

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Sistema de apoio à decisão para planeamento de júris de mestrados

Miguel Martingo Barbosa

DISSERTAÇÃO



Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Professora Doutora Maria Antónia Carravilla

27 de Julho de 2020

Resumo

O planeamento manual de júris de mestrado para as defesas de dissertação do Departamento de Engenharia e Gestão Industrial é um processo complexo de optimização combinatória que ocorre todos os anos, tal complexidade provém do elevado número de restrições e, também, da existência de vários objectivos conflituosos.

Esta dissertação teve como objectivo um Sistema de Apoio à Decisão (SAD) que permita a calendarização automática das defesas de dissertação, acompanhados de um conjunto de medidas de desempenho que permitam ao decisor avaliar as soluções alternativas e, eventualmente, acrescentar ou retirar restrições permitindo a realização de uma análise "what-if".

Deste modo, foi realizada uma pesquisa da literatura em problemas de horários, particularmente na educação.

Posteriormente, foi criado o modelo de programação matemático para o planeamento de júris de mestrado.

Assim sendo e com base no modelo matemático, após um estudo mais pormenorizado dos Algoritmos Genéticos, mais especificamente do Algoritmo Genético de Chaves Aleatórias Tendenciosas, ou Biased Random-Key Genetic Algorithms (BRKGA), foi criado e desenvolvido um algoritmo com o propósito de obter um planeamento das defesas de dissertação com o menor número de conflitos possível.

Por fim, e não menos importante, foram realizados testes para analisar e avaliar os resultados das soluções finais e melhorar, se possível, o algoritmo desenvolvido.

Como resultado desta dissertação, obteve-se uma ferramenta que permite automatizar e otimizar o processo do planeamento de júris de mestrado.

Keywords: Algoritmo Genético; Algoritmo Genético de Chaves Aleatórias Tendenciosas; calendarização, heurística; modelo matemático; Sistema de Apoio à Decisão

Abstract

The manual planning of Master's degree juries for dissertation defenses from the Department of Industrial Engineering and Management is a complex process of combinatorial optimization that occurs every year, such complexity comes from the high number of restrictions and, also, from several conflicting goals.

This dissertation aimed at a Decision Support System (DSS) that allows the automatic scheduling of dissertation defenses, followed by a set of performance standards that allow the decision-maker to assess alternative solutions and, eventually, add or remove restrictions allowing for a "what-if" analysis.

In this way, a literature survey was conducted on timetables problems, especially in education.

Subsequently, a mathematical programming model was created for the planning of Masters' juries.

After a study of the Genetic Algorithms, more precisely Biased Random-Key Genetic Algorithms (BRKGA), and based on the mathematical model, an algorithm was created and developed to obtain the planning of the dissertation defenses with the fewer conflicts as possible.

Last but not least, tests were performed to analyze and evaluate the results of the final solutions and to improve, if possible, the developed algorithm.

As a result of this dissertation, a tool was obtained which allows automation and optimization process of the planning of Master's juries.

Keywords: Biased Random-Key Genetic Algorithm; Decision Support System; heuristic; Genetic Algorithm; mathematical model; timetabling

Agradecimentos

A presente dissertação representa o culminar de todo um percurso académico.

Gostaria de agradecer a todos aqueles que de uma forma directa ou indirecta contribuíram para a finalização do trabalho apresentado, em especial:

À Professora Doutora Maria Antónia Carravilla, professora orientadora, por me ter proporcionado a realização desta dissertação, pela sua confiança e disponibilidade para esclarecer dúvidas, revisão da dissertação, assim como pela sua competente direcção e perspicácia na superação dos diversos obstáculos que surgiram.

A todos os docentes da Faculdade de Engenharia da Universidade do Porto, pela sabedoria e conhecimentos transmitidos, sem os quais não poderia chegar a esta fase.

Aos meus pais e irmã, por me inculcaram a importância do estudo e por me terem proporcionado todas as condições para eu chegar até aqui.

À minha namorada pelo apoio constante, carinho, encorajamento e força para continuar.

A todos, muito obrigado!

Miguel Barbosa

“Eighty percent of success is showing up.”

Woody Allen

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objectivos da dissertação	2
1.3	Estrutura da dissertação	2
2	Revisão bibliográfica	3
2.1	Problemas de horários no ensino	3
2.1.1	Calendarização de exames	4
2.1.2	Calendarização do horário das turmas	4
2.2	Métodos de resolução	5
2.2.1	Arrefecimento Simulado	5
2.2.2	Pesquisa Tabu	7
2.2.3	Optimização da Colónia de Formigas	9
2.2.4	Algoritmo Genético	11
2.3	Algoritmo Genético Baseado em Chaves Aleatórias Tendenciosas	13
2.3.1	Introdução	13
2.3.2	Descrição da meta-heurística BRKGA	13
2.3.3	Exemplos e comparação dos Algoritmos Genéticos	19
3	Descrição do problema	23
3.1	Parâmetros	23
3.2	Objectivos	25
3.3	Restrições	25
4	Modelo proposto para o problema de planeamento de júris de mestrado	27
4.1	Índices e Parâmetros	27
4.2	Variáveis de decisão	28
4.3	Restrições fortes	28
4.4	Restrições fracas	29
4.5	Função Objectivo	29
5	Procedimento computacional	33
5.1	Primeira abordagem	34
5.1.1	Heurística da primeira abordagem	34
5.1.2	BRKGA da primeira abordagem	35
5.2	Segunda abordagem	38
5.2.1	Cromossoma	38
5.2.2	Função de avaliação	38

5.2.3	Restrições	39
6	Sistema de Apoio à Decisão	41
6.1	Introdução à framework	41
6.2	Parâmetros	41
6.2.1	Parte independente	41
6.2.2	Parte dependente	42
6.3	Leitura de dados	43
6.4	Planeamento final	43
6.4.1	Critério de paragem	43
6.4.2	Pesos das restrições na função objectivo	44
6.4.3	Resultado	44
6.5	Criação de blocos	45
6.5.1	Critério de paragem	45
6.5.2	Pesos das restrições na função objectivo	46
6.5.3	Resultado	46
7	Testes computacionais	47
7.1	Plano de testes	47
7.2	Análise Gráfica	47
7.3	Análise Estatística	51
7.3.1	Primeira abordagem	51
7.3.2	Segunda abordagem	54
7.4	Comparação com o planeamento manual	59
8	Conclusão e Trabalho futuro	61
8.1	Conclusão	61
8.2	Trabalho futuro	61
A	Algoritmo Genético de Chaves Aleatórias Tendenciosas	65

Lista de Figuras

2.1	Algoritmo SA	6
2.2	Algoritmo TS	8
2.3	Algoritmo ACO	10
2.4	Algoritmo GA	12
2.5	Evolução do BRKGA	15
2.6	Exemplo de um cruzamento no BRKGA	16
2.7	Hipercubo do BRKGA	18
2.8	Processo do BRKGA	18
2.9	Gráfico com a distribuição do valor das funções objectivo da população, adaptado do exemplo no artigo [Gonçalves and Resende, 2011]	20
2.10	Gráfico da probabilidade dos algoritmos encontrarem a solução pretendida, adaptado do exemplo no artigo [Gonçalves and Resende, 2011]	21
5.1	Diferentes abordagens para o planeamento de júris de mestrado	33
5.2	Heurística para a criação dos blocos	34
5.3	Caracterização de uma população para o planeamento final	35
5.4	Caracterização de uma população para a criação de blocos	38
7.1	Gráfico da evolução da função <i>fitness</i> para o planeamento final	48
7.2	Gráfico da evolução da função <i>fitness</i> para a criação dos blocos	49
7.3	Planeamento final da defesa das dissertações	50
A.1	Código correspondente ao Algoritmo Genético de Chaves Aleatórias Tendenciosas	65

Lista de Tabelas

2.1	Terminologia básica para problemas de geração de horários	3
2.2	Configuração dos parâmetros SA	7
2.3	Configuração dos parâmetros TS	8
2.4	Configuração dos parâmetros ACO	11
2.5	Configuração dos parâmetros GA	12
2.6	Parâmetros do BRKGA	14
4.1	Valores da componente 4.14 da função objectivo	31
4.2	Valores da componente 4.22 da função objectivo	32
4.3	Valores da componente 4.23 da função objectivo	32
6.1	Parâmetros independentes do problema	42
6.2	Parâmetros dependentes do problema	43
7.1	Resultado para diferentes sementes, primeira abordagem	51
7.2	Resultado estatísticos da diferentes sementes, primeira abordagem	52
7.3	Variação de valores dos parâmetros independentes, primeira abordagem	52
7.4	Valores dos parâmetros independentes e consequentes resultados para a mesma semente, primeira abordagem	53
7.5	Resultado estatísticos para diferentes parâmetros independentes e mesma semente, primeira abordagem	53
7.6	Resultado para diferentes sementes para a criação de blocos, segunda abordagem	54
7.7	Resultado estatísticos de diferentes sementes para a criação de blocos, segunda abordagem	54
7.8	Variação de valores dos parâmetros independentes para a criação de blocos, segunda abordagem	55
7.9	Valores dos parâmetros independentes e consequentes resultados para a criação de blocos, segunda abordagem	55
7.10	Resultado estatísticos para diferentes parâmetros independentes e mesma semente para a criação de blocos, segunda abordagem	56
7.11	Resultado para diferentes sementes para o planeamento final, segunda abordagem	57
7.12	Resultado estatísticos da diferentes sementes para o planeamento final, segunda abordagem	57
7.13	Variação de valores dos parâmetros independentes para o planeamento final, segunda abordagem	57
7.14	Valores dos parâmetros independentes e consequentes resultados para o planeamento final, segunda abordagem	58
7.15	Resultado estatísticos para diferentes parâmetros independentes e mesma semente para o planeamento final, segunda abordagem	58

7.16	Comparação dos diferentes métodos para a criação de blocos	59
7.17	Comparação dos diferentes métodos para o planeamento final	59

Acrónimos

ACO	Ant Colony Optimization
B2	Blocos constituídos por 2 dissertações
B3	Blocos constituídos por 3 dissertações
B4	Blocos constituídos por 4 dissertações
BRKGA	Biased Random-Key Genetic Algorithms
DEGI	Departamento de Engenharia e Gestão Industrial
FEUP	Faculdade de Engenharia da Universidade do Porto
GA	Genetic Algorithm
GRFortes	Geração em que restrições fortes foram cumpridas
GRFracas	Geração em que restrições fracas foram cumpridas
MIEGI	Mestrado Integrado em Engenharia e Gestão Industrial
RKGA	Random-Key Genetic Algorithms
SA	Simulated Annealing
SAD	Sistema de Apoio à Decisão
TS	Tabu Search

Capítulo 1

Introdução

Neste capítulo, apresenta-se a motivação para o problema a tratar, os objectivos que se pretendem atingir e termina-se com a descrição da estrutura da dissertação.

1.1 Motivação

Os problemas de geração de horários consistem em agendar uma sequência de actividades, num determinado período de tempo, satisfazendo um conjunto de restrições de vários tipos.

O planeamento manual destes problemas, dependendo da complexidade do problema, é um processo demorado. Além disso, a solução obtida pode não ser satisfatória em determinados aspectos. Por estes motivos, o planeamento automatizado começou a ganhar bastante importância em diversas áreas da sociedade, foram efectuadas diversas pesquisas e há um elevado número de artigos na literatura, nas áreas de investigação operacional e inteligência artificial, que abordam a diversidade dos problemas de calendarização [Schaefer, 1999].

Estes problemas são designados como problemas de optimização, *optimization problems*, ou seja, é necessário um planeamento que satisfaça todas as restrições *fortes*, restrições que devem ser cumpridas para obter uma solução aceitável, e minimize, ou maximize, uma determinada função objectivo que incorpore as restrições *fracas*, se cumpridas melhoram a respectiva função objectivo [Petrovic and Burke, 2014].

Nos últimos anos, o interesse em abordagens meta-heurísticas para a resolução de problemas de calendarização, como o Arrefecimento Simulado, Pesquisa Tabu ou Algoritmos Genéticos, aumentou devido à capacidade de obter uma melhor solução comparativamente às heurísticas. Uma heurística é uma abordagem algorítmica para encontrar uma solução de boa qualidade com mais rapidez, sem garantir que essa é a melhor solução.

Uma solução inicial é construída através de uma heurística apropriada e, posteriormente, é efectuada uma melhoria através de meta-heurísticas [Burke et al., 2006].

1.2 Objectivos da dissertação

A diversidade de problemas de optimização combinatória, a variedade de restrições e os requisitos específicos fazem com que seja difícil encontrar uma solução geral e eficaz de planeamento [Oude Vrielink et al., 2019].

Deste modo, para o problema desta dissertação é necessário uma ferramenta capaz de atingir os seguintes objectivos:

- Respeitar as restrições *fortes* impostas;
- Garantir, na medida do possível, o cumprimento das restrições *fracas*.

1.3 Estrutura da dissertação

Esta dissertação é composta pelos seguintes capítulos:

- **Capítulo 1** - introdução à dissertação;
- **Capítulo 2** - revisão bibliográfica de problemas similares;
- **Capítulo 3** - descrição detalhada do problema em questão;
- **Capítulo 4** - descrição do modelo matemático proposto para o problema de planeamento de júris de mestrado;
- **Capítulo 5** - integração do modelo matemático ao problema computacional;
- **Capítulo 6** - descrição do Sistema de Apoio à Decisão desenvolvido ao longo da dissertação;
- **Capítulo 7** - descrição e comparação dos testes computacionais realizados;
- **Capítulo 8** - conclusão do trabalho desenvolvido.

Capítulo 2

Revisão bibliográfica

Neste capítulo, é descrito o desenvolvimento de soluções para problemas de optimização. Inicialmente, o problema é explicado para uma melhor compreensão das soluções possíveis. De seguida, são descritos métodos de resolução e as respectivas comparações. Por fim, é realizado um estudo mais detalhado à meta-heurística utilizada durante a dissertação.

2.1 Problemas de horários no ensino

A geração de horários consiste em planear uma sequência de actividades, num determinado período de tempo e sujeita a um conjunto de restrições, de modo a satisfazer, tanto quanto possível, um conjunto de objectivos [Carter and Laporte, 1998].

As restrições estão divididas em duas categorias, as restrições *fortes*, que devem ser cumpridas sob qualquer circunstância, e as restrições *fracas*, que devem ser atendidas o máximo possível [Colomi, 1998].

Um problema geral de geração horários nas universidades consiste em atribuir vários eventos, como exames, palestras ou aulas, a um número limitado de intervalos de tempo e salas, minimizando as violações de um conjunto de restrições [Hartmanis and Leeuwen, 2002]. Na tabela 2.1 está representado a terminologia básica para este tipo de problemas.

Tabela 2.1: Terminologia básica para problemas de geração de horários

Evento	Actividade a ser planeada
Intervalo de tempo	Período onde um evento pode ser alocado
Recurso	Recursos utilizados para os eventos
Restrição	Restrição ou requisito de um evento
Indivíduo	Pessoa interveniente no evento
Conflito	Dois eventos estão em conflito se o mesmo indivíduo ou recurso for alocado em ambos os eventos no mesmo intervalo de tempo

Estes problemas são classificados, em termos de complexidade computacional, como *NP-difíceis*, o que implica grande exigência de capacidade de processamento.

Ao longo desta secção são estudados dois tipos de problemas, na área da educação, um para a calendarização de exames e outro para a calendarização do horário das turmas.

2.1.1 Calendarização de exames

A calendarização de exames é um problema muito comum, não só para escolas secundárias como também para as universidades.

O problema de geração de horários de exames tem como principal objectivo evitar a sobreposição de exames do mesmo curso, e exames de cursos com estudantes em comum, e obter o maior espaçamento possível entre os exames de cada estudante.

Dado um conjunto de exames, um conjunto de intervalos de tempo, um conjunto de estudantes e um conjunto de matrículas para os exames, pretende-se atribuir exames aos intervalos de tempo, satisfazendo um conjunto de restrições.

Tal como descrito no artigo [Pillay and Qu, 2018], para este exemplo, um conflito de primeira ordem, designado como restrição *forte*, é quando um aluno dispõe de dois exames no mesmo intervalo de tempo. Um conflito de segunda ordem, restrição *fraca*, surge quando um aluno dispõe de dois exames em horários próximos, em intervalos de tempo consecutivos.

Sempre que um estudante possuir dois exames agendados em x intervalos de tempo consecutivos é definida a atribuição de um custo de proximidade $w(x)$. O custo de cada restrição é multiplicado pelo número de estudantes que possuam os mesmos dois exames. O custo para as restrições *fracas* diminui de forma logarítmica da seguinte maneira: $w(1)=16$, $w(2)=8$, $w(3)=4$. O custo das restrições *fortes* é 1000. A função de custo é normalizada com base no número total de estudantes.

Para a resolução deste problema existem diversas meta-heurísticas na literatura, como o Algoritmo de Refecimento Simulado, Pesquisa Tabu, Algoritmos Genéticos e Colónia de Formigas que serão abordados na secção 2.2.

2.1.2 Calendarização do horário das turmas

Na calendarização de turmas das unidades curriculares no ensino universitário, um conjunto de unidades curriculares, ou disciplinas, são planeadas para um determinado número de salas, num determinado período de tempo e, ao mesmo tempo, são designados professores para que as aulas se possam realizar [Abdullah, 2006].

Tal como na calendarização de exames, este planeamento também apresenta restrições *fortes* e *fracas*. Para este problema, exemplos de restrições *fortes* são:

- Cada estudante e cada professor não pode pertencer a duas aulas no mesmo intervalo de tempo;
- Apenas uma aula numa sala pode ser atribuída a um intervalo de tempo;
- A capacidade da sala de aula deve ser igual ou superior ao número de alunos que frequentam a respectiva aula;

- A sala deve satisfazer os recursos exigidos pela unidade curricular.

Algumas restrições *fracas* que influenciam a solução final deste problema são as seguintes:

- Os estudantes não devem ter apenas uma aula num dia;
- Os estudantes não devem frequentar mais de duas aulas consecutivas, isto é, sem intervalo;
- Os estudantes não devem ter aulas no último intervalo de tempo do dia.

Tal como referido anteriormente, vários algoritmos meta-heurísticos foram estudados para a resolução deste problema, no entanto o Arrefecimento Simulado, os Algoritmos Genéticos e a Pesquisa Tabu são aqueles que obtiveram mais atenção dos investigadores e que foram aplicados com sucesso [Abdullah, 2006].

2.2 Métodos de resolução

Problemas relacionados com a calendarização de exames ou do horário das turmas numa escola são resolvidos utilizando meta-heurísticas. Nesta secção descrevem-se quatro meta-heurísticas como possíveis resoluções para o problema de calendarização de exames, abordadas no artigo [Pillay and Qu, 2018].

As meta-heurísticas, introduzidas nos últimos trinta anos por Glover e Kochenberger, citado no artigo [Gonçalves and Resende, 2011], são métodos que combinam uma interacção entre procedimentos de melhoria local e uma estratégia de nível superior, capaz de realizar um pesquisa robusta de um espaço de solução e capaz de evitar os óptimos locais [Buriol et al., 2010b].

Estas meta-heurísticas foram projectadas para resolver problemas complexos de optimização, onde as heurísticas clássicas não conseguem ser eficazes e eficientes [Osman and Laporte, 1996].

2.2.1 Arrefecimento Simulado

Cerny, [Gerny, 1985], e Kirkpatrick, [Kirkpatrick et al., 1983], ambos citados no artigo de [Pillay and Qu, 2018], propuseram o uso do Arrefecimento Simulado, *Simulated Annealing*, para problemas combinatórios.

A meta-heurística Arrefecimento Simulado fundamenta-se numa analogia com a termodinâmica, é o processo utilizado para fundir um metal, onde este é aquecido a uma temperatura elevada e em seguida é arrefecido lentamente, de modo a que o produto final seja uma massa homogénea.

Este algoritmo implementa um arrefecimento na heurística com o objectivo de reduzir a probabilidade de aceitar soluções que piorem a função objectivo, deste modo as restrições do tipo *fortes* são submetidas a este processo para melhorar as restrições do tipo *fracas*.

Para a implementação deste método, a temperatura inicial é calculada através da seguinte expressão:

$$P_a = e^{-\frac{\Delta C}{T_{begin}}} \quad (2.1)$$

$$\Leftrightarrow T_{begin} = -\frac{\Delta C}{\ln P_a} \quad (2.2)$$

Onde ΔC : representa a média de deterioração, é a divisão do valor total dos movimentos agravantes da solução inicial com o número de movimentos que provocaram a deterioração da função objectivo.

