter dos piores valores de custos e de fins-de-semana, não se revelando também dos melhores em termos de cumprimento

de mínimo de folgas, é dos que melhor cumpre a restrição de alocações consecutivas, tendo também bons valores nas restantes restrições (o melhor em preferências de alocações). A penalização de fins-de-semana não se afasta, ainda assim, muito dos restantes valores, não sendo um pico fortemente destacável.

O conjunto de pesos $T2$ tem os segundos melhores custos de alocações, penalizações de fins-de-semana altas, embora melhores que o anterior, os melhores valores de penalizações de mínimos de folgas, alocações consecutivas e dias livres consecutivos, mantendo um bom equilíbrio entre os horários. Não permite tão bons resultados nas preferências de alocações e de contactos, especialmente entre enfermeiros.

O conjunto $T4$ prima também pelos reduzidos custos de alocações, pelos bons valores de penalização de fins-de-semana e equilibrados valores de alocações consecutivas e dias livres consecutivos. A penalização de mínimos de folgas é a mais prejudicada para este conjunto relativamente aos demais.

Os restantes conjuntos foram excluídos por os seus valores de penalizações, apesar de para certas restrições particulares poderem destacar-se por serem bons, noutras, especialmente nas de alocações consecutivas e dias livres consecutivos, serem maus resultados, e estas costumam ser as “soft constraints” que primeiramente se pretendem satisfeitas, por dizerem respeito a cargas de trabalho no horizonte temporal. Alguns dos conjuntos chegam a ultrapassar os 22 dias de alocações permitidos ao longo do horizonte temporal.

Esta selecção não foi fácil de realizar, por uns conjuntos se revelarem claramente melhores para determinadas restrições e outros para outras, havendo ainda alguns, que, apesar de não se destacarem como bons para alguma restrição em particular, se revelaram bastante equilibrados. Assim, houve a necessidade de dar maior importância a algumas “soft constraints”, tendo sido seleccionadas as “soft constraints” relativas à carga de trabalho no horizonte temporal anteriormente referidas. Mas, em casos particulares da implementação deste algoritmo, a instituição que pretender utilizar o sistema pode querer calibrar os pesos de outra forma, dando maior valor a outras restrições.

Dos três conjuntos de pesos seleccionados pretendia-se, em seguida, seleccionar um, para ser apontado como o melhor e ser implementado no final. Com esse intuito, e para que a escolha fosse bem fundamentada, esses conjuntos foram testados para mais duas populações iniciais. Desses testes resultaram soluções com valores de penalizações e custos anotados na Tabela 6.19, com três soluções para cada conjunto de pesos, correspondendo às três populações iniciais diferentes criadas.

6.4. O ALGORITMO GENÉTICO FINAL

Id	F.A.	custos	f.d.s.	folgas	maxa	mxo	mlc	equil.	pref.	cont_e	cont_d	S.C.
T1.1	49756	53730	120	59	0	15	31	38	102	-952	-5684	3974
T1.2	52445	54540	126	61	0	18	21	19	57	-1036	-5289	2095
T1.3	49746	53700	131	62	0	24	32	18	84	-364	-5712	3954
T2.1	48033	54555	117	43	0	14	27	38	159	-420	-5358	6522
T2.2	50636	53910	108	42	0	16	37	33	75	-868	-5628	3274
T2.3	50486	53520	112	53	0	15	26	31	93	-1260	-5806	3034
T4.1	47889	54510	83	86	0	16	29	46	138	-784	-5385	6621
T4.2	49604	53610	95	52	0	22	23	29	78	-756	-5808	4006
T4.3	49162	53865	91	75	0	20	24	34	99	-1036	-5651	4703

Tabela 6.19: Resultados dos testes aos três conjuntos de pesos seleccionados, com diferentes populações iniciais.

Analisando os resultados, o conjunto que aparenta ser mais equilibrado e mais vantajoso é o identificado por $T2$, tendo sempre bons resultados de penalizações de fins-de-semana, os melhores nas penalizações de mínimos de folgas e alocações consecutivas, mantendo bons resultados para dias livres consecutivos (valores, em geral baixos), equilíbrio de horários e de contactos entre enfermeiros e doentes. Quanto a preferências de alocações, não revelam resultados destacáveis, embora se mantenham bons, especialmente tendo em conta que, em geral, nas populações iniciais esta restrição é das mais violadas. Nos contactos entre enfermeiros, os resultados dependem bastante da população inicial, o que revela ser o ponto fraco deste conjunto. Mas, em termos gerais, este é o melhor conjunto de pesos.

Voltando um pouco a trás, no que respeita à escolha da dimensão das populações, como já foi referido, escolheu-se apenas utilizar populações com mais de 500 indivíduos nos testes iniciais, pensando na diversidade que isso permite às várias gerações pelas quais o *Algoritmo Genético* passa durante a pesquisa. Esta particularidade é de muita importância na versão final do *Algoritmo Genético*, uma vez que, além de se estar a otimizar a solução em função das “hard constraints”, está-se também a otimizar em função de muitas e complexas “soft constraints”. Na Tabela 6.20 pode constatar-se que, apesar de a diferença de valores não ser muito elevada, ela existe e revela que é mais compensador ter uma população de 500 indivíduos do que uma com 300 ou 400 indivíduos apenas. Essa compensação é visível nos valores de penalizações totais às “soft constraints”, que decrescem quando se utiliza os 500 indivíduos (decrescem um valor perto de 300), nos valores de Função Avaliadora que crescem valores entre 400 e 700 e nos custos de alocação, que decrescem com os 500 indivíduos. Por outro lado, apesar de o tempo de pesquisa aumentar, tendo em conta que se mantém inferior a uma hora de execução, pode considerar-se satisfatório e compensatório usar os 500

6.4. O ALGORITMO GENÉTICO FINAL

indivíduos. O acréscimo entre os 300 e os 500 indivíduos pouco excede os 1200 segundos (20 minutos), sendo significativo nos valores a que diz respeito e podendo ser importante considerar a opção pelos 300 indivíduos, caso se pretenda obter uma solução, não tão boa, mas rápida.

Nº de indivíduos	F.A.	custos	Total de Penalizações			Tempo de Pesquisa (em segundos)			
500	48033	54555	6522			2958			
400	47378	54240	6862			2367			
300	47603	54405	6802			1744			
Nº de indivíduos			Penalizações						
	f.d.s.	folgas	maxa	mxo	mlc	equil.	pref.	cont_e	cont_d
500	117	43	0	14	27	38	159	-420	-5358
400	124	59	0	21	35	48	132	-672	-5486
300	155	66	0	14	26	40	144	-1232	-5456

Tabela 6.20: Resultados dos testes a diferentes números de indivíduos em cada geração, no *Algoritmo Genético* final.

Por último, testou-se o funcionamento do *Algoritmo Genético* com a primeira abordagem concebida para a restrição de contactos entre enfermeiros e doentes. O seu tempo de pesquisa é muito próximo do tempo de pesquisa para a primeira abordagem e o seu contributo para a solução foi bom, melhorando quase todas as penalizações, e, as que piora (máximo e mínimo de alocações consecutivas e dias livres consecutivos), embora sejam das mais valorizadas nesta abordagem ao problema, não se distanciam significativamente dos valores do segundo caso. Assim, será proveitoso utilizar a primeira abordagem em vez da segunda, já que garante melhor o cumprimento da restrição: não serve apenas de compensador, caso a restrição seja cumprida, para além disso aplica uma penalização de cada vez que um doente não conhece nenhum enfermeiro a trabalhar durante um dia inteiro. A Tabela 6.21 tem registados os resultados obtidos para as duas abordagens testadas.

Para referência na comparação aos trabalhos abordados na literatura, sem a restrição de contacto entre os enfermeiros e os doentes, o tempo de pesquisa do *Algoritmo Genético* foi de 1114, 1148 e 1240 segundos (aproximadamente 19 minutos, nos dois primeiros casos e 21 no último) para as três populações iniciais (utilizando os pesos de penalizações do conjunto $T2$). Isto corresponde a cerca de 20 minutos de pesquisa por uma boa solução, o que se revela satisfatório quando comparado com os valores divulgados na literatura. Sem as duas restrições de contactos (entre enfermeiros e entre enfermeiros e doentes), é possível reduzir ainda mais o tempo de pesquisa, para menos de 15 minutos (775 segundos para uma das populações).

Os testes realizados ao longo do trabalho descrito neste capítulo foram

6.4. O ALGORITMO GENÉTICO FINAL

corridos num computador pessoal, com um processador Intel Core 2 Duo a 1,80GHz, com 2038Mb de memória RAM e um sistema operativo Windows Vista a 32 bits.

Abordagem	F.A.	custos	Total de Penalizações			Tempo de Pesquisa (em segundos)			
Primeira	49645	54900	5255			3073			
Segunda	48033	54555	6522			2958			
Abordagem		Penalizações							
	f.d.s.	folgas	maxa	mxo	mlc	equil.	pref.	cont_e	cont_d
Primeira	98	38	0	18	29	37	141	-896	-5139
Segunda	117	43	0	14	27	38	159	-420	-5358

Tabela 6.21: Resultados dos testes às duas abordagens para a restrição de contactos.

Capítulo 7

Conclusão

Com este trabalho pretendeu-se definir um problema de *Nurse Rostering*, desenvolver e validar um modelo matemático que o resolvesse e, finalmente, conceber e implementar uma heurística que permitisse obter uma boa solução, num curto intervalo de tempo.

A concepção do problema foi uma confluência entre o que geralmente é abordado e o caso particular que previamente se tinha identificado no Projecto Chronos. Assim, atingiu-se desde cedo o primeiro objectivo proposto: o de definir um problema de *Nurse Rostering* que, além de representar um caso geral, particularizasse o que se identificou ser uma restrição destacada na instituição de Cuidados Continuados para a qual foi desenvolvido o Projecto Chronos.

A modelização desse problema, recorrendo a programação inteira mista (MIP), permitiu consolidar e aprofundar o seu conhecimento, já proveniente do Projecto Chronos e da revisão bibliográfica inicial. Esta revelou-se uma fase importante do trabalho, facilitando a compreensão e análise da pesquisa levada a cabo pelo *Algoritmo Genético*, posteriormente desenvolvido, e a influência das restrições do problema na solução obtida.

Com os modelos matemáticos compreendeu-se quais os limites aceitáveis para valores dos parâmetros que definem restrições do problema. Confirmou-se terem sido desenvolvidas uma boa interpretação e uma abordagem ao problema proposto, conseguindo-se solucionar instâncias com diferentes dimensões de necessidades de cobertura.

A implementação do *Algoritmo Genético* para mais rapidamente resolver o problema, obtendo uma boa solução, exigiu o estudo destes algoritmos. Este estudo permitiu um aprofundamento considerável da compreensão, por parte do autor, do funcionamento das heurísticas e da sua importância em Investigação Operacional. Foi essencial para o sucesso na concepção e im-

plementação do *Algoritmo Genético* desenvolvido para o problema da dissertação.

A implementação do *Algoritmo Genético* que resolve o problema foi dividida em três fases. Primeiro projectou-se e implementou-se um *Algoritmo Genético* simples, apenas considerando as “hard constraints” na pesquisa por uma solução. Esta primeira implementação permitiu afinar os parâmetros que definem o funcionamento do *Algoritmo Genético*, para identificar o seu melhor desempenho. Na segunda fase implementou-se as várias “soft constraints”, uma de cada vez, testando a sua influência na pesquisa levada a cabo pelo *Algoritmo Genético*. Serviu também para afinar e identificar a melhor forma de inserir cada uma na função avaliadora das soluções. Finalmente, confluiu-se o trabalho destas três fases e testou-se o *Algoritmo Genético* final, considerando todas as “soft constraints”. Foi nesta última fase que foi incluída a nova restrição de contacto entre enfermeiros e doentes. Foi possível alcançar boas soluções para as instâncias testadas em intervalos de tempo de pesquisa que se enquadram nos referidos na literatura, o que revela o bom desempenho do algoritmo. Com a introdução da nova restrição (de contacto entre enfermeiros e doentes) esse tempo aumenta significativamente, embora permaneça em valores inferiores a uma hora (3600 segundos), que revelam a permanência de um bom desempenho temporal.

Ao longo dos testes realizados ao *Algoritmo Genético*, houve algumas decisões que tiveram de ser tomadas, deixando para trás possíveis desenvolvimentos de certos pormenores, por limitações temporais. Um desses desenvolvimentos que merece destaque é a pormenorização do estudo da calibração dos pesos das diferentes “soft constraints” incluídas no *Algoritmo Genético*. Pode ser realizado um estudo que defina conjuntos de pesos óptimos que permitam ao utilizador (elaborador dos horários) definir quais destas restrições devem ser mais relevantes para o escalonamento que pretende calcular.

Como nota final, é importante referir que foi muito importante para o autor todo o trabalho intelectual que este estudo exigiu. A percepção e compreensão de problemas na área de Investigação Operacional foram muito desenvolvidas e trabalhadas ao longo da dissertação, especialmente na pesquisa bibliográfica, que, além de estudos sobre *Nurse Rostering*, também incluiu estudos realizados noutras áreas de escalonamento de pessoal, aqui não referidos por não se enquadrarem no âmbito da dissertação.

