

Luís Filipe dos Santos Coelho Paquete

Algoritmos Evolutivos Multiobjectivo para
Afectação de Recursos e sua Aplicação à
Geração de Horários em Universidades

Dissertação realizada sob a orientação do Prof. Doutor Carlos Manuel Mira da Fonseca,
para a obtenção do grau de Mestre em Engenharia de Sistemas e Computação.

Faculdade de Ciências e Tecnologia
UNIVERSIDADE DO ALGARVE
FARO - PORTUGAL
2001

DECLARAÇÃO DE ORIGINALIDADE

Em cumprimento do disposto na alínea b) do artigo 5º do Decreto-Lei nº 216/92 de 13 de Outubro, Luís Filipe dos Santos Coelho Paquete declara que todo o trabalho efectuado desenvolvido na presente dissertação é de sua autoria.

Mestrando

(Luís Filipe dos Santos Coelho Paquete)

Supervisor

(Carlos Manuel Mira da Fonseca)

Aos meus pais

AGRADECIMENTOS

A realização desta dissertação contou com o apoio de um conjunto de pessoas às quais quero prestar os meus agradecimentos.

Ao Prof. Doutor Carlos Fonseca pela orientação, estímulo e empenho pessoal que manifestou durante a realização deste trabalho.

Ao Prof. Doutor Efigénio Rebelo, presidente do Conselho Directivo da Faculdade de Economia por me ter permitido conciliar o meu trabalho como funcionário da Faculdade com a minha actividade de estudante.

Ao Prof. Doutor António Branco, presidente do Conselho Directivo da Faculdade de Ciências Humanas e Sociais, por me ter facultado os dados sobre a Faculdade.

À Doutora Vivianne Grunert da Fonseca pelo apoio prestado na avaliação estatística de desempenho dos algoritmos multiobjectivo.

Ao Prof. Doutor Fernando Lobo pela troca de ideias sobre o operador de recombinação.

Ao Dr. Sérgio Baltazar e Dr. Luís Sousa por me terem elucidado sobre o funcionamento da base de dados de horários na Universidade do Algarve.

Ao Tony, Renato, João e David por terem assegurado o bom funcionamento do Centro de Informática da Faculdade de Economia quando não me era possível estar presente.

Aos amigos Sérgio, Fernando, João e Paulo por me terem incentivado a continuar com este projecto.

Um obrigado muito especial aos meus pais e à Maria Celeste por todo o importante apoio emocional que me deram durante a parte lectiva do Mestrado e dissertação.

RESUMO

Esta dissertação tem por objectivo aplicar algoritmos evolutivos multiobjectivo a problemas de afectação de recursos, particularmente a problemas de geração de horários de exames e problemas de geração de horários de aulas em Universidades. Estes problemas são normalmente caracterizados pela existência de múltiplos objectivos conflituosos. Neste sentido, uma formalização multiobjectivo para estes problemas é apresentada, com base no conceito de metas e prioridades.

Vários aspectos dos algoritmos evolutivos são propostos e analisados para esta classe de problemas, nomeadamente, métodos de selecção e tipo e parâmetros de operadores de mutação. A escolha da representação e dos operadores utilizados é feita tendo em conta a necessidade de não privilegiar demasiadamente certos objectivos em relação a outros ao nível dos mecanismos de exploração.

São apresentados estudos comparativos entre os algoritmos propostos por meio de métodos de inferência estatística em problemas reais na Universidade do Algarve. O conceito de função de aproveitamento é utilizado para avaliação de algoritmos evolutivos multiobjectivo. Finalmente, a análise da evolução do custo das soluções encontradas ao longo do tempo de execução através de funções de aproveitamento é apresentada.

ABSTRACT

The aim of this study is the application of multiobjective evolutionary algorithms to resource allocation problems, such as university examination timetabling and course timetabling problems. Usually, these problems are characterized by multiple conflicting objectives. A multiobjective formalization of these problems is presented, based on goals and priorities.

Various aspects of evolutionary algorithms are proposed and studied for these problems, particularly, selection methods and types and parameters of mutation operator. The choice of both representation and operators is made so as not to favour excessively certain objectives with respect to others at the level of the exploration mechanism.

A comparative study of performance is presented for the proposed algorithms by means of statistical inference, based on real problems of the University of Algarve. The notion of attainment functions is used as a base for the assessment of performance of multiobjective evolutionary algorithms. Finally, the evolution of the solution cost during the runs is analysed by means of attainment functions, as well.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Organização da Tese	2
1.3	Contribuições	2
2	Elaboração de Horários: Um Problema de Afectação de Recursos	4
2.1	Introdução	4
2.2	Formalização	5
2.2.1	Problema de Geração de Horários de Aulas	7
2.2.2	Problema de Geração de Horários de Exames	8
2.3	Restrições	9
2.4	Complexidade	11
2.5	Técnicas de Resolução do Problema de Geração de Horários	11
2.5.1	Abordagens de Investigação Operacional	12
2.5.2	Progamação de Restrições	13
2.5.3	Meta-heurísticas	15
3	Algoritmos Evolutivos	18
3.1	Introdução	18
3.2	Representação	19
3.3	Avaliação e Atribuição de Aptidão	21
3.4	Seleccção	22
3.5	Mutação	23

3.6	Recombinação	25
3.7	População	26
3.8	Restrições	28
3.9	Algoritmos Evolutivos Multiobjective	29
3.9.1	Optimização Multiobjective	30
3.9.2	Revisão dos Algoritmos Evolutivos Multiobjective	32
3.9.3	Abordagens de Optimização Multiobjective para Problemas de Geração de Horários	33
3.10	Conclusões	34
4	Um Algoritmo Evolutivo Multiobjective para Problemas de Afectação de Recursos	36
4.1	Introdução	36
4.2	Representação	37
4.3	Qualidade da Solução	37
4.4	Atribuição de Aptidão	39
4.5	Mutação	39
4.6	Recombinação	43
4.7	Seleção e Reinserção	44
4.8	Conclusões	44
5	Abordagem Multiobjective ao Problema de Elaboração de Horários na Universidade do Algarve	46
5.1	Introdução	46
5.2	Problema de Geração de Horários de Exames	48
5.2.1	Caracterização do Problema	48
5.2.2	Definição de Prioridades e Metas	51
5.2.3	Definição da Qualidade da Solução	51
5.3	Problema de Geração de Horários de Aulas	52
5.3.1	Caracterização do Problema	52
5.3.2	Definição de Prioridades e Metas	53
5.3.3	Definição da Qualidade das Soluções	54