A razão do valor aceite pelo número total de movimentos ocorridos é representado pelo parâmetro P , Acceptance Fraction.

A temperatura na iteração t é obtida através da temperatura da iteração anterior da seguinte maneira:

$$T_t = R * T_{t-1} \quad (2.3)$$

O parâmetro R representa a taxa de arrefecimento.

O critério de paragem é satisfeito quando o valor da temperatura se aproximar de 0.

2.2.1.1 Algoritmo

O algoritmo do Arrefecimento Simulado está representado na figura 2.1.

```

1  Selecionar estado inicial i;
2  Selecionar temperatura inicial T > 0;
3  Definir contador de mudança de temperatura t = 0;
4  while (Critério_de_paragem != TRUE) {
5      Definir contador de repetição n = 0;
6      while (n != N(t)) {
7          Gerar estado j, vizinho de i;
8          Calcular  $\delta = f(j) - f(i)$ ;
9          if ( $\delta < 0$ )
10             i = j;
11             if (random(0,1) < exp(- $\delta/T$ ))
12                 i = j;
13             n = n + 1;
14         }
15         t = t + 1;
16         T = T(t);
17     }

```

Figura 2.1: Algoritmo SA

2.2.1.2 Parâmetros

Acceptance Fraction (A) é o parâmetro usado para fixar a temperatura inicial, percentagem de movimentos aceites obtidos ao executar 1000 ciclos da solução inicial. Os valores atribuídos a este parâmetro são:

$$A \mid 0.30 \mid 0.50 \mid 0.70$$

Cooling Rate (R) é o parâmetro que representa a fracção pela qual a temperatura é reduzida e são atribuídos os seguintes valores:

$$R \mid 0.70 \mid 0.80 \mid 0.90 \mid 0.99$$

O melhor par de parâmetros entre os acima referidos é identificado na tabela 2.2.

Tabela 2.2: Configuração dos parâmetros SA

Parâmetro	Valor
A	0.5
R	0.99

2.2.2 Pesquisa Tabu

A Pesquisa Tabu, *Tabu Search*, foi concebida por Glover, [Glover, 1986] citado no artigo [Pillay and Qu, 2018]. Esta meta-heurística é baseada nos princípios de inteligência artificial, a ideia é iniciar a partir de uma solução aleatória e comparar essa solução, sucessivamente, com um dos seus vizinhos actuais. Cada vez que esta comparação é executada, no caso do problema de calendarização de exames, os pares [exame, período] são adicionados à lista tabu, lista que inclui os movimentos inibidos, e o intervalo de tempo desse exame não poderá ser alterado.

Uma nova mudança, presente na lista tabu, só é aceite se diminuir o valor actual da função objectivo.

Este processo é repetido enquanto o critério de paragem não for verdadeiro. O critério de paragem deste algoritmo é um número limitado de iterações entre iteração actual e a iteração da melhor solução.

2.2.2.1 Algoritmo

O algoritmo da Pesquisa Tabu está representado na figura 2.2.

```

1  s = solução inicial em X;
2  n = 0; /* iteração corrente */
3  best_n = 0; /* iteração quando melhor
4  solução é encontrada*/
5  best_sol = s; /* melhor solução */
6  T = 0; /* lista tabu */
7  Inicializar a função de aspiração A;
8  while ((f(s) > f*) AND (n - best_n < n_max)) {
9      n = n + 1;
10     Gerar o conjunto V* de soluções si em N(s)
11     que não pertençam à lista tabu ou que
12     A(f(s) >= f(si));
13     Escolher solução s* que minimize f no
14     conjunto V*;
15     Atualizar a função de aspiração A e a lista
16     tabu T;
17     if (f(s*) < F(best_sol)) {
18         best_sol = S*;
19         best_n = n;
20         s = s*;
21     }
22 }

```

Figura 2.2: Algoritmo TS

2.2.2.2 Parâmetros

Comprimento da lista tabu (L) é um parâmetro que indica o tamanho da lista tabu, um número fixo. Apresenta os seguintes valores atribuídos:

$$L \mid 10 \mid 20 \mid 30 \mid \frac{n}{3} \mid \frac{n}{2}$$

Memória de longo prazo (G) é um parâmetro que determina se uma memória a longo prazo é utilizada ou não.

Critério de paragem, ou número de ciclos máximo sem melhoria, ($Nmax$) é o parâmetro que define o número entre a iteração actual e a iteração onde a melhor solução é alcançada. Composta pelos seguintes valores atribuídos:

$$Nmax \mid 5 \mid 10 \mid 20 \mid 30$$

Na tabela 2.3 está representada a melhor configuração dos parâmetros anteriormente referidos.

Tabela 2.3: Configuração dos parâmetros TS

Parâmetro	Valor
L	$\frac{n}{3}$
G	Não
$Nmax$	30

2.2.3 Optimização da Colónia de Formigas

O algoritmo de Optimização da Colónia de Formigas, *Ant Colony Optimization*, foi concebido por Marco Dorigo, Vittorio Maniezzo e Alberto Coloni, [Dorigo et al., 1991] citado no artigo [Pillay and Qu, 2018]. Este algoritmo de optimização é baseado no comportamento de forrageamento de formigas reais, através de um mecanismo probabilístico de construção de soluções assente em estigmergia. Estigmergia é um mecanismo de coordenação indirecta entre agentes ou acções, ou seja, o traço deixado no ambiente por qualquer acção estimula o desempenho de outra acção, pelo mesmo agente ou por agentes diferentes.

Neste método, cada formiga segue uma lista de exames, E , e para cada exame, e , escolhe um intervalo de tempo, t . Cada formiga começa com uma atribuição vazia. As atribuições parciais, A_i , são construídas probabilisticamente após a construção A_{i-1} . O intervalo de tempo, t , é escolhido aleatoriamente de acordo com a probabilidade, p , que depende da matriz de feromônios, τ , e das informações heurísticas, η .

A função da probabilidade, p , está representada na seguinte função:

$$p_{e,t}(\tau(A_{i-1}), \eta(A_{i-1})) = \frac{(\eta_{(e,t)}(A_{i-1}))^\alpha * (\tau_{(e,t)}(A_{i-1}))^\beta}{\sum_{\theta \in T} (\eta_{(e,\theta)}(A_{i-1}))^\alpha * (\tau_{(e,\theta)}(A_{i-1}))^\beta} \quad (2.4)$$

Um método simples para para calcular as informações heurísticas é o seguinte:

$$\eta_{(e,t)}(A_{i-1}) = \frac{1.0}{1.0 + V_{e,t}(A_{i-1})} \quad (2.5)$$

Onde $V_{e,t}(A_{i-1})$ é o número adicional de violações provocadas pela adição (e,t) à atribuição parcial $A(i-1)$. A função V é uma soma ponderada das restrições *fortes* e *fracas*.

A seguinte regra é utilizada quando A é a atribuição da melhor solução C encontrada desde o início.

$$\tau(e,t) = \begin{cases} (1-p) * \tau_{(e,t)} + 1 & \text{if } A_{globalBest}(e) = t \\ (1-p) * \tau_{(e,t)} & \text{otherwise} \end{cases} \quad (2.6)$$

2.2.3.1 Algoritmo

O algoritmo de Optimização da Colónia de Formigas está representado na figura 2.3.

```

1  Input: Uma instância I
2   $\tau_0 = 1/\rho$ ;
3   $\tau(e,t) = \tau_0$ 
4  while (Tempo_limite != TRUE) {
5      for (a = 1; a == m; a++) {
6          /* Processo de construção da formiga a */
7          A0 = 0;
8          for (i = 1; i == |E|; i++) {
9              /* escolha de um intervalo de tempo aleatório, de
10               acordo com uma probabilidade, para o exame e */
11               $A_i = A(i-1)U\{e,t\}$ 
12          }
13          C = solução;
14          C_melhor_iteração = melhor_(C, C_melhor_iteração);
15      }
16      C_melhor_iteração = solução após aplicar pesquisa
17                          local a C_melhor_iteração;
18      C_melhor_global = melhor_(C_melhor_iteração, C_melhor_global);
19      Atualização de feromônios para  $\tau$  usando C_melhor_global;
20  }
21  Output: Uma solução candidata otimizada, C_melhor_global, para I;

```

Figura 2.3: Algoritmo ACO

2.2.3.2 Parâmetros

Taxa de evaporação, ρ , é um parâmetro cujo intervalo pertence a $[0;1]$ com os seguintes valores atribuídos:

$$\rho \mid 0.2 \mid 0.4 \mid 0.6 \mid 0.8 \mid 0.99$$

O parâmetro α indica a importância do rastreamento de feromônios e os valores atribuídos a este parâmetro são os seguintes:

$$\alpha \mid 0.2 \mid 0.4 \mid 0.6 \mid 0.8 \mid 1$$

As informações heurísticas são representadas pelo parâmetro β e os valores atribuídos são os seguintes:

$$\beta \mid 0.2 \mid 0.4 \mid 0.6 \mid 0.8 \mid 1$$

O parâmetro m indica o número de formigas e apresenta os seguintes valores atribuídos:

$$m \mid 10 \mid 20 \mid \frac{n}{3} \mid \frac{n}{2} \mid n$$

Na tabela 2.4 está representada a melhor configuração dos parâmetros mencionados anteriormente.

Tabela 2.4: Configuração dos parâmetros ACO

Parâmetro	Valor
ρ	0.8
α	1
β	0.4
m	n

2.2.4 Algoritmo Genético

O Algoritmo Genético, *Genetic Algorithm*, foi concebido por Holland, [Pillay and Qu, 2018]. O Algoritmo Genético é uma meta-heurística evolutiva baseada na população, onde todas as soluções são representadas por uma codificação genética, denominada de indivíduo.

Inicialmente, é atribuído aleatoriamente um conjunto de soluções, população, onde os indivíduos são acasalados aleatoriamente, originando a recombinação das suas codificações. A estes novos indivíduos pode ser atribuída uma probabilidade de mutação específica. Deste modo, a nova população é obtida através de um processo que elimina, probabilisticamente, as piores soluções.

A aptidão dos indivíduos é representada por uma função, *fitness function* ($f.f.$), que está relacionada com a função objectivo a otimizar. Esta função quantifica a qualidade da solução para o problema em questão.

Nos Algoritmos Genéticos, os indivíduos também são designados de cromossomas e a posição no cromossoma é designada por gene. O valor que um gene assume é o valor alélico, ou alelo.

P é a população de N cromossomas.

$P(0)$ é a população inicial, atribuída aleatoriamente.

$P(t)$ é a população no instante t .

$P(t+1)$ é originada através de $P(t)$ com operadores genéticos. Os quatro operadores genéticos são os seguintes:

- Reprodução: Atribui à população $P(t+1)$ um número crescente de cópias dos indivíduos que apresentam um valor de aptidão maior que a média da população $P(t)$;
- Selecção dos pais: Os cromossomas dos pais são seleccionados de acordo com a razão de aptidão;
- *Crossover*: O operador genético é activado com probabilidade p . Apresenta como entrada dois indivíduos escolhidos (pais) que combinados originam filhos. Nesta abordagem, o cruzamento de um ou dois pontos é baseado em processos aleatórios;
- Mutação: Operação que, com probabilidade p , provoca a alteração do valor alélico de um gene seleccionado aleatoriamente.

2.2.4.1 Algoritmo

O algoritmo do Algoritmo Genético está representado na figura 2.4.

```

1  Inicializar; /* Esta rotina cria uma população
2      com N indivíduos aleatórios */
3  while (Fim_de_teste != TRUE) {
4      /* Fim de teste é um número de iterações
5      definidas */
6      Calcular f.f. para cada indivíduo;
7      Aplicar 'reprodução';
8      Aplicar 'seleção de pais';
9      Aplicar 'crossover' com probabilidade pc;
10     Aplicar 'mutação' com probabilidade pm;
11 }

```

Figura 2.4: Algoritmo GA

2.2.4.2 Parâmetros

Os valores atribuídos ao parâmetro que representam o tamanho da população (N) são os seguintes:

$$N \mid 10 \mid 50 \mid 100 \mid 200$$

P_m é o parâmetro que representa a probabilidade da taxa de mutação, os valores atribuídos estão a seguir representados:

$$P_m \mid 0.02 \mid 0.1 \mid 0.5$$

P_c é o parâmetro que representa a probabilidade da taxa de *crossover*, os valores atribuídos são os seguintes:

$$P_c \mid 0.05 \mid 0.8 \mid 1$$

A melhor configuração dos parâmetros mencionados está representada na tabela 2.5.

Tabela 2.5: Configuração dos parâmetros GA

Parâmetro	Valor
N	100
P_m	0.02
P_c	0.8

2.3 Algoritmo Genético Baseado em Chaves Aleatórias Tendenciosas

2.3.1 Introdução

Após uma análise cuidadosa, concluiu-se que a meta-heurística Algoritmo Genético é aquela que melhor se enquadra com o problema em questão e permite uma resolução mais eficiente e eficaz deste tipo de problemas, uma vez que foi utilizada com sucesso para problemas semelhantes. Desta forma, foi efectuado um estudo mais pormenorizado do Algoritmo Genético, nomeadamente do Algoritmo Genético de Chaves Aleatórias Tendenciosas.

Um Algoritmo Genético de Chaves Aleatórias Tendenciosas, ou Biased Random-Key Genetic Algorithms (BRKGA), [Gonçalves and Resende, 2011], é uma meta-heurística de pesquisa para obter as melhores soluções para problemas de optimização combinatória.

2.3.2 Descrição da meta-heurística BRKGA

Os Algoritmos Genéticos, [Gonçalves and Resende, 2011], são métodos adaptativos usados para resolver problemas de optimização combinatória, através da associação de soluções de problemas de optimização a indivíduos de uma população. Ao longo de várias gerações, as populações naturais evoluem de acordo com o princípio de selecção natural de Charles Darwin, [Darwin, 1859] citado no artigo [Gonçalves and Resende, 2012], designado como sobrevivência do indivíduo mais apto [Buriol et al., 2010a].

O Algoritmo Genético de Chaves Aleatórias, ou Random-Key Genetic Algorithms (RKGA), foi introduzido por Bean, [Bean, 1994] citado em [Gonçalves and Resende, 2015], para resolver problemas de sequenciamento. Num RKGA, os cromossomas são representados como vectores de números reais gerados aleatoriamente no intervalo $[0, 1]$. Um decodificador é um algoritmo determinístico que recebe como entrada um cromossoma e resulta numa solução com um valor objectivo.

No Algoritmo Genético de Chaves Aleatórias, uma população de vectores de chave aleatória evolui ao longo de várias gerações (iterações). A população inicial é composta por P vectores e R chaves aleatórias. Cada componente do vector da solução, ou chave aleatória, é gerado independentemente e aleatoriamente no intervalo real $[0, 1]$. Depois, a aptidão de cada indivíduo é calculada pelo decodificador na geração G , a população é dividida em dois grupos de indivíduos: um pequeno grupo de indivíduos Elite, P_e , ou seja, aqueles com os melhores valores de aptidão e o restante conjunto, $P - P_e$, de indivíduos Não Elite. Para evoluir a população da geração G é produzida uma nova geração ($G + 1$) de indivíduos. Todos os indivíduos Elite da população da geração G são copiados, sem modificação, para a população da geração $G + 1$. Os RKGA implementam a mutação introduzindo mutantes na população.

Um mutante é um vector de chaves aleatórias que é gerado da mesma maneira que um elemento da população inicial. O seu objectivo é idêntico à mutação no Algoritmo Genético por Goldberg, [Goldberg, 1989] no artigo [Gonçalves and Resende, 2012], ou seja, introduzir um ruído na população e evitar a convergência de toda a população para um óptimo local. A cada geração,

um pequeno número de mutantes, P_m , é introduzido na população. Com os indivíduos $P_e + P_m$ representados na população $G + 1$, $P - P_e - P_m$ indivíduos precisam de ser reproduzidos para completar os indivíduos P que compõem a população $G + 1$. Essa reprodução é realizada através da produção de soluções de filhos $P - P_e - P_m$ com o processo de acasalamento, ou cruzamento.

Tabela 2.6: Parâmetros do BRKGA

Parâmetro	Descrição
n	Número de alelos por cromossoma
P	Número de cromossomas numa população
P_e	Percentagem de indivíduos Elite numa população
P_m	Percentagem de indivíduos mutantes introduzidos na população em cada geração
ρ	Probabilidade de um alelo ser do pai do grupo Elite

Um Algoritmo Genético de Chaves Aleatórias Tendenciosas, ou BRKGA, difere de um Algoritmo Genético de Chaves Aleatórias, RKGA, na maneira como os pais são seleccionados para a reprodução e como o cruzamento, ou acasalamento, é realizado. Enquanto no RKGA de Bean, [Bean, 1994] citado em [Gonçalves and Resende, 2015], ambos os pais são seleccionados aleatoriamente em toda a população actual, no BRKGA cada elemento é gerado combinando um pai seleccionado aleatoriamente da população Elite na população actual e outro indivíduo da restante população. A repetição na selecção de um pai é permitida e, portanto, um indivíduo pode produzir mais de um filho na mesma geração.

Quando a próxima população estiver completa, ou seja, quando houver P indivíduos, os valores de aptidão são calculados para todos os vectores de chaves aleatórias recém-criados e a população é dividida em indivíduos de Elite e de Não Elite para iniciar uma nova geração.

2.3.2.1 Processo evolutivo

Para especificar um Algoritmo Genético de Chaves Aleatórias Tendenciosas, é necessário especificar os seus parâmetros, como as soluções são codificadas e decodificadas e como são calculados os valores da função de aptidão correspondentes.

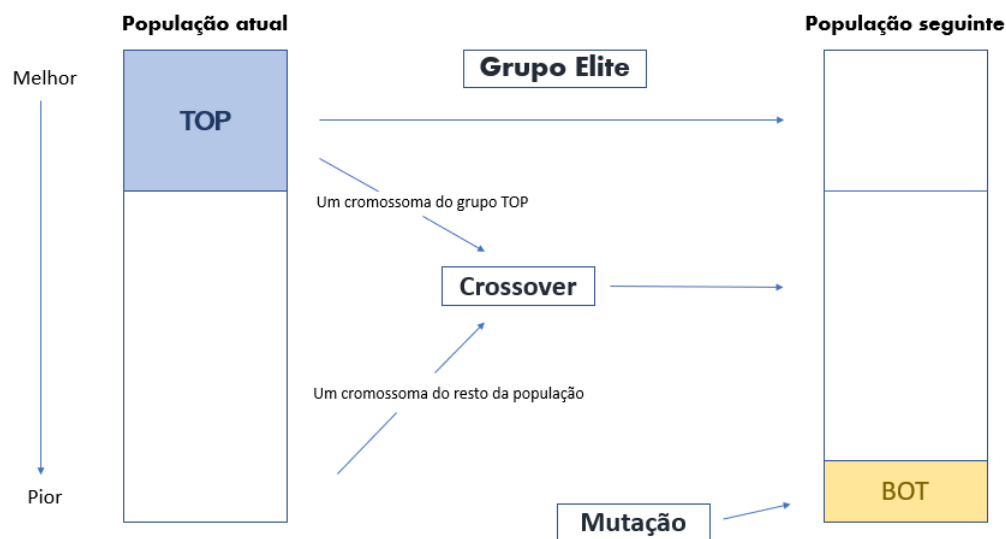


Figura 2.5: Evolução do BRKGA

A ilustração da figura 2.5 exemplifica a dinâmica do processo evolutivo entre gerações consecutivas do Algoritmo Genético de Chaves Aleatórias Tendenciosas. A população actual é ordenada do indivíduo melhor classificado para o pior classificado através da função de aptidão. Os indivíduos melhor classificados são colocados no conjunto TOP, equivalente ao grupo Elite, enquanto que os outros são colocados no conjunto REST. Os indivíduos do grupo Elite são copiados sem alteração para a próxima população.

Os indivíduos mutantes são gerados aleatoriamente de maneira semelhante à população inicial e uma pequena percentagem é colocada no conjunto BOT na próxima população. Os indivíduos restantes da próxima população são gerados através da operação de cruzamento com indivíduos seleccionados aleatoriamente do conjunto TOP na população actual e indivíduos seleccionados aleatoriamente do conjunto REST da população actual.