Bibliografia

- [1] *E. K. Burke, P. De Causmaecker e G. Vanden Berghe*, “A Hybrid Tabu Search Algorithm for the *Nurse Rostering* Problem”, *Simulated Evolution and Learning 199, Lecture Notes in Artificial Intelligence*, **1585**: 187–194, 1999
- [2] *E. K. Burke, P. De Causmaecker, S. Petrovic e G. Vanden Berghe*, “Variable Neighbourhood Search for *Nurse Rostering* Problems”, submitted to MIC, 2001
- [3] *E. K. Burke, P. De Causmaecker, S. Petrovic e G. Vanden Berghe*, “Fitness Evaluation for *Nurse Scheduling* Problems”, *Proceedings of CEC2001, Seoul, IEEE Press*, 1139–1146, 2001
- [4] *E. K. Burke e G. Vanden Berghe*, “A Memetic Approach to the *Nurse Rostering* Problem”, *Applied Intelligence soecial issue on Simulated Evolution and Learning, Vol. 15, 3*: 199–214, 2001
- [5] *E. K. Burke, P. De Causmaecker, S. Petrovic e G. Vanden Berghe*, “A Multi Criteria Meta-Heuristic Approach to *Nurse Rostering*”, *Proceedings of Congress on Evolutionary Computation, CEC2002, Honolulu, IEEE Press*, 1197–1202, 2002
- [6] *Greet Vanden Berghe*, “Soft Constraints for the *Nurse Rostering* Problem”, <http://www.cs.nott.ac.uk/~gvb/constraints.ps>, 2001
- [7] *M. Moz e M. Vaz Pato*, “An Integer Multicommodity Flow Model Applied to the Rerostering of Nurse Schedules”, *Annals of Operations Research*, **119**: 285–301, 2003
- [8] *M. Moz e M. Vaz Pato*, “A genetic algorithm approach to a nurse rostering problem”, *Computers & Operations Research*, **34**: 667–691, 2007

- [9] *B. Cheang, H. Li, A. Lim e B. Rodrigues*, “Nurse Rostering problems – a bibliographic survey”, *European Journal of Operational Research*, **151**: 447–460, 2003
- [10] *K. A. Dowsland*, “Nurse Scheduling with a tabu search and strategic oscillation”, *European Journal of Operations Research*, **106**: 393–407, 1998
- [11] *K. A. Dowsland e J. M. Thompson*, “Solving a Nurse Scheduling Problem with knapsacks, networks and tabu search”, *Journal of Operational Research Society*, **51**: 825–833, 2000
- [12] *U. Aickelin e K. A. Dowsland*, “Exploiting problem structure in a Genetic Algorithm approach to a Nurse Rostering Problem”, *Journal of Scheduling*, **3**: 139–153, 2000
- [13] *U. Aickelin e K. A. Dowsland*, “An indirect Genetic Algorithm for a Nurse Scheduling Problem”, *Computers and Operations Research*, **31**: 761–778, 2004
- [14] *Edmund K. Burke et al.*, “The State of Art of Nurse Rostering”, *Journal of Shceduling*, **7**: 441–499, 2004
- [15] *B. Jaumard, F. Semet e T. Vovor*, “A Generalized Linear Programming Model for Nurse Rostering”, *European Journal of Operational Research*, **107**: 1–18,1998
- [16] *A. Meisels e A. Schaerf*, “Modelling and Solving *Employee Timetabling Problems*”, *Annals of Mathematics and Artificial Intelligence*, **39**: 41–59, 2003
- [17] *A. T. Ernst, H. Jiang, M. Krishnamoorthy e D. Sier*, “Staff Scheduling and Rostering: A Review of Applications, Methods and Models”, *European Journal of Operational Research*, **153**: 3–27, 2004
- [18] *Saul I. Glass e Carl M. Harris*, “Encyclopedia of Operations Research and Management Science”, Kluwer Academic Publishers, 1996
- [19] *Brant E. Fries*, “Bibliography of Operations Research in Health-Care Systems”, *Operations Research*, Vol. **24**, N^o **5**: 801–814, 1976
- [20] *Lawrence Davis*, “Handbook of Genetic Algorithms”, Van Nostrand Reinhold, New York, 1991

- [21] *M. Pirlot*, “General Local Search Heuristics in Combinatorial Optimization: a tutorial”, *Belgian Journal of Operations Research, Statistics and Computer Science*, **32**: 7–67, ...
- [22] *Marek Obitko*, “Introduction to Genetic Algorithms”, <http://cs.felk.cvut.cz/~xobitko/ga/index.html>, at Hochschule für Technik und Wirtschaft Dresden (University of Applied Sciences), 1998
- [23] *Paul C. H. Chu*, “A Genetic Algorithm Approach for Combinatorial Optimization Problems”, Thesis for the degree of PhD, University of London, and the Diploma of the Imperial College, January 1997
- [24] *J. E. Beasley*, “OR-Library”, <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>
- [25] *José Coelho Rodrigues*, “Sistema de Apoio à elaboração de horários em Unidades de Cuidados Continuados – Relatório do Trabalho de Fim de Curso”, Julho de 2007
- [26] “CPLEX LP Files”, in <http://lpsolve.sourceforge.net/5.5/CPLEX-format.html>
- [27] “LP File Format”, in <http://lpsolve.sourceforge.net/5.5/lp-format.html>

Apêndice A

Glossário

Durante o Projecto Chronos, para garantir o bom entendimento por todas as partes e a coerência da linguagem utilizada ao longo do trabalho desenvolvido para a empresa *Carlton Life*, construiu-se o seguinte glossário:

- **Unidade de Negócio** — Estrutura física que agrega todas as valências necessárias ao bom funcionamento de uma unidade de cuidados continuados;
- **Unidade Residencial** — Sub-unidade de uma *Unidade de Negócio*;
- **Unidade de Dia** — Sub-unidade de uma *Unidade de Negócio*;
- **Unidade de Apoio** — Sub-unidade de uma *Unidade de Negócio*;
- **Unidade de Diagnóstico** — Unidade que compõe uma *Unidade de Negócio*;
- **Núcleo** — Subdivisão de uma *Unidade Residencial*;
- **Sub Núcleo** — Subdivisão de um *Núcleo*;
- **Cuidador** — Todos os funcionários da *Carlton Life*;
- **Cliente** — Pessoa a quem está a ser prestado o serviço (utilizadores dos serviços de uma *Unidade de Negócio* da *Carlton Life*);
- **Produto Nuclear** — Serviço básico prestado por um ou mais *Cuidadores* a um ou mais *Clientes*;
- **Intensidade de um *Produto Nuclear*** — Quantidade de minutos dispendidos por um *Cuidador* a um *Cliente* na prestação de um serviço associado a um *Produto Nuclear*;

-
- **Produto** — Conjunto de *Produtos Nucleares*, com intensidades pré-definidas, prestados pelos *Cuidadores* a *Cientes*.

Apêndice B

Ficheiro de Criação dos Cenários de Teste

Neste anexo estão apresentadas todas as folhas do ficheiro MS Excel que serviu para a criação dos cenários de teste para este problema. Cada figura apresenta uma folha de cálculo.

Na Figura B.2, além das tabelas a preencher com os valores dos parâmetros, está definido o que representa cada parâmetro e apresentada uma estimativa da quantidade de enfermeiros a trabalhar por dia, dependente dos parâmetros aí definidos.

Na Figura B.6 existem três matrizes: uma primeira relativa às necessidades do primeiro turno de cada dia do horizonte temporal (manhã), a segunda relativa ao segundo turno (tarde) e a última relativa ao terceiro turno (noite).

Existem três folhas de cálculo como a da Figura B.7 no ficheiro MS Excel. A primeira diz respeito ao primeiro turno de cada dia (manhã), a segunda ao segundo turno de cada dia (tarde) e a terceira ao último turno (noite).

Os resultados exibidos na Figura B.8 só são apresentados após se carregar no botão “Ler Dados” na folha “Índices”. Nesta folha, além de aparecerem três matrizes com as alocações de cada enfermeiro a cada dia da semana (uma matriz para cada turno – manhã, tarde e noite), em que as alocações são assinaladas com um 1 na célula correspondente à alocação e com esta célula preenchida a verde, aparecem também outros dados: o valor da Função Objectivo da solução, preenchido automaticamente, e células prontas para que o utilizador acrescente dados que possam ser importantes para analisar e comparar o resultado obtido com outros (como o tempo de execução, a memória utilizada e o número de linha e colunas do problema).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															

Índices	Quantidade
Enfermeiros	20
Dias	28
Turnos	3
Graus	3

e	d
t	g

a modificação deste valor exige uma alteração manual das matrizes com este índice
a modificação deste valor exige uma alteração manual das matrizes com este índice

Passos para a Criação do Problema	
1º	Preencha os índices acima
2º	Gere Matrizes
3º	Exporte matrizes
4º	Leia Dados

Gerar Matrizes
Exportar Matrizes
Ler Dados

Figura B.1: Folha de cálculo em que se definem os índices do cenário a testar e onde, após essa definição, se criam as matrizes de dados a preencher nas restantes folhas, se exportam os dados e se lê os resultados obtidos.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1																								
2																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
10																								
11																								
12																								
13																								
14																								
15																								
16																								
17																								
18																								
19																								
20																								
21																								
22																								
23																								
24																								
25																								
26																								
27																								
28																								
29																								
30																								
31																								
32																								
33																								
34																								
35																								
36																								
37																								
38																								
39																								
40																								
41																								
42																								
43																								
44																								
45																								
46																								
47																								
48																								

Figura B.2: Folha de cálculo em que se definem os parâmetros do cenário a testar.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															

Penalização		
C	Valor	Descrição
c1	60	custo de alocação de um enfermeiro de grau 1 a um turno
c2	45	custo de alocação de um enfermeiro de grau 2 a um turno
c3	30	custo de alocação de um enfermeiro de grau 3 a um turno
p	12	penalização por desrespeito de uma preferência
a12	15	custo de enfermeiro de grau 1 trabalhar em grau 2
a2	15	custo de enfermeiro de grau 2 trabalhar em grau 3
a13	20	custo de enfermeiro de grau 1 trabalhar em grau 3

 $a12 \leq a2 < a13$

Figura B.3: Folha de cálculo em que se definem os custos para o cenário a testar e onde as siglas que nomeiam os custos estão explicadas. .

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																

Figura B.4: Folha de cálculo em que se definem graus de cada enfermeiro contemplado por este problema.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1																															
2																															
3																															
4																															
5																															
6																															
7																															
8																															
9																															
10																															
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															
32																															
33																															
34																															
35																															
36																															
37																															
38																															
39																															
40																															
41																															
42																															
43																															
44																															
45																															
46																															
47																															
48																															
49																															
50																															
51																															
52																															
53																															
54																															
55																															
56																															
57																															
58																															
59																															
60																															
61																															
62																															
63																															

Figura B.6: Folha de cálculo em que se definem as necessidades de cada dia do horizonte temporal do problema.

[]

31 de Março de 2008

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD AE																							
1																							
2																							
3																							
4																							
5																							
6																							
7																							
8																							
9																							
10																							
11																							
12																							
13																							
14																							
15																							
16																							
17																							
18																							
19																							
20																							
21																							
22																							
23																							
24																							
25																							
26																							
27																							
28																							
29																							
30																							
31																							
32																							
33																							
34																							
35																							
36																							
37																							
38																							
39																							
40																							
41																							
42																							
43																							
44																							
45																							
46																							
47																							
48																							
49																							
50																							
51																							
52																							
53																							
54																							
55																							
56																							
57																							
58																							
59																							
60																							
61																							
62																							
63																							
64																							
65																							
66																							
67																							
68																							
69																							
70																							
71																							
72																							
73																							
74																							
75																							
76																							
77																							
78																							
79																							
80																							
81																							
82																							
83																							
84																							
85																							
86																							
87																							
88																							
89																							
90																							
91																							
92																							
93																							
94																							
95																							
96																							
97																							
98																							
99																							
100																							

Figura B.8: Folha de cálculo onde são apresentados os resultados do problema.

Apêndice C

Resultados Obtidos com o “Modelo Grande”

C.1 Escalonamentos para o “Cenário 1”

Todos os escalonamentos obtidos para os testes efectuados, relativos ao “Cenário 1”, e que não estão apresentados nem comentados directamente ao longo dos capítulos deste trabalho, são exibidos neste anexo. Estes são escalonamentos do “Modelo Grande”. O primeiro escalonamento apresentado em cada figura é o escalonamento da solução obtida pelo “glpk” corrido num computador pessoal e o segundo o escalonamento da solução óptima obtida através do CPLEX, corrido num computador de maior capacidade.

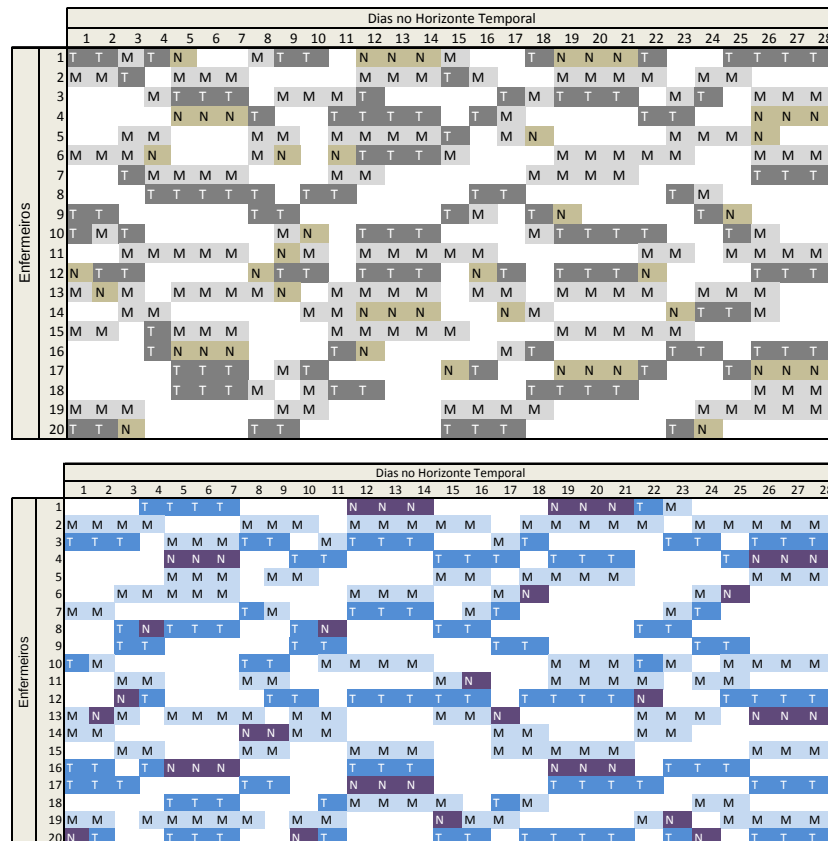


Figura C.1: Escalonamentos obtidos para o teste 1.7. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