5.4	Implementação Computacional	55
5.4.1	Introdução	55
5.4.2	Pré-Processamento	55
5.4.3	Descrição da Aplicação	58
5.4.4	Tempo de Execução	60
6	Resultados Experimentais	62
6.1	Introdução	62
6.2	Avaliação de Desempenho de Algoritmos Evolutivos Multiobjectivo	63
6.2.1	Funções de Aproveitamento	63
6.2.2	Testes Estatísticos	64
6.3	Metodologia Experimental	66
6.4	Resultados	68
6.4.1	PGHE da UCEH - Análise de Custo de Solução em Tempo Fixo	68
6.4.2	PGHE da UCEE - Análise de Custo de Solução em Tempo Fixo	71
6.4.3	PGHE da UCEH - Análise de Custo de Solução / Tempo de Execução	75
6.4.4	PGHA da UCEH	83
6.5	Discussão	89
6.6	Conclusões	91
7	Conclusões	93
7.1	Algoritmos Evolutivos Multiobjectivo para Afectação de Recursos	93
7.2	Perspectivas Futuras	95
7.2.1	Processo de Decisão Multiobjectivo	95
7.2.2	Algoritmos Evolutivos Multiobjectivo	96
7.2.3	Aplicação em Problemas de Afectação de Recursos	96
7.2.4	Operadores de Mutação e Cruzamento	97
7.2.5	Avaliação Estatística de Algoritmos Estocásticos	98

Lista de Figuras

6.1	Valores de p para comparações múltiplas de parâmetros β em termos de agregação de objectivos na UCEH na análise do custo de solução em tempo fixo	72
6.2	Valores de p para comparações múltiplas de parâmetros β em termos de funções de aproveitamento na UCEH na análise do custo de solução em tempo fixo	73
6.3	Valores de p para comparações múltiplas de parâmetros β em termos de agregação de objectivos na UCEE na análise do custo de solução em tempo fixo	76
6.4	Valores de p para comparações múltiplas de parâmetros β em termos de funções de aproveitamento na UCEE na análise do custo de solução em tempo fixo	77
6.5	Comparação de métodos de selecção na UCEH em termos de agregação de objectivos na análise do custo de solução/tempo de execução	79
6.6	Comparação de métodos de selecção na UCEH em termos de separação de objectivos na análise do custo de solução/tempo de execução	80
6.7	Comparação de métodos de selecção na UCEH em termos de agregação de objectivos na análise do custo de solução/tempo de execução	81
6.8	Comparação de tipos de mutação na UCEH em termos de separação de objectivos na análise do custo de solução/tempo de execução	82
6.9	Valores de p para comparações múltiplas de parâmetros β em termos de separação de objectivos na UCEH na análise do custo de solução/tempo de execução	83
6.10	Convergência do algoritmo nas abordagens +agregação e -agregação	85
6.11	N ^o de dias de aulas por semana <i>versus</i> n ^o de horas de aulas por semana na abordagem manual (docentes)	86
6.12	N ^o de dias de aulas por semana <i>versus</i> n ^o de horas de aulas por semana na abordagem –agregação (docentes)	87

6.13	N ^o de dias de aulas por semana <i>versus</i> n ^o de horas de aulas por semana na abordagem +agregação (docentes)	87
6.14	N ^o de manhãs e tardes de aulas por semana <i>versus</i> n ^o de horas de aulas por semana na abordagem manual (turmas)	88
6.15	N ^o de manhãs e tardes de aulas por semana <i>versus</i> n ^o de horas de aulas por semana na abordagem –agregação (turmas)	88
6.16	N ^o de manhãs e tardes de aulas por semana <i>versus</i> n ^o de horas de aulas por semana na abordagem +agregação (turmas)	89

Lista de Tabelas

5.1	Número de exames na UCEH	49
5.2	Número de exames na UCEE	50
5.3	Número de aulas na FCHS	52
6.1	Valores de p para comparação de métodos de selecção na UCEH na análise do custo de solução em tempo fixo	70
6.2	Valores de p para comparação de tipos de mutação na UCEH na análise do custo de solução em tempo fixo	70
6.3	Valores de p para comparação de parâmetros β na UCEH na análise do custo de solução em tempo fixo	71
6.4	Valores de p para comparação de métodos de selecção na UCEE na análise do custo de solução em tempo fixo	73
6.5	Valores de p para comparação de tipos de mutação na UCEE na análise do custo de solução em tempo fixo	74
6.6	Valores de p para comparação de parâmetros β na UCEE na análise do custo de solução em tempo fixo	75
6.7	Valores de p para comparação de métodos de selecção na UCEH na análise do custo de solução/tempo de execução	78
6.8	Valores- p para comparação de tipos de mutação na UCEH na análise do custo de solução/tempo de execução	80
6.9	Número de Conflitos no PGHA	84

Capítulo 1

Introdução

1.1 Motivação

Um problema de afectação de recursos consiste em determinar a afectação de m recursos a n itens de modo a otimizar uma determinada função objectivo f que representa a qualidade da solução através da satisfação de um máximo de r restrições. Num problema de geração de horários, por exemplo, os itens consistem num conjunto de eventos, como aulas ou exames, e os recursos num conjunto de intervalos de tempo e salas. A intenção é afectar os intervalos de tempo e salas disponíveis aos eventos, minimizando simultaneamente uma determinada função objectivo que represente o grau de violação das restrições. Embora aparentemente simples, esta classe de problemas pertence aos denominados problemas NP-completos [50].

Este factor de dificuldade materializa este estudo num desafio gratificante do ponto de vista intelectual em que o investigador deve encontrar mecanismos que resolvam o problema em tempo considerado útil, como é geralmente pretendido na vida real. Aliás, estes mecanismos deverão abranger um largo espectro de aplicações e não devem ser dependentes de determinados grupos de problemas, o que torna este estudo um duplo desafio. A dificuldade de construir um conjunto de soluções que satisfaçam, de algum modo, vários objectivos conflituosos entre si, altera o modo convencional de pesquisa de soluções óptimas e adiciona um terceiro desafio.

A utilização de mecanismos existentes na Natureza, ainda que simplificados, para a resolução destes problemas é um móbil inspirador como tema de investigação. Esta nova classe de algoritmos, da qual os algoritmos evolutivos fazem parte, tem demonstrado bom desempenho na resolução de

problemas considerados intratáveis. É de assinalar o número crescente de publicações que reflectem a eficiência dos algoritmos evolutivos neste âmbito. A adaptação deste mecanismo ao contexto dos problemas atrás referidos coloca o último desafio.

1.2 Organização da Tese

Esta tese pretende adaptar os algoritmos evolutivos multiobjectivo propostos por Fonseca e Fleming [47] para resolução de entidades reais do Problema de Afecção de Recursos onde existem vários objectivos. Os tipos de Problemas de Afecção de Recursos considerados neste estudo correspondem a duas instâncias do Problema de Geração de Horários, respectivamente, o Problema de Geração de Horários de Aulas (PGHA) e o Problema de Geração de Horários de Exames (PGHE). A Universidade do Algarve foi a instituição escolhida para testar a metodologia proposta nesta tese.

Com o objectivo de contextualizar esta tese, é apresentado no Capítulo 2 uma formalização dos problemas em estudo e uma revisão dos métodos utilizados para a sua resolução. O Capítulo 3 sumaria certas características dos algoritmos evolutivos no âmbito da resolução desta classe de problemas e introduz os conceitos de optimização multiobjectivo. O Capítulo 4 apresenta uma possível abordagem a Problemas de Afecção de Recursos que apresentam objectivos múltiplos através de algoritmos evolutivos multiobjectivo. O Capítulo 5 particulariza a metodologia proposta no Capítulo anterior para resolução do PGHE e PGHA existente na Universidade do Algarve. O Capítulo 6 descreve o procedimento estatístico utilizado e os resultados obtidos. Finalmente, o Capítulo 7 delinea as conclusões finais e perspectivas futuras.