Como exigimos que $P_e < P - P_e$, a probabilidade de um determinado indivíduo de Elite ser seleccionado para cruzamento, $\frac{1}{P_e}$, é maior do que um indivíduo de Não Elite, $\frac{1}{P - P_e}$, e, portanto, um indivíduo de Elite tem maior probabilidade de transmitir suas características para as gerações futuras do que um indivíduo que pertence ao grupo Não Elite.

Também contribui para este fim o cruzamento uniforme parametrizado por Spears e DeJong, [Spears and DeJong, 1991] citado no artigo [Gonçalves and Resende, 2015], o mecanismo usa a combinação nos BRKGA e o facto de que um dos pais é sempre seleccionado do grupo Elite. Seja $P_e > 0,5$ o parâmetro escolhido, este parâmetro é a probabilidade de um filho herdar o alelo de seu pai de Elite.

Seja n o número de genes no cromossoma de um indivíduo. Para $i = 1, \dots, n$, o i -ésimo alelo $f(i)$ do filho, f , assume o valor do i -ésimo alelo $p(i)$ do pai de Elite, p , com probabilidade P_e e o valor de o i -ésimo alelo $q(i)$ do pai Não Elite, q , com probabilidade $1 - P_e$. Dessa maneira, é mais provável que os filhos herdem características dos pais de Elite do que dos pais Não Elite.

Como é assumido que qualquer vector de chaves aleatórias pode ser decodificado numa solução, o filho resultante do cruzamento é sempre válido, ou seja, pode ser decodificado numa solução do problema de optimização combinatória.

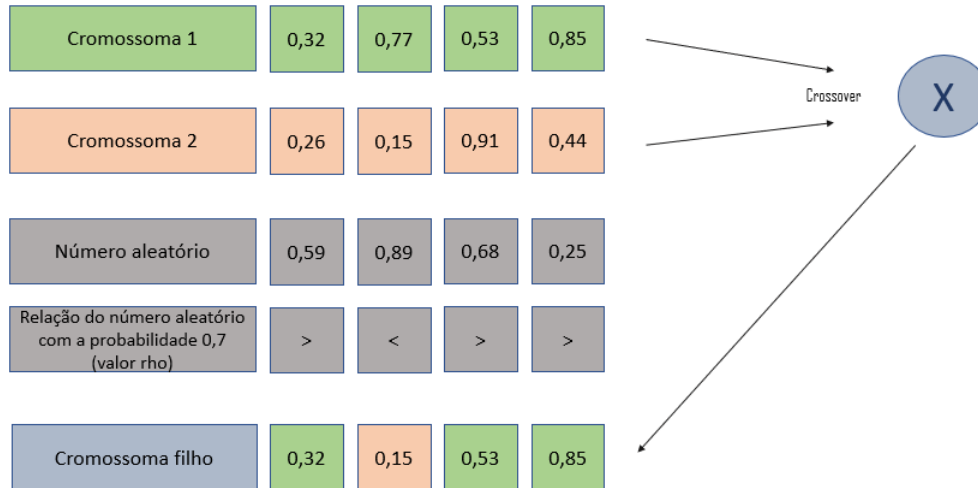


Figura 2.6: Exemplo de um cruzamento no BRKGA

A figura 2.6 ilustra o processo de cruzamento para dois vectores de chaves aleatórias com quatro genes cada um.

O cromossoma 1 refere-se a um indivíduo do grupo Elite e o cromossoma 2 pertence ao grupo Não Elite. Neste exemplo, $P_e = 0,7$, ou seja, o filho herda o alelo do progenitor de Elite com probabilidade 0,7 e do outro progenitor com probabilidade 0,3.

Um número real gerado aleatoriamente no intervalo $[0, 1]$ simula o lançamento de uma moeda tendenciosa. Se o resultado for menor ou igual a 0,7, o filho herda o alelo do pai de Elite. Caso contrário, o filho herda o alelo do outro pai. Neste exemplo, o filho herda o alelo do pai de Elite no primeiro, terceiro e quarto gene. Apresenta mais características do pai de Elite do que do outro pai. Quando a próxima população estiver concluída, ou seja, quando houver P indivíduos, os valores de aptidão serão calculados para todos os vectores de chaves aleatórias recém-criados e a população será dividida em indivíduos de Elite e Não Elite para iniciar uma nova geração.

Os BRKGA iniciam e posteriormente desenvolvem uma população contendo exactamente P cromossomas, cada um com N chaves aleatórias, por várias gerações, até que algum critério de paragem seja respeitado.

Resumindo, numa determinada geração G , esse processo evolutivo é guiado pelas seguintes etapas, descritas em [Gonçalves and Resende, 2012]:

- Cada cromossoma é decodificado para revelar a sua função de aptidão, isto é, o valor objectivo do problema de optimização. Um cromossoma quando associado à sua função de aptidão ou função *fitness* é chamado de indivíduo;

- A população é dividida em dois grupos: um conjunto Elite contendo indivíduos com os melhores valores de aptidão e um conjunto Não Elite com os restantes indivíduos;
- A população da geração $G + 1$ consistirá em:
 - Indivíduos do conjunto de Elite da geração actual;
 - Cromossomas gerados aleatoriamente designados mutantes;
 - $P_o = P - P_m - P_e$, com P igual ao tamanho da população, P_m o número de indivíduos mutantes, P_e igual ao número de indivíduos do grupo Elite e P_o o número de cromossomas produzidos por cruzamento de dois pais da geração actual, seleccionados aleatoriamente, um do conjunto de Elite e outro do conjunto de Não Elite. Para definir o j -ésimo alelo do filho i , é gerado um número aleatório uniforme $r_{ij} \in [0, 1)$: se $r_{ij} \leq \rho_e$, com $\rho_e > 0,7$, o filho i herda o j -ésimo alelo de seu pai de Elite, caso contrário, ele herda o j -ésimo alelo dos pais não pertencentes à Elite.

2.3.2.2 Representação dos cromossomas e decodificação

Os Algoritmos Genéticos, como já referido, adequadamente codificados são capazes de desenvolver soluções para problemas de optimização [Gonçalves and Resende, 2012]. Uma codificação, ou representação, para o problema é planeado e depois é definido um algoritmo genético. Também é necessária uma função de aptidão ou função *fitness*, que atribui um valor a cada solução codificada. Os pais são seleccionados para reprodução e recombinação para gerar filhos.

Num Algoritmo Genético, uma solução é codificada como um conjunto de parâmetros, conhecidos como genes, unidos para formar uma cadeia de valores designada cromossoma. O conjunto de parâmetros representados num determinado cromossoma é designado como indivíduo.

A função aptidão, ou função *fitness*, de um indivíduo depende da representação do seu cromossoma. Isto é, durante a fase de reprodução, os indivíduos da população são seleccionados e recombinação entre si, produzindo filhos, que compõem a população da próxima geração. Os pais são seleccionados aleatoriamente da população, dando prioridade aos indivíduos mais aptos. Uma vez seleccionados, os cromossomas dos dois pais são recombinação, através de mecanismos de cruzamento. A mutação é ocasionalmente aplicada a alguns indivíduos para garantir a diversidade da população.

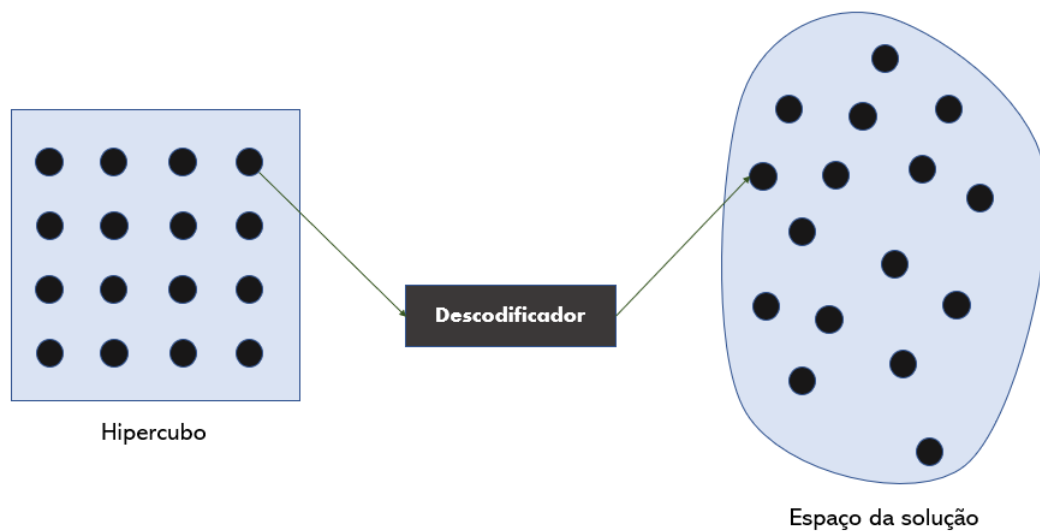


Figura 2.7: Hiper-cubo do BRKGA

A figura 2.7 ilustra o papel do decodificador. As heurísticas BRKGA são baseadas numa estrutura meta-heurística genérica.

Um BRKGA pesquisa indirectamente no espaço da solução do problema de optimização combinatória, através do Hiper-cubo r -dimensional contínuo, usando o decodificador para mapear soluções no espaço da solução do problema de optimização combinatória em que a função de aptidão é avaliada [Gonçalves and Resende, 2011].

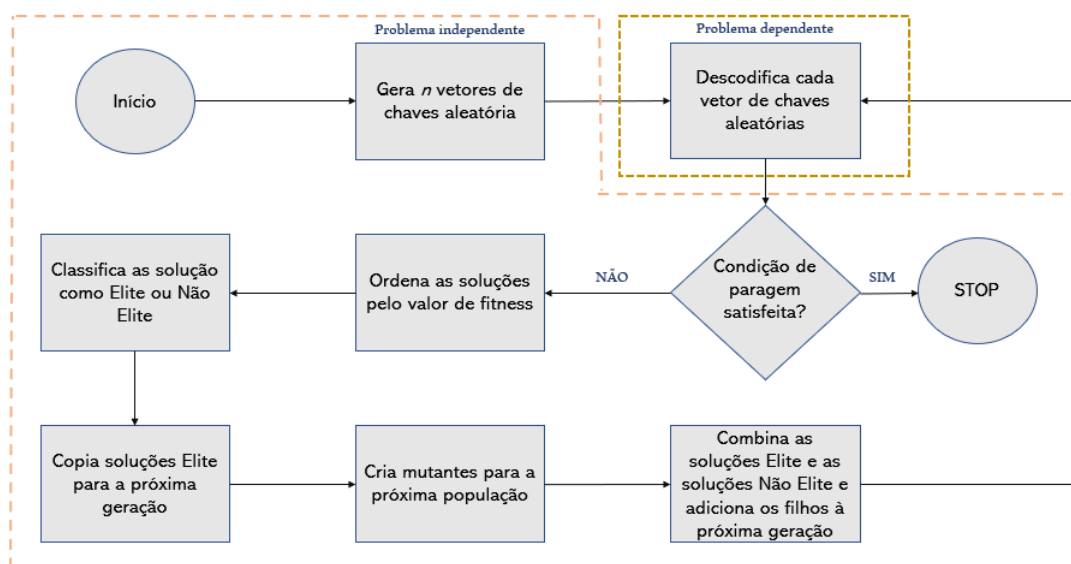


Figura 2.8: Processo do BRKGA

Na estrutura da figura 2.8, existe uma divisão entre a parte independente do algoritmo e a parte

dependente do problema. A parte independente do problema não tem conhecimento do problema que está a ser resolvido. É limitado a pesquisar no Hipercubo. A única conexão com o problema de optimização combinatória que está a ser resolvido é a parte dependente do problema do algoritmo, onde o decodificador produz soluções a partir dos vectores de chaves aleatórias e calcula o valor dessas soluções.

Portanto, para especificar uma heurística BRKGA, é necessário definir a representação dos cromossomas e o decodificador.

2.3.2.3 Função de Aptidão

Para avaliar a qualidade de uma solução para o processo evolutivo, é necessário definir uma medida de aptidão da solução ou medida de qualidade. A função de condicionamento natural para esse tipo de problema é a percentagem do valor total dos conflitos.

2.3.2.4 Estratégia de multi população

A estratégia de multi população consiste em desenvolver várias populações independentes em paralelo. Após um número predeterminado de gerações, todas as populações trocam cromossomas de boa qualidade.

No entanto, nesta dissertação apenas foi desenvolvida uma população.

2.3.3 Exemplos e comparação dos Algoritmos Genéticos

Embora os BRKGA usem apenas chaves geradas aleatoriamente, são mais eficazes para encontrar melhores soluções do que um algoritmo puramente aleatório.

Podemos analisar, no exemplo demonstrado em [\[Gonçalves and Resende, 2011\]](#), a variação dos valores das funções objectivo para uma população de 100 indivíduos, e as suas consequentes gerações, para diferentes Algoritmos Genéticos.

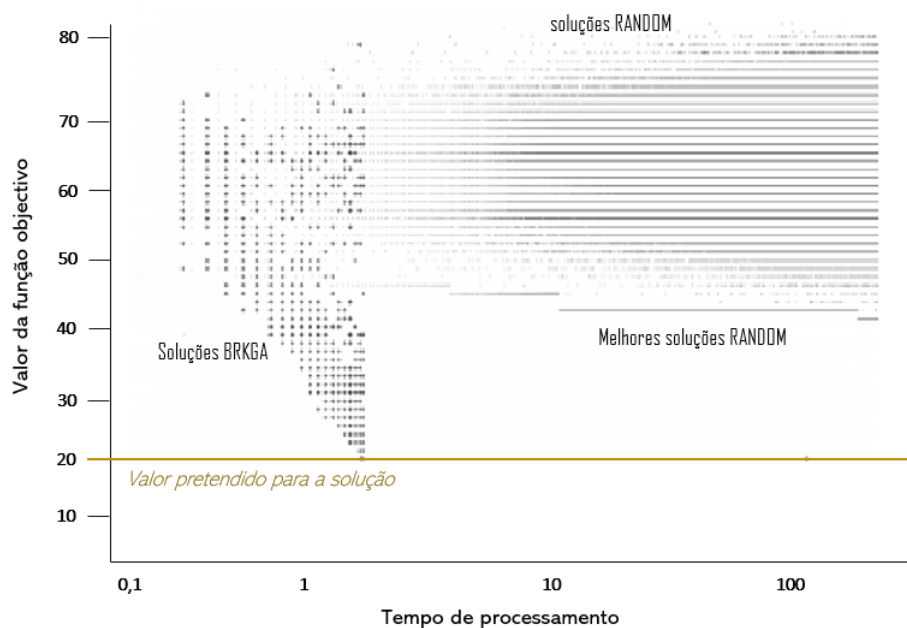


Figura 2.9: Gráfico com a distribuição do valor das funções objectivo da população, adaptado do exemplo no artigo [Gonçalves and Resende, 2011]

Esta problema tem como objectivo obter uma função objectivo com o valor 21. Observa-se que, enquanto o BRKGA encontra rapidamente a solução pretendida em 2 segundos, a heurística com as melhores soluções *Random*, após 1000 segundos, ainda está muito longe da ideal, tendo encontrado apenas a melhor solução com valor 38.

Como discutido anteriormente, um Algoritmo Genético de Chaves Aleatórias Tendenciosas e um Algoritmo Genético de Chaves Aleatórias diferem ligeiramente na maneira como seleccionam os pais para a reprodução. A variante tendenciosa selecciona um dos pais do conjunto de soluções de Elite, enquanto a variante imparcial selecciona os dois pais de toda a população.

Dessa forma, os filhos produzidos pela versão tendenciosa têm mais probabilidade de herdar características das soluções de Elite. Essa probabilidade é ainda mais enfatizada através do cruzamento uniforme parametrizado, usado nos dois métodos, para combinar os pais e produzir o filho.

Embora essa seja, aparentemente, apenas uma diferença muito pequena, normalmente resulta numa grande diferença do desempenho. Os BRKGA tendem a encontrar soluções melhores que os RKGA, com o mesmo tempo de execução, têm uma probabilidade muito maior de encontrar uma solução com um valor de solução pretendido em menos tempo.

Para ilustrar isso, a figura 2.10 mostra o gráfico com a probabilidade de cada meta-heurística encontrar o valor desejado em função do tempo.

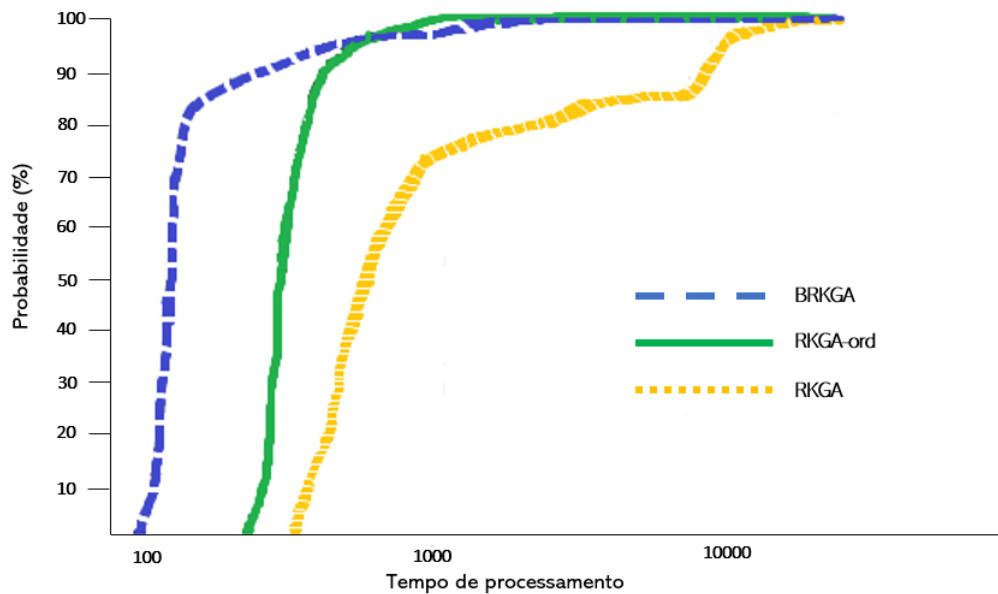


Figura 2.10: Gráfico da probabilidade dos algoritmos encontrarem a solução pretendida, adaptado do exemplo no artigo [Gonçalves and Resende, 2011]

O gráfico compara a probabilidade do algoritmo encontrar a solução pretendida com o tempo de processamento. Esta probabilidade foi obtida através de 200 execuções independentes para um BRKGA, um RKGA e uma heurística RKGA-ord, que é semelhante a um RKGA, com a exceção que o filho herda o alelo do melhor dos dois pais com uma determinada probabilidade maior que 0,5.

A figura mostra claramente que o BRKGA encontra as soluções pretendidas em menos tempo. Por exemplo, 400 segundos é aproximadamente o tempo que o RKGA demora para resolver uma tentativa. A ordenação dos pais, como é feito no RKGA-ord, melhora o RKGA, mas não o suficiente para fazer melhor que o BRKGA. Por exemplo, 216 segundos é o tempo que o RKGA-ord leva para resolver qualquer uma das suas 200 tentativas, nesse tempo, o BRKGA resolve 176 das suas 200 tentativas.

Embora ilustrado para apenas uma instância deste tipo de problemas, este comportamento é típico para uma ampla gama de problemas [Gonçalves and Resende, 2011].

Capítulo 3

Descrição do problema

Actualmente, o planeamento de júris de mestrado, do Departamento de Engenharia e Gestão Industrial (DEGI) da Faculdade de Engenharia da Universidade do Porto (FEUP), é realizado manualmente através de uma análise de um conjunto de dados fornecidos pelos júris, alunos e instituição.

No Mestrado Integrado em Engenharia e Gestão Industrial (MIEGI), no final do ano lectivo, cerca de 70 alunos defendem as suas dissertações no período de duas semanas, originando um elevado número de opções para alocar cada defesa de dissertação num determinado intervalo de tempo, numa respectiva sala e com a presença de um determinado júri.

Deste modo, o planeamento de júris de mestrado é um processo complexo que deve cumprir com determinados requisitos e, ao mesmo tempo, ter em atenção as restrições impostas. Assim sendo, a construção manual deste calendário é um processo demorado.

Esta dissertação tem como objectivo o desenvolvimento de uma ferramenta que realize o planeamento automático de júris de mestrado com uma solução aceitável e o mais eficiente possível.

Para este processo, existem dois tipos de restrições, as restrições *fortes*, que têm que ser cumpridas obrigatoriamente, e as restrições *fracas* que, apesar de não serem obrigatórias, devem ser cumpridas o máximo possível.