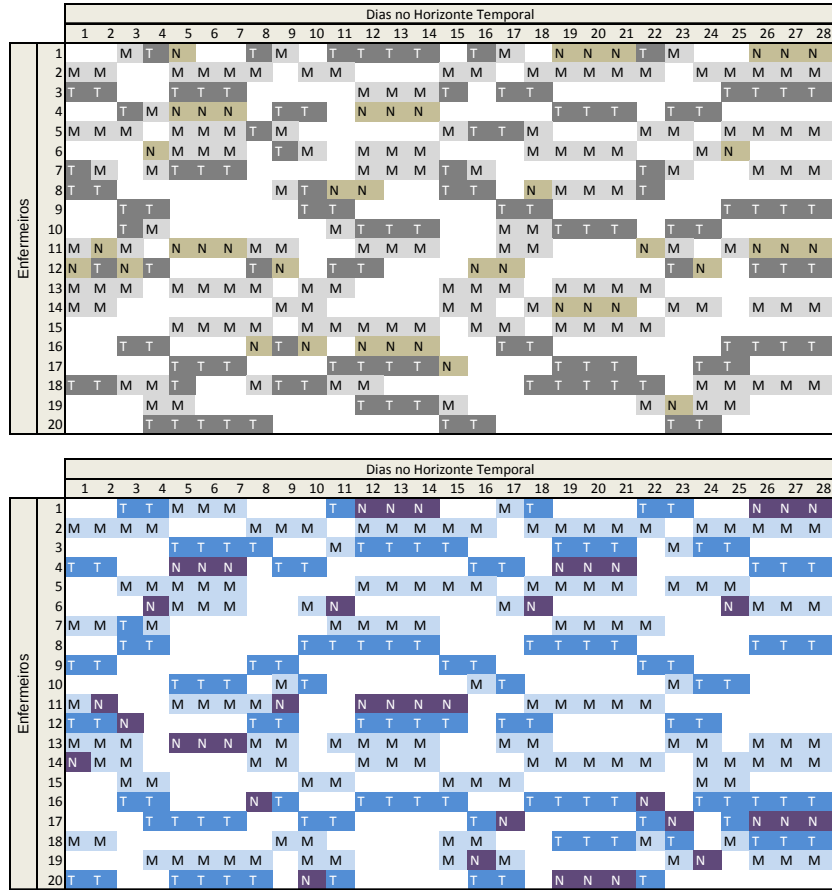


Figura C.2: Escalonamentos obtidos para o teste 1.8. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

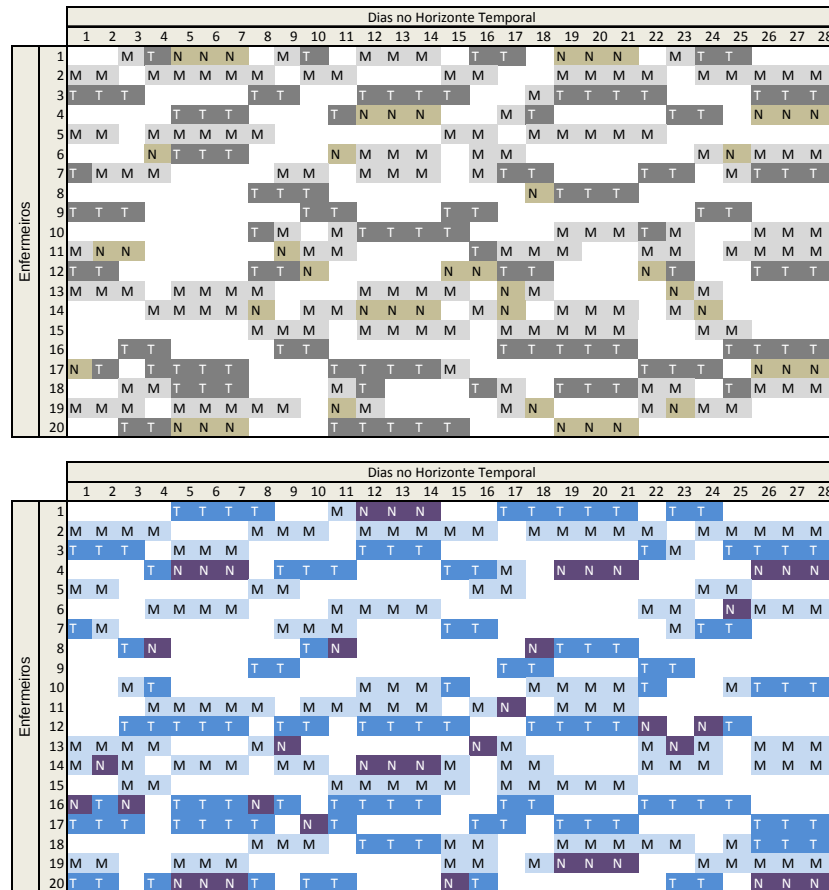


Figura C.3: Escalonamentos obtidos para o teste 1.9. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

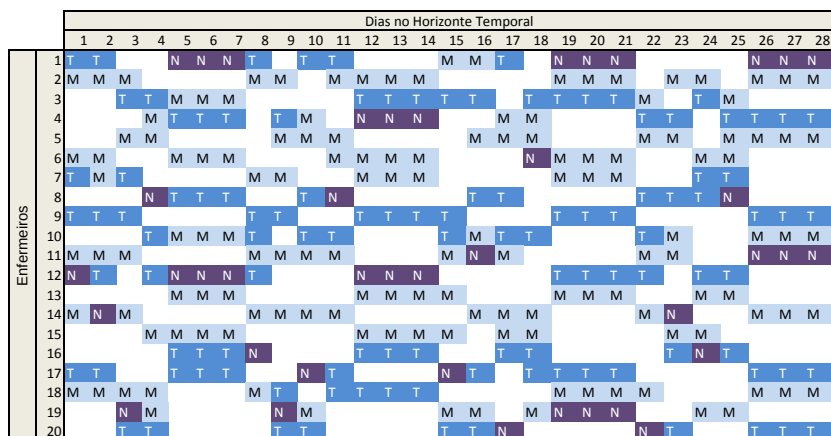


Figura C.4: Escalonamentos obtidos para o teste 1.11. Unicamente apresentado o escalonamento óptimo, obtido com o CPLEX, por não ter sido alcançada qualquer solução com o “glpk”.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

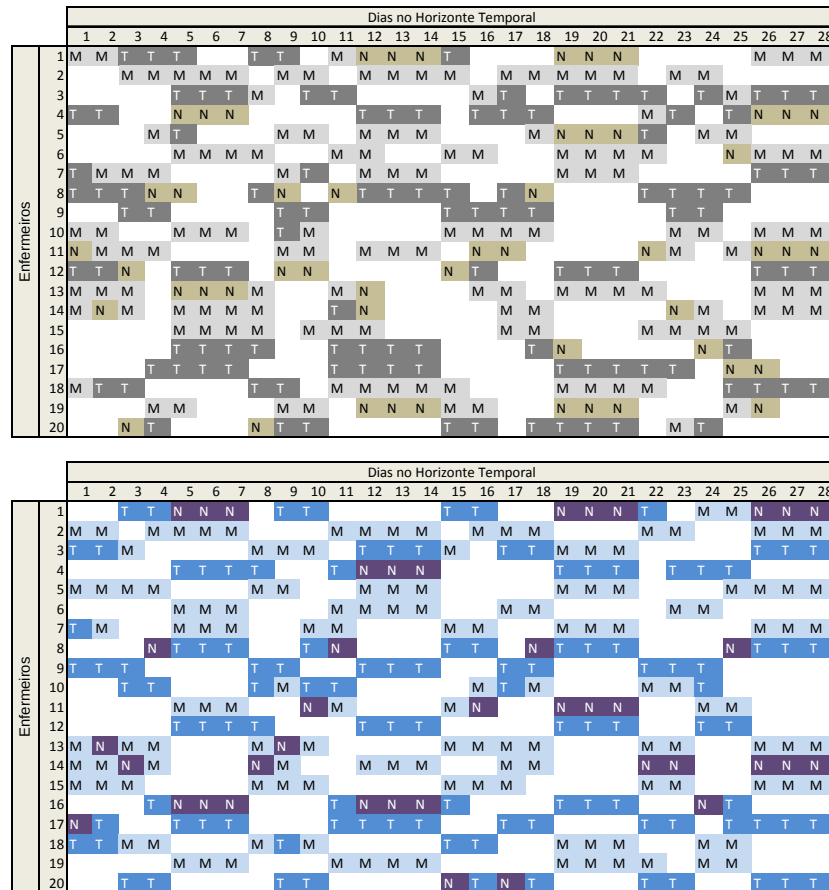


Figura C.5: Escalonamentos obtidos para o teste 1.12. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

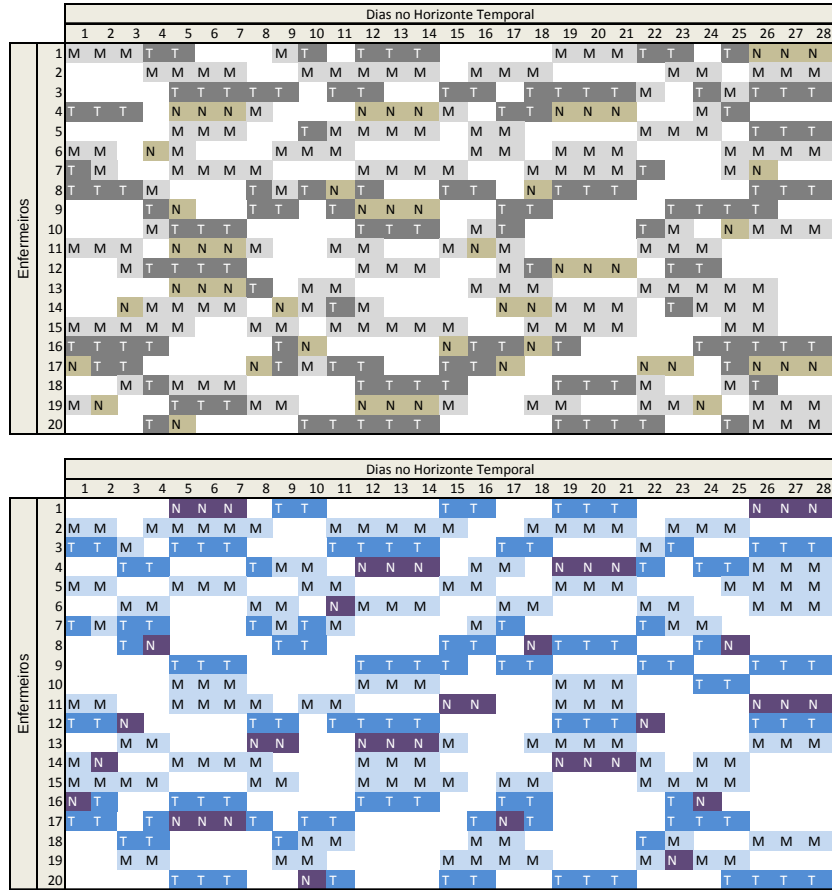


Figura C.6: Escalonamentos obtidos para o teste 1.13. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

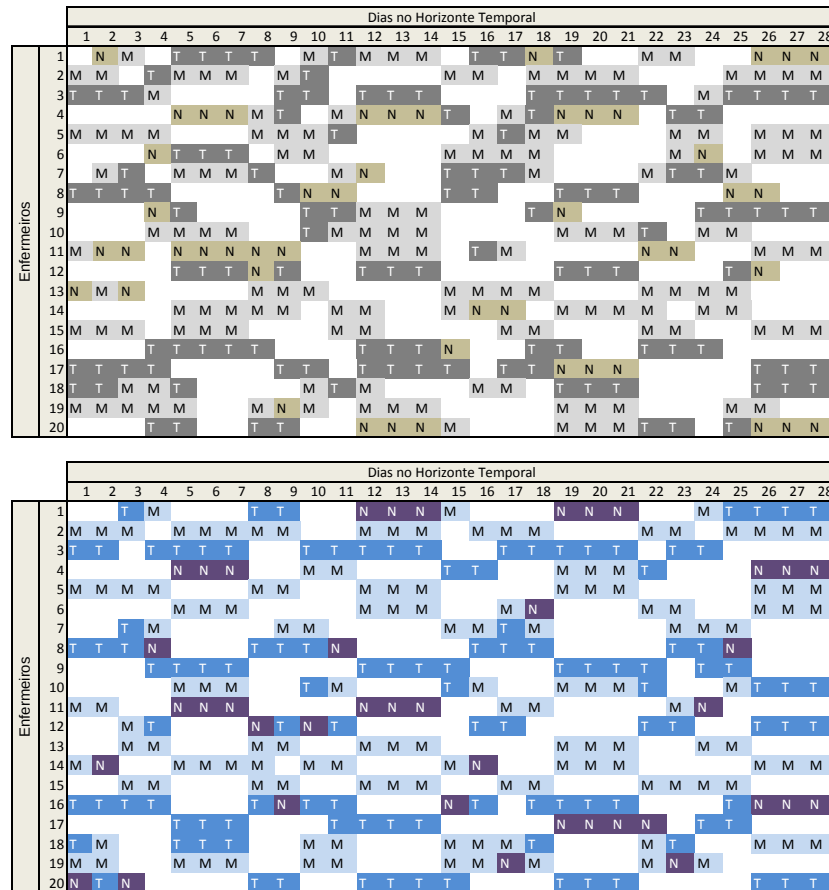


Figura C.7: Escalonamentos obtidos para o teste 1.14. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