1.3 Contribuições

O objectivo principal deste estudo é a aplicação de algoritmos evolutivos multiobjectivo a Problemas de Afecção de Recursos, particularmente na geração de horários de aulas e de exames onde existem múltiplos objectivos. Contribuições originais neste estudo incluem:

- revisão bibliográfica de métodos de optimização para resolução de Problema de Geração de Horários;

- desenvolvimento de uma abordagem evolutiva para Problemas de Geração de Horários baseada em representação directa das soluções e num novo operador de mutação;
- caracterização do desempenho dos algoritmos evolutivos implementados simultaneamente em termos de tempo e custo, com base em funções de aproveitamento;
- comparação do desempenho dos vários algoritmos evolutivos implementados através de procedimentos de testes fechados para hipóteses formuladas sobre funções de aproveitamento;

Os algoritmos propostos foram utilizados para geração de calendários de exames em uso na Universidade do Algarve, respectivamente na Unidade de Ciências Exactas e Humanas no ano lectivo 1999/2000 e na Faculdade de Ciência e Tecnologia no ano lectivo 2000/2001.

Capítulo 2

Elaboração de Horários: Um Problema de Afecção de Recursos

2.1 Introdução

Um Problema de Afecção de Recursos apresenta a seguinte definição [31]:

Um Problema de Afecção de Recursos é definido por um conjunto de tarefas T onde cada tarefa é um intervalo numa linha real, um conjunto de recursos R , e um mapeamento de cada tarefa $t_i \in T$ a um conjunto $r_i \subseteq R$ que especifica os recursos que podem ser utilizados para executar a tarefa. O problema é encontrar uma afecção do recurso para cada tarefa tal que o recurso afecto pertença ao conjunto de recursos da tarefa e que a duas tarefas em sobreposição não seja afecto o mesmo recurso.

Num PGHA, como exemplo de uma instância real de um Problema de Afecção de Recursos, pretende-se afectar um conjunto de salas e um conjunto de intervalos de tempo a um conjunto de aulas, evitando a ocorrência de aulas sobrepostas na mesma sala ou de aulas sobrepostas para o mesmo estudante. Um outro exemplo de Problema de Afecção de Recursos real é o Problema de Afecção de Frequências, cujo objectivo é afectar frequências de rádio a um determinado número de transmissores sob determinadas restrições, como mínima interferência [22]. A afecção representações a estados no Problema de Afecção de Estados é também uma instância do Problema de Afecção de Recursos, em que se pretende minimizar a distância entre estados que estejam no

mesmo conjunto de sucessores de um dado estado, no mesmo conjunto de antecessores de um dado estado com um determinada condição de entrada, ou que estejam na mesma partição para uma dada saída [2].

Este estudo centra-se nas instâncias do PGHE e PGHA do Problema de Geração de Horários, como exemplo de Problema de Afectação de Recursos. Outras instâncias do Problema de Geração de Horários incluem o Problema de Geração de Horários de Turnos de Empregados [74], [78], o Problema de Geração de Horários para Competições Desportivas [101], [105] e o Problema de Geração de Horários de Transportes Públicos [80], [109].

Esta área de investigação, desde a sua formalização at ao desenvolvimento de metodologias de resolução, tem sido estudada pelas comunidades científicas de Investigação Operacional, Inteligência Artificial e Computação e tem recebido um grande acolhimento por parte de um elevado número de investigadores, o que comprovado pelas três conferências internacionais sobre geração automática de horários [13], [15], [16], realizadas desde 1996.

A secção 2.2 tem por objectivo apresentar uma formalização do Problema de Afectação de Recursos e das suas instâncias que estão no âmbito deste estudo, ou seja, o Problema de Geração de Horários de Aulas e o Problema de Geração de Horários de Exames. Uma perspectiva sobre as restrições e alguns aspectos sobre a complexidade deste tipo de problemas será delineada nas secções 2.3 e 2.4 . Finalmente será apresentado uma revisão das abordagens actualmente utilizadas para resolução dos problemas atrás referidos na secção 2.5.

2.2 Formalização

Um Problema Geral de Afectação de Recursos [38], admitindo a existência de m recursos que devem ser afectos a n itens e sujeitos a r restrições, pode ser formalizado pelas seguintes expressões, supondo um problema de minimização:

$$\text{Min } F(x)$$

Sujeito a :

$$G_R(x) \leq 0, \quad 1 \leq R \leq r$$

$$x \in X(a)$$

em que

$$X(a) = \left\{ x : \sum_{j \in J_i} x_{ij} = a_i \quad 1 \leq i \leq n \right.$$

$$\left. a = [a_1, a_2, \dots, a_n]^T \quad a_i \geq 1 \right.$$

e onde as variáveis de decisão x_{ij} são definidas por

$$x_{ij} = \begin{cases} 1 & \text{se o item } i \text{ estiver afecto ao recurso } j \\ 0 & \text{caso contrário} \end{cases} \quad 1 \leq i \leq n, \quad j \in J_i,$$

e $J_i \subset \{1, 2, \dots, m\}$ é o conjunto de recursos disponíveis para o item i .

Nesta formalização observa-se uma hierarquia de importância das restrições, distinguindo-se entre as denominadas restrições *hard* e restrições *soft*. As restrições *hard* são restrições que têm de ser obrigatoriamente satisfeitas. Neste sentido, uma restrição *hard* não satisfeita implica a inadmissibilidade da solução. Por outro lado, as violações das restrições *soft* devem ser minimizadas, isto é, o seu grau de violação representa a qualidade da solução. Nesta formalização as restrições *hard* são representadas por $G_R(x)$ e as restrições *soft* por $F(x)$.

A situação ideal seria a satisfação total de todas as restrições. Contudo, em problemas excessivamente restringidos, poderão não existir sequer soluções admissíveis. Neste caso, é necessário recorrer ao relaxamento das restrições, obtendo uma atribuição de recursos aos itens que represente a maior proximidade do óptimo, em que esta proximidade pode ser definida sob várias formas.

Muitos problemas combinatórios podem ser considerados como instâncias do Problema de Afectação de Recursos. Considerando, por exemplo, o problema de coloração de grafos [83], definido como um grafo $G = (V, E)$, um inteiro k e um mapeamento $\chi : V \rightarrow \{1, 2, \dots, k\}$ tal que se $[v, u] \in E$ então $\chi(v) \neq \chi(u)$, os vértices podem ser considerados como itens e as cores como recursos.

Outro exemplo típico de Problema de Afectação de Recursos é o Problema Geração de Horários. Um Horário (E, T, L) numa Instituição de Ensino é definido por um mapeamento $t : E \rightarrow T \times L$, onde $E = \{e_1, e_2, \dots, e_n\}$ é um conjunto de eventos, $T = \{t_1, t_2, \dots, t_l\}$ é um conjunto de intervalos de tempo e $L = \{l_1, l_2, \dots, l_m\}$ é um conjunto de localizações. Um Problema de Geração de Horários (E, T, L, f) para uma Instituição de Ensino é a tarefa de encontrar um mapeamento óptimo t_{opt} , em que a função de custo $f : H \rightarrow R^+$, onde $\forall t \in H : f(t_{opt}) \leq f(t)$ e H é o conjunto de horários definidos em (E, T, L) . Os elementos de E são itens e os elementos de T e L são considerados recursos.