Assim sendo, a primeira fase consiste em encontrar uma solução viável, que respeite todas as restrições *fortes*. Posteriormente, a ferramenta em questão deve fazer cumprir o máximo de restrições *fracas* de modo a encontrar uma melhor solução final, minimizando o número total de conflitos.

3.1 Parâmetros

Para a resolução deste problema, é necessário ter em conta um conjunto de parâmetros associados aos intervenientes do processo.

Dissertações

As dissertações, realizadas por alunos, são classificadas com uma ou mais palavras-chave, que distingue o tema da dissertação, e identificadas com o respectivo orientador.

Orientadores

Cada orientador está sujeito a uma determinada disponibilidade horária.

Júris

Os júris são compostos por um presidente de júri e um arguente externo que têm em consideração os seguintes factores:

- Disponibilidade horária;
- Palavras-chave, que identifica os temas que o júri avalia;
- Prioridade de preferência pela sala para a defesa da dissertação;
- Número de dissertações a presidir.

Salas

Cada sala, para a defesa das dissertações, tem as seguintes características:

- Disponibilidade horária;
- Ordem de preferência, que classifica as salas em termos de qualidade para efectuar uma apresentação;

Blocos

Um bloco é constituído por 3 dissertações com pelo menos uma palavra-chave em comum, presididas pelo mesmo júri, presidente de júri e respectivo arguente externo, e na mesma sala. Cada bloco é dividido em 3 períodos, a cada período é associada uma dissertação.

Intervalos de tempo

Em cada dia existem 3 intervalos de tempo, um de manhã e dois de tarde, com duração de 02:30 horas onde, a cada intervalo de tempo, podem ser atribuídos um ou mais blocos. A disponibilidade horária para marcação das defesas de dissertação é a seguinte:

- 09:30h até 12:00h;
- 13:30h até 16:00h;
- 16:30h até 19:00h;

Excepcionalmente, no caso de existir um ou mais blocos com um número de dissertações diferente de 3, a duração dos intervalos de tempo é adaptada.

Época de avaliação

A época de avaliação das dissertações é realizada em duas semanas, 10 dias úteis.

3.2 Objectivos

O objectivo deste problema é obter um planeamento compacto, que cumpra com todas as restrições *fortes* impostas e garanta o maior número de restrições *fracas* possível, de modo a minimizar o número de conflitos.

3.3 Restrições

As restrições são agrupadas em restrições *fortes* e restrições *fracas*, como já referido anteriormente. Para o problema em questão:

Restrições fortes, que têm que ser cumpridas de forma a obter uma solução admissível:

- As dissertações têm de estar agrupadas em blocos constituídos por 3 dissertações;
- Cada apresentação tem que ser realizada num período com uma sala disponível;
- Garantir a disponibilidade do orientador;
- Garantir a disponibilidade do júri, presidente de júri e arguente;
- Um professor não pode ser presidente de júri e orientador da mesma dissertação;
- Um orientador não pode pertencer a dois blocos em simultâneo, ou seja, não pode pertencer a dois blocos que sejam compostos pelo mesmo intervalo tempo;
- Um júri, constituído pelo presidente de júri e arguente, não pode pertencer a dois blocos em simultâneo;
- Um professor não pode estar presente como orientador num bloco e ser presidente de júri de outro bloco no mesmo intervalo de tempo;
- Cada sala só pode ter um bloco num determinado período de tempo;
- Dar prioridade às melhores salas;
- Limite máximo de dissertações a presidir por cada júri tem de ser respeitado.

Restrições fracas que, se cumpridas, melhoram a solução final:

- Limite mínimo de dissertações a presidir por cada júri tem de ser respeitado.
- Cada orientador não deve pertencer a mais de 2 blocos no mesmo dia;
- Cada júri, presidente de júri e arguente, não deve avaliar mais de 1 bloco no mesmo dia;
- Um professor não deve ser presidente de júri de 1 bloco e pertencer a mais de 1 bloco como orientador no mesmo dia.

Capítulo 4

Modelo proposto para o problema de planeamento de júris de mestrado

Para o planeamento de júris de mestrado foi elaborado um modelo matemático para a resolução do problema em questão, tendo como base a descrição do problema apresentada no capítulo anterior.

4.1 Índices e Parâmetros

$d = \{0, \dots, D\}$	Índice para o conjunto $\mathcal{D}$ das dissertações
$b = \{0, \dots, B\}$	Índice para o conjunto $\mathcal{B}$ dos blocos
$o = \{0, \dots, O\}$	Índice para o conjunto $\mathcal{O}$ dos orientadores
$j = \{0, \dots, J\}$	Índice para o conjunto $\mathcal{J}$ dos júris
$t = \{0, \dots, T\}$	Índice para o conjunto $\mathcal{T}$ dos intervalos de tempo disponíveis para apresentação das dissertações
$h = \{0, \dots, H\}$	Índice para o conjunto $\mathcal{H}$ dos dias disponíveis para apresentação das dissertações
$s = \{0, \dots, S\}$	Índice para o conjunto $\mathcal{S}$ das salas
MIN_j	Número mínimo de dissertações a avaliar pelo júri j
MAX_j	Número máximo de dissertações a avaliar pelo júri j
PRI_{js}	Disponibilidades do júri j na sala s ($= 0$) ou não ($= 1$)
AVA_{dj}	Disponibilidades da dissertação d ser avaliada pelo júri j ($= 0$) ou não ($= 1$)
$DISO_{ot}$	Disponibilidades do orientador o no intervalo de tempo t ($= 0$) ou não ($= 1$)
$DISJ_{jt}$	Disponibilidades do júri j no intervalo de tempo t ($= 0$) ou não ($= 1$)
$DISS_{st}$	Disponibilidades da sala s no intervalo de tempo t ($= 0$) ou não ($= 1$)

4.2 Variáveis de decisão

x_{bd}	= 1 se o bloco b contém a dissertação d
y_{dj}	= 1 se a dissertação d é avaliada pelo júri j
k_{bot}	= 1 se no bloco b está alocado o orientador o no intervalo de tempo t
w_{bjt}	= 1 se o bloco b é alocado ao júri j no intervalo de tempo t
z_{bst}	= 1 se o bloco b foi alocado à sala s no intervalo de tempo t
e_{boh}	= 1 se o orientador o pertence ao bloco b no dia h
u_{bjh}	= 1 se o júri j pertence ao bloco b no dia h

4.3 Restrições fortes

As restrições fortes devem ser cumpridas obrigatoriamente de forma a obter uma solução viável.

Cada bloco deverá conter 3 dissertações:

$$\forall b \in \mathcal{B} \quad \sum_{d \in \mathcal{D}} x_{db} = 3 \quad (4.1)$$

Cada dissertação num bloco tem de ser avaliada por um júri com pelo menos uma palavra-chave comum a cada dissertação:

$$\forall j \in \mathcal{J} \quad \sum_{d \in \mathcal{D}} y_{dj} * AVA_{dj} = 0 \quad (4.2)$$

Disponibilidade do orientador:

$$\forall o \in \mathcal{O} \quad \forall t \in \mathcal{T} \quad \sum_{b \in \mathcal{B}} k_{bot} * DISO_{ot} = 0 \quad (4.3)$$

Disponibilidade do júri, presidente de júri e arguente:

$$\forall j \in \mathcal{J} \quad \forall t \in \mathcal{T} \quad \sum_{b \in \mathcal{B}} w_{bjt} * DISJ_{jt} = 0 \quad (4.4)$$

Cada bloco de apresentação tem de ser realizado num período com uma sala disponível:

$$\forall s \in \mathcal{S} \quad \forall t \in \mathcal{T} \quad \sum_{b \in \mathcal{B}} z_{bst} * DISS_{st} = 0 \quad (4.5)$$

Um orientador só pode pertencer a um bloco num determinado intervalo de tempo:

$$\forall o \in \mathcal{O} \quad \forall t \in \mathcal{T} \quad \sum_{b1=1}^{B-1} \sum_{b2=b1+1}^B k_{b1ot} * k_{b2ot} = 0 \quad (4.6)$$

Um júri só pode pertencer a um bloco num determinado intervalo de tempo:

$$\forall j \in \mathcal{J} \quad \forall t \in \mathcal{T} \quad \sum_{b1=1}^{B-1} \sum_{b2=b1+1}^B w_{b1jt} * w_{b2jt} = 0 \quad (4.7)$$

Cada sala só pode ter um bloco num determinado intervalo de tempo:

$$\forall s \in \mathcal{S} \quad \forall t \in \mathcal{T} \quad \sum_{b1=1}^{B-1} \sum_{b2=b1+1}^B z_{b1st} * z_{b2st} = 0 \quad (4.8)$$

Um júri não pode avaliar mais teses do que o seu número máximo:

$$\forall j \in \mathcal{J} \quad \sum_{d \in \mathcal{D}} y_{dj} \leq MAX_j \quad (4.9)$$

4.4 Restrições fracas

As restrições fracas que, ao serem cumpridas, melhoram a solução final do problema.

Um júri tem de avaliar um número mínimo de dissertações:

$$\forall j \in \mathcal{J} \quad \sum_{d \in \mathcal{D}} y_{dj} \geq MIN_j \quad (4.10)$$

Respeitar as prioridades das salas:

$$\forall s \in \mathcal{S} \quad \forall j \in \mathcal{J} \quad \sum_{b \in \mathcal{B}} i_{bjs} * PRI_{js} = 0 \quad (4.11)$$

Cada orientador deverá pertencer no máximo a 2 blocos por dia:

$$\forall o \in \mathcal{O} \quad \forall h \in \mathcal{H} \quad \sum_{b1=1}^{B-2} \sum_{b2=b1+1}^{B-1} \sum_{b3=b2+1}^B e_{b1oh} * e_{b2oh} * e_{b3oh} = 0 \quad (4.12)$$

Cada júri, presidente de júri e arguente, deverá pertencer a exclusivamente 1 bloco por dia:

$$\forall j \in \mathcal{J} \quad \forall h \in \mathcal{H} \quad \sum_{b1=1}^{B-1} \sum_{b2=b1+1}^B u_{b1jh} * u_{b2jh} = 0 \quad (4.13)$$

4.5 Função Objectivo

min:

(3 dissertações por bloco)

$$\sum_{b=1}^B \left| \left[\sum_{d=1}^D x_{db} \right] - 3 \right| \quad (4.14)$$

+

(Capacidade do júri para avaliar todos as dissertações do bloco)

$$\sum_{j=1}^J \sum_{d=1}^D y_{dj} * AVA_{dj} \quad (4.15)$$

+

(Disponibilidade do orientador)

$$\sum_{o=1}^O \sum_{t=1}^T \sum_{b=1}^B k_{bot} * DISO_{ot} \quad (4.16)$$

+

(Disponibilidade do júri)

$$\sum_{j=1}^J \sum_{t=1}^T \sum_{b=1}^B w_{bjt} * DISJ_{jt} \quad (4.17)$$

+

(Disponibilidade da sala)

$$\sum_{s=1}^S \sum_{t=1}^T \sum_{b=1}^B z_{bst} * DISS_{st} \quad (4.18)$$

+

(Orientador só num bloco num determinado intervalo de tempo)

$$\sum_{o=1}^O \sum_{t=1}^T \sum_{b1=1}^{B-1} \sum_{b2=b1+1}^B k_{b1ot} * k_{b2ot} \quad (4.19)$$

+

(Júri só num bloco num determinado intervalo de tempo)

$$\sum_{j=1}^J \sum_{t=1}^T \sum_{b1=1}^{B-1} \sum_{b2=b1+1}^B w_{b1jt} * w_{b2jt} \quad (4.20)$$

+

(Sala só num bloco num determinado intervalo de tempo)

$$\sum_{s=1}^S \sum_{t=1}^T \sum_{b1=1}^{B-1} \sum_{b2=b1+1}^B z_{b1st} * z_{b2st} \quad (4.21)$$

+

(Um júri não avaliar mais do que o número máximo de dissertações)

$$\sum_{j=1}^J \frac{|\sum_{b=1}^B y_{dj} - MAX_j|}{\sum_{b=1}^B y_{dj} - MAX_j} \quad (4.22)$$

+

(Um júri não avaliar menos do que o número mínimo de dissertações)

$$\sum_{j=1}^J \frac{|MIN_j - \sum_{b=1}^B y_{dj}|}{MIN_j - \sum_{b=1}^B y_{dj}} \quad (4.23)$$

+

(Respeitar as prioridades das salas)

$$\sum_{j=1}^J \sum_{s=1}^S \sum_{b=1}^B i_{bjs} * PRI_{js} \quad (4.24)$$

+

(Orientador avaliar no máximo 2 dissertações por dia)

$$\sum_{o=1}^O \sum_{h=1}^H \sum_{b1=1}^{B-2} \sum_{b2=b1+1}^{B-1} \sum_{b3=b2+1}^B e_{b1oh} * e_{b2oh} * e_{b3oh} \quad (4.25)$$

+

(Júri avaliar no máximo 1 dissertações por dia)

$$\sum_{j=1}^J \sum_{h=1}^H \sum_{b1=1}^{B-2} \sum_{b2=b1+1}^{B-1} \sum_{b3=b2+1}^B u_{b1jh} * u_{b2jh} * u_{b3jh} \quad (4.26)$$

Para o cálculo da função objectivo deste problema são utilizadas as expressões da função objectivo, da 4.14 à 4.26. Como o objectivo é obter o valor mínimo possível, quando uma restrição não for cumprida, incrementa-se um valor à função objectivo.

A primeira expressão, 4.14, diz respeito ao número de dissertações num bloco. Os valores que influenciam o resultado da função objectivo estão de acordo com a tabela 4.1.

Tabela 4.1: Valores da componente 4.14 da função objectivo

Número de dissertações num bloco	Resultado
0	3
1	2
2	1
3	0
4	1
5	2

A expressão seguinte, 4.15, é utilizada para verificar a capacidade e a viabilidade do júri avaliar todas as dissertações do bloco a que foi atribuído. Sempre que uma dissertação for avaliada por um júri, $y = 1$, sem capacidade para tal, $AVA = 1$, consequentemente, será incrementado um valor à função objectivo.

As expressões 4.16, 4.17 e 4.18 dizem respeito à disponibilidade dos júris, orientadores e das salas. Para estas componentes, pretende-se verificar se alguma sala, júri ou orientador foi alocado num intervalo de tempo indisponível. Deste modo, sempre que esse conflito acontecer, o resultado da multiplicação será incrementado à função objectivo.

As três parcelas seguintes, expressões 4.19, 4.20 e 4.21, são relativas à existência de algum conflito num determinado intervalo de tempo. Como o caso do mesmo orientador ou júri ser alocado para diferentes blocos no mesmo intervalo de tempo, ou a mesma sala seleccionada para mais que um bloco no mesmo intervalo de tempo. O método para o cálculo na função objectivo é executado da mesma maneira que nas expressões anteriores, 4.16, 4.17 e 4.18.

As expressões seguintes, 4.22 e 4.23, são executadas para calcular a relação entre o número de dissertações que um júri avalia e o número máximo e mínimo previamente definido. Em ambas as componentes, para o caso da restrição ser cumprida será decrementado um valor à função objectivo, caso contrário será incrementado.

Na tabela 4.2 estão detalhados os valores a atribuir à função objectivo de acordo com o número máximo de dissertações a avaliar por cada júri.

Tabela 4.2: Valores da componente 4.22 da função objectivo

Número máximo	Número dissertações	Resultado
3	2	-1
3	3	-1
3	4	1

Na tabela 4.3 estão exemplificados os valores a atribuir à função objectivo relativamente ao número mínimo de dissertações a avaliar por cada júri.

Tabela 4.3: Valores da componente 4.23 da função objectivo

Número mínimo	Número dissertações	Resultado
3	2	1
3	3	-1
3	4	-1

A parcela seguinte, 4.24, calcula a relação entre a prioridade das salas e a prioridade dos júris. Um determinado júri não pode avaliar dissertações em salas com valor de prioridade inferior. A variável auxiliar PRI toma o valor de 1 se existe conflito no júri j e sala s . Nesse caso, será incrementado um valor à função objectivo.

As últimas duas parcelas, 4.25 e 4.26, têm como objectivo não permitir que um júri avalie mais que um bloco no mesmo dia, e que um orientador não pertença a mais de dois blocos num dia. O valor da função objectivo aumenta para o caso da restrição não ser cumprida.

Capítulo 5

Procedimento computacional

Neste capítulo abordam-se os dados do problema, as diferentes restrições e a função objectivo em termos de linguagem algorítmica para as duas abordagens realizadas.

Cada abordagem está dividida em duas fases, a primeira fase onde são criados os blocos, de preferência com três dissertações, avaliadas pelo mesmo júri. Na segunda fase, definida como planeamento final, ocorre a atribuição de cada bloco a um respectivo intervalo de tempo e uma sala, garantindo as restrições impostas inicialmente.

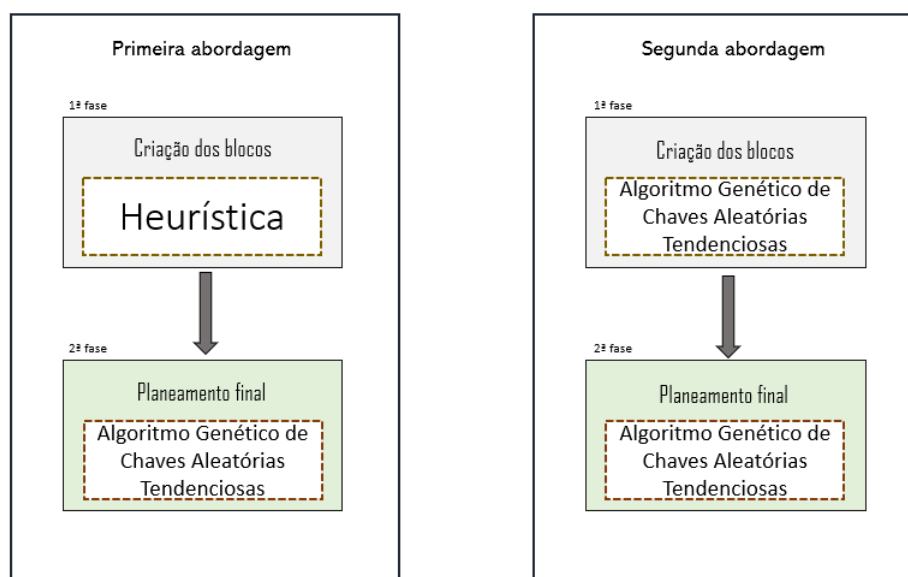


Figura 5.1: Diferentes abordagens para o planeamento de júris de mestrado

Como se observa na figura 5.1, na primeira abordagem, a criação dos blocos para defesa das dissertações é realizada através de uma heurística, descrita na secção 5.1, posteriormente, o planeamento final é realizado através do Algoritmo Genético de Chaves Aleatórias Tendenciosas.

Na segunda abordagem, o Algoritmo Genético de Chaves Aleatórias Tendenciosas é executado duas vezes, inicialmente, para a elaboração dos blocos e, de seguida, para o planeamento final.

Em ambas as fases, o processo para o planeamento das defesas de dissertação é dividido em duas fases, isso acontece devido à complexidade do Algoritmo Genético de Chaves Aleatórias Tendenciosas, em simultâneo, atribuir os júris às dissertações, criar os blocos, alocar os blocos nos intervalos de tempo e nas salas e ainda garantir o cumprimento de outras restrições.

5.1 Primeira abordagem

Esta abordagem é composta por duas fases: uma heurística que determina os blocos de dissertações atribuídas a cada júri e, em seguida, são definidos os horários para cada bloco.

5.1.1 Heurística da primeira abordagem

Na primeira fase, são agrupadas as dissertações e os júris aos blocos, cada bloco é constituído por um júri e por três dissertações onde o respectivo júri e cada dissertação têm pelo menos uma palavra-chave em comum.