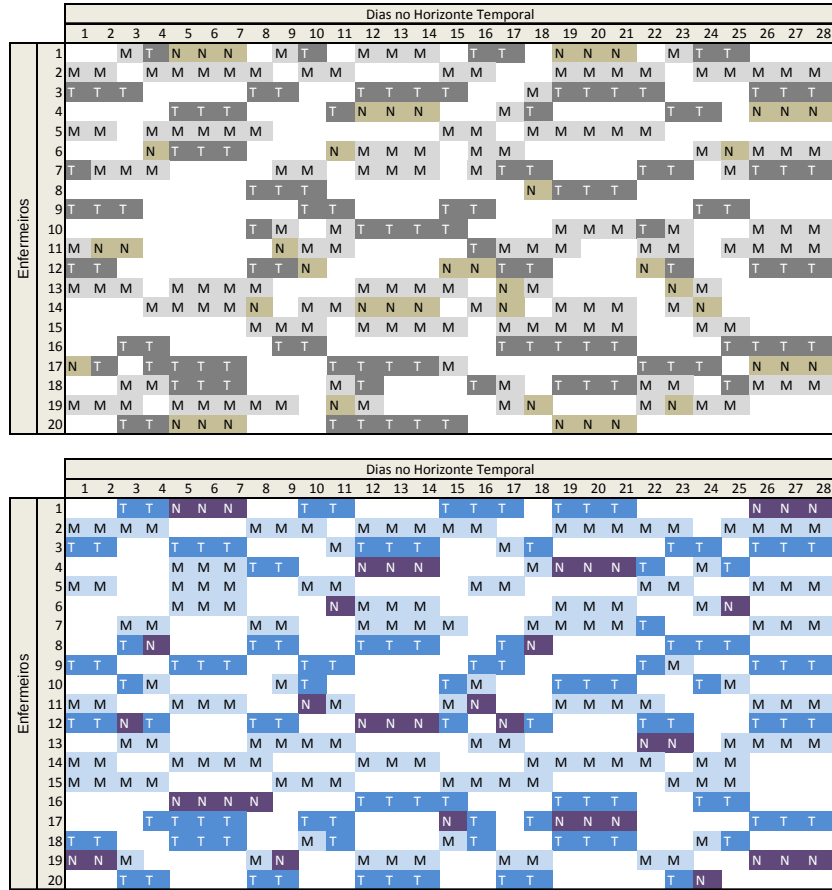


Figura C.8: Escalonamentos obtidos para o teste 1.15. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

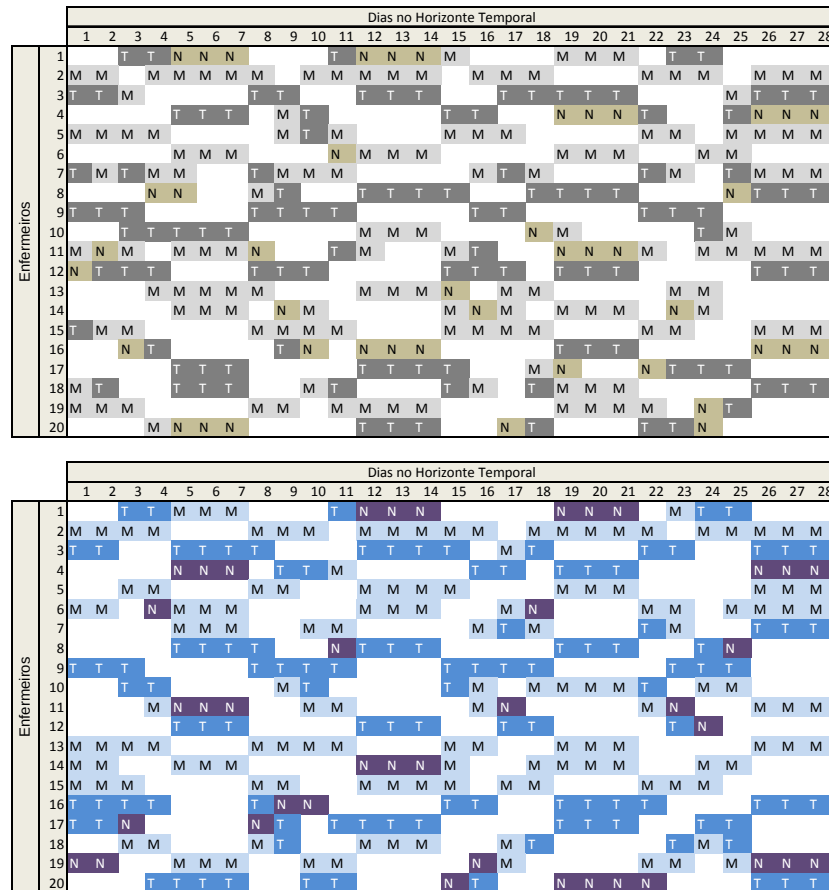


Figura C.9: Escalonamentos obtidos para o teste 1.16. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

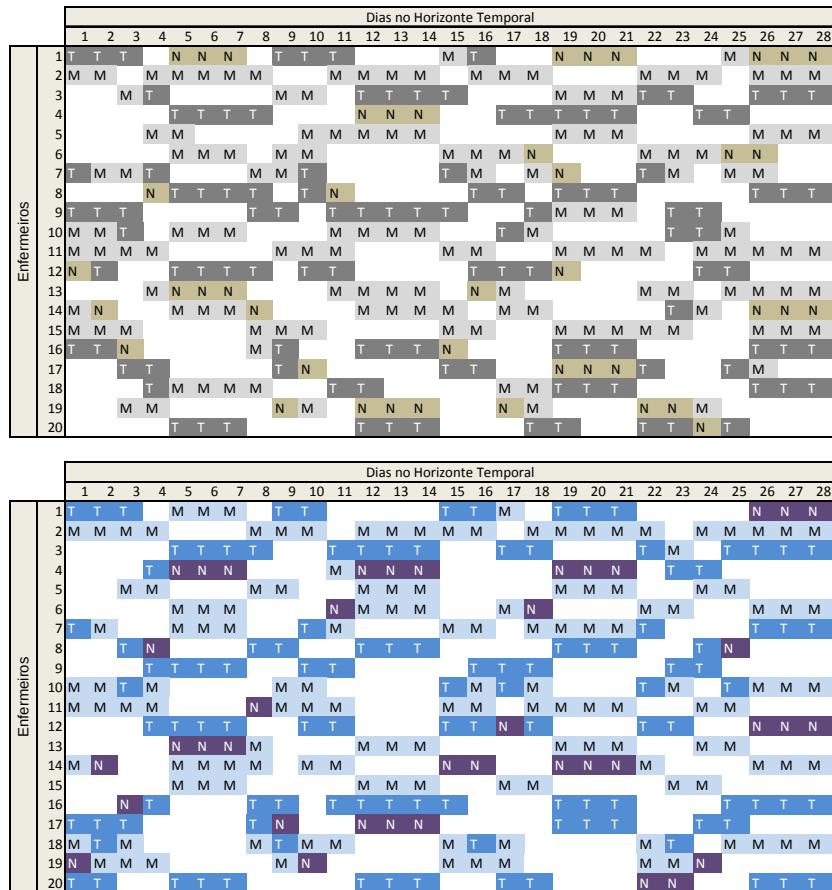


Figura C.10: Escalonamentos obtidos para o teste 1.17. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.1. ESCALONAMENTOS PARA O “CENÁRIO 1”

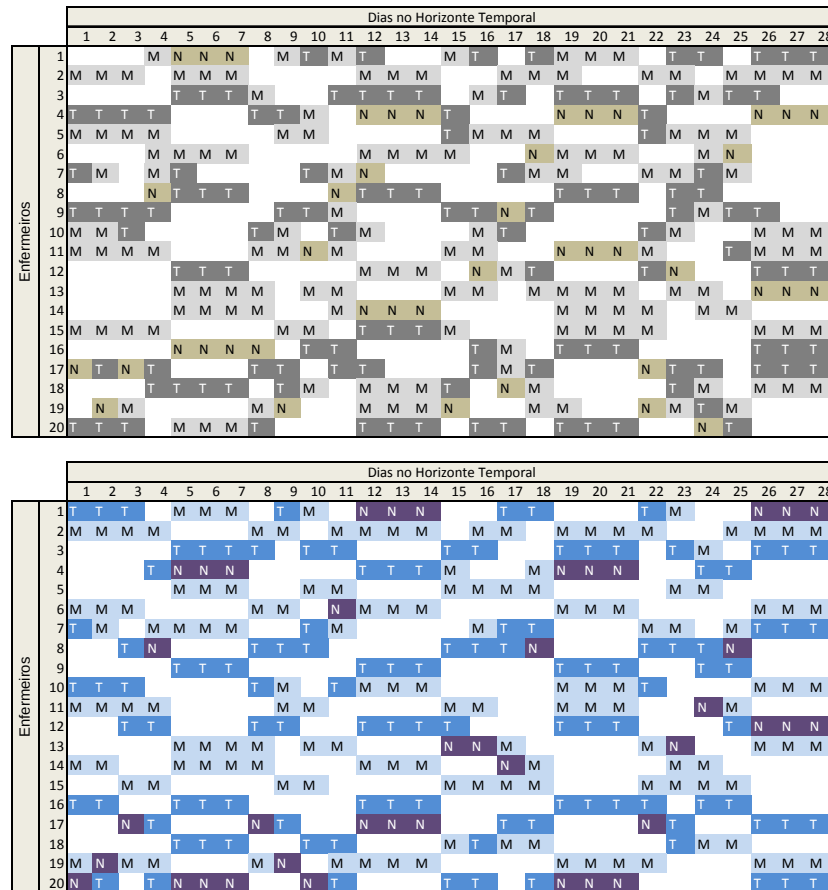


Figura C.11: Escalonamentos obtidos para o teste 1.20. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.2 Escalonamentos para o “Cenário 4”

Todos os escalonamentos obtidos para os testes efectuados, relativos ao “Cenário 4”, e que não estão apresentados nem comentados directamente ao longo dos capítulos deste trabalho, são exibidos neste anexo. Estes são escalonamentos do “Modelo Grande”. O primeiro escalonamento apresentado em cada figura é o escalonamento da solução obtida pelo “glpk” corrido num computador pessoal e o segundo, o escalonamento da solução óptima obtida através do CPLEX, corrido num computador de maior capacidade.

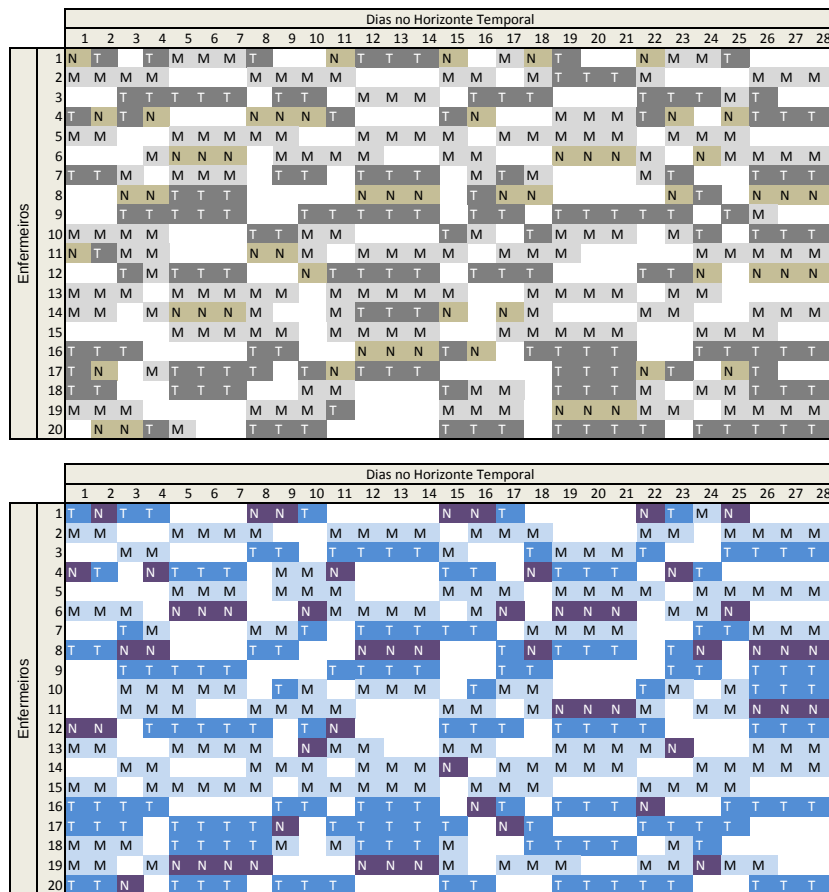


Figura C.12: Escalonamentos obtidos para o teste 2.9. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.2. ESCALONAMENTOS PARA O “CENÁRIO 4”

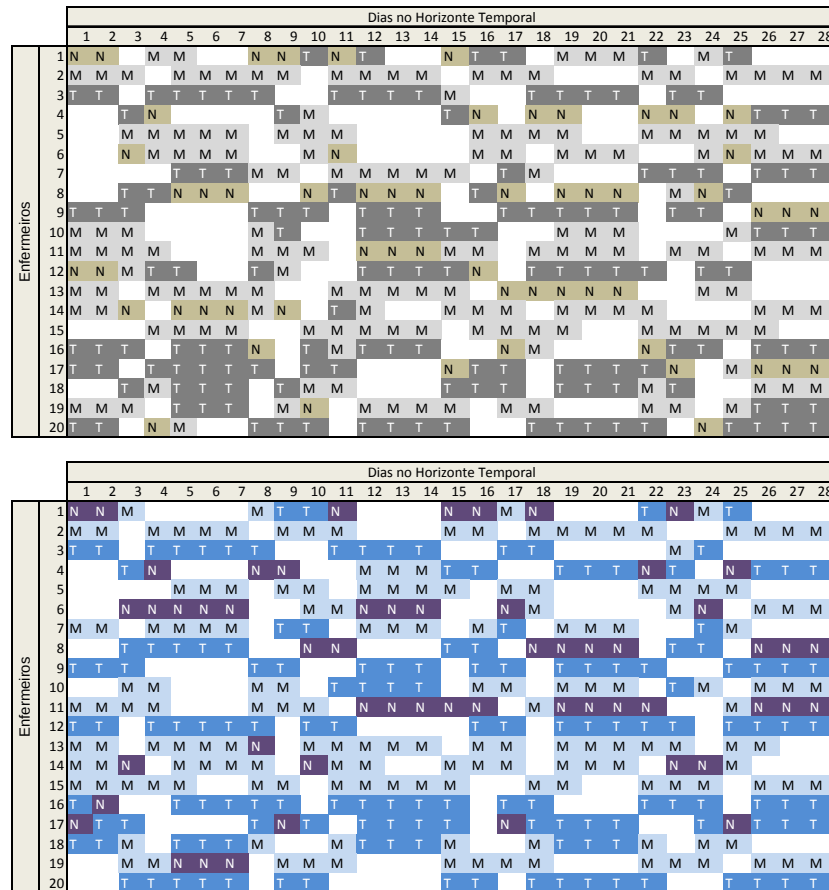


Figura C.13: Escalonamentos obtidos para o teste 2.10. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.2. ESCALONAMENTOS PARA O “CENÁRIO 4”