2.2.1 Problema de Geração de Horários de Aulas

O PGHA consiste na afectação de um conjunto de períodos e um conjunto de salas a um conjunto de aulas, evitando a violação de um determinado conjunto de restrições. Estas restrições são violadas se, por exemplo, o mesmo estudante tiver que frequentar duas ou mais aulas simultaneamente.

Considerando um PGHA à luz do modelo definido anteriormente obtém-se uma formalização simplificada:

$$\text{Min}_{x \in X(1)} P(x) = F(x)$$

Sujeito a :

$$V_1(x) \leq 0$$

$$V_2(x) \leq 0$$

em que

$$F(x) = \sum_{i=1}^n \sum_{j \in J_i} c_{ij} x_{ij}$$
$$V_1(x) = \sum_{j=1}^m \sum_{(i,k) \in \Gamma_j} x_{ij} x_{kj}$$
$$V_2(x) = \sum_{j=1}^m \sum_{(p,q) \in \Pi_j} x_{pj} x_{qj}$$

onde

$$c_{ij} = \begin{cases} 1 & \text{se a aula } i \text{ a iniciar a } j \text{ não constar das preferências do docente} \\ 0 & \text{caso contrário} \end{cases}$$

$$\Gamma_j = \{(i, k): \text{aulas } i \text{ e } k \text{ que ocorrem na mesma turma e no tempo } j\}$$

$$\Pi_j = \{(p, q): \text{aulas } p \text{ e } q \text{ que ocorrem na mesma sala e no tempo } j\}$$

A função $P(x)$ representa a minimização das violações das preferências dos docentes ($F(x)$), sujeita à não sobreposição temporal entre aulas da mesma turma ($V_1(x)$), e à não sobreposição temporal de afectações de salas ($V_2(x)$).

Esta formalização é uma simplificação do que acontece em casos reais nas instituições de ensino em Portugal. Na realidade, nesta formalização não se consideram outras restrições igualmente

importantes como a antecedência das aulas teóricas em relação às aulas práticas, intervalos para almoço e número máximo de horas de leccionação por dia. Na formalização de um PGHA numa escola secundária de tamanho médio através de programação não-linear binária, Pato e Carrasco [85] deduziram que seria necessário definir aproximadamente 200 mil variáveis, tornando a formalização demasiado extensa.

A maioria da literatura lida com um PGHA com diferentes características do que acontece na realidade em Portugal. Noutros países, como o Reino Unido, os estudantes do ensino superior têm maior flexibilidade na escolha das disciplinas que pretendem. Por isso, um dos objectivos que preocupa quem se dedica à elaboração de horários de aulas nas instituições de ensino superior é tentar assegurar que as aulas frequentadas por cada estudante não estejam sobrepostas. Em Portugal, como os estudantes estão afectos a turmas com currículo fixo, estas questões são colocadas de outro modo. Pretende-se, assim, assegurar a não sobreposição de aulas da mesma turma e de turmas de anos consecutivos, tentando reduzir o número de estudantes que tenham aulas sobrepostas. Naturalmente, os alunos com maior número de disciplinas atrasadas de anos anteriores serão os mais prejudicados.

2.2.2 Problema de Geração de Horários de Exames

Segundo Carter [20], um PGHE consiste numa afectação de um número fixo de períodos a um conjunto de exames tal que nenhum estudante realize mais do que um exame ao mesmo tempo. Um PGHE simplificado pode ser seguidamente formalizado por:

$$\underset{x \in X(1)}{\text{Min}} P(x) = F(x)$$

Sujeito a :

$$V_1(x) \leq 0$$

em que

$$F(x) = \sum_{i=1}^n \sum_{j \in J_i} c_{ij} x_{ij}$$

$$V_1(x) = \sum_{j=1}^m \sum_{(i,k) \in \Gamma_j} x_{ij} x_{kj}$$

onde

$$c_{ij} = \begin{cases} 1 & \text{se o exame } i \text{ a iniciar a } j \text{ não constar das preferências do docente} \\ 0 & \text{caso contrário} \end{cases}$$

$$\Gamma_j = \{(i, k): \text{ exames } i \text{ e } k \text{ que ocorrem na mesma turma e no tempo } j\}$$

Assim, a função $P(x)$ corresponde à minimização das violações às preferências dos docentes ($F(x)$), sujeita à não sobreposição temporal de exames na mesma turma ($V_1(x)$). Neste problema não se considera a marcação de salas, como no PGHA, pois normalmente esta afectação é realizada posteriormente. Outra formulação alternativa é apresentada por Bullheimer, [8] que propõe uma função objectivo que maximiza o tempo de estudo do estudante com base num modelo de afectação quadrática.

Tal como no problema anterior, esta formalização lida com restrições simples que não correspondem à realidade da elaboração de exames numa instituição de ensino superior.

Como já se tinha verificado nos PGHAs, os PGHEs no ensino superior português recorrem normalmente ao mesmo conceito de turmas.

2.3 Restrições

Os Problemas de Geração de Horários reais são um caso típico de problemas com diferentes tipos de restrições. Cheng *et al.* [24] apresentaram 20 restrições num PGHA, das quais se realçam :

- Não é possível marcar mais do que uma aula ao mesmo tempo para o mesmo aluno;
- Não é possível marcar mais do que uma aula ao mesmo tempo para o mesmo docente;
- As salas de aula têm de apresentar capacidade igual ou superior ao número de alunos a que estão afectas;
- As salas de aula não podem ser marcadas em intervalos de tempo reservados para outras actividades;
- As aulas não podem ser marcadas quando os docentes não estão disponíveis;
- Eventos da mesma disciplina que ocorrem mais do que uma vez por semana, devem ocorrer em dias diferentes;

- Certas aulas têm de ocorrer simultaneamente.

Num inquérito realizado às universidades no Reino Unido sobre o processo de elaboração de horários de exames, elaborado por Burke *et al.* [14], foi possível apurar 32 restrições diferentes, demonstrando a complexidade deste tipo de problemas. As restrições mais consideradas no processo de geração de exames foram as seguintes, por ordem decrescente de importância:

- Não devem existir mais estudantes do que lugares disponíveis na sala atribuída;
- Exames com questões em comum devem ser marcados no mesmo período;
- Certos exames devem ser marcados num determinado conjunto de períodos;
- Só exames com a mesma duração é que podem ser marcados simultaneamente na mesma sala;
- Os exames com maior número de estudantes devem ser marcados mais cedo;
- Certos exames só podem ocorrer em determinadas salas;
- Deve ser dada preferência à marcação em salas de grande capacidade;
- Determinado exame deve ser marcado antes de outro com determinada antecedência.