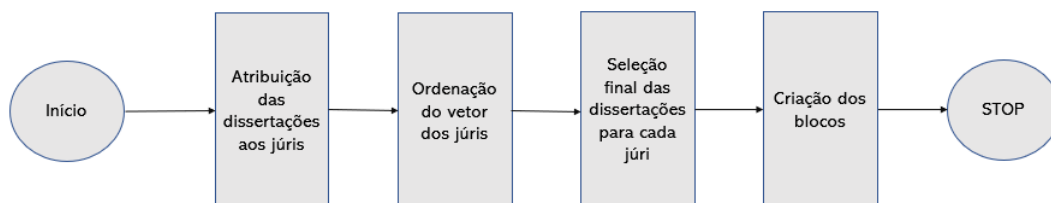


Figura 5.2: Heurística para a criação dos blocos

Tal como ilustrado na figura 5.2, esta alocação é realizada com uma heurística dividida em 4 etapas. Na primeira etapa são atribuídas a cada júri todas as dissertações que este pode avaliar. Na segunda etapa, o vector com os júris é ordenado por ordem crescente do número de dissertações que cada júri pode avaliar. Na terceira, são seleccionadas as dissertações que cada júri vai avaliar, respeitando o limite máximo de cada júri, para tal é percorrido o vector com os júris e, para cada júri, são eliminadas dissertações até esse júri ficar com o seu número máximo de dissertações que pode avaliar. As dissertações que serão removidas são aquelas que podem ser avaliadas por mais júris. As dissertações que ficaram alocadas a um júri, são eliminadas nos outros júris. Na quarta e última etapa, o júri e as suas respectivas dissertações são organizados em blocos de 3 dissertações.

Após esta primeira fase concluída, o Algoritmo Genético de Chaves Aleatórias Tendenciosas é executado para o planeamento final do problema, ou seja, atribuir os intervalos de tempo e as salas aos júris, orientadores e estudantes e garantir outras restrições já mencionadas.

5.1.2 BRKGA da primeira abordagem

5.1.2.1 Cromossoma

O cromossoma, do Algoritmo Genético de Chaves Aleatórias Tendenciosas, para o planeamento final da primeira abordagem é constituído por um conjunto de genes com dimensão igual ao número de blocos. Tal como ilustrado na figura 5.3.

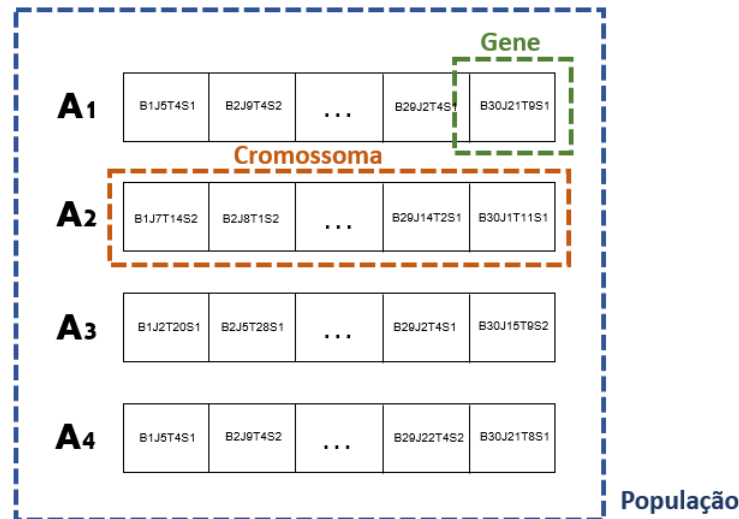


Figura 5.3: Caracterização de uma população para o planeamento final

Cada gene é constituído por um vector com a informação relativa a cada bloco. Deste modo um gene é constituído pela informação sobre o seu bloco, respectivo júri, intervalo de tempo e a sala atribuída.

5.1.2.2 Função de avaliação

No geral, como o valor da função de avaliação é inversamente proporcional ao número de conflitos, o objectivo é minimizar o número de conflitos e, consequentemente, maximizar o valor desta função.

Uma primeira fase é determinante para cumprir as restrições *fortes*, como a disponibilidade dos júris, orientadores e das salas, apresentar uma correta associação das dissertações aos respectivos júris, ou seja, garantir que as dissertações são avaliadas por júris com capacidades para tal e cada júri, orientador e sala só pode pertencer a um bloco num determinado intervalo de tempo.

Depois de cumpridas estas restrições *fortes*, o objectivo desta função passa por minimizar as restrições designadas como restrições *fracas*, onde os júris devem avaliar um número de dissertações compreendidas no intervalo de valores previamente estabelecidos, um orientador não deve pertencer mais de dois blocos no mesmo dia, um júri não deve avaliar mais de um bloco por dia e deve ser respeitada a prioridade das salas.

Para avaliar esta função foi definido uma combinação linear das restrições *fortes* e posteriormente das restrições *fracas*. Inicialmente verifica-se se o número de conflitos para as restrições *fortes* é nulo, caso se confirme, são avaliadas as restrições *fracas*.

5.1.2.3 Restrições

Para o obter um planeamento eficaz e eficiente é necessário uma função objectivo com o valor máximo possível, igual a 1, deste modo para qualquer uma das restrições, sempre que alguma destas condições não for cumprida, será incrementado um valor ao número de conflitos.

Fortes:

- Cada bloco deverá conter 3 dissertações:

Através do vector com os blocos verificar o número de dissertações presentes em cada bloco.

- Cada júri deverá ter capacidade para avaliar todas as dissertações do bloco a que foi atribuído:

Através do vector com os blocos verificar, para cada bloco, se uma palavra-chave do júri é compatível com pelo menos uma palavra-chave de cada dissertação do bloco.

- Disponibilidade do orientador:

Através do vector com os blocos, aceder à informação que diz respeito aos orientadores de cada bloco e ao intervalo de tempo atribuído ao bloco e verificar, no vector com os orientadores, se existe disponibilidade no respectivo intervalo de tempo.

- Disponibilidade do júri, presidente de júri e arguente:

Através do vector com os blocos, aceder à informação que diz respeito ao júri de cada bloco e ao intervalo de tempo atribuído ao bloco e verificar, no vector com os júris, se existe disponibilidade no respectivo intervalo de tempo.

- Cada apresentação tem de ser realizada num período com uma sala disponível:

Através do vector com os blocos, aceder à informação que diz respeito à sala e ao intervalo de tempo e verificar, no vector com as salas, se existe disponibilidade no respectivo intervalo de tempo.

- Um professor não pode ser presidente de júri e orientador da mesma dissertação:

Através do vector com os blocos, para cada dissertação, verificar se o nome do orientador e o nome do presidente de júri são iguais;

- Um orientador só pode pertencer a um bloco num determinado intervalo de tempo:

Através do vector com os blocos verificar se existe um orientador que pertença ao mesmo intervalo de tempo em blocos diferentes.

- Um júri só pode pertencer a um bloco num determinado intervalo de tempo:

Através do vector com os blocos verificar se existe um júri que pertença ao mesmo intervalo de tempo em blocos diferentes.

- Um professor não pode estar presente como orientador num bloco e ser presidente de júri de outro bloco no mesmo intervalo de tempo:

Através do vector com os blocos verificar se existe um orientador que também é presidente de júri noutro bloco no mesmo intervalo de tempo.

- Cada sala só pode ter um bloco num determinado intervalo de tempo:

Através do vector com os blocos verificar se existe uma sala que esteja alocada para dois blocos no mesmo intervalo de tempo.

- Cada júri não pode avaliar mais dissertações do que o número previamente estabelecido:

Através do vector com os blocos, comparar o número máximo de dissertações de cada júri com o número máximo que esse júri deve avaliar, informação essa presente no vector dos júris.

Fracas:

- Cada júri não pode avaliar menos dissertações do que o número previamente estabelecido:

Através do vector com os blocos, comparar o número mínimo de dissertações de cada júri com o número máximo que esse júri deve avaliar, informação essa presente no vector dos júris.

- Respeitar as prioridades das salas:

Através do vector com os blocos, comparar a prioridade da sala com a prioridade do júri que avalia o bloco.

- Evitar a sobrecarga diária:

Através do vector com os blocos, somar o número de dissertações que cada júri e cada orientador é alocado num dia e comparar com os limites previamente estabelecidos.

- No caso de apenas um bloco num determinado intervalo de tempo, atribuir esse bloco à melhor sala,;

Através do vector com os blocos, verificar, nos casos em que só há um bloco atribuída a um intervalo de tempo, se a sala seleccionada é a melhor.

5.2 Segunda abordagem

Numa fase mais avançada da dissertação, foi realizada uma segunda abordagem que difere da primeira no modo com agrupa as dissertações e os júris nos blocos.

Nesta abordagem, a alocação das dissertações e dos júris aos blocos é realizada, independentemente, através do mesmo Algoritmo Genético de Chaves Aleatórias Tendenciosas utilizado para o planeamento final, explicado na secção anterior.

Após um determinado número de gerações ou quando todas as restrições forem cumpridas, ou seja, quando os júris e dissertações estiverem alocados nos blocos, o Algoritmo Genético de Chaves Aleatórias Tendenciosas é realizado novamente, exactamente como na primeira abordagem, para a alocar os blocos nos intervalos de tempo, nas salas e garantir outras restrições.

Ao longo desta secção serão explorados os tópicos para a construção dos blocos.

5.2.1 Cromossoma

Tal como explicado anteriormente, cada cromossoma para a criação dos blocos é constituído por um conjunto de genes. Cada gene é composto pela dissertação e pelo respectivo júri, como representado na figura 5.4.

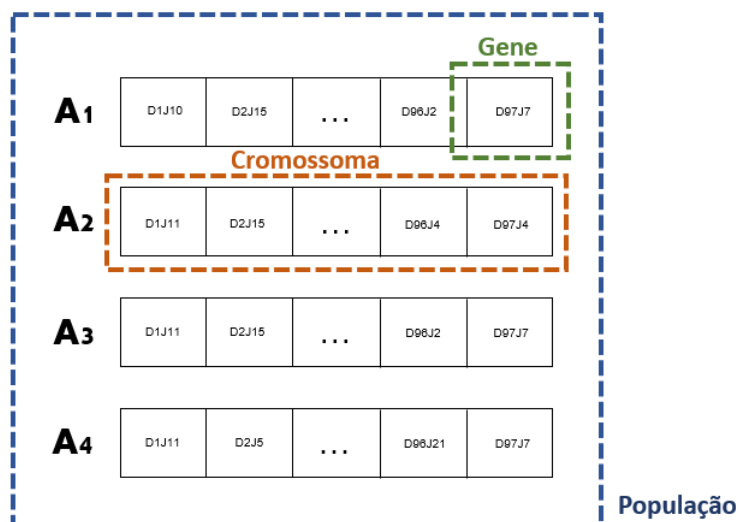


Figura 5.4: Caracterização de uma população para a criação de blocos

5.2.2 Função de avaliação

A função de avaliação para a criação de blocos é semelhante à função de avaliação para o planeamento final, explicado na secção anterior. Numa primeira fase é necessário cumprir com as restrições *fortes*, isto é, garantir que cada júri só avalia dissertações com pelo menos uma palavra-chave em comum e que não ultrapassa o número de dissertações previamente estabelecido.

A segunda fase, cumprir com as restrições *fracas*, tem como propósito garantir que os júris avaliam pelo menos o número mínimo de dissertações previamente estabelecido e que cada júri avalia um número de dissertações múltiplo de 3, de modo a que cada bloco seja constituído por 3 dissertações.

5.2.3 Restrições

Este propósito é que todas as restrições sejam cumpridas, inicialmente as *fortes* e depois as *fracas*, sempre que isso não acontecer será incrementado um valor à função objectivo.

Fortes

- Cada júri deverá ter capacidade para avaliar todas as dissertações a que foi atribuído:

Através do vector com as defesas da dissertação verificar pelo menos uma palavra-chave da dissertação é compatível a uma palavra-chave do júri atribuído.

- Cada júri não pode avaliar mais dissertações do que o número previamente estabelecido:

Através do vector com as defesas da dissertação, comparar o número de dissertações a que um júri foi atribuído e verificar se é menor ou igual ao número previamente estabelecido.

Fracas:

- Cada júri não pode avaliar menos dissertações do que o número previamente estabelecido:

Através do vector com as defesas da dissertação, comparar o número de dissertações a que um júri foi atribuído e verificar se é maior ou igual ao número previamente estabelecido.

- Cada júri deve avaliar um número de dissertações múltiplo de 3:

Através do vector com as defesas da dissertação, comparar o número de dissertações a que um júri foi atribuído e verificar se é múltiplo de 3.

Capítulo 6

Sistema de Apoio à Decisão

Neste capítulo será retratado o sistema de apoio à decisão utilizado nesta dissertação, desde o algoritmo, a leitura dos dados, os parâmetros, as restrições, o critério de paragem e o resultado pretendido.

6.1 Introdução à framework

O Algoritmo Genético de Chaves Aleatórias Tendenciosas (BRKGA) é uma meta-heurística de pesquisa geral para encontrar as melhores soluções possíveis para problemas de optimização combinatória.

Os BRKGA têm três recursos principais especializados em algoritmos genéticos citados em [Silva et al., 2015]:

- "Um cromossoma com um vector de n chaves aleatórias ou alelos no intervalo $[0, 1]$, onde o valor de n depende da instância do problema de optimização";
- "Um processo evolutivo bem definido que adopta o cruzamento uniforme parametrizado para gerar filhos e, assim, evoluir a população";
- "Introdução de novos cromossomas chamados mutantes no lugar do operador de mutação geralmente encontrado em algoritmos genéticos".

As meta-heurísticas BRKGA são baseadas numa framework meta-heurística de uso geral [Toso and Resende, 2015]. Como já referido anteriormente na secção 2.3.2.2, nesta framework há uma divisão clara entre a parte independente do problema e a parte dependente do problema.

6.2 Parâmetros

6.2.1 Parte independente

A parte independente não está directamente relacionada com o problema a ser resolvido, pode ser universal para vários problemas. Estes parâmetros são os mesmos para as duas abordagens,

para o planeamento final e para a criação dos blocos, excepto o número máximo de gerações, que são 250 gerações para o planeamento final e 1500 para a construção dos blocos.

Estes parâmetros não influenciam directamente o valor final da função objectivo, estão relacionados com as operações do algoritmo genético.

Os parâmetros são:

- P : número de indivíduos de uma população;
- N : número de genes de cada cromossoma, que está directamente relacionado com o número de defesas;
- P_e : percentagem de indivíduos Elite na população;
- P_m : fracção da população constituída por mutantes;
- A : número de indivíduos seleccionados para ser um dos pais, e posteriormente escolher o melhor deles;
- ρ : probabilidade do cromossoma filho receber o gene do pai Elite;
- $GMAX$: número máximo de gerações.

Tabela 6.1: Parâmetros independentes do problema

Parâmetro	Valor
P	100
N (criação de blocos)	97 (número de dissertações)
N (planeamento final)	{31; 32} (número de blocos)
P_e	0.10
P_m	0.05
A	3
ρ	0.70
$GMAX$	{250; 1500}

6.2.2 Parte dependente

Os parâmetros dependentes são valores pré-definidos pelo utilizador que influenciam directamente o valor da função objectivo e podem ser alterados conforme a solução pretendida.

Podemos definir estes parâmetros para o problema de planemaneto de júris de mestrado como o número máximo de dissertações, ou blocos, que um júri deve avaliar num dia. A prioridade das salas, número de intervalos de tempo por dia ou o número máximo de dissertações por bloco.

Tabela 6.2: Parâmetros dependentes do problema

Parâmetro	Valor
Número máximo de blocos de um júri a avaliar por dia	1
Número máximo de blocos de um orientador por dia	2
Número base de dissertações num bloco	3

6.3 Leitura de dados

A leitura de dados para a resolução deste problema é através da biblioteca Pandas da linguagem de programação Python. Através desta biblioteca é retida a informação contida num ficheiro Excel.

Os dados extraídos consistem em:

- Lista com os dias e intervalos de tempo disponíveis para avaliação das dissertações;
- Lista das salas e respectiva disponibilidade horária;
- Lista das dissertações para avaliação, esta lista inclui a informação do aluno, informação relativa à dissertação e respectivo orientador;
- Lista dos júris;
- Disponibilidade dos júris e orientadores;
- Palavras-chave dos júris que permitem identificar que dissertações podem avaliar.

6.4 Planeamento final

Na primeira abordagem, o Algoritmo Genético de Chaves Aleatórias Tendenciosas é, apenas, executado para o planeamento final, após a organização dos blocos ser realizada através de um heurística autónoma, descrita na secção 5.1.

6.4.1 Critério de paragem

Para obter o máximo valor da função *fitness*:

$$\text{Fitness} = \sum \frac{1}{N_{\text{Conflitos}} + 1} * \text{Cromossoma}[i]$$

Onde $N_{\text{Conflitos}}$ é o número de conflitos, *fortes* e *fracos*, de cada cromossoma.

Deste modo, o critério de paragem será cumprido quando o valor da função *fitness* for igual a 1, ou seja, número de conflitos igual a 0. Ou após um número de gerações previamente definido.

6.4.2 Pesos das restrições na função objectivo

Uma vez que nem todas as restrições têm a mesma importância para a solução final do problema, é importante estabelecer valores para cada restrição. Como já referido anteriormente, o número de conflitos é uma combinação linear das restrições fortes e das restrições fracas, onde o peso das restrições fracas só é contabilizado após as restrições fortes serem cumpridas.

6.4.2.1 Restrições Fortes

Uma vez que todas as restrições fortes devem ser cumpridas na totalidade, as restrições apresentam o mesmo valor de conflito porque são todas igualmente importantes para ser cumpridas.

- Garantir que um júri ou orientador não está presente em mais que um bloco no mesmo intervalo de tempo;
- Garantir que uma sala não foi alocada a mais que um bloco no mesmo intervalo de tempo;
- Garantir que o horário atribuído a um bloco respeita a disponibilidade horária dos júris e dos orientadores previamente estabelecidos;
- No caso de apenas um bloco num determinado intervalo de tempo, atribuir esse bloco à melhor sala;

6.4.2.2 Restrições Fracas

Após cumpridas as restrições fortes, serão calculadas as restrições fracas, sendo que estas têm valores de prioridade diferentes:

- Garantir a prioridade dos júris e da salas, que será atribuído um valor de 5 em caso de conflito;
- Um valor de 10 será atribuído se um mesmo júri avaliar duas dissertações no mesmo dia e um valor de 25 caso seja atribuído a 3 ou mais blocos num dia;
- Caso um orientador pertença a mais de dois blocos no mesmo dia, o número de conflitos será incrementado com 5 valores.
- Caso um professor avalie um bloco e pertença a outro bloco como orientador no mesmo dia, é adicionado um valor de 15 à função objectivo.

6.4.3 Resultado

Após a leitura dos dados fornecidos, o algoritmo é executado de forma cíclica 250 vezes ou até obter 3 diferentes planeamentos sem conflitos.

No entanto, ao longo de cada geração criada é possível observar informações detalhadas do funcionamento do algoritmo:

- Tamanho da população (número de cromossomas);
- Valor fitness de cada cromossoma;
- Número de restrições de cada cromossoma;
- Verificar se as restrições fortes já foram cumpridas;
- Planeamento simplificado.

O resultado final é imprimido num ficheiro Excel onde é possível analisar informações detalhadas relativas a cada dissertação, tais como:

- Número da dissertação;
- Curso;
- Número do estudante;
- Nome do estudante;
- Título da dissertação;
- Empresa responsável pela dissertação;
- Orientador;
- Júri;
- Data;
- Período;
- Sala.

6.5 Criação de blocos

Na segunda abordagem, antes da utilização do Algoritmo Genético de Chaves Aleatórias Tendenciosas para o planeamento final, tal como descrito na secção anterior, também se aplicou esta meta-heurística para a criação de blocos.

6.5.1 Critério de paragem

O critério de paragem é obtido da mesma maneira que o planeamento final, através de um determinado número de gerações ou quando o valor da seguinte função objectivo for igual 1, ou seja, número de conflitos igual a 0.

$$\text{Fitness} = \sum \frac{1}{N_{\text{Conflitos}} + 1} * \text{Cromossoma}[i]$$

Onde $N_{\text{Conflitos}}$ é o número de conflitos, *fortes* e *fracos*, de cada cromossoma.

6.5.2 Pesos das restrições na função objectivo

Nesta abordagem, as restrições apresentam diferentes valores que influenciam a solução final e, após cumpridas as restrições *fortes*, são avaliadas as restrições *fracas*.