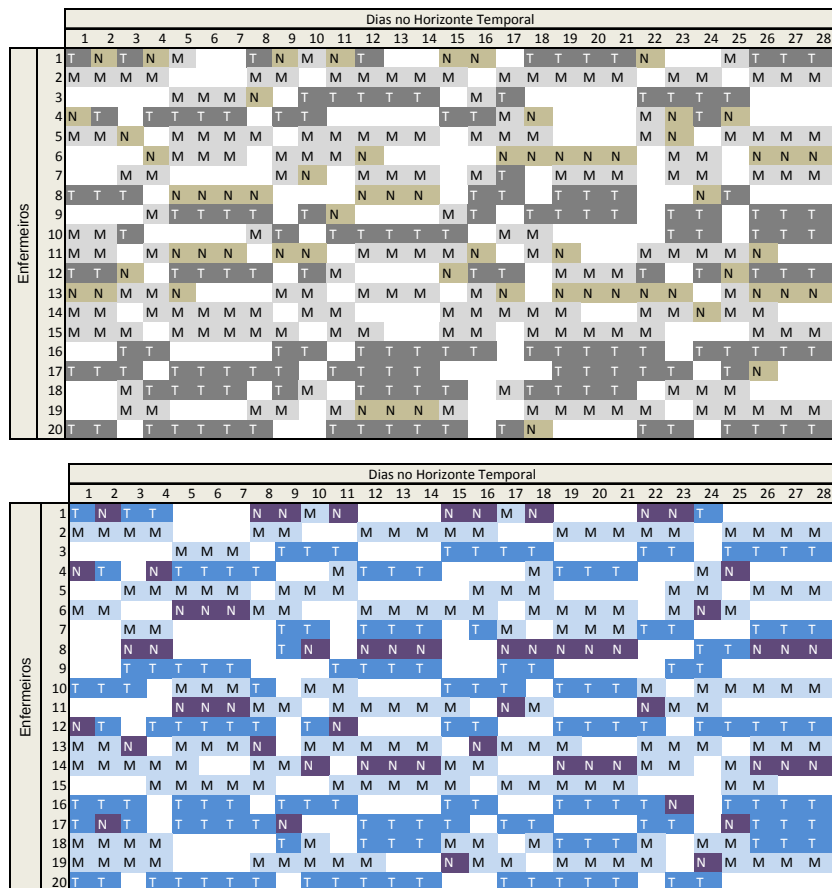


Figura C.14: Escalonamentos obtidos para o teste 2.11. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.2. ESCALONAMENTOS PARA O “CENÁRIO 4”

		Dias no Horizonte Temporal																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
Enfermeiros	1	T	T	T	N				N	T		N	M	M	M		T	M	T				N	N	M					
	2	M	M		M	M	M	M		M	M	M				M	M			M	M	M	M			M	M	M	M	
	3																													
	4	N	N	M	T				T	N	T	T				N	N	T	N					T	T	N	T	T	T	
	5	M	M	M					M			M	M	M	M					M	M	M	M		M	M	M			
	6				N	M	M	M	M		M	N		M	M	M	M			N	N	N	N		N	N				
	7				M	T	T	T		T	M	T				T	M			M	M	M	M					T	T	T
	8	T	T	N		N	N	N				N	N	N	N		T	N	T					T	T	T		N	N	N
	9				T	T	T	T		T	T					T	T			T	T	T		T	T	T				
	10				T	M				T	M		M	T	T	T		M	M			T	T	T	M			M	M	M
	11	M	M	N								M	N	N	N	N		M	M			N	N	N	M		M	M	M	M
	12	T	N	T		T	T	T	T		T	T				T	N	T				T	T	T		T	T	T	T	T
	13	N	M	M	M				M	N	N		M	M	M	M		M	M	M			M	M		M	N	N	N	N
	14				M	M	M		N	M	M		M	M	M	M							M	M		M	M	M	M	M
	15	M	M	M		M	M	M	M		M	M				M	M			M	M	M	M		M	M		M	M	M
	16	T	T	T		T	T	T	T		T	T	T	T		T	T			T	T	T	T		N	T				
	17	T	T	T	T				T	T	T		T	T	T	T								N	T		T	T	T	T
	18	M	T	M	M				M	M	M	M				T	T		M		T	T	T	T	T	M		T	T	T
	19	M	M		M	N	N	N				M	M	M	M		M	M			M	M	M	M		M	M	M	M	M
	20				T	T	T	T		T	T		T	T	T	N		N	T				N	T	T		T	T	T	T

Figura C.15: Escalonamentos obtidos para o teste 2.13. Unicamente apresentado o escalonamento óptimo, obtido com o CPLEX, por não ter sido alcançada qualquer solução com o “glpk”.

C.3 Escalonamentos para Cenários com Diferentes Necessidades

Todos os escalonamentos obtidos para os testes efectuados, relativos aos cenários com diferentes necessidades, e que não estão apresentados nem comentados directamente ao longo dos capítulos deste trabalho, são exibidos neste anexo. Estes são escalonamentos do “Modelo Grande”. O primeiro escalonamento apresentado em cada figura é o escalonamento da solução obtida pelo “glpk” corrido num computador pessoal e o segundo, o escalonamento da solução óptima obtida através do CPLEX, corrido num computador de maior capacidade.

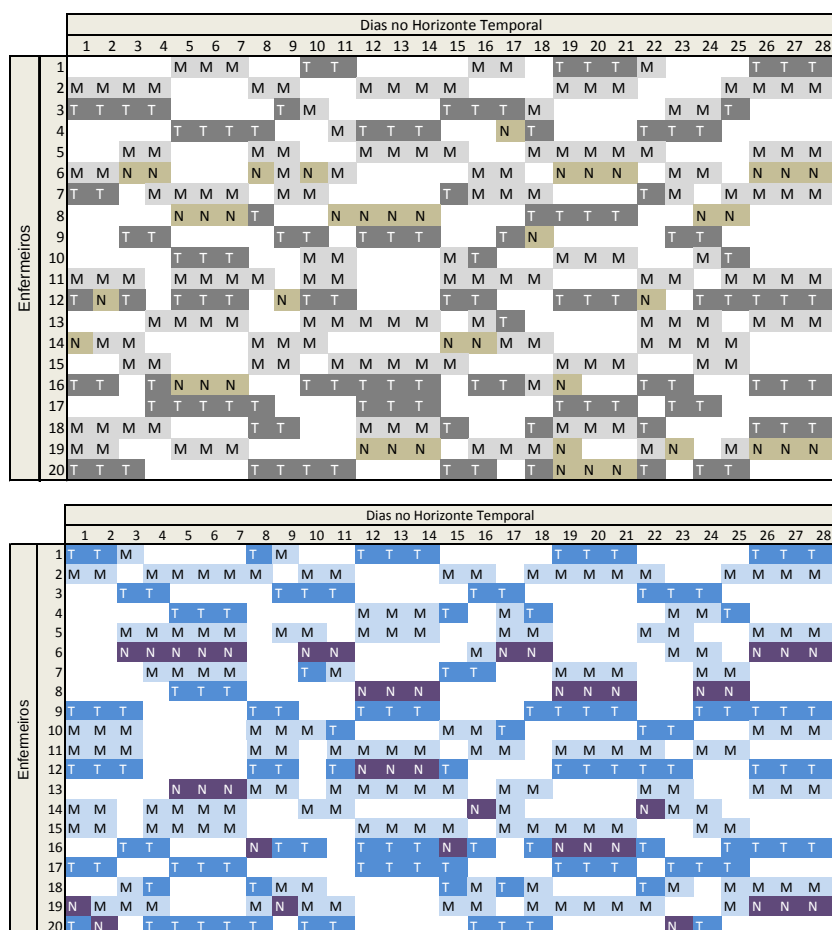


Figura C.16: Escalonamentos obtidos para o teste 3.2. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.3. ESCALONAMENTOS PARA CENÁRIOS COM DIFERENTES NECESSIDADES

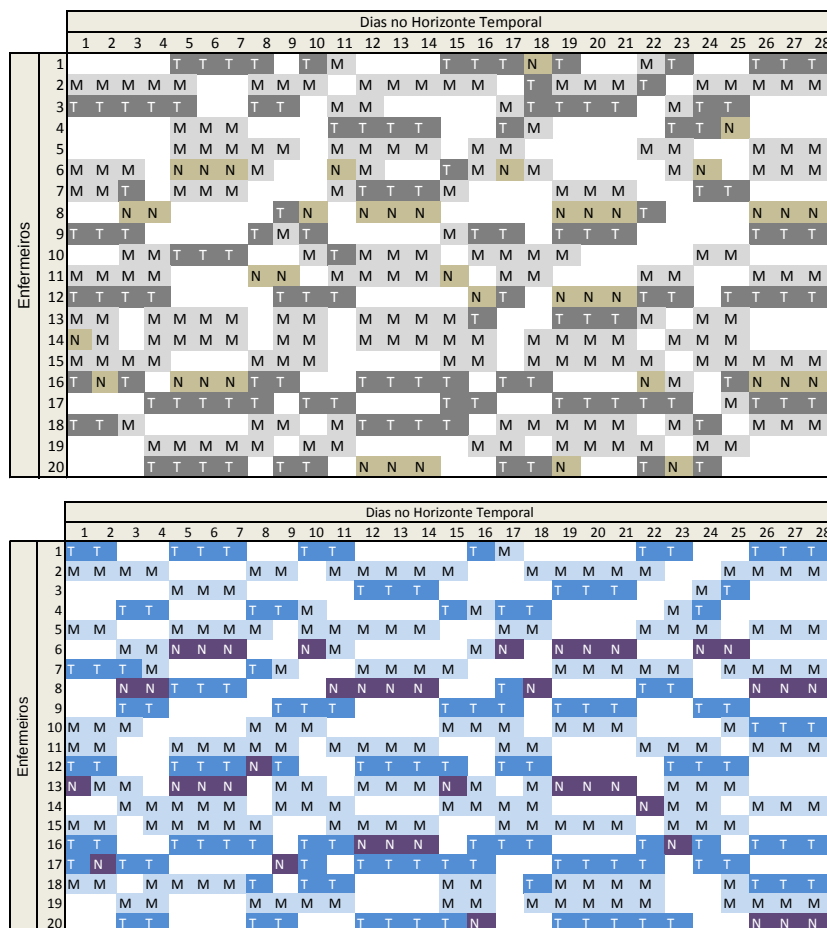


Figura C.17: Escalonamentos obtidos para o teste 3.3. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.3. ESCALONAMENTOS PARA CENÁRIOS COM DIFERENTES NECESSIDADES

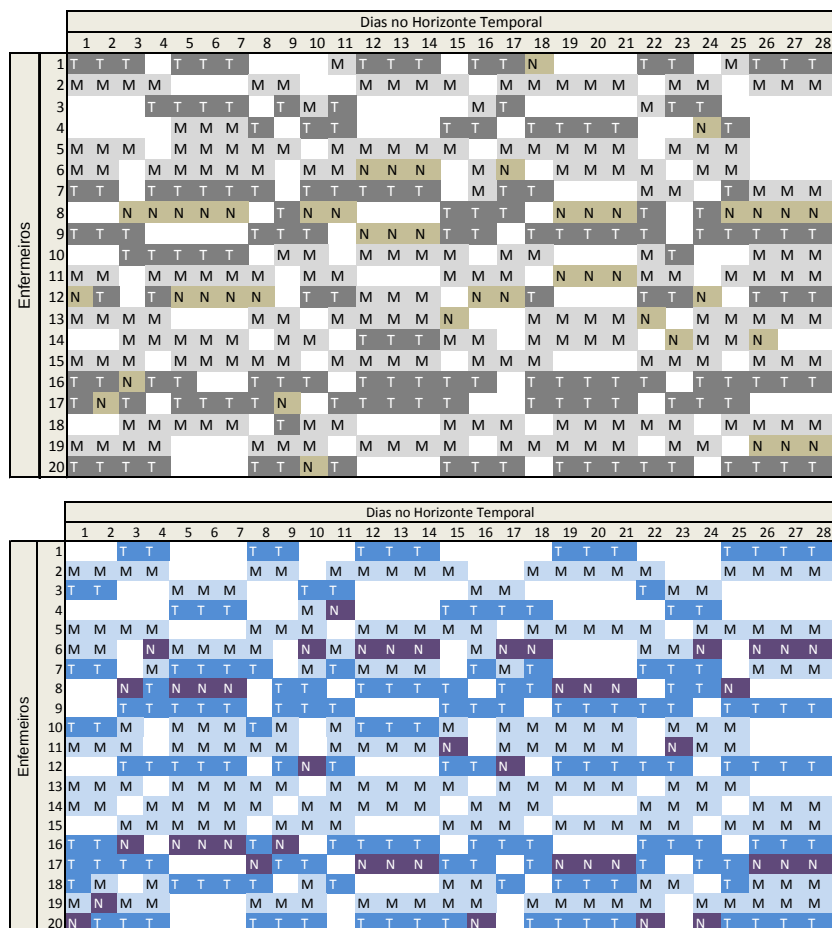


Figura C.18: Escalonamentos obtidos para o teste 3.5. Em cima o escalonamento obtido com o “glpk” e em baixo o escalonamento óptimo, obtido com o CPLEX.

C.4 Dias de Trabalho

Neste anexo está apresentada a tabela que compara os dias que cada enfermeiro trabalha nos cenários com necessidades diárias totais diferentes – Tabela C.1. Esta tabela encontra-se em anexo por não ter sido relevante para a análise dos resultados apresentada anteriormente.

Enf	3.1		3.2		3.3		3.4		3.5		3.6	3.7
	glpk	opt	glpk	opt	glpk	opt	glpk	opt	glpk	opt	opt	opt
1	20	16	14	14	16	14	18	16	19	14	20	20
2	20	22	17	20	23	20	20	20	20	20	22	22
3	17	16	13	10	17	11	15	15	12	12	21	21
4	18	14	13	12	12	11	15	17	14	11	21	21
5	18	14	16	17	16	19	17	18	21	22	22	21
6	17	13	18	15	18	14	18	19	20	21	20	22
7	15	15	18	13	16	19	21	16	21	21	20	21
8	16	13	14	11	14	16	17	16	20	19	21	20
9	13	11	11	17	15	13	18	16	21	20	21	20
10	14	14	12	15	16	16	16	17	18	21	21	21
11	16	15	19	17	18	19	19	21	21	21	21	21
12	20	12	20	18	18	16	18	21	21	20	21	21
13	17	10	17	17	19	20	20	19	21	21	20	22
14	14	16	14	13	20	18	19	20	20	21	21	21
15	13	15	14	17	19	19	22	20	21	20	21	21
16	15	12	20	19	20	20	19	20	23	21	22	21
17	17	13	13	15	18	17	18	19	20	22	22	22
18	14	19	18	15	20	20	19	22	20	21	20	20
19	15	14	18	19	15	17	22	21	21	22	22	20
20	15	18	16	14	15	17	19	15	20	22	21	22
Méd	16.2	14.6	15.75	15.4	17.25	16.8	18.5	18.4	19.7	19.6	21	21

Tabela C.1: Número de dias que cada enfermeiro trabalha, durante todo o horizonte temporal, para todas as soluções obtidas, relativas aos cenários com diferentes necessidades.