Corne *et al.* [28] apresentaram uma classificação das restrições em Problemas de Geração de Horários. Segundo estes autores, as restrições podem ser:

- unárias, que envolvem um só evento e podem tomar a forma de exceções ou de especificações;
- binárias, que envolvem dois eventos e podem ser restrições de aresta quando se referem à sobreposição temporal ou espacial de dois eventos, ou de justa-posição quando se define uma determinada ordem e/ou intervalo de tempo entre eventos;
- de capacidade, que se refere à ocupação máxima dos recursos afectos aos eventos;
- de dispersão, que se refere à dispersão temporal e/ou espacial dos eventos;
- de agentes, que podem ser definidas de acordo com as preferências do agente ou com a sua carga horária.

2.4 Complexidade

Antes de abordar o método de solução do problema, é útil compreender a dificuldade em resolvê-lo. Muitos dos problemas combinatórios, tal como o Problema de Geração de Horários, são considerados problemas NP-completos [50], que significa que não podem ser resolvidos em tempo polinomial. Cooper e Kingston [26] demonstraram que o PGHA é, na realidade, constituído por vários sub-problemas NP-completos, respectivamente, *K* Coloração de Grafos, *Bin Packing*, *Exact Cover By 3-Sets* e *3 Dimensional Matching*.

Recentemente, alguns investigadores têm tentado estudar a dificuldade de resolução de problemas combinatórios com base em fenómenos de transição de fase, em alternativa aos métodos clássicos de complexidade computacional. A transição de fase resulta de uma analogia com determinados fenómenos físicos, e ocorre algures entre um problema com poucas restrições e de fácil resolução e um problema com um número elevado de restrições, cuja insolubilidade é fácil de determinar, como descrito por Cheeseman *et al.* [23]. Gent e Walsh [52] observaram o comportamento de determinadas heurísticas para estudar a transição de fase em problemas da vida real. Estes autores observaram que os PGHEs mais difíceis de resolver surgiam tanto na região de insolubilidade como na região com poucas restrições, sem retirar grandes conclusões das observações.

Ross *et al.* [89] [90] implementaram algoritmos evolutivos e algoritmos de arrefecimento simulado em PGHE aleatórios solúveis para localizar a transição de fase, variando a homogeneidade e conectividade dos problemas. Estes autores detectaram várias regiões de transição de fase em que os algoritmos utilizados apresentaram um desempenho distinto. Recentemente, Erben [35] detectou as mesmas regiões de transição de fase através de um algoritmo genético de agrupamento.

O estudo da transição de fase, embora em estado embrionário e principalmente aplicado a problemas aleatórios, tem sido fortemente investigado e poderá trazer algumas contribuições importantes, tanto para o entendimento da dificuldade dos problemas como para implementação de métodos de resolução mais apropriados.

2.5 Técnicas de Resolução do Problema de Geração de Horários

A abundante literatura de estudos sobre o Problema de Geração de Horários tem demonstrado que estes problemas continuam a motivar um grande número de investigadores. A enorme diversidade

de formalizações e de métodos de resolução destes problemas originou uma grande quantidade de publicações, o que torna bastante difícil a extração dos marcos mais importantes em cada metodologia. Neste estudo foi realizada a revisão da aplicação de abordagens de investigação operacional, programação com restrições e meta-heurísticas, pois acredita-se que este é um dos modos de classificação de algoritmos mais difundido neste tipo de problemas. Maior ênfase será colocado na aplicação de algoritmos evolutivos no Capítulo seguinte pois foi a metodologia escolhida para abordar os problemas em estudo.

2.5.1 Abordagens de Investigação Operacional

Uma das áreas de maior dimensão na Investigação Operacional é a programação matemática, que consiste num conjunto de teorias e métodos aplicados a problemas de determinação de extremos de funções de variáveis reais ou inteiras, sujeitas a restrições lineares e não lineares. Estas restrições são representadas sob a forma de equações e desigualdades. A programação linear considera a linearidade das restrições e utiliza o método simplex. Contudo, este método apresenta tempo exponencial no pior caso. Uma extensão à programação linear é a programação inteira em que uma restrição adicional define que as variáveis do modelo sejam inteiras.

Nos Problemas de Geração de Horários, a formalização apresentada pela comunidade de Investigação Operacional tem sido baseada em programação linear binária e em grafos. No entanto, os métodos de resolução baseados em técnicas tradicionais de Investigação Operacional têm sido pouco aplicados devido à complexidade dos problemas em estudo [85]. As abordagens baseiam-se na decomposição do problema em vários sub-problemas menos complexos e na implementação de heurísticas utilizadas em problemas de grafos.

Werra [33] sugere que os Problemas de Geração de Horários sejam formalizados como grafos e resolvidos por heurísticas associadas a problemas de fluxo de redes. A cada arco é atribuído um limite inferior e superior, que é modificado durante a resolução. O problema é resolvido pela sua divisão em vários sub-problemas de fluxo máximo, um por cada período. Kiaer e Teller [68] propõem uma heurística de coloração de vértices para problemas de PGHAs em que os vértices são coloridos de acordo com a regra de coloração do vértice mais difícil em primeiro lugar. Esta dificuldade é determinada com base numa função que tem em conta os custos associados a cada vértice, a soma das ponderações dos arcos adjacentes a cada vértice e a cor já utilizada num vértice adjacente. Contudo, estes métodos dificultam a introdução de outros tipos de restrições.

Carter *et al.* [21] utilizou um conjunto de procedimentos de regras de ordenação em conjunto com um algoritmo de *backtracking* para resolver PGHEs aleatórios e reais, com base numa formalização em grafos. As regras algorítmicas de ordenação consideradas para escolha da próxima marcação foram as seguintes:

- Maior grau, correspondendo ao maior número de exames em conflito;
- Grau de saturação, correspondendo ao número de períodos em conflito;
- Maior grau ponderado, correspondendo ao número de estudantes em conflito;
- Maior número de inscrições para cada exame;
- Ordenação aleatória;

Estes autores observaram que a regra de ordenação por grau de saturação em conjunto com o algoritmo de *backtracking* foi a mais favorável.

2.5.2 Progamação de Restrições

A Programação de Restrições é uma ferramenta que lida com problema de satisfação de restrições. Seguindo um certo paralelismo com a área de Investigação Operacional, aborda igualmente problemas de escalonamento e de afectação de recursos e outros problemas combinatórios. Embora esta área seja proveniente da área da Inteligência Artificial, as suas fronteiras têm vindo a alargar para uma interligação com certos aspectos da área de Investigação Operacional [59].

Um problema de satisfação de restrições é definido sobre um conjunto de variáveis $X = \{x_1, \dots, x_n\}$, em que cada x_i toma valores num dado domínio. Associado ao problema existe um conjunto de restrições que condicionam os valores que as variáveis podem tomar simultaneamente. Uma solução para um problema de satisfação de restrições consiste numa afectação de um valor do domínio a cada variável, tal que a restrição seja satisfeita [102]. As soluções de problemas de satisfação de restrições são usualmente obtidas por métodos de pesquisa sistemática, através de afectação de valores do domínio às variáveis com o recurso a heurísticas.