6.5.2.1 Restrições Fortes

Para este caso, as restrições *fortes* apresentam diferentes valores de conflito:

- Garantir que cada júri avalia uma dissertação com pelo menos uma palavra-chave em comum, caso contrário tem um peso de 100;
- Cada júri avaliar um número de dissertações igual ou inferior ao número máximo estabelecido inicialmente, tem um peso igual à diferença entre o número máximo e o número de dissertações a que foi atribuído;
- Cada júri avaliar um número de dissertações igual ou superior ao número mínimo estabelecido inicialmente, tem um peso igual à diferença entre o número mínimo e o número de dissertações a que foi atribuído;
- Garantir que nenhum júri avalia só uma dissertação, peso igual a 3;

6.5.2.2 Restrições Fracas

Posteriormente, as restrições *fracas* servem para garantir que:

- Garantir que todos os blocos são constituídos por 3 dissertações, com peso igual ao resto da divisão do número de dissertações que cada júri avalia por 3;

6.5.3 Resultado

O resultado final desta abordagem é um vector de blocos constituídos pelas dissertações e pelo respectivo júri.

Capítulo 7

Testes computacionais

Neste capítulo, o objectivo é apresentar os testes realizados ao algoritmo desenvolvido ao longo da dissertação. Para as duas abordagens são realizados diferentes testes, com diferentes parâmetros de modo a perceber como obter o melhor planeamento. De seguida é feita uma comparação entre os resultados obtidos.

7.1 Plano de testes

Através deste tipo de problemas de optimização combinatória é possível obter um infinito número de soluções, ou seja, a semente para o problema é gerada de forma aleatória que tem como consequência diferentes caminhos para obter a solução final.

A semente é um número aleatório que inicia o algoritmo, se for usada a mesma semente, em duas ou mais execuções sucessivas de um código, será executada a mesma sequência de números aleatórios. Esta característica é utilizada para comparar o efeitos de alguns parâmetros, nomeadamente os parâmetros independentes.

A execução destes testes foi abordada de duas maneiras:

- Variar a semente e manter os parâmetros independentes;
- Manter a semente e variar os parâmetros independentes.

De seguida será efectuada uma avaliação aos resultados obtidos.

7.2 Análise Gráfica

A criação dos gráficos para posterior análise foi realizada com a biblioteca *Matplotlib* do Python. Para imprimir os planeamentos no Excel foi utilizada a biblioteca *Pandas* do Python.

Assim sendo, para cada metodologia é possível obter um gráfico para a evolução, ao longo das gerações, da função *fitness* do melhor cromossoma de cada geração, que corresponde à linha azul, e também é possível concluir quando é que as restrições *fortes* são cumpridas, ou seja, quando a linha laranja atingir o valor 1.

Observa-se um exemplo da evolução da função *fitness* para o planeamento final da primeira abordagem na figura 7.1.

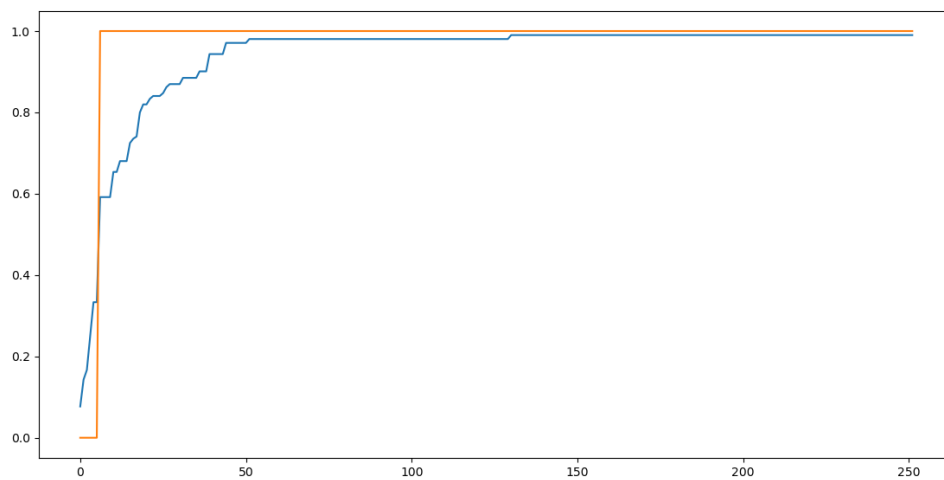


Figura 7.1: Gráfico da evolução da função *fitness* para o planeamento final

De acordo com o gráfico da figura 7.1 é possível analisar que o valor da função *fitness* apresenta uma subida muito significativa até próximo das primeiras 50 gerações e depois disso mantêm-se praticamente constante com um valor muito próximo de 1, com uma ligeira melhoria ao fim de 140 gerações. Conclui-se, também, que as restrições *fortes* são cumpridas ao fim de, aproximadamente, 10 gerações.

Para a criação de blocos, na segunda abordagem, é, também, possível obter um gráfico com a evolução da função *fitness* ao longo da constituição dos blocos.

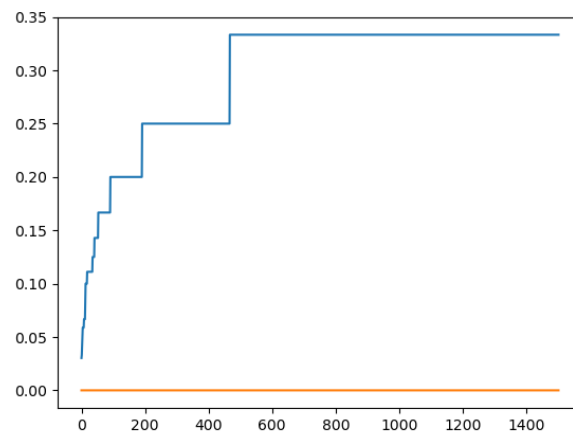


Figura 7.2: Gráfico da evolução da função *fitness* para a criação dos blocos

Como demonstrado na figura 7.2 o valor final da função *fitness* é de, aproximadamente, 0,35, e conclui-se que esse valor corresponde apenas às restrições *fortes* uma vez que estas restrições não foram cumpridas

Assim que cumprido o critério de paragem, são obtidos 3 calendários para as defesas da dissertação. Em cada calendário é possível observar a informação relativa a cada dissertação, como o respectivo júri, a sala e a data, tal como demonstrado na figura 7.3.

Planeamento de dissertações										
Curso	Numero Estudo	Nome estudante	Título	Empresa	Orientador	Juri	Data	Perí	Sala	
MIEMI	201403769	Luís Miguel Vaz de C	Tactical time slot management under fulfillment consideration	LTPiabs, Lda	Fátima Barbosa	Jorge Rui Guimarães Freire de	Segunda 09:30 - 11:00	1	L202A	
MIEMI	201404347	Pedro Miguel Costa	A Aplicação do Método Analítico Hierárquico na Definição da O.Sika Portugal	Produtos	Eduardo José Rego Gil da C	Jorge Rui Guimarães Freire de	Segunda 09:30 - 11:00	1	L202A	
MIEMI	201404094	Ana Camões de Araújo	Melhoria dos Processos de Logística Interna de uma Indústria Amorim & Irmãos, S.A.		Eduardo José Rego Gil da C	Jorge Rui Guimarães Freire de	Segunda 09:30 - 11:00	1	L202A	
MIEMI	201406349	Adriano Rafael Silva	Manutenção Preventiva - Ferramenta de Apoio ao Planeame	CONTINENTAL MABOR - I	Armando Luís Ferreira Leitão	Manuel Augusto de Pina Marc	Segunda 14:30 - 16:00	1	L202A	
MIEMI	201208141	Luís Manuel Araújo	Sc Metodologia para Melhorar das Políticas de Manutenção	CONTINENTAL MABOR - I	Armando Luís Ferreira Leitão	Manuel Augusto de Pina Marc	Segunda 14:30 - 16:00	2	L202A	
MIEMI	201709022	Orlando Alexandre Ca	Análise e Melhoria de Procedimentos de Manutenção na Indi	Amorim & Irmãos, S.A.	Armando Luís Ferreira Leitão	Manuel Augusto de Pina Marc	Segunda 14:30 - 16:00	3	L202A	
MIEMI	201709013	João Nuno da Silva M	Otimização da rede de laboratórios através de metodologias	Kaizen Institute Portugal	Carlos Alberto Bragança de	Maria Antónia da Silva Lopes e	Segunda 16:30 - 18:00	1	L202A	
MIEMI	201406361	José Manuel Lino Sá	C Melhoria de uma linha de PVD através da Gestão da Qualida	de Kaizen Institute Portugal	Mário Filipe Amorim Faria	Maria Antónia da Silva Lopes e	Segunda 16:30 - 18:00	2	L202A	
MIEMI	201306484	Luís Ferreira Fagundes	Redução supply Time-to-market in a Luxury Digital Marketplace	Farfetch Portugal	Unipei Maria João Afonso Gil	Pires Maria Antónia da Silva Lopes e	Segunda 16:30 - 18:00	3	L202A	
MIEMI	201502842	Rita Tavares da Silva	A aplicação da metodologia PDCA no desenho de processos de	SUPER BOCK BEBIDAS, S	Eduardo José Rego Gil da C	Maria Gabriela Beirão dos San	Terça 14:30 - 16:00	1	L11B	
MIEMI	201405293	Sara Santos Teixeira	Desenvolvimento de Atividades de Suporte à Implementaçã	Tapeçarias Ferreira de Sá,	Eduardo José Rego Gil da C	Maria Gabriela Beirão dos San	Terça 14:30 - 16:00	2	L11B	
MIEMI	201407682	Teresa Eugénia Carme	Mapeamento dos processos e monitorização operacional de	Rangel Distribuição e Logi	Eduardo José Rego Gil da C	Maria Gabriela Beirão dos San	Terça 14:30 - 16:00	3	L11B	
MIEMI	201403130	Pedro Carinhos Marq	Eficiência das Operações Logísticas de Armazéns	Colep Portugal, S.A.	António Miguel da Fonseca	Jorge Rui Guimarães Freire de	Terça 14:30 - 16:00	1	L202A	
MIEMI	201404625	José Maria Lencastre	The impact of the "Minimum Life on Receipt" criteria in suppl	IESC TEC	Sara Sofia Baltazar Martins	Jorge Rui Guimarães Freire de	Terça 14:30 - 16:00	2	L202A	
MIEMI	201407051	Joana Rodrigues Quei	Dimensionamento de carga em contentor com uma perspeti	Amorim Cork Composites	Pedro Alexandre Rodrigues	Jorge Rui Guimarães Freire de	Terça 14:30 - 16:00	3	L202A	
MIEMI	201405355	Daniela Timóteo Mart	A Simulation-Optimization Model to Determine Fashion Deliv	LTPiabs, Lda	José Fernando da Costa Ol	Bernardo Sobrinho Simões de	Terça 16:30 - 18:00	1	L202A	
MIEMI	201405110	Rafael Leonardo Reib	An optimization approach to production planning with sched	LTPiabs, Lda	José Fernando da Costa Ol	Bernardo Sobrinho Simões de	Terça 16:30 - 18:00	2	L202A	
MIEMI	201403688	Filipe Manuel de Pinh	The impact of better logistics of POP marketing materials in	I kaizen Institute Portugal	Jorge Rui Guimarães Freire	Bernardo Sobrinho Simões de	Terça 16:30 - 18:00	3	L202A	
MIEMI	201404590	Hugo Pereira Santos	Melhoria da Disponibilidade de uma linha de Aglomeração	de Amorim Revestimentos, S	Paulo Luís Cardoso Osswald	Armando Luís Ferreira Leitão	Quarta 09:30 - 11:00	1	L11B	
MIEMI	201204995	Gonçalo Gomes Carva	Melhoria da disponibilidade em linhas de produção de caixas	Europac Embalagem, S.A.	Paulo Luís Cardoso Osswald	Armando Luís Ferreira Leitão	Quarta 09:30 - 11:00	2	L11B	
MIEMI	201405970	Francisco de Almeida	Otimização de serviços de manutenção através da implemen	cação Kaizen Institute Portugal	Mário Filipe Amorim Faria	Armando Luís Ferreira Leitão	Quarta 09:30 - 11:00	3	L11B	
MIEMI	201404668	Luís Pedro Martins da	Application of lean methodologies to the logistics operation	C PREZITE, Ferramentas de	Eduardo José Rego Gil da C	José António Rodrigues Perer	Quarta 09:30 - 11:00	1	L202A	
Opção 1 Opção 2 Opção 3										
MIEMI	201305913	Maria Vitória Sousa	Aumento da produtividade de uma linha de fundição em are	l Felino - Fundição e Cones	Eduardo José Rego Gil da C	José António Rodrigues Perer	Quarta 09:30 - 11:00	2	L202A	
MIEMI	201302814	Maria Beatriz Gouveia	Programa de Melhorias de Postos de Trabalho na Área de Re	Hydro Extruded Solutions	Eduardo José Rego Gil da C	José António Rodrigues Perer	Quarta 09:30 - 11:00	3	L202A	
MIEMI	201303531	Luís Pedro Silva Mans	Modelo de custeio e investimento para uma nova linha de m	ATEP - Amkor Technology	Luís Filipe Ribeiro dos Sant	Américo Lopes Azevedo	Quarta 16:30 - 18:00	1	L202A	
MIEMI	201404395	Rui Fernando Teixeira	Reestruturação do Supply Network Plan numa Empresa de P	Sonae Arauco Portugal	S.A. Luís Filipe Ribeiro dos Sant	Américo Lopes Azevedo	Quarta 16:30 - 18:00	2	L202A	
MIEMI	201402981	Matilde Pregoña Ares	Model delivery pricing model applied to two-sample hypot	Farfetch Portugal	Unipei Alciobades Paulo Soares	Gu Américo Lopes Azevedo	Quarta 16:30 - 18:00	3	L202A	
MIEMI	201404367	Ana Filipa Moreira de	Melhoria do controlo de produção numa empresa de compo	GRUPO SALVADOR CAETA	Paulo Luís Cardoso Osswald	António Miguel da Fonseca	Fe Quinta 09:30 - 11:00	1	L11B	
MIEMI	201403345	Marco Samuel Carvall	Melhoria no processo de fabrico de reservatórios metálicos	XIC Consultores, S.A.	Paulo Luís Cardoso Osswald	António Miguel da Fonseca	Fe Quinta 09:30 - 11:00	2	L11B	
MIEMI	201403990	Ricardo Manuel Azev	Aumento da produtividade de uma linha de PVD utilizando	Ti Kaizen Institute Portugal	Mário Filipe Amorim Faria	António Miguel da Fonseca	Fe Quinta 09:30 - 11:00	3	L11B	
MIEMI	201404839	Catarina Matos Fern	Improving Packaging Assignment on a fashion multi-partner	Farfetch Portugal	Unipei Bernardo Sobrinho Simões	João Alberto Vieira Campos	Pe Quinta 09:30 - 11:00	1	L202A	
MIEMI	201304301	Alexandre Sinde Fere	Análise de Variabilidade no Desenvolvimento de Produtos co	Kaizen Institute Portugal	Maria Teresa Galvão Dias	João Alberto Vieira Campos	Pe Quinta 09:30 - 11:00	2	L202A	
MIEMI	201304310	João Pedro Graça Mo	Development of an end-to-end Supply Chain monitoring and	Farfetch Portugal	Unipei Eduardo José Rego Gil da C	João Alberto Vieira Campos	Pe Quinta 09:30 - 11:00	3	L202A	
MIEMI	201303266	Miguel Cardoso de Ca	Análise de Variabilidade de um Processo Industrial na Indú	stri Kaizen Institute Portugal	Nuno Domingues Mateus	Maria Henriqueta Dourado	Eu Quinta 16:30 - 18:00	1	L202A	
MIEMI	201305218	Sofia Chibante Teixei	Digitalização de informação e aplicação de metodologia	SMEI Deloitte Consultores, S.A.	Manuel Augusto de Pina M	Maria Henriqueta Dourado	Eu Quinta 16:30 - 18:00	2	L202A	
MIEMI	201305582	Ana Luísa Rodrigues	L Desenvolvimento de uma ferramenta de controlo de custos	X Amorim Cork Composites	José António de Sousa Ban	Maria Henriqueta Dourado	Eu Quinta 16:30 - 18:00	3	L202A	
MIEMI	201305859	Paulo Filipe Gomes S	Mapeamento e Reengenharia de Processos Financeiros para	Sonae Capital	Manuel Augusto de Pina M	João Alberto Vieira Campos	Pe Sexta 09:30 - 11:00	1	L202A	
MIEMI	201406184	João Pedro de Azered	Melhoria do processo de receção de mercadorias e do respo	Medlog - Logística Farmac	Manuel Augusto de Pina M	João Alberto Vieira Campos	Pe Sexta 09:30 - 11:00	2	L202A	
MIEMI	201404137	João Paulo Ferreira	Diagnóstico e otimização da função de Procurement numa e	PRICEWATERHOUSECOOP	Manuel Augusto de Pina M	João Alberto Vieira Campos	Pe Sexta 09:30 - 11:00	3	L202A	
MIEMI	201404185	Ana Cristina Neto Anc	Store Attribute Weighting for Clustering in Fast Fashion	KSR - Retail Consult	Paulo Luís Cardoso Osswald	Luís Filipe Ribeiro dos Santos	C Sexta 16:30 - 18:00	1	L202A	
MIEMI	201404875	Ana Rita Amorim de A	Item attribute weight in fashion assortment planning	KSR - Retail Consult	Paulo Luís Cardoso Osswald	Luís Filipe Ribeiro dos Santos	C Sexta 16:30 - 18:00	2	L202A	
MIEMI	201404654	Maria João dos Santos	Forecast of Short Lifecycle Items in the Fashion Industry	KSR - Retail Consult	Paulo Luís Cardoso Osswald	Luís Filipe Ribeiro dos Santos	C Sexta 16:30 - 18:00	3	L202A	
MIEMI	201405014	Daniel Amadeu Ribeir	Melhoria do fluxo de produção e do layout de uma unidade	Industrial Amorim & Irmãos, S.A.	Maria Henriqueta Dourado	Jorge Manuel Pinho de Sousa	Sexta 16:30 - 18:00	1	L11B	
MIEMI	201403112	Maria Leonor Bento	The vehicle scheduling problem of electric buses	Faculdade de Engenharia	Maria Teresa Galvão Dias	Jorge Manuel Pinho de Sousa	Sexta 16:30 - 18:00	2	L11B	
MIEMI	201405461	Joana Alves Moreira	F Estudo e Melhoria do Sequenciamento da Produção numa	F Amorim Revestimentos, S	José António de Sousa Ban	Jorge Manuel Pinho de Sousa	Sexta 16:30 - 18:00	3	L11B	
MIEMI	201404236	Paulo Renato Martins	Visão holística do crescimento da base de Clientes numa em	NS Comunicações, S.A.	Maria João Afonso Gil	Pires Maria Dulce Soares Lopes	Segunda 09:30 - 11:00	1	L202A	
Opção 1 Opção 2 Opção 3										
MIEMI	201404864	João António Abreu	Developing business analytics to support Uber-zooxoters sha	ri Wunder Mobility GmbH	Vera Lucia Miguéis Oliveira	Maria Dulce Soares Lopes	Segunda 09:30 - 11:00	1	L202A	
MIEMI	201404733	Francisco Manuel Gor	Early adoption of trends in food-based retailing	Sonae MC	Vera Lucia Miguéis Oliveira	Maria Dulce Soares Lopes	Segunda 09:30 - 11:00	2	L202A	
MIEMI	201403106	Luís de Sousa Soares	Pull by Pushing: aplicação de um modelo híbrido de planeam	ca Kaizen Institute Portugal	José Luís Cabral Moura Bor	Ana Maria Cunha Ribeiro dos	Segunda 09:30 - 11:00	1	L11B	
MIEMI	201403978	Catarina Alexandra B	Desenho de um Manufacturing Execution System (MES)	RELVAS II - ROLHAS DE CH	Pedro Alexandre Rodrigues	Ana Maria Cunha Ribeiro dos	Segunda 09:30 - 11:00	2	L11B	
MIEMI	201709020	Nuno Gonçalo Coutin	Pós "Igo live" de linha produtiva e otimização da mesma	Amorim Cork Composites	Pedro Alexandre Rodrigues	Ana Maria Cunha Ribeiro dos	Segunda 09:30 - 11:00	3	L11B	
MIEMI	201403757	Tiago Manuel Medein	Order Cycle Optimization in a Luxury Fashion Marketplace	Farfetch Portugal	Unipei Bernardo Sobrinho Simões	Carlos Alberto Bragança de	Oil Segunda 16:30 - 18:00	1	L11B	
MIEMI	201406185	José Luís Faria de Sá	Product Catalogue Process Improvement and Advanced Mkt	Farfetch Portugal	Unipei Bernardo Sobrinho Simões	Carlos Alberto Bragança de	Oil Segunda 16:30 - 18:00	2	L11B	
MIEMI	201303632	Francisco Maria Sousa	How to improve procurement on a digitalization app?	BA Glass, S.A.	Jorge Rui Guimarães Freire	Carlos Alberto Bragança de	Oil Segunda 16:30 - 18:00	3	L11B	
MIEMI	201303965	Manuel Morais Castro	Aplicação da fabricação aditiva na construção de ferramentas	Peugeot Citroen Automot	José António de Sousa Ban	Maria Henriqueta Dourado	Eu Segunda 16:30 - 18:00	1	L202A	
MIEMI	201303933	João Afonso Sá de Mi	Análise de variabilidade e melhoria de um processo de estam	Colep Portugal, S.A.	Paulo Luís Cardoso Osswald	Maria Henriqueta Dourado	Eu Segunda 16:30 - 18:00	2	L202A	
MIEMI	201305715	Luís José Preto Pere	Assessing the impact of operational performance improve	me Farfetch Portugal	Unipei Isabel Maria Noronha de R	Maria Henriqueta Dourado	Eu Segunda 16:30 - 18:00	3	L202A	
MIEMI	201403566	Elser Miguel Santos	O Marketing Analítico no Setor de Distribuição Farmacéuti	ca - A Medlog - Logística Farmac	João Alberto Vieira Campos	Vera Lucia Miguéis Oliveira e	S Segunda 14:30 - 16:00	1	L202A	
MIEMI	201403921	Pedro Barros de Olive	Improving customer experience - Predictive Model of Billing	ENOS Comunicações, S.A.	Maria Gabriela Beirão dos	Vera Lucia Miguéis Oliveira e	S Segunda 14:30 - 16:00	2	L202A	
MIEMI	201403910	José João Guimarães	Returns in the luxury fashion e-commerce: predicting and	Unipei Farfetch Portugal	Unipei Jorge Daniel Grenha	Luís Filipe Ribeiro dos Santos	C Sexta 14:30 - 16:00	3	L202A	
MIEMI	201404820	Inês Sousa Fernandes	Implementação de um novo canal de comércio eletrónico no	Medlog - Logística Farmac	João Alberto Vieira Campos	Vera Lucia Miguéis Oliveira e	S Segunda 14:30 - 16:00	1	L202A	
MIEMI	201000711	Ana Catarina da Fons	Mapeamento, Análise e Melhoria de Processos num Operad	r Rangel	José António Rodrigues	Pe Nuno Domingues Mateus	Ped Terça 09:30 - 11:00	1	L11B	
MIEMI	201404897	Margarida Lopes Gue	Designing a Performance Management System for Creative C	Farfetch Portugal	Unipei Ana Maria Cunha Ribeiro	d Nuno Domingues Mateus	Ped Terça 09:30 - 11:00	2	L11B	
MIEMI	201404539	Francisca Inês Fint	de uma metodologia para avaliação de desempenho de um	pro ITGROW - Software e Siste	José Luís Cabral Moura Bor	Nuno Domingues Mateus	Ped Terça 09:30 - 11:00	3	L11B	
MIEMI	201307031	Gonçalo Pedro Better	Metodologias Kaizen aplicadas à Indústria Têxtil - Caso de	e Kaizen Institute Portugal	António Miguel da Fonseca	José António de Sousa Barros	Terça 09:30 - 11:00	1	L202A	
MIEMI	201402680	José Miguel Gonçalves	Criação de Fluxo numa Indústria de Acabados	Kaizen Institute Portugal	António Miguel da Fonseca	José António de Sousa Barros	Terça 09:30 - 11:00	2	L202A	
MIEMI	201709010	Francisco Eduardo Ve	Implementação de Melhoria Contínua Baseada Na Metodolo	gia Amorim Cork Composites	Pedro Alexandre Rodrigues	José António de Sousa Barros	Terça 09:30 - 11:00	3	L202A	
MIEMI	201303350	Beatriz Gonçalo Meir	An approach for evaluating the impact of fulfillment operati	on Farfetch Portugal	Unipei Jorge Manuel Pinho de	Sol Bernardo Sobrinho Simões de	Terça 14:30 - 16:00	1	L202A	
MIEMI	201404141	Juliana de Sousa Coel	Melhoria de eficiência de uma equipa de instalação de equi	pa Kaizen Institute Portugal	Jorge Rui Guimarães Freire	Bernardo Sobrinho Simões de	Terça 14:30 - 16:00	2	L202A	
MIEMI	201403673	Joana Pinto de Sousa	Transport Planning in the Glass Industry	BA Glass, S.A.	Jorge Rui Guimarães Freire	Bernardo Sobrinho Simões de	Terça 14:30 - 16:00	3	L202A	
Opção 1 Opção 2 Opção 3										
MIEMI	201404221	Carlos Pedro Cabral	De Desenho de Processo para o Tratamento de Divergências	de I Medlog - Logística Farmac	João Alberto Vieira Campos	José Fernando da Costa Ol	Terça 16:30 - 18:00	1	L202A	
MIEMI	201709026	Rafael Gomes Ferreira	Definição e Monitorização de Indicadores Chave de Desempe	Amorim & Irmãos, S.A.	Eduardo José Rego Gil da C	José Fernando da Costa Ol	Terça 16:30 - 18:00	2	L202A	
MIEMI	201406245	Duarte da Costa Lima	Aplicação de Metodologias Kaizen & TPM na Indústria de	Tra Kaizen Institute Portugal	Jorge Rui Guimarães Freire	José Fernando da Costa Ol	Terça 16:30 - 18:00	3	L202A	
MIEMI	201306079	Luís Carlos Lopes Sou	Definição do Fluxo de Abastecimento às Áreas de Produção	Caetano Aeronautic, S.A.	José António de Sousa Ban	Maria Teresa Galvão Dias	Quarta 14:30 - 16:00	1	L11B	
MIEMI	201404429	Beatriz de Azevedo G	Reverse logistics optimization model for a multi-warehouse	n HUUB	Maria Teresa Peixoto Bragi	Maria Teresa Galvão Dias	Quarta 14:30 - 16:00	2	L11B	
MIEMI	201404357	Vasco Lencastre de F	Decision Making Solution for Dynamic Stock Allocation	HULIB	Maria Teresa Peixoto Bragi	Maria Teresa Galvão Dias	Quarta 14:30 - 16:00	3	L11B	
MIEMI	201404073	Sebastião Afonso Nar	Aumento de Eficiência na Indústria de Extração Mineral	Kaizen Institute Portugal	Carlos Alberto Bragança de	Armando Luís Ferreira Leitão	Quarta 14:30 - 16:00	1	L202A	
MIEMI	201405058	André Filipe Coutinho	Planeamento da manutenção numa indústria de decoração	e BA Glass, S.A.	Maria João Afonso Gil	Pires Armando Luís Ferreira Leitão	Quarta 14:30 - 16:00	2	L202A	
MIEMI	201305955	Carlos Daniel Santos	Metodologias de Kaizen Diário em Unidade Industrial	Amorim Cork Composites	Pedro Alexandre Rodrigues	Armando Luís Ferreira Leitão	Quarta 14:30 - 16:00	3	L202A	
MIEMI	201403702	José Pedro da Silva	S8 Order Tracking model for enhanced delivery experience	- a pri Farfetch Portugal	Unipei Isabel Maria Noronha de R	José António Rodrigues Perer	Quarta 16:30 - 18:00	1	L202A	
MIEMI	201303618	Catarina Lima Carmei	Implementação de metodologias de Gestão da Qualidade To	i Kaizen Institute Portugal	José Luís Cabral Moura Bor	José António Rodrigues Perer	Quarta 16:30 - 18:00	2	L202A	
MIEMI	201306044	Ricardo Fernando For	Melhoria da Eficiência de uma Linha de Enchimento de Barri	l SUPER BOCK BEBIDAS, S	Eduardo José Rego Gil da C	José António Rodrigues Perer	Quarta 16:30 - 18:00	3	L202A	
MIEMI	201305570	José Pedro Cruz Cast	Lean Project	SOFT CRAFTS	Pedro Alexandre Rodrigues	José António de Sousa Barros	Quarta 16:30 - 18:00	1	L11B	
MIEMI	201405211	Pedro Tomas Goncal	Balançamento de uma linha de produção e implementação	de BI - SILOUE, Produtos de	Pedro Alexandre Rodrigues	José António de Sousa Barros	Quarta 16:30 - 18:00	2	L11B	
MIEMI	201303614	Bruno Miguel de Oliveira	Analysis and improvement of processes using variability anal	ysis JPM - Automação e Equip	Américo Lopes Azevedo	José António de Sousa Barros	Quarta 16:30 - 18:00	3	L11B	
MIEMI	201404226	Gonçalo Barbosa Cam	The impact of enhanced product imagery on an e-commerce	Farfetch Portugal	Unipei Mário Filipe Amorim Faria	José Luís Cabral Moura Bor	Ger Quinta 09:30 - 11:00	1	L11B	
MIEMI	201405038	Basílio Miguel da Sil	Performance Analytics Solution in an E-Commerce Business	I Deloitte - WeShare - Servi	Ana Maria Cunha Ribeiro	d José Luís Cabral Moura Bor	Ger Quinta 09:30 - 11:00	2	L11B	
MIEMI	201402919	Miguel Constantino C	Ensuring Quality of Material Master Data in ERP Systems	- A CBA Glass, S.A.	António Miguel da Fonseca	José Luís Cabral Moura Bor	Ger Quinta 09:30 - 11:00	3	L11B	
MIEMI	201404343	Ana Cláudia Marques	Ferramenta de apoio à decisão para controlo de inventário	g Europac Embalagem, S.A.	Paulo Luís Cardoso Osswald	Manuel Augusto de Pina Marc	Quarta 09:30 - 11:00	1	L202A	
MIEMI	201404355	Filipa José Almeida	Pe Melhoria dos processos de aprovisionamento de compra	s na BA Glass, S.A.	Maria João Afonso Gil	Pires Manuel Augusto de Pina	Manuel Augusto de Pina Marc	Quarta 09:30 - 11:00	2	L202A
MIEMI	201403954	Alexandre Silva Sousa	Melhoria do processo de planeamento e gestão de stocks	na Sika Portugal	Produtos C	Eduardo José Rego Gil da C	Manuel Augusto de Pina Marc	Quarta 09:30 - 11:00	3	L202A
MIEMI	201304232	Nuno Alexandre Fern	Reorganização de Fluxos no Abastecimento numa Unidade d	Europac Cartão Ovar, S. Paulo	Luís Cardoso Osswald	Alciobades Paulo Soares	Guede Sexta 14:30 - 16:00			