Apêndice D

Anexos à Implementação do *Algoritmo Genético*

D.1 Novas Folhas no Ficheiro de Criação dos Cenários de Teste

Aqui estão apresentadas as folhas adicionadas ao ficheiro MS Excel de criação de cenários de teste para o problema de *Nurse Rostering*, necessárias para definir os novos dados relativos às restrições adicionadas com a implementação do *Algoritmo Genético*.

31 de Março de 2008

223

Figura D.1: Nova folha de cálculo em que se definem os parâmetros do cenário a testar com o *Algoritmo Genético*. Esta folha inclui o novo parâmetro contemplado: *Mnlc* – Mínimo de dias livres consecutivos permitidos.

D.1. NOVAS FOLHAS NO FICHEIRO DE CRIAÇÃO DOS CENÁRIOS DE TESTE

	Dt+1	Dt+2	Dt+3	Dt+4	Dt+5	Dt+6	Dt+7	Dt+8	Dt+9	Dt+10	Dt+11	Dt+12	Dt+13	Dt+14	Dt+15	Dt+16	Dt+17	Dt+18	Dt+19	Dt+20	Dt+21	Dt+22
Enr1																						
Enr2																						
Enr3																						
Enr4																						
Enr5																						
Enr6																						
Enr7																						
Enr8																						
Enr9																						
Enr10																						
Enr11																						
Enr12																						
Enr13																						
Enr14																						
Enr15																						
Enr16																						
Enr17																						
Enr18																						
Enr19																						
Enr20																						

Legenda:
 não está disponível para trabalhar
 está disponível para trabalhar

Ribbon: Parametros / Custos / Graus / Fds / Necessidades / Preferencias_1 / Preferencias_2 / Disponibilidade / Contas

Figura D.2: Nova folha de cálculo em que se definem as disponibilidades dos enfermeiros para trabalhar em cada dia do horizonte temporal. Esta folha foi concebida para os dados relativos à restrição de disponibilidades dos enfermeiros.

D.1. NOVAS FOLHAS NO FICHEIRO DE CRIAÇÃO DOS CENÁRIOS DE TESTE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1																							
2																							
3																							
4																							
5																							
6																							
7																							
8																							
9																							
10																							
11																							
12																							
13																							
14																							
15																							
16																							
17																							
18																							
19																							
20																							
21																							
22																							
23																							
24																							
25																							
26																							
27																							
28																							
29																							
30																							
31																							
32																							
33																							
34																							
35																							
36																							
37																							
38																							
39																							
40																							
41																							
42																							
43																							
44																							
45																							
46																							
47																							
48																							

Legenda:

- 1: não haver contacto
- 0: indiferente
- 1: haver contacto

NOTA: Matriz triangular superior

Graus / Fds / Necessidades / Preferências :1 / Preferências :2 / Preferências :3 / Disponibilidade / Contacto enfermeiros / Cont / 1 / III

Figura D.3: Nova folha de cálculo em que se definem quais os contactos desejados (tutorias, por exemplo), os indesejados e os indiferentes entre os enfermeiros. Esta folha foi concebida para os dados relativos à restrição de contactos entre enfermeiros.

Legende:	
1	enfermiers embauchés du district
0	enfermiers d'embouchure au district

D.2 Um Escalonamento para o *Algoritmo Genético*

Um escalonamento obtido com o *Algoritmo Genético* e com os valores de pesos para as “soft constraints” recomendados (os da Tabela D.1) está apresentado na Figura D.5, com os respectivos valores de penalizações e custos associados registados na Tabela D.2.

f.d.s.	folgas	maxa	mxs	mlc	equil.	pref.	cont_e	cont_d
30	20	10	40	40	40	30	1	1

Tabela D.1: Pesos para as restrições.

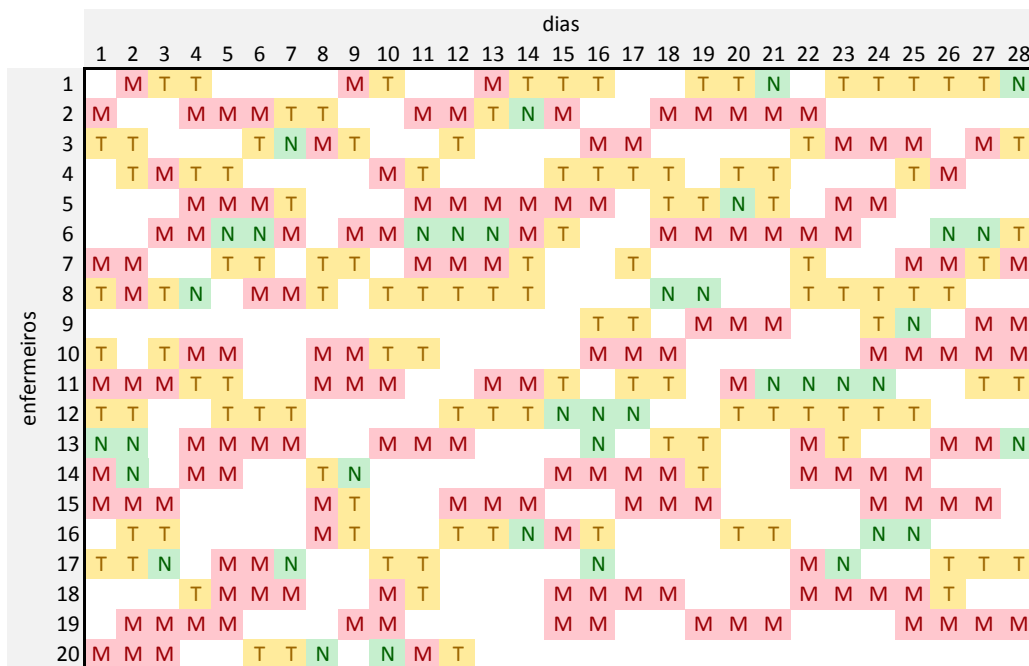


Figura D.5: Escalonamento obtido com o *Algoritmo Genético*, com definições de pesos das “soft constraints” como indicado na Tabela D.1.

D.2. UM ESCALONAMENTO PARA O ALGORITMO GENÉTICO

F.A.	custos	f.d.s.	folgas	maxa	mxo	mlc	equil.	pref.	cont_e	cont_d	S.C.
48033	54555	117	43	0	14	27	38	159	-420	-5358	6522

Tabela D.2: Valores dos custos e penalizações associados ao escalonamento da Figura D.5.

D.3 Populações Iniciais

Neste anexo estão expostos os valores das funções avaliadores dos cromossomas constituintes de duas populações iniciais do *Algoritmo Genético*, uma para o “**Cenário 1**” e outra para o cenário alargado, utilizados nos vários testes efectuados ao *Algoritmo Genético*.

Valores das funções avaliadores dos cromossomas constituintes de uma população inicial, para testes com o “**Cenário 1**”:

FA[1] = 16530	FA[48] = 16395	FA[95] = 16320	FA[142] = 16755	FA[189] = 17145
FA[2] = 16140	FA[49] = 16470	FA[96] = 17025	FA[143] = 16785	FA[190] = 16650
FA[3] = 17130	FA[50] = 16590	FA[97] = 16695	FA[144] = 16800	FA[191] = 16470
FA[4] = 16455	FA[51] = 16815	FA[98] = 16395	FA[145] = 16350	FA[192] = 16650
FA[5] = 16875	FA[52] = 16530	FA[99] = 17070	FA[146] = 16380	FA[193] = 16665
FA[6] = 16530	FA[53] = 16635	FA[100] = 16575	FA[147] = 16680	FA[194] = 17025
FA[7] = 16530	FA[54] = 17010	FA[101] = 16980	FA[148] = 16410	FA[195] = 16695
FA[8] = 16785	FA[55] = 17220	FA[102] = 16065	FA[149] = 17130	FA[196] = 16950
FA[9] = 16935	FA[56] = 16395	FA[103] = 16770	FA[150] = 16905	FA[197] = 17415
FA[10] = 16650	FA[57] = 17100	FA[104] = 16455	FA[151] = 17205	FA[198] = 16575
FA[11] = 16725	FA[58] = 16785	FA[105] = 16605	FA[152] = 16695	FA[199] = 16950
FA[12] = 17160	FA[59] = 16455	FA[106] = 16170	FA[153] = 16830	FA[200] = 16380
FA[13] = 16845	FA[60] = 16785	FA[107] = 16425	FA[154] = 17055	FA[201] = 16245
FA[14] = 17250	FA[61] = 16365	FA[108] = 16740	FA[155] = 16815	FA[202] = 16635
FA[15] = 16635	FA[62] = 16605	FA[109] = 16380	FA[156] = 16920	FA[203] = 16530
FA[16] = 16680	FA[63] = 17115	FA[110] = 17205	FA[157] = 16695	FA[204] = 16620
FA[17] = 16620	FA[64] = 17085	FA[111] = 16785	FA[158] = 16320	FA[205] = 17085
FA[18] = 16830	FA[65] = 16440	FA[112] = 16755	FA[159] = 16695	FA[206] = 16695
FA[19] = 16365	FA[66] = 16275	FA[113] = 16860	FA[160] = 16905	FA[207] = 16875
FA[20] = 16635	FA[67] = 16725	FA[114] = 16500	FA[161] = 16050	FA[208] = 16440
FA[21] = 16905	FA[68] = 17250	FA[115] = 16770	FA[162] = 16395	FA[209] = 16515
FA[22] = 17025	FA[69] = 16830	FA[116] = 16680	FA[163] = 17175	FA[210] = 16890
FA[23] = 16605	FA[70] = 16410	FA[117] = 16755	FA[164] = 16755	FA[211] = 16785
FA[24] = 16710	FA[71] = 16815	FA[118] = 16875	FA[165] = 16290	FA[212] = 17025
FA[25] = 16305	FA[72] = 16725	FA[119] = 16665	FA[166] = 17220	FA[213] = 16545
FA[26] = 16665	FA[73] = 17370	FA[120] = 16380	FA[167] = 16680	FA[214] = 16380
FA[27] = 16545	FA[74] = 17055	FA[121] = 17160	FA[168] = 17160	FA[215] = 16635
FA[28] = 17010	FA[75] = 16980	FA[122] = 16995	FA[169] = 17220	FA[216] = 16245
FA[29] = 16395	FA[76] = 16605	FA[123] = 16935	FA[170] = 16875	FA[217] = 16755
FA[30] = 17235	FA[77] = 16275	FA[124] = 17145	FA[171] = 16350	FA[218] = 16800
FA[31] = 16650	FA[78] = 16470	FA[125] = 17145	FA[172] = 16725	FA[219] = 16650
FA[32] = 16410	FA[79] = 16875	FA[126] = 16860	FA[173] = 16650	FA[220] = 16650
FA[33] = 16920	FA[80] = 16815	FA[127] = 16560	FA[174] = 16875	FA[221] = 16575
FA[34] = 16905	FA[81] = 16365	FA[128] = 17115	FA[175] = 16110	FA[222] = 16410
FA[35] = 16665	FA[82] = 16515	FA[129] = 16455	FA[176] = 16830	FA[223] = 16560
FA[36] = 16935	FA[83] = 16365	FA[130] = 16155	FA[177] = 16170	FA[224] = 16905
FA[37] = 16920	FA[84] = 16605	FA[131] = 16440	FA[178] = 16980	FA[225] = 16860
FA[38] = 16530	FA[85] = 17175	FA[132] = 16650	FA[179] = 16800	FA[226] = 16530
FA[39] = 16440	FA[86] = 16005	FA[133] = 16755	FA[180] = 16800	FA[227] = 16845
FA[40] = 16890	FA[87] = 16470	FA[134] = 17355	FA[181] = 16905	FA[228] = 16890
FA[41] = 16680	FA[88] = 17190	FA[135] = 16710	FA[182] = 16230	FA[229] = 16785
FA[42] = 16755	FA[89] = 16950	FA[136] = 16845	FA[183] = 16425	FA[230] = 16680
FA[43] = 17130	FA[90] = 16710	FA[137] = 16335	FA[184] = 16695	FA[231] = 16695
FA[44] = 16470	FA[91] = 16950	FA[138] = 16560	FA[185] = 16395	FA[232] = 16620
FA[45] = 16845	FA[92] = 16350	FA[139] = 17100	FA[186] = 17010	FA[233] = 16785
FA[46] = 16185	FA[93] = 16815	FA[140] = 17100	FA[187] = 16920	FA[234] = 17160
FA[47] = 16545	FA[94] = 16695	FA[141] = 17085	FA[188] = 16590	FA[235] = 16800