Uma das vantagens deste método em relação às abordagens clássicas de Investigação Operacional reside na representação, pois as variáveis de um problema de satisfação de restrições correspondem directamente às entidades dos problemas e as restrições podem ser incorporadas no modelo

sem ser necessário traduzi-las em desigualdades lineares [94]. A declaratividade inerente a este tipo de formalização resulta numa maior flexibilidade na manipulação das variáveis e facilita o desenvolvimento rápido e eficiente do mecanismo de pesquisa para resolução de problemas. Para além disso, os algoritmos de resolução deste tipo de problemas apresentam, muitas vezes, um desempenho mais rápido do que é verificado com os métodos de programação linear [94].

A programação lógica de restrições consiste na integração dos métodos desenvolvidos na programação de restrições com a programação lógica, dando origem a uma linguagem com a flexibilidade necessária para o utilizador representar o problema. Um programa consiste num conjunto de cláusulas que contêm restrições declaradas a alto nível. Este método foi abordado por Guéret *et al.* [54] para resolver um PGHE com o recurso ao CHIP (Constraint Handling In Prolog). Nesta aplicação, a redução do domínio é efectuada com base em técnicas clássicas de consistência e de algoritmos de filtragem. Neste problema foi também utilizada a restrição cumulativa implementada no CHIP que assegura que, a cada instante i , a quantidade consumida de recursos não deve exceder um determinado limite. Estes autores, contudo, reportaram a necessidade de implementar um mecanismo eficiente de relaxamento de restrições nas linguagens de programação lógica de restrições.

Outras alternativas de linguagens de programação de restrições são o Eclipse e Oz. Panagiotis *et al.* [96] implementaram o sistema ACTS (Automatic Course Timetabling System) num PGHA com base no Eclipse combinado com heurísticas que mimificam as escolhas efectuadas manualmente pelos responsáveis pela elaboração de horários. Henz e Wurts [63] implementaram a resolução de um PGHA através de Oz num colégio alemão. Segundo estes autores, Oz apresenta vantagens em relação a outros modelos pois permite o desenvolvimento de novas técnicas de pesquisa. Neste problema, o processo de optimização baseou-se no *first-fail* e *branch-and-bound*.

Recentemente tem havido algum interesse na combinação de métodos utilizados na programação de restrições com meta-heurísticas. White e Zhang [107] implementaram métodos de satisfação de restrições para criar soluções iniciais para uma Pesquisa Tabu. Estes autores observaram que a utilização sequencial destes dois métodos foi mais eficiente do que só com os métodos de satisfação de restrições e mais rápido do que verificado só com Pesquisa Tabu.

A programação de restrições é a metodologia que está a ser mais intensivamente aplicada a este tipo de problemas, como se verifica pelo número de comunicações na última conferência sobre geração automática de horários [16]. A existência de linguagens e aplicações que facilitam a mo-

delação destes problemas como ILOG, Oz, CHIP e Eclipse são um factor fundamental para a sua disseminação.

2.5.3 Meta-heurísticas

Meta-heurísticas são algoritmos que são guiados por métodos de uso geral para produzir soluções de boa qualidade. Recentemente, tem-se assistido a significativos avanços no desenho de meta-heurísticas para resolução de problemas combinatórios. Nesta família de técnicas incluem-se os Algoritmos de Arrefecimento Simulado, Algoritmos de Pesquisa Tabu, Redes Neurais e Algoritmos Genéticos.

Neste estudo será apresentada uma revisão de meta-heurísticas em Problemas de Geração de Horários onde será dado maior ênfase aos Algoritmos de Arrefecimento Simulado, Pesquisa Tabu e Algoritmos Evolutivos, pois são os métodos que têm sido mais utilizados.

Algoritmo de Arrefecimento Simulado

O nome deste algoritmo resulta da analogia com a simulação do arrefecimento de sólidos. Normalmente, este procedimento permite o movimento para soluções de vizinhança, piores que a actual, com base numa probabilidade maior que zero, permitindo que se escape de mínimos locais. A temperatura é o parâmetro que controla a probabilidade de aceitação de uma transição para uma solução inferior. Este parâmetro decresce após um dado número de N iterações se não for encontrada a melhor solução. Quando decrescer até um dado limite, o procedimento pára e a melhor solução encontrada é fornecida.

Thompson e Dowsland [100] e Dowsland [34] apresentaram várias variantes do Algoritmo de Arrefecimento Simulado para resolução de PGHEs numa perspectiva multiobjectivo. O problema é definido como um Problema de Coloração de Grafos em que cada vértice representa um exame que deve ser afecto a um intervalo de tempo, ou seja, a uma cor. Cada aresta descreve um par de exames que não podem ocorrer simultaneamente. Uma coloração admissível do grafo é a coloração em que nenhum par de vértices adjacentes apresenta a mesma cor. O processo de optimização é dividido em múltiplas fases, cuja primeira cria uma solução admissível enquanto que as fases seguintes optimizam os restantes objectivos de acordo com a ordem de prioridades.

Algoritmo de Pesquisa Tabu

O Algoritmo de Pesquisa Tabu é um procedimento iterativo que se inicia com uma solução e se movimenta pelo espaço de pesquisa à procura de uma solução óptima. A cada iteração é definida uma vizinhança, ou conjuntos de movimentos, que é aplicada a uma dada solução para produzir uma outra. Para desencorajar a possibilidade de obter unicamente um óptimo local e prevenir a visita a soluções já anteriormente visitadas, um certo número de movimentos são considerados proibidos, denominados movimentos tabu. Este conjunto de movimentos no espaço de pesquisa é determinado por uma ou mais condições e é baseado num histórico da sequência de movimentos.

Na aplicação do Algoritmo de Pesquisa Tabu é natural distinguir entre dois tipos de funções de memória, i. e., funções de memória de curto e de longo prazo. A primeira consiste numa memória dos movimentos mais recentes elaborados pelo procedimento, prevenindo a visita da mesma solução. A função de memória de longo prazo é uma função que se baseia na frequência de soluções já visitadas para prevenir ciclos de longo termo. Contudo, a escolha destes parâmetros depende das características do problema.

Recentemente, Gaspero e Schaefer [51] testaram as capacidades do Algoritmo de Pesquisa Tabu para resolver um PGHE. O algoritmo apresentado combina algumas noções do problema de coloração de grafos. A vizinhança é definida com base na mudança da cor de um dos nós. Para identificar os melhores movimentos em cada iteração é mantida uma lista que contém os nós que estão envolvidos em pelo menos uma violação. Os autores implementaram também outra lista de menores dimensões que guarda os nós que estão envolvidos em violações de restrições mais importantes. Nos diversos estádios da pesquisa, os nós são seleccionados com base numa pesquisa exaustiva das duas listas.