7.3 Análise Estatística

Nesta secção, são analisados os testes realizados ao algoritmo e as consequentes conclusões.

Para cada teste é possível constatar a semente utilizada em cada teste, o número da geração em que as restrições *fortes* e *fracas* foram cumpridas e o valor *fitness* final. Para a criação de blocos considera-se, também, o número de dissertações presentes em cada bloco.

7.3.1 Primeira abordagem

Na primeira abordagem, através da heurística autónoma consegue-se obter 31 blocos constituídos por 3 dissertações e 1 bloco com 4 dissertações.

Assim sendo, a organização dos blocos é sempre igual nesta primeira abordagem. Não há nenhum caso que exceda o número máximo de dissertações a avaliar por um júri. Apenas um júri não avalia o número mínimo de dissertações pré-definido.

Após a organização dos blocos, para o Algoritmo Genético de Chaves Aleatórias Tendenciosas foram realizados 10 testes para a segunda etapa, ou seja, para o planamaneto final, onde é possível observar as seguintes informações: a semente utilizada, geração em que restrições *fortes* foram cumpridas, *GRFortes*, geração em que restrições *fracas* foram cumpridas, *GRFracas*, e o Valor *fitness* final.

O símbolo * significa que ao fim de 250 gerações, as restrições não foram cumpridas.

Tabela 7.1: Resultado para diferentes sementes, primeira abordagem

Teste	Semente	<i>GRFortes</i>	<i>GRFracas</i>	Valor <i>fitness</i> final
1	1076	13	*	0,9804
2	1141	10	74	1.0000
3	1157	15	138	1.0000
4	1725	8	*	0.9901
5	2500	10	*	0.9804
6	4570	11	*	0.9901
7	4596	10	111	1.0000
8	7059	8	119	1.0000
9	8901	6	52	1.0000
10	9870	5	*	0.9901

De acordo da tabela 7.1, conclui-se que em metade dos testes realizados, com diferentes sementes, a meta-heurística utilizada conseguiu garantir todas as restrições *fracas* impostas. Com base nestes valores adquiridos, obtiveram-se as estatísticas representadas na tabela 7.2.

Tabela 7.2: Resultado estatísticos da diferentes sementes, primeira abordagem

Informação	GRFortes	GRFracas	Valor <i>fitness</i> final
Mínimo	5,00	52,00	0,9804
Máximo	15,00	138,00	1,0000
Média	9,60	98,00	0,9934
Desvio Padrão	2,87	31,30	0,0076
Moda	10,00	-	1,0000

Através do tabela 7.2 pode-se extrair a informação relativa às restrições *fortes* e as restrições *fracas* e ao valor *fitness* final. Conclui-se, desta maneira, que o número da geração em que as restrições, tanto as *fortes* como as *fracas*, são cumpridas não influencia o resultado do valor *fitness* final.

De seguida, para a mesma semente, com valor 6531, foi realizada a alteração dos parâmetros independentes. O tamanho da população, P , manteve-se constante, no entanto ocorreu a variação dos valores relativos à fracção da população de indivíduos Elite, P_e , à fracção da população de mutantes, P_m , ao número de indivíduos seleccionados para ser um dos pais, A , para, posteriormente, escolher o melhor deles, e à probabilidade do cromossoma filho receber o gene do pai Elite, ρ .

A variação destes parâmetros está representada na tabela 7.3.

Tabela 7.3: Variação de valores dos parâmetros independentes, primeira abordagem

Parâmetros do BRKGA	Variação
P	100
P_e	{0,1; 0,15; 0,2; 0,25}
P_m	{0,05; 0,1; 0,15; 0,2}
A	{2; 3; 4; 5}
ρ	{0,5; 0,7; 0,8}

Consideram-se como valores base os valores atribuídos para o primeiro teste. Nos testes número 2, 3 e 4 efectuou-se uma variação da percentagem dos indivíduos Elite na população. Para os testes 5, 6 e 7 alterou-se o valor da percentagem de indivíduos mutantes na população. Nos testes 8, 9, 10 e 11 ocorreu a variação do número de indivíduos seleccionados para ser um dos pais. Nos teste número 12, 13 e 14 alterou-se a percentagem de um filho herdar o alelo do pai do grupo Elite. E por fim, nos testes 15 e 16, houve uma variação de dois parâmetros simultaneamente.

Tabela 7.4: Valores dos parâmetros independentes e consequentes resultados para a mesma semente, primeira abordagem

Teste	P	P_e	P_m	A	ρ	$GRFortes$	$GRFracas$	Valor <i>fitness</i> final
1	100	0,10	0,10	3	0,70	8	216	1,0000
2	100	0,15	0,10	3	0,70	8	163	1,0000
3	100	0,20	0,10	3	0,70	14	149	1,0000
4	100	0,25	0,10	3	0,70	15	249	1,0000
5	100	0,10	0,05	3	0,70	5	138	1,0000
6	100	0,10	0,15	3	0,70	15	*	0,9901
7	100	0,10	0,20	3	0,70	8	*	0,8929
8	100	0,10	0,10	1	0,70	13	216	1,0000
9	100	0,10	0,10	2	0,70	10	*	0,9804
10	100	0,10	0,10	4	0,70	14	*	0,9901
11	100	0,10	0,10	5	0,70	8	128	1,0000
12	100	0,10	0,10	3	0,50	11	112	1,0000
13	100	0,10	0,10	3	0,60	9	*	0,9901
14	100	0,10	0,10	3	0,80	8	*	0,9901
15	100	0,10	0,20	3	0,80	42	*	0,9091
16	100	0,15	0,10	3	0,80	9	*	0,9901

Após uma análise cuidadosa observa-se que o interesse da presença de indivíduos mutantes é, exclusivamente, evitar a convergência de toda a população para um óptimo local, uma vez que um número elevado na percentagem de indivíduos mutantes influencia negativamente o resultado final.

Com base nestes valores obtidos, obtiveram-se as estatísticas presentes na tabela 7.5.

Tabela 7.5: Resultado estatísticos para diferentes parâmetros independentes e mesma semente, primeira abordagem

Informação	$GRFortes$	$GRFracas$	Valor <i>fitness</i> final
Mínimo	5,00	112,00	0,8929
Máximo	42,00	249,00	1,0000
Média	12,31	158,88	0,9833
Desvio Padrão	8,21	46,00	0,0318
Moda	8,00	-	1,0000

Com a ajuda da tabela 7.5 conclui-se que, na maioria dos testes, obteve-se um planeamento final que cumpre com todas as restrições. Nos casos em que não é possível obter um planeamento 100% eficaz, ou seja, que cumpra com todas restrições *fortes* e garanta todas as restrições *fracas*, deve-se ao facto de um júri avaliar dois blocos no mesmo dia, um orientador pertencer a mais de 3 blocos no mesmo dia ou um professor ser presidente de júri num determinado bloco e pertencer a 2 blocos como orientador no mesmo dia.

7.3.2 Segunda abordagem

Tal como descrito no capítulo 5, o Algoritmo Genético de Chaves Aleatórias Tendenciosas é aplicado numa primeira fase para a criação de blocos e numa segunda fase para o planeamento final.

7.3.2.1 Criação dos blocos

Inicialmente será analisado o resultado para a criação dos bloco.

Para diferentes sementes, verifica-se o número da geração em que restrições *fortes* foram cumpridas, *GRFortes*, o número da geração em que restrições *fracas* foram cumpridas, *GRFracas*, o valor *fitness* final, número de blocos com 2 dissertações, *B2*, número de blocos com 3 dissertações, *B3*, e número de blocos com 4 dissertações *B4*.

O símbolo * significa que ao fim de 1500 gerações, as restrições não foram cumpridas.

Tabela 7.6: Resultado para diferentes sementes para a criação de blocos, segunda abordagem

Teste	Semente	<i>GRFortes</i>	<i>GRFracas</i>	Valor <i>fitness</i> final	<i>B2</i>	<i>B3</i>	<i>B4</i>
1	1124	*	*	0,3333	2	27	3
2	1738	*	*	0,3333	2	27	3
3	1853	*	*	0,3333	2	27	3
4	2957	*	*	0,2500	2	27	3
5	6668	*	*	0,3333	1	29	2
6	7648	*	*	0,2500	3	25	4
7	8642	*	*	0,3333	1	29	2
8	9378	*	*	0,3333	2	27	3
9	9442	*	*	0,3333	2	27	3
1	9838	*	*	0,3333	2	27	3

Com suporte nos valores adquiridos, constata-se que em nenhum dos testes tanto as restrições *fracas* como as restrições *fortes* foram cumpridas. Também é possível averiguar o número de blocos constituídos por 2, 3 e 4 dissertações, sendo que o propósito é obter o maior número de blocos compostos por 3 dissertações. De acordo com valores adquiridos, obteve-se as estatísticas presentes na tabela 7.7.

Tabela 7.7: Resultado estatísticos de diferentes sementes para a criação de blocos, segunda abordagem

Informação	<i>GRFortes</i>	<i>GRFracas</i>	Valor <i>fitness</i> final	<i>B2</i>	<i>B3</i>	<i>B4</i>
Mínimo	*	*	0,2500	1,00	25,00	2,00
Máximo	*	*	0,3333	3,00	29,00	4,00
Média	*	*	0,3166	1,90	27,20	2,90
Desvio Padrão	*	*	0,0333	0,54	1,08	0,54
Moda	*	*	0,3333	2,00	27,00	3,00

Conforme os resultados estatísticos conclui-se que, para além de não ser possível cumprir com as restrições *fortes*, esta meta-heurística não é capaz de gerar mais que 29 blocos com 3 dissertações.

Ainda para a criação dos blocos, foram realizados testes com a mesma semente, de valor 6061, e variação dos parâmetros independentes, como na primeira abordagem.

Tabela 7.8: Variação de valores dos parâmetros independentes para a criação de blocos, segunda abordagem

Parâmetros	Variação
P	100
P_e	{0,1; 0,15; 0,2; 0,25}
P_m	{0,05; 0,1; 0,15; 0,2}
A	{2; 3; 4; 5}
ρ	{0,5; 0,7; 0,8}

Tal como na tabela 7.4, foram realizados 16 testes para a mesma semente com variação dos parâmetros independentes, tal como demonstrado na tabela 7.9.