D.3. POPULAÇÕES INICIAIS

FA[236] = 16275	FA[290] = 16860	FA[344] = 16590	FA[398] = 16560	FA[452] = 17205
FA[237] = 16620	FA[291] = 16635	FA[345] = 16665	FA[399] = 16860	FA[453] = 16485
FA[238] = 17040	FA[292] = 16785	FA[346] = 17190	FA[400] = 16935	FA[454] = 16995
FA[239] = 16890	FA[293] = 16635	FA[347] = 16425	FA[401] = 16485	FA[455] = 16995
FA[240] = 16755	FA[294] = 16680	FA[348] = 17130	FA[402] = 16995	FA[456] = 16935
FA[241] = 16935	FA[295] = 16425	FA[349] = 16725	FA[403] = 16785	FA[457] = 16575
FA[242] = 16410	FA[296] = 16740	FA[350] = 16785	FA[404] = 16605	FA[458] = 16905
FA[243] = 16545	FA[297] = 16725	FA[351] = 17055	FA[405] = 16620	FA[459] = 16425
FA[244] = 16575	FA[298] = 16665	FA[352] = 16395	FA[406] = 16755	FA[460] = 16650
FA[245] = 16905	FA[299] = 17115	FA[353] = 16500	FA[407] = 16470	FA[461] = 16650
FA[246] = 17145	FA[300] = 16770	FA[354] = 16230	FA[408] = 16050	FA[462] = 16950
FA[247] = 16350	FA[301] = 16635	FA[355] = 17040	FA[409] = 17205	FA[463] = 16125
FA[248] = 16845	FA[302] = 16785	FA[356] = 16515	FA[410] = 17055	FA[464] = 16755
FA[249] = 16965	FA[303] = 16350	FA[357] = 16620	FA[411] = 16800	FA[465] = 16710
FA[250] = 16695	FA[304] = 16170	FA[358] = 16500	FA[412] = 16980	FA[466] = 17010
FA[251] = 17025	FA[305] = 16350	FA[359] = 16995	FA[413] = 16665	FA[467] = 16785
FA[252] = 16665	FA[306] = 16605	FA[360] = 16515	FA[414] = 16995	FA[468] = 17040
FA[253] = 16800	FA[307] = 17010	FA[361] = 16560	FA[415] = 16290	FA[469] = 16770
FA[254] = 16995	FA[308] = 16830	FA[362] = 16740	FA[416] = 16650	FA[470] = 17055
FA[255] = 17115	FA[309] = 16425	FA[363] = 16725	FA[417] = 16935	FA[471] = 16695
FA[256] = 16575	FA[310] = 16215	FA[364] = 16935	FA[418] = 16140	FA[472] = 16815
FA[257] = 16335	FA[311] = 16440	FA[365] = 16560	FA[419] = 16440	FA[473] = 17220
FA[258] = 16275	FA[312] = 16590	FA[366] = 16410	FA[420] = 16710	FA[474] = 17115
FA[259] = 16875	FA[313] = 16695	FA[367] = 16815	FA[421] = 16665	FA[475] = 16785
FA[260] = 16125	FA[314] = 17145	FA[368] = 16950	FA[422] = 17295	FA[476] = 16890
FA[261] = 16935	FA[315] = 16860	FA[369] = 16830	FA[423] = 16575	FA[477] = 16695
FA[262] = 17370	FA[316] = 16845	FA[370] = 17040	FA[424] = 17190	FA[478] = 16290
FA[263] = 16575	FA[317] = 16860	FA[371] = 16740	FA[425] = 17085	FA[479] = 16830
FA[264] = 16650	FA[318] = 16530	FA[372] = 16920	FA[426] = 16035	FA[480] = 17295
FA[265] = 16380	FA[319] = 16410	FA[373] = 16650	FA[427] = 16725	FA[481] = 17265
FA[266] = 17070	FA[320] = 17130	FA[374] = 16365	FA[428] = 16200	FA[482] = 17070
FA[267] = 16785	FA[321] = 16995	FA[375] = 16965	FA[429] = 16425	FA[483] = 16170
FA[268] = 16275	FA[322] = 16860	FA[376] = 16680	FA[430] = 16560	FA[484] = 16605
FA[269] = 16845	FA[323] = 16305	FA[377] = 16410	FA[431] = 16575	FA[485] = 17055
FA[270] = 16110	FA[324] = 16800	FA[378] = 16110	FA[432] = 16545	FA[486] = 16575
FA[271] = 16395	FA[325] = 16545	FA[379] = 16515	FA[433] = 16890	FA[487] = 16410
FA[272] = 17430	FA[326] = 16545	FA[380] = 16920	FA[434] = 17175	FA[488] = 17520
FA[273] = 15930	FA[327] = 17115	FA[381] = 16875	FA[435] = 16560	FA[489] = 16920
FA[274] = 16815	FA[328] = 16920	FA[382] = 16815	FA[436] = 16365	FA[490] = 16410
FA[275] = 16965	FA[329] = 16740	FA[383] = 16620	FA[437] = 17205	FA[491] = 16620
FA[276] = 16635	FA[330] = 16635	FA[384] = 16620	FA[438] = 16245	FA[492] = 16575
FA[277] = 16965	FA[331] = 16875	FA[385] = 16785	FA[439] = 16830	FA[493] = 16800
FA[278] = 16965	FA[332] = 16650	FA[386] = 16455	FA[440] = 16410	FA[494] = 16920
FA[279] = 16980	FA[333] = 16695	FA[387] = 16770	FA[441] = 16950	FA[495] = 16785
FA[280] = 16800	FA[334] = 16725	FA[388] = 16815	FA[442] = 16560	FA[496] = 16560
FA[281] = 17025	FA[335] = 16590	FA[389] = 17055	FA[443] = 16725	FA[497] = 16425
FA[282] = 16935	FA[336] = 17070	FA[390] = 16755	FA[444] = 16680	FA[498] = 16860
FA[283] = 16350	FA[337] = 16215	FA[391] = 16455	FA[445] = 16455	FA[499] = 17010
FA[284] = 17220	FA[338] = 16305	FA[392] = 16575	FA[446] = 16860	FA[500] = 16455
FA[285] = 17010	FA[339] = 16725	FA[393] = 17040	FA[447] = 17085	
FA[286] = 17220	FA[340] = 16845	FA[394] = 16695	FA[448] = 17010	
FA[287] = 16800	FA[341] = 16500	FA[395] = 16680	FA[449] = 16380	
FA[288] = 16980	FA[342] = 16650	FA[396] = 16890	FA[450] = 16755	
FA[289] = 16725	FA[343] = 17070	FA[397] = 16815	FA[451] = 17160	

D.3. POPULAÇÕES INICIAIS

Valores das funções avaliadores dos cromossomos constituintes de uma população inicial, para testes com o cenário alargado:

FA[1] = 24720	FA[58] = 24510	FA[115] = 26415	FA[172] = 24930	FA[229] = 26025
FA[2] = 26070	FA[59] = 26325	FA[116] = 25560	FA[173] = 25410	FA[230] = 25830
FA[3] = 24960	FA[60] = 26130	FA[117] = 24480	FA[174] = 25950	FA[231] = 26565
FA[4] = 26130	FA[61] = 26535	FA[118] = 24915	FA[175] = 26340	FA[232] = 25800
FA[5] = 25380	FA[62] = 25125	FA[119] = 25995	FA[176] = 26265	FA[233] = 26340
FA[6] = 26115	FA[63] = 25125	FA[120] = 25905	FA[177] = 25215	FA[234] = 25890
FA[7] = 25680	FA[64] = 26205	FA[121] = 25440	FA[178] = 26070	FA[235] = 25080
FA[8] = 25770	FA[65] = 25455	FA[122] = 24960	FA[179] = 25440	FA[236] = 24510
FA[9] = 25980	FA[66] = 26160	FA[123] = 25455	FA[180] = 25425	FA[237] = 25815
FA[10] = 25695	FA[67] = 25470	FA[124] = 25635	FA[181] = 25665	FA[238] = 25965
FA[11] = 25155	FA[68] = 25095	FA[125] = 26130	FA[182] = 25350	FA[239] = 25125
FA[12] = 25830	FA[69] = 25710	FA[126] = 26115	FA[183] = 26400	FA[240] = 25395
FA[13] = 24735	FA[70] = 26010	FA[127] = 25515	FA[184] = 25920	FA[241] = 25725
FA[14] = 26205	FA[71] = 24495	FA[128] = 25755	FA[185] = 26175	FA[242] = 25410
FA[15] = 26655	FA[72] = 26040	FA[129] = 25635	FA[186] = 26730	FA[243] = 26205
FA[16] = 25920	FA[73] = 25635	FA[130] = 25965	FA[187] = 25725	FA[244] = 25545
FA[17] = 25005	FA[74] = 26400	FA[131] = 25890	FA[188] = 26310	FA[245] = 25965
FA[18] = 25680	FA[75] = 25500	FA[132] = 27120	FA[189] = 24930	FA[246] = 26010
FA[19] = 26595	FA[76] = 26385	FA[133] = 25965	FA[190] = 26640	FA[247] = 26490
FA[20] = 25860	FA[77] = 26100	FA[134] = 24720	FA[191] = 26370	FA[248] = 26520
FA[21] = 25665	FA[78] = 25575	FA[135] = 25635	FA[192] = 25755	FA[249] = 25470
FA[22] = 25695	FA[79] = 25680	FA[136] = 25665	FA[193] = 26160	FA[250] = 25440
FA[23] = 26715	FA[80] = 25950	FA[137] = 26205	FA[194] = 25740	FA[251] = 25155
FA[24] = 25350	FA[81] = 26610	FA[138] = 26250	FA[195] = 26430	FA[252] = 25140
FA[25] = 25890	FA[82] = 26175	FA[139] = 24885	FA[196] = 25155	FA[253] = 25770
FA[26] = 24915	FA[83] = 26610	FA[140] = 25755	FA[197] = 25620	FA[254] = 25725
FA[27] = 25335	FA[84] = 25695	FA[141] = 26250	FA[198] = 25365	FA[255] = 26175
FA[28] = 25965	FA[85] = 25365	FA[142] = 25485	FA[199] = 25830	FA[256] = 25770
FA[29] = 26265	FA[86] = 25410	FA[143] = 24945	FA[200] = 26010	FA[257] = 26190
FA[30] = 25740	FA[87] = 24870	FA[144] = 24930	FA[201] = 25635	FA[258] = 25005
FA[31] = 25035	FA[88] = 25710	FA[145] = 26130	FA[202] = 24690	FA[259] = 26250
FA[32] = 25005	FA[89] = 25725	FA[146] = 24840	FA[203] = 24330	FA[260] = 26190
FA[33] = 25410	FA[90] = 26415	FA[147] = 25620	FA[204] = 25275	FA[261] = 25215
FA[34] = 26100	FA[91] = 25830	FA[148] = 25605	FA[205] = 26430	FA[262] = 25980
FA[35] = 25395	FA[92] = 25860	FA[149] = 25305	FA[206] = 25155	FA[263] = 26040
FA[36] = 25320	FA[93] = 26295	FA[150] = 25620	FA[207] = 24945	FA[264] = 26175
FA[37] = 26745	FA[94] = 26565	FA[151] = 26355	FA[208] = 25710	FA[265] = 25620
FA[38] = 25530	FA[95] = 26280	FA[152] = 25995	FA[209] = 25695	FA[266] = 25275
FA[39] = 26100	FA[96] = 25485	FA[153] = 25395	FA[210] = 26175	FA[267] = 25980
FA[40] = 25140	FA[97] = 24795	FA[154] = 25995	FA[211] = 25485	FA[268] = 26355
FA[41] = 25020	FA[98] = 25650	FA[155] = 25440	FA[212] = 25500	FA[269] = 25755
FA[42] = 25545	FA[99] = 25935	FA[156] = 26265	FA[213] = 25530	FA[270] = 24840
FA[43] = 25890	FA[100] = 26655	FA[157] = 25860	FA[214] = 26115	FA[271] = 25470
FA[44] = 25245	FA[101] = 26775	FA[158] = 24735	FA[215] = 25605	FA[272] = 25515
FA[45] = 25860	FA[102] = 25845	FA[159] = 26370	FA[216] = 25860	FA[273] = 26415
FA[46] = 26220	FA[103] = 26640	FA[160] = 25665	FA[217] = 25065	FA[274] = 26250
FA[47] = 25110	FA[104] = 25995	FA[161] = 25890	FA[218] = 25965	FA[275] = 26175
FA[48] = 25860	FA[105] = 25320	FA[162] = 24840	FA[219] = 26220	FA[276] = 25800
FA[49] = 27660	FA[106] = 25215	FA[163] = 25725	FA[220] = 25575	FA[277] = 25650
FA[50] = 25890	FA[107] = 25470	FA[164] = 26205	FA[221] = 26025	FA[278] = 25440
FA[51] = 25965	FA[108] = 24390	FA[165] = 24945	FA[222] = 25545	FA[279] = 25095
FA[52] = 25110	FA[109] = 26775	FA[166] = 25680	FA[223] = 25275	FA[280] = 25455
FA[53] = 26415	FA[110] = 25860	FA[167] = 26115	FA[224] = 25875	FA[281] = 26025
FA[54] = 24645	FA[111] = 25905	FA[168] = 26370	FA[225] = 24810	FA[282] = 26160
FA[55] = 25485	FA[112] = 26655	FA[169] = 25935	FA[226] = 26505	FA[283] = 26205
FA[56] = 24930	FA[113] = 26655	FA[170] = 26205	FA[227] = 25605	FA[284] = 25920
FA[57] = 25755	FA[114] = 25290	FA[171] = 25890	FA[228] = 24825	FA[285] = 25575