White e Xie [108] apresentaram um Algoritmo de Pesquisa Tabu com quatro fases e com memória de longo prazo para resolver um PGHE. A primeira fase estima os exames que apresentaram maior mobilidade e gera uma solução inicial com base numa heurística de *bin packing*. Esta solução só é avaliada em termos de conflitos de primeira ordem. Se for admissível, será considerada solução inicial para o Algoritmo de Pesquisa Tabu, caso contrário, o intervalo de tempo será alargado. As componentes da função objectivo relativas aos conflitos de menor ordem são adicionados nas fases posteriores até se obter uma solução que minimize simultaneamente todos os conflitos. A definição de vizinhança considerada pelos autores consiste no conjunto de soluções

que estão sujeitas a um movimento atómico, ou seja, a mudança de um exame para outro intervalo de tempo. Estes autores observaram que a utilização de memória de longo prazo combinada com memória de curto prazo fornecia melhores resultados.

Algoritmos Evolutivos

Os algoritmos evolutivos são métodos que se baseiam na pesquisa de uma solução com base numa iteração contínua sobre uma população de soluções iniciais. Esta iteração tem a forma de operadores genéticos, das quais se destacam o cruzamento, mutação e selecção.

Esta metodologia foi a escolhida para a resolução do problema em estudo. O facto de lidar com uma população de soluções torna-o um candidato interessante à abordagem de problemas de optimização multiobjectivo. No capítulo seguinte será apresentada uma revisão das características principais dos algoritmos evolutivos assim como da sua aplicabilidade na resolução de Problemas de Afectação de Recursos e, em particular, de Problemas de Geração de Horários.

Capítulo 3

Algoritmos Evolutivos

3.1 Introdução

Os algoritmos evolutivos são um classe de algoritmos dos quais fazem parte os algoritmos genéticos, estratégias evolutivas, programação evolutiva e programação genética. O comum entre estes algoritmos é a analogia com mecanismos existentes na Natureza tais como o conceito de reprodução, competição, variação aleatória e selecção de indivíduos de uma população [40].

De uma forma geral, os algoritmos evolutivos podem ser definidos como um processo colectivo de aprendizagem de um conjunto de indivíduos que formam uma população. Cada indivíduo representa, ou codifica, um determinado ponto no espaço de soluções para a resolução de um problema. Para avaliar estes indivíduos no ambiente em que estão inseridos, é necessário atribuir-lhes um determinado valor de aptidão. Com base neste valor, é efectuado um processo de selecção que favorece os melhores indivíduos. Estes estarão sujeitos a processos aleatórios, como a mutação e o cruzamento, para produzir descendentes. Estes últimos são inseridos na população dando início à geração seguinte.

Uma das áreas de aplicação desta metodologia é a optimização. A metodologia da evolução fornece inspiração para o desenvolvimento de algoritmos para resolver problemas considerados intratáveis. Os tipos de problemas normalmente abordados pelos algoritmos evolutivos são problemas descontínuos, não-diferenciáveis, multimodais ou ruidosos, pois é onde o desempenho desta metodologia prevalece sobre o de outras.

Seguidamente se apresentam alguns aspectos considerados importantes na implementação de

um algoritmo evolutivo e a sua aplicação na resolução de Problemas de Geração de Horários. A secção 3.2 abordará os métodos de representação utilizados nesta classe de problemas e a secção 3.3 abordará alguns aspectos sobre a avaliação e atribuição de aptidão. Os operadores de selecção, mutação e recombinação serão referidos, respectivamente, nas secções 3.4, 3.5 e 3.6. O modo como as restrições têm sido abordadas nesta classe de problemas será referido na secção 3.7 e os vários modelos de população e sua inicialização serão abordados na secção 3.8. A secção 3.9 efectuará uma apresentação dos conceitos de optimização multiobjectivo e do modo como esta questão tem sido tratada por algoritmos evolutivos multiobjectivo e como os problemas em estudo têm sido abordados de uma perspectiva multiobjectivo. Finalmente, a secção 3.10 delineará alguns comentários finais sobre a metodologia evolutiva para resolver os problemas abordados neste estudo.

3.2 Representação

A primeira questão que se levanta quando se pretende resolver um problema de uma perspectiva computacional é como representar o problema. Os diferentes algoritmos evolutivos utilizam diferentes tipos de representação. Nos algoritmos genéticos, as variáveis de decisão são muitas vezes codificadas como uma cadeia de caracteres binários. As estratégias evolutivas e programação evolutiva utilizam, sobretudo, variáveis de decisão de valores reais para além de um conjunto de parâmetros de estratégia. Na programação genética, a solução é representada por uma árvore que corresponde a um bloco de código que representa um algoritmo para desempenhar uma determinada tarefa.

Contudo, existem algumas propostas de representações mais específicas para determinadas classes de problemas. Problemas combinatórios têm sido representados como um vector de inteiros em que cada elemento do vector representa um nó do problema. É necessário, contudo, ter um cuidado especial quando se pretende, por exemplo, com permutação de elementos, em que cada elemento só pode ocorrer uma só vez, como se verifica nas abordagens ao problema do caixeiro viajante.

A grande proporção das implementações de algoritmos evolutivos para resolução de Problemas de Geração de Horários baseiam-se em representações que se podem distinguir entre directas e indirectas. Na representação directa cada gene corresponde a um evento ao qual é afecto um tempo inicial pertencente ao conjunto de tempos iniciais disponíveis para esse evento. Por outro lado, na representação indirecta, ou implícita, o cromossoma representa uma lista ordenada de eventos que

são colocados num horário de acordo com um método pré-definido.

Burke *et al.* [9][10] realçaram a utilização de representações directas em problemas de PGHE, assim como Ross *et al.* [86][87] em Problemas de Geração de Horários. Os primeiros consideram a geração inicial de soluções admissíveis e adicionam um tempo por cada evento que não cumpre a satisfação das restrições. A estratégia consiste em aplicar operadores que preservam a satisfação de todas as restrições minimizando o número de tempos extras. Os últimos autores consideram a violação das restrições e penalizam as soluções através de uma função de penalidade.

Paechter *et al.* [81] apresentam um trabalho distinto dos autores anteriores. Estes autores consideram que o cromossoma deve fornecer instruções para a construção de um horário com base numa representação indirecta. O objectivo é minimizar o número de eventos não marcados, que transformam o horário numa solução inadmissível. Os autores apresentam duas alternativas para representação do cromossoma num PGHA, respectivamente, o método de permutação espaço-tempo e o método "coloca-pesquisa". No primeiro, o cromossoma é representado por uma permutação de eventos de uma matriz em que cada elemento representa a utilização de uma sala num determinado intervalo de tempo. O cromossoma indica o evento que deverá ser colocado em cada elemento dessa matriz. Se, contudo, o evento não poder ser colocado nesse intervalo de tempo, então é redireccionado para outro intervalo. Esta modificação é escrita no próprio cromossoma para que instâncias futuras possam beneficiar dessa informação. No método "coloca-pesquisa", o cromossoma é constituído por vários genes correspondentes a cada evento. Cada gene contém informação sobre o actual intervalo de tempo atribuído ao evento, a localização do próximo intervalo de tempo caso não possa ser colocado no intervalo actual, o próximo gene a processar e o próximo gene que não contém eventos atribuídos. Na comparação com um algoritmo evolutivo convencional, estes autores observaram um melhor desempenho dos métodos por eles propostos [81].