Tabela 7.9: Valores dos parâmetros independentes e consequentes resultados para a criação de blocos, segunda abordagem

Teste	P	P_e	P_m	A	ρ	$GRFortes$	$GRFracas$	Valor <i>fitness</i> final	$B2$	$B3$	$B4$
1	100	0,10	0,10	3	0,70	*	*	0,2000	1	25	5
2	100	0,15	0,10	3	0,70	*	*	0,3333	2	27	3
3	100	0,20	0,10	3	0,70	*	*	0,2000	1	29	2
4	100	0,25	0,10	3	0,70	*	*	0,3333	2	27	3
5	100	0,10	0,05	3	0,70	*	*	0,3333	2	27	3
6	100	0,10	0,15	3	0,70	*	*	0,2000	1	25	5
7	100	0,10	0,20	3	0,70	*	*	0,1667	4	23	5
8	100	0,10	0,10	1	0,70	*	*	0,2500	3	25	4
9	100	0,10	0,10	2	0,70	*	*	0,3333	2	27	3
10	100	0,10	0,10	4	0,70	*	*	0,3333	2	27	3
11	100	0,10	0,10	5	0,70	*	*	0,3333	1	29	2
12	100	0,10	0,10	3	0,50	*	*	0,3333	1	29	2
13	100	0,10	0,10	3	0,60	*	*	0,3333	0	27	4
14	100	0,10	0,10	3	0,80	*	*	0,3333	1	29	2
15	100	0,10	0,20	3	0,80	*	*	0,1429	2	23	6
16	100	0,15	0,10	3	0,80	*	*	0,2500	2	27	3

Tal como os testes realizados para diferentes sementes, com a variação dos parâmetros independentes, também, não é possível cumprir com as restrições *fortes* e consequentemente as restrições *fracas*. Constata-se uma diminuição do valor *fitness* com o aumento da percentagem de indivíduos mutantes, já explicado anteriormente.

Tabela 7.10: Resultado estatísticos para diferentes parâmetros independentes e mesma semente para a criação de blocos, segunda abordagem

Informação	GRFortes	GRFracas	Valor <i>fitness</i> final	B2	B3	B4
Mínimo	*	*	0,1429	0,00	23,00	2,00
Máximo	*	*	0,3333	4,00	29,00	6,00
Média	*	*	0,2756	1,69	26,63	3,44
Desvio Padrão	*	*	0,0698	0,92	1,90	1,22
Moda	*	*	0,3333	2,00	27,00	3,00

Com base na análise da tabela 7.10, o número máximo de blocos com 3 dissertações é igual a 29 e o máximo valor da função *fitness* manteve-se nos 0,3333, ou seja, com alteração dos parâmetros independentes não ocorreu nenhuma melhoria.

Através destes resultados, observa-se que, tal como no exemplo anterior, as restrições *fortes* e, consequentemente, as restrições *fracas* não foram cumpridas.

Observa-se também que se obtiveram piores resultados que a heurística realizada na primeira abordagem. Nesta segunda abordagem, na maioria dos testes obtiveram-se 27 blocos com 3 dissertações, 2 blocos com 2 dissertações e 3 blocos com 4 dissertações. Consequentemente, para alguns júris não foi cumprido o número mínimo de dissertações a avaliar. Deste modo, este método é pior que o anterior, onde se obtém 31 blocos com 3 dissertações e apenas 1 bloco com 4 dissertações.

7.3.2.2 Planeamento final

Nesta secção, foram realizados testes para o planeamento final com o Algoritmo Genético de Chaves Aleatórias Tendenciosa, onde são analisados os resultados para a atribuição dos intervalos de tempo, das salas e para o cumprimento das restantes restrições *fracas*.

É, também, possível analisar o número de dissertações presentes nos blocos que foi obtido com a criação dos blocos através do Algoritmo Genético de Chaves Aleatórias Tendenciosa.

Em primeiro lugar foram realizados os testes para diferentes sementes.

O símbolo * significa que ao fim de 250 gerações, as restrições não foram cumpridas.

Tabela 7.11: Resultado para diferentes sementes para o planeamento final, segunda abordagem

Teste	Semente	GRFortes	GRFracas	Valor <i>fitness</i> final	B2	B3	B4
1	1302	8	*	0,9901	2	27	3
2	3409	8	*	0,9901	2	27	3
3	6058	12	*	0,9804	1	29	0
4	6700	12	*	0,9259	2	27	3
5	6711	8	*	0,9901	2	27	3
6	7158	10	*	0,9615	2	27	3
7	7192	5	*	0,9709	2	27	3
8	7603	10	*	0,9709	2	27	3
9	8059	10	*	0,9804	2	27	3
10	9065	12	*	0,9804	0	27	4

Segundo os valores obtidos, chegou-se à conclusão que as restrições *fortes* foram sempre cumpridas, no entanto em nenhum dos testes realizados é possível garantir com as restrições *fracas*.

Tabela 7.12: Resultado estatísticos da diferentes sementes para o planeamento final, segunda abordagem

Informação	GRFortes	GRFracas	Valor <i>fitness</i> final	B2	B3	B4
Mínimo	5,00	*	0,9259	0,00	27,00	0,00
Máximo	12,00	*	0,9901	2,00	29,00	4,00
Média	9,50	*	0,9741	1,70	27,20	2,80
Desvio Padrão	2,16	*	0,0184	0,64	0,60	0,98
Moda	8,00	*	0,9901	2,00	27,00	3,00

De acordo com resultados estatísticos o valor de *fitness* final é próximo de 1, ou seja, a maioria das restrições *fracas* foram cumpridas. No entanto, tal como demonstrado na secção 7.3.2.1, em média, só é possível obter um número máximo de 29 blocos constituídos por 3 dissertações.

Numa segunda fase, foram realizados testes com a mesma semente, de valor 2029, e variação dos parâmetros independentes, como já mostrado anteriormente.

Tabela 7.13: Variação de valores dos parâmetros independentes para o planeamento final, segunda abordagem

Parâmetros	Variação
P	100
P_e	{0,1; 0,15; 0,2; 0,25}
P_m	{0,05; 0,1; 0,15; 0,2}
A	{2; 3; 4; 5}
ρ	{0,5; 0,7; 0,8}

Os resultados obtidos para os 16 testes efectuados foram:

Tabela 7.14: Valores dos parâmetros independentes e consequentes resultados para o planeamento final, segunda abordagem

Teste	P	P_e	P_m	A	ρ	$GRFortes$	$GRFracas$	Valor <i>fitness</i> final	$B2$	$B3$	$B4$
1	100	0,10	0,10	3	0,70	6	*	0,9901	2	27	3
2	100	0,15	0,10	3	0,70	11	*	0,9901	2	27	3
3	100	0,20	0,10	3	0,70	15	*	0,9901	2	27	3
4	100	0,25	0,10	3	0,70	13	*	0,9804	2	27	3
5	100	0,10	0,05	3	0,70	8	*	0,9804	2	27	3
6	100	0,10	0,15	3	0,70	10	*	0,9615	4	23	5
7	100	0,10	0,20	3	0,70	26	*	0,8197	5	21	6
8	100	0,10	0,10	1	0,70	18	*	0,9804	2	27	3
9	100	0,10	0,10	2	0,70	15	*	0,9804	2	27	3
10	100	0,10	0,10	4	0,70	13	*	0,9901	3	25	4
11	100	0,10	0,10	5	0,70	8	*	0,9709	3	24	4
12	100	0,10	0,10	3	0,50	13	167	1,0000	2	27	3
13	100	0,10	0,10	3	0,60	12	*	0,9901	2	27	3
14	100	0,10	0,10	3	0,80	8	*	0,9804	2	27	3
15	100	0,10	0,20	3	0,80	30	*	0,9091	5	21	6
16	100	0,15	0,10	3	0,80	8	*	0,9901	2	27	3

Conclui-se que, após os 16 testes apresentados na tabela 7.14, em apenas um caso foi possível obter o valor máximo da função *fitness*, igual a 1. Observa-se, também, uma redução do valor da função *fitness* em função do aumento da percentagem de indivíduos mutantes.

Tabela 7.15: Resultado estatísticos para diferentes parâmetros independentes e mesma semente para o planeamento final, segunda abordagem

Informação	$GRFortes$	$GRFracas$	Valor <i>fitness</i> final	$B2$	$B3$	$B4$
Mínimo	6,00	167	0,8197	2,00	21,00	3,00
Máximo	30,00	167	1,0000	5,00	27,00	6,00
Média	13,38	167	0,9690	2,63	25,69	3,63
Desvio Padrão	6,38	0	0,0435	1,05	2,14	1,05
Moda	8,00	*	0,9901	2,00	27,00	3,00

Depois de analisa a tabela 7.15 a média do valor *fitness* é próximo de 1, no entanto, entanto esse valor só é 1 uma única vez.

Após uma análise cuidadosa dos resultados, como não são positivos. Um dos principais objectivos é obter blocos com 3 dissertações e desta maneira não é possível. Consequentemente, os resultados para o planeamento final são ligeiramente piores que na primeira abordagem, frutos dos maus resultados obtidos na criação dos bloco, ou seja, influenciam negativamente o planeamento final.

Enquanto que na primeira abordagem todos os blocos são constituídos por 3 dissertações, excepto 1 bloco que é constituído por 4. Nesta segunda abordagem, na maioria das vezes existem 27 blocos com 3 dissertações, 2 blocos com 2 dissertações e 3 blocos com 4 dissertações.

Em termos do valor da função *fitness* final, a primeira abordagem apresenta valor mais alto, mais próximo de 1, ou seja, menos conflitos.

7.4 Comparação com o planeamento manual

Todos os anos o planeamento dos júris de mestrado é efectuado manualmente o que provoca um elevado investimento de tempo e, para além disso, face às inúmeras possibilidades que existem é difícil garantir que todos os requisitos são cumpridos e, também, avaliar a solução final.

Cada dissertação pode ser avaliada, em média, por $7,87 \simeq 8$ júris, ou seja, para as 97 dissertações, sem respeitar os limites máximos e mínimos a avaliar por cada júri, existem $7,73 \times 10^{86}$ possibilidades diferentes de alocar os júris às dissertações.

Com o algoritmo utilizado avaliou-se o planeamento manual realizado para estas instâncias.

Tabela 7.16: Comparação dos diferentes métodos para a criação de blocos

Informação	B2	B3	B4	Restrições fortes cumpridas	Valor <i>fitness</i> médio final
Primeira abordagem	0	31	1	Não	0,5000
Segunda abordagem	2	27	3	Não	0,3333
Planeamento manual	0	23	7	Não	0,0769

Com base nos resultados obtidos na tabela 7.16, pode-se concluir que em nenhum dos casos foi possível cumprir com as restrições *fortes*, tal deve-se ao facto de não ser possível obter apenas blocos com 3 dissertações.

No entanto, em nenhum dos casos um júri avaliou mais dissertações do que o seu máximo permitido.

Tabela 7.17: Comparação dos diferentes métodos para o planeamento final

Informação	Restrições fortes cumpridas	Valor <i>fitness</i> médio final
Primeira abordagem	Sim	1,0000
Segunda abordagem	Sim	0,9690
Planeamento manual	Não	0,1000

Para o planeamento final, tal como demonstrado na tabela 7.17, na primeira abordagem é possível obter uma solução que cumpra com todos os requisitos. Na segunda, existem conflitos quando algum orientador está presente em mais do que dois blocos no mesmo dia, um júri avalia mais de dois blocos no mesmo dia ou professor é presidente de júri num determinado bloco e pertence a outros 2 blocos como orientador no mesmo dia. Na terceira abordagem, as restrições fortes não são cumpridas porque alguns orientadores foram alocados em horários onde, inicialmente, não estavam disponíveis.

Após a realização de todos os testes e consecutiva análise, conclui-se que a primeira abordagem é aquela que permite obter um planeamento com maior eficiência e eficácia, com um valor mais alto da função *fitness*, ou seja, permite atingir um planeamento que, após cumpridas todas as restrições *fortes*, garanta um melhor cumprimento das restrições *fracas*.

Capítulo 8

Conclusão e Trabalho futuro

Neste capítulo conclui-se o trabalho desenvolvido ao longo da dissertação e são propostas algumas ideias futuras para melhorar a ferramenta desenvolvida.

8.1 Conclusão

Tendo em conta as dificuldades encontradas para realizar o planeamento manual das defesas da dissertação, a presente dissertação teve como objectivo a criação de uma ferramenta que permita a calendarização das defesas da dissertação de um modo mais rápido e eficiente.

No início da dissertação foi efectuado um estudo bibliográfico para uma melhor compreensão do problema em questão e foram estudados métodos de resolução de problemas similares na área da educação.

Finalizada a ferramenta para o planeamento das defesas da dissertação, foram executados testes ao Algoritmo Genético de Chaves Aleatórias Tendenciosas, demonstrado na figura [A.1](#) no anexo [A](#), para as duas abordagens utilizadas. Após os testes serem consumados, conclui-se que a ferramenta utilizada permite criar um calendário em menos de 1 hora, onde todas as restrições *fortes* são cumpridas e as restrições *fracas* são garantidas na maioria dos testes.

Posto isto, esta dissertação permite automatizar um processo que, até então, para além de demorado, não conseguia cumprir com todas as condições impostas inicialmente.

8.2 Trabalho futuro

Este trabalho pode servir de base para trabalhos futuros nesta área que está em constante crescimento.

Apesar do algoritmo já cumprir com os requisitos pré-definidos, a presente ferramenta ainda poderia ser aperfeiçoada, num desenvolvimento futuro, em alguns aspectos:

- Um software aplicativo que permitisse aos presidentes de júri e orientadores introduzirem os dados e, após verificação e confirmação do utilizador, o planeamento era criado;

- Cada bloco apresentar o menor número possível de orientadores diferentes;
- Adaptar os intervalos de tempo, ou seja, atrasar ou adiantar os intervalos de tempo, por exemplo, em vez de começar às 10 horas da manhã, começar às 11 horas;
- Utilização de outra meta-heurística que conseguisse melhorar o processo;
- Optimizar o algoritmo desenvolvido.

Bibliografia

- [Abdullah, 2006] Abdullah, S. (2006). Heuristic approaches for university timetabling problems.
- [Bean, 1994] Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization.
- [Buriol et al., 2010a] Buriol, L. S., Hirsch, M. J., Pardalos, P. M., Querido, T., Resende, M. G., and Ritt, M. (2010a). A biased random-key genetic algorithm for road congestion minimization. *Optimization Letters*.
- [Buriol et al., 2010b] Buriol, L. S., Hirsch, M. J., Pardalos, P. M., Querido, T., Resende, M. G., and Ritt, M. (2010b). *Scatter Search and Path-Relinking: Fundamentals, Advances, and Applications*.
- [Burke et al., 2006] Burke, E. K., Petrovic, S., and Qu, R. (2006). Case-based heuristic selection for timetabling problems. *Journal of Scheduling*.
- [Carter and Laporte, 1998] Carter, M. W. and Laporte, G. (1998). Recent developments in practical course timetabling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- [Colorni, 1998] Colorni, A. (1998). Metaheuristics for high school timetabling. *Kluwer Academic Publishers*.
- [Darwin, 1859] Darwin, C. (1859). On the origin of species through natural selection.
- [Dorigo et al., 1991] Dorigo, M., Maniezzo, V., and Colorni, A. (1991). Positive feedback as a search strategy. *Technical Report*.
- [Gerny, 1985] Gerny, V. (1985). A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory Application*.
- [Glover, 1986] Glover, F. (1986). The general employee scheduling problem: an integration of management science and artificial intelligence. *Computers and Operation Research*.
- [Goldberg, 1989] Goldberg, D. (1989). Genetic algorithms in search optimization and machine learning.

- [Gonçalves and Resende, 2011] Gonçalves, J. F. and Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*.
- [Gonçalves and Resende, 2012] Gonçalves, J. F. and Resende, M. G. (2012). A parallel multi-population biased random-key genetic algorithm for a container loading problem. *Computers and Operations Research*.
- [Gonçalves and Resende, 2015] Gonçalves, J. F. and Resende, M. G. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. *European Journal of Operational Research*.
- [Hartmanis and Leeuwen, 2002] Hartmanis, J. and Leeuwen, J. V. (2002). *LNCS 2740 - Practice and Theory of Automated Timetabling IV*.
- [Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, J. C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*.
- [Osman and Laporte, 1996] Osman, I. H. and Laporte, G. (1996). Metaheuristics: A bibliography.
- [Oude Vrielink et al., 2019] Oude Vrielink, R. A., Jansen, E. A., Hans, E. W., and van Hillegersberg, J. (2019). Practices in timetabling in higher education institutions: a systematic review. *Annals of Operations Research*.
- [Petrovic and Burke, 2014] Petrovic, S. and Burke, E. (2014). University timetabling university timetabling.
- [Pillay and Qu, 2018] Pillay, N. and Qu, R. (2018). Examination timetabling problems. *Natural Computing Series*.
- [Schaerf, 1999] Schaerf, A. (1999). A survey of automated timetabling. *Kluwer Academic Publishers*.
- [Silva et al., 2015] Silva, R. M., Resende, M. G., and Pardalos, P. M. (2015). A python/c++ library for bound-constrained global optimization using a biased random-key genetic algorithm. *Journal of Combinatorial Optimization*.
- [Spears and Dejong, 1991] Spears, W. and Dejong, K. A. (1991). On the virtues of parameterized-uniform crossover. in proceedings of the fourth international conference on genetic algorithms.
- [Toso and Resende, 2015] Toso, R. F. and Resende, M. G. (2015). A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*.

Anexo A

Algoritmo Genético de Chaves Aleatórias Tendenciosas

Figura A.1: Código correspondente ao Algoritmo Genético de Chaves Aleatórias Tendenciosas

```
519 class GeneticAlgorithm:
520     def evolve(self, population): return self._mutate_population(self._crossover_population(population))
521     def _crossover_population(self, pop):
522         crossover_pop = Population(0)
523         for i in range(NUMB_OF_ELITE_SCHEDULES):
524             crossover_pop.get_schedules().append(pop.get_schedules()[i])
525         i = NUMB_OF_ELITE_SCHEDULES
526         while i < POPULATION_SIZE:
527             schedule1 = self._select_tournament_populationELITE(pop).get_schedules()[0]
528             schedule2 = self._select_tournament_population(pop).get_schedules()[0]
529             crossover_pop.get_schedules().append(self._crossover_schedule(schedule1, schedule2))
530             i += 1
531         return crossover_pop
532     def _mutate_population(self, population):
533         for i in range(NUMB_OF_ELITE_SCHEDULES, POPULATION_SIZE):
534             self._mutate_schedule(population.get_schedules()[i])
535         return population
536     def _crossover_schedule(self, schedule1, schedule2):
537         crossoverSchedule = Schedule().initialize()
538         for i in range(0, len(crossoverSchedule.get_defesas())):
539             if (rnd.random() < 0.7):
540                 crossoverSchedule.get_defesas()[i] = schedule1.get_defesas()[i]
541             else:
542                 crossoverSchedule.get_defesas()[i] = schedule2.get_defesas()[i]
543         crossoverSchedule.get_fitness()
544         return crossoverSchedule
545     def _mutate_schedule(self, mutateSchedule):
546         schedule = Schedule().initialize()
547         for i in range(0, len(mutateSchedule.get_defesas())):
548             if (MUTATION_RATE > rnd.random()):
549                 mutateSchedule.get_defesas()[i] = schedule.get_defesas()[i]
550         mutateSchedule.get_fitness()
551         return mutateSchedule
552     def _select_tournament_populationELITE(self, pop):
553         tournament_pop = Population(0)
554         i = 0
555         while i < TOURNAMENT_SELECTION_SIZE:
556             tournament_pop.get_schedules().append(pop.get_schedules()[rnd.randrange(0, NUMB_OF_ELITE_SCHEDULES)])
557             i += 1
558         tournament_pop.get_schedules().sort(key=lambda x: x.get_fitness(), reverse=True)
559         return tournament_pop
560     def _select_tournament_population(self, pop):
561         tournament_pop = Population(0)
562         i = 0
563         while i < TOURNAMENT_SELECTION_SIZE:
564             tournament_pop.get_schedules().append(pop.get_schedules()[rnd.randrange(0, POPULATION_SIZE)])
565             i += 1
566         tournament_pop.get_schedules().sort(key=lambda x: x.get_fitness(), reverse=True)
567         return tournament_pop
```