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

FA[286] = 25260	FA[330] = 25005	FA[374] = 26415	FA[418] = 24735	FA[462] = 25620
FA[287] = 25410	FA[331] = 25230	FA[375] = 25980	FA[419] = 26205	FA[463] = 26160
FA[288] = 25695	FA[332] = 25785	FA[376] = 25140	FA[420] = 25245	FA[464] = 26175
FA[289] = 25695	FA[333] = 26250	FA[377] = 25545	FA[421] = 25905	FA[465] = 25320
FA[290] = 25830	FA[334] = 25785	FA[378] = 26100	FA[422] = 25785	FA[466] = 25860
FA[291] = 25230	FA[335] = 26115	FA[379] = 26610	FA[423] = 26085	FA[467] = 25410
FA[292] = 26100	FA[336] = 26055	FA[380] = 27090	FA[424] = 26760	FA[468] = 25740
FA[293] = 25680	FA[337] = 25845	FA[381] = 26280	FA[425] = 25320	FA[469] = 25095
FA[294] = 26295	FA[338] = 26100	FA[382] = 26790	FA[426] = 25665	FA[470] = 26250
FA[295] = 25095	FA[339] = 26040	FA[383] = 25680	FA[427] = 26310	FA[471] = 25500
FA[296] = 25410	FA[340] = 25725	FA[384] = 25350	FA[428] = 25905	FA[472] = 24945
FA[297] = 25110	FA[341] = 25815	FA[385] = 25290	FA[429] = 24945	FA[473] = 26145
FA[298] = 27285	FA[342] = 24915	FA[386] = 26610	FA[430] = 25500	FA[474] = 25665
FA[299] = 25980	FA[343] = 25920	FA[387] = 25755	FA[431] = 25260	FA[475] = 25980
FA[300] = 25485	FA[344] = 25440	FA[388] = 24645	FA[432] = 25785	FA[476] = 26220
FA[301] = 26640	FA[345] = 25695	FA[389] = 25050	FA[433] = 25635	FA[477] = 25365
FA[302] = 25470	FA[346] = 25710	FA[390] = 25305	FA[434] = 25350	FA[478] = 25140
FA[303] = 26385	FA[347] = 26505	FA[391] = 26355	FA[435] = 26355	FA[479] = 25305
FA[304] = 25920	FA[348] = 25230	FA[392] = 25785	FA[436] = 25485	FA[480] = 25620
FA[305] = 25875	FA[349] = 25035	FA[393] = 25395	FA[437] = 24720	FA[481] = 25665
FA[306] = 25470	FA[350] = 26685	FA[394] = 26175	FA[438] = 25485	FA[482] = 25005
FA[307] = 25665	FA[351] = 25605	FA[395] = 26055	FA[439] = 25770	FA[483] = 25035
FA[308] = 26160	FA[352] = 25530	FA[396] = 26085	FA[440] = 24990	FA[484] = 25575
FA[309] = 25470	FA[353] = 25155	FA[397] = 25260	FA[441] = 25935	FA[485] = 25875
FA[310] = 26715	FA[354] = 26430	FA[398] = 24855	FA[442] = 25515	FA[486] = 26490
FA[311] = 26325	FA[355] = 24870	FA[399] = 24735	FA[443] = 26580	FA[487] = 24990
FA[312] = 26610	FA[356] = 26280	FA[400] = 26040	FA[444] = 25920	FA[488] = 24825
FA[313] = 26235	FA[357] = 26430	FA[401] = 25350	FA[445] = 25155	FA[489] = 25635
FA[314] = 26085	FA[358] = 25005	FA[402] = 26460	FA[446] = 26370	FA[490] = 25260
FA[315] = 26010	FA[359] = 26190	FA[403] = 25425	FA[447] = 25560	FA[491] = 25980
FA[316] = 26730	FA[360] = 25395	FA[404] = 25605	FA[448] = 25815	FA[492] = 25890
FA[317] = 25680	FA[361] = 26565	FA[405] = 25305	FA[449] = 25440	FA[493] = 25515
FA[318] = 25920	FA[362] = 25920	FA[406] = 25965	FA[450] = 25740	FA[494] = 26115
FA[319] = 25065	FA[363] = 26250	FA[407] = 26055	FA[451] = 25845	FA[495] = 25140
FA[320] = 27180	FA[364] = 25875	FA[408] = 25560	FA[452] = 25470	FA[496] = 24900
FA[321] = 26970	FA[365] = 25395	FA[409] = 25695	FA[453] = 26310	FA[497] = 26835
FA[322] = 25875	FA[366] = 26295	FA[410] = 25680	FA[454] = 25350	FA[498] = 25080
FA[323] = 25575	FA[367] = 25725	FA[411] = 25440	FA[455] = 26205	FA[499] = 26580
FA[324] = 25590	FA[368] = 25650	FA[412] = 25185	FA[456] = 25650	FA[500] = 25995
FA[325] = 25935	FA[369] = 25710	FA[413] = 25515	FA[457] = 25740	
FA[326] = 26310	FA[370] = 25560	FA[414] = 25170	FA[458] = 26280	
FA[327] = 25110	FA[371] = 26010	FA[415] = 25740	FA[459] = 26115	
FA[328] = 25485	FA[372] = 25365	FA[416] = 25215	FA[460] = 25860	
FA[329] = 25845	FA[373] = 25575	FA[417] = 25470	FA[461] = 25170	

D.4 Penalizações Parciais de Fins-de-Semana

Na Tabela D.3 encontram-se discriminados todos os valores de penalizações parciais relacionadas com fins-de-semana, para os cromossomas constituintes de uma população inicial. A instância utilizada para o teste iniciado com esta população foi a construída com o “**Cenário 1**”.

Estão apresentados os valores relativos às penalizações de fins-de-semana completos (*fscompletos*), ao número total de fins-de-semana alocados permitido (*fstotal*) e ao número de fins-de-semana consecutivos (*fsconsecutivos*).

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
1	163	15	69
2	157	16	74
3	162	11	67
4	145	15	72
5	143	11	59
6	148	13	64
7	152	15	69
8	148	15	73
9	166	15	71
10	167	16	71
11	149	13	69
12	142	10	63
13	143	13	65
14	138	11	59
15	145	14	67
16	147	12	67
17	154	14	69
18	140	12	65
19	150	13	71
20	156	14	70
21	154	14	69
22	157	12	67
23	147	14	69
24	153	11	65
25	131	11	66
26	154	13	70
27	154	15	71
28	151	12	69
29	151	14	68
30	157	13	67
31	163	15	73
32	157	14	69
33	169	17	75
34	145	14	66
35	167	14	71
36	154	10	64
37	162	15	71
38	144	13	69
39	158	17	74
40	155	15	69
41	143	16	72
42	145	12	64
43	164	15	71
44	150	11	64
45	172	13	68
46	161	16	74
47	146	13	69
48	149	15	72
49	144	13	65
50	151	15	71
51	149	13	68
52	148	15	73
53	162	15	72
54	161	14	71
55	142	11	63
56	162	16	74
57	167	16	73

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
58	158	13	71
59	143	15	71
60	156	15	69
61	135	10	66
62	158	16	71
63	182	15	72
64	176	16	74
65	142	15	68
66	148	15	73
67	158	13	68
68	148	12	67
69	154	13	70
70	148	15	72
71	142	10	65
72	154	14	69
73	152	12	66
74	154	12	68
75	166	14	70
76	144	13	69
77	142	11	60
78	150	14	69
79	155	14	70
80	160	15	72
81	137	13	68
82	165	15	72
83	150	15	73
84	136	10	62
85	152	12	67
86	136	13	65
87	152	15	71
88	169	15	70
89	148	14	70
90	155	14	70
91	168	14	71
92	150	14	69
93	156	11	66
94	152	14	71
95	137	12	63
96	157	14	70
97	159	16	73
98	136	13	70
99	149	14	69
100	155	14	70
101	157	14	71
102	152	16	73
103	130	12	65
104	155	15	69
105	160	13	68
106	147	13	69
107	150	14	71
108	146	13	70
109	149	13	68
110	156	12	66
111	151	14	68
112	144	12	66
113	159	13	68
114	152	14	68

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
115	147	13	68
116	151	12	67
117	146	14	68
118	145	12	68
119	157	16	73
120	167	16	73
121	167	14	71
122	147	12	63
123	161	15	71
124	162	12	65
125	156	13	65
126	163	14	71
127	140	15	69
128	151	13	66
129	144	12	68
130	140	12	67
131	144	13	67
132	152	14	70
133	155	12	68
134	165	13	64
135	160	15	71
136	138	14	69
137	146	14	68
138	161	14	70
139	164	14	71
140	169	15	72
141	145	10	60
142	165	13	68
143	141	13	69
144	157	14	70
145	159	15	73
146	149	14	69
147	158	14	68
148	149	13	67
149	162	15	67
150	148	15	69
151	165	14	67
152	147	13	69
153	162	13	71
154	138	12	65
155	163	12	68
156	152	13	69
157	142	13	70
158	139	12	62
159	150	13	70
160	161	16	74
161	141	16	71
162	160	16	71
163	135	7	60
164	167	14	70
165	160	14	70
166	156	14	68
167	160	12	64
168	146	10	62
169	154	14	66
170	145	12	68
171	144	13	65

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
172	163	15	71
173	155	13	69
174	143	14	70
175	137	14	70
176	153	15	69
177	158	16	74
178	158	13	67
179	142	12	64
180	151	13	67
181	164	15	70
182	146	14	69
183	150	14	68
184	145	11	67
185	149	15	70
186	139	10	64
187	154	13	67
188	138	9	61
189	155	14	68
190	149	14	71
191	144	14	66
192	147	14	70
193	156	13	69
194	162	14	68
195	141	14	68
196	156	15	67
197	163	11	65
198	157	15	72
199	128	10	58
200	144	12	68
201	148	15	72
202	155	14	70
203	144	11	67
204	149	13	70
205	166	13	68
206	166	17	75
207	161	14	69
208	150	14	72
209	162	16	74
210	155	13	70
211	149	15	71
212	159	14	67
213	128	11	66
214	140	12	67
215	148	10	64
216	143	16	70
217	145	12	67
218	158	13	68
219	151	16	74
220	144	13	65
221	143	13	65
222	142	14	66
223	162	15	72
224	157	15	73
225	156	15	68
226	150	13	69
227	154	12	69
228	142	12	65

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
229	158	13	69
230	139	14	68
231	176	16	74
232	138	13	67
233	147	14	70
234	165	12	65
235	146	12	70
236	150	16	74
237	157	13	68
238	170	15	68
239	152	15	70
240	137	14	69
241	137	13	67
242	150	15	71
243	156	12	64
244	153	12	65
245	150	14	67
246	140	12	61
247	155	14	69
248	152	15	68
249	157	15	69
250	154	13	66
251	146	11	63
252	155	14	68
253	148	12	68
254	147	12	64
255	145	11	66
256	168	15	71
257	143	15	70
258	143	14	72
259	158	12	67
260	157	14	71
261	164	14	68
262	149	9	60
263	151	14	70
264	147	11	65
265	140	13	66
266	169	13	70
267	146	13	66
268	144	13	69
269	145	14	69
270	136	15	72
271	158	14	71
272	157	13	66
273	131	14	70
274	139	11	65
275	147	12	67
276	167	17	75
277	160	15	72
278	152	13	64
279	167	13	70
280	146	11	63
281	151	11	62
282	162	16	73
283	151	13	65
284	155	15	72
285	156	12	64

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
286	147	13	68
287	149	13	68
288	167	16	73
289	145	13	65
290	164	15	71
291	135	8	56
292	157	13	68
293	145	12	64
294	139	12	64
295	142	14	72
296	141	13	66
297	146	13	68
298	150	14	68
299	151	10	66
300	152	14	66
301	142	12	65
302	145	12	67
303	145	14	71
304	163	17	75
305	148	13	69
306	142	13	67
307	152	12	66
308	147	13	67
309	147	13	70
310	164	13	69
311	141	12	62
312	142	11	67
313	159	14	69
314	162	14	69
315	150	12	64
316	173	14	70
317	140	12	66
318	159	16	69
319	146	13	64
320	166	14	70
321	144	11	67
322	158	12	66
323	161	16	74
324	153	12	69
325	137	15	70
326	150	15	71
327	151	13	69
328	159	14	71
329	148	10	62
330	140	12	64
331	136	11	62
332	157	12	69
333	151	14	68
334	140	12	63
335	146	13	69
336	148	10	62
337	150	14	70
338	151	12	65
339	151	12	67
340	143	13	68
341	142	12	65
342	142	12	65

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
343	145	11	64
344	138	11	66
345	144	11	64
346	152	14	69
347	164	14	72
348	164	15	70
349	149	12	68
350	149	13	68
351	161	14	71
352	142	11	64
353	159	13	68
354	151	15	71
355	163	13	67
356	168	15	72
357	148	14	71
358	146	12	66
359	153	15	69
360	162	16	73
361	169	15	72
362	157	11	67
363	155	15	70
364	152	14	70
365	157	14	70
366	154	14	70
367	144	10	63
368	143	11	63
369	154	14	68
370	150	9	55
371	150	16	73
372	131	10	62
373	139	10	62
374	144	15	71
375	137	12	65
376	162	13	69
377	151	14	71
378	132	12	68
379	146	14	70
380	166	14	69
381	144	13	67
382	158	13	68
383	137	10	63
384	155	14	71
385	145	10	64
386	152	15	70
387	164	15	72
388	161	16	73
389	150	11	67
390	158	12	66
391	154	16	74
392	159	14	71
393	156	11	62
394	164	16	74
395	169	17	74
396	159	13	69
397	143	15	71
398	151	14	70
399	156	15	70

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
400	135	12	66
401	137	12	65
402	156	11	66
403	149	12	66
404	139	16	70
405	165	14	70
406	162	15	69
407	148	13	67
408	144	14	72
409	154	12	67
410	162	12	67
411	151	14	70
412	171	14	70
413	147	15	70
414	150	13	65
415	157	14	70
416	151	13	66
417	146	12	65
418	135	14	68
419	145	13	65
420	155	12	65
421	136	11	62
422	139	11	63
423	132	14	65
424	152	12	64
425	147	12	65
426	158	16	73
427	144	14	71
428	154	15	73
429	144	15	72
430	131	11	59
431	154	12	64
432	154	11	63
433	154	15	70
434	159	13	67
435	153	12	66
436	136	12	67
437	153	11	62
438	147	14	71
439	159	16	72
440	148	14	67
441	157	15	68
442	155	14	69
443	150	15	72
444	146	11	62
445	144	16	71
446	159	13	70
447	154	13	69
448	146	12	64
449	124	12	61
450	146	9	60
451	159	13	67
452	156	13	68
453	148	16	71
454	154	14	67
455	141	12	63
456	149	12	66

continua na próxima página

D.4. PENALIZAÇÕES PARCIAIS DE FINS-DE-SEMANA

Tabela D.3: Valores de penalizações parciais relacionadas com fins-de-semana para cromossomas de uma população inicial.

Cromossoma	fscompletos	fstotal	fsconsecutivos
457	151	14	69
458	171	15	70
459	160	11	66
460	157	14	69
461	145	14	72
462	157	13	70
463	146	14	67
464	155	13	65
465	137	13	67
466	143	11	61
467	144	10	59
468	160	14	71
469	154	14	70
470	141	14	70
471	156	13	68
472	150	15	70
473	153	13	66
474	152	10	63
475	158	13	68
476	141	12	64
477	164	14	70
478	141	12	68
479	151	11	62
480	157	13	65
481	135	11	64
482	164	14	69
483	164	16	72
484	149	15	69
485	173	16	71
486	149	14	70
487	152	15	71
488	149	10	61
489	152	13	69
490	157	15	71
491	147	12	69
492	163	15	71
493	155	13	69
494	151	15	71
495	159	15	70
496	141	13	67
497	151	13	64
498	153	12	67
499	161	13	65
500	143	12	67