Corne *et al.* [28] apresentaram várias alternativas de representações do cromossoma. A representação em espaços ricos em conflitos apresenta o cromossoma como uma lista de eventos fixos atribuídos a unidades de tempo. Trata-se na realidade de uma representação directa, em que o espaço de procura se torna rico em violações. A outra abordagem, representação indirecta em espaços livres de conflitos, utiliza o algoritmo genético para pesquisar o espaço de permutações de eventos. Com o recurso a uma heurística, os eventos são atribuídos ao intervalo de tempo mais recente sem violar nenhuma restrição. A terceira abordagem, representação de espaços com conflitos dispersos, semelhante à primeira, descodifica o cromossoma através de um algoritmo que elimina a maioria

dos conflitos. A representação indirecta demonstrou ser a melhor opção, também verificada por Corne *et al.* [30] onde se realizou uma comparação aprofundada das duas representações. Contudo, como é realçado por Hart e Corne [62], esta discussão sobre a melhor representação ainda está por ser estudada mais convenientemente.

3.3 Avaliação e Atribuição de Aptidão

Num problema de optimização existe uma função objectivo que avalia a solução no contexto do problema. Nos algoritmos evolutivos o resultado da avaliação, ou custo da solução, é mapeado num valor de aptidão que discrimina a qualidade das soluções.

Existem dois tipos de mapeamento para aptidão: atribuição de aptidão proporcional e por seriação [56]. No primeiro caso, a aptidão é calculada como uma função linear do custo da solução. A desvantagem desta implementação reside na distribuição da aptidão relativa pelos indivíduos. Um indivíduo com um valor de aptidão muito elevado pode rapidamente dominar toda a população através do operador de selecção e promover a convergência prematura num óptimo local.

Na seriação, o mapeamento é efectuado pela ordenação das soluções com base no seu custo. Os valores de aptidão são atribuídos aos indivíduos de acordo com a sua posição na população. Deste modo, ao melhor indivíduo é sempre atribuída a mesma aptidão, impedindo que um "super-indivíduo" se reproduza excessivamente e possibilitando a afectação do melhor indivíduo à mesma aptidão numa população quase homogénea. Numa população de tamanho N , atribuindo posição zero ao melhor indivíduo e posição $N - 1$ ao pior e representando a classificação por r e aptidão relativa por $f = f(r)$, o mapeamento linear de aptidão com base em seriação pode ser definido pela seguinte expressão:

$$f(r) = s - (s - 1) \frac{2r}{N - 1}$$

em que s , $1 < s \leq 2$ corresponde à aptidão relativa desejada para o melhor indivíduo [45]. Contudo, nas abordagens à resolução de Problemas de Geração de Horários, estes aspectos não têm tido a importância necessária.

Nesta classe de problemas, o valor da avaliação é normalmente baseado no custo da violação das restrições do problema, como é exemplo a seguinte equação utilizada por Erben e Keppler [36] e Ross *et al.* [86]:

$$aval(f) = \frac{1}{1 + V(f)}$$

em que f é uma solução e $V(f)$ é uma função de penalidade representada pela soma ponderada do número de violações das restrições verificada em f . Burke *et al.* [12] implementaram uma função de aptidão mais complexa, adequada à metodologia por eles desenvolvida:

$$aval(f) = \frac{E}{U + \sum_{i=0}^{P-1} V(p_i, p_{i+1})}$$

em que E representa o número de eventos, U é o número de eventos que não foram marcados e V é uma função que representa o número de violações de restrições entre dois períodos no mesmo dia.

3.4 Selecção

O objectivo principal da selecção é favorecer as melhores soluções numa população de acordo com valor de aptidão. Os vários operadores de selecção propostos na literatura pretendem dar maior probabilidade de escolha a uma solução com maior valor de aptidão. A diferença entre estes operadores reside no modo como as cópias são atribuídas às melhores soluções.

Associado à selecção está o conceito de pressão selectiva que determina a velocidade com que a melhor solução da população inicial irá dominar toda a população com aplicações repetidas do operador de selecção [4] [58]. Normalmente, é entendido como o número de descendentes que o melhor indivíduo conseguirá obter na selecção. A pressão selectiva não pode ser excessivamente elevada, pois pode promover a convergência prematura devido à falta de diversidade. Um valor ideal de pressão selectiva é o que promove uma convergência suficientemente lenta, permitindo a interacção entre os operadores de cruzamento e mutação para pesquisar o espaço de procura, mas não tão baixa que a pesquisa seja dominada por fenómenos de deriva genética.

Os métodos de selecção mais comuns são o *Roulette Wheel Sampling* (RWS) [56], *Stochastic Universal Sampling* (SUS) [5] e o Torneio [58]. O operador de selecção RWS consiste numa sequência de selecções independentes em que os indivíduos são mapeados em segmentos contíguos de uma linha. Cada segmento do indivíduo é igual em tamanho à sua aptidão normalizada. Um número aleatório é gerado e o indivíduo é seleccionado se o segmento ao qual pertence corresponder ao

valor gerado. O processo continua até se obter um número desejado de indivíduos. Uma das desvantagens deste método é a ocorrência de erros de selecção elevados. Visto que o número de descendentes deverá ser inteiro, este será necessariamente superior ou inferior ao valor desejado. No caso do RWS, o número actual de descendentes de a um indivíduo pode ficar muito longe do valor esperado.

O operador de selecção SUS consiste numa só triagem de n indivíduos que estão mapeados em segmentos contíguos do mesmo modo que no RWS. Contudo, neste operador, considera-se a existência de n ponteiros igualmente espaçados e colocados sobre os segmentos contíguos. Considerando nP como o número de indivíduos a serem seleccionados, a distância entre ponteiros será $1/nP$ e a posição do primeiro é gerada aleatoriamente entre $[0, 1/nP[$. A vantagem deste operador em relação ao RWS é que garante um desvio mínimo em relação ao valor desejado.

De um modo simplista, o operador de selecção Torneio consiste na selecção aleatória de um par de indivíduos da população que são comparados entre si. O melhor indivíduo ganha o torneio e é seleccionado. Nas implementações práticas pode existir uma maior componente probabilística. A este operador é associado um valor aleatório r e um determinado parâmetro k em que, se $r < k$, o melhor indivíduo é seleccionado, caso contrário, o pior.

Ross *et al.* [87] foram os únicos que compararam o desempenho de vários operadores de selecção em Problemas de Geração de Horários. Estes autores observaram que a selecção Torneio seria a melhor opção para Problemas de Geração de Horários, seguido da selecção baseada em seriação e da selecção proporcional à aptidão. A escolha deste operador também é realçada devido à redução do tempo computacional na avaliação, pois só os indivíduos escolhidos para Torneio é que seriam avaliados. Contudo, os autores afirmam que seria necessário executar mais simulações com vários parâmetros nos operadores de selecção para confirmarem a generalização dos resultados obtidos.

3.5 Mutaç o

A muta o   um operador que altera, com uma probabilidade controlada, o valor de cada gene num cromossoma de um indiv duo. Algumas conclus es emp ricas foram determinadas pela comunidade cient fica sobre o papel da muta o.   de real ar os resultados obtidos por Fogarty [39], que observou uma melhoria no desempenho do algoritmo quando a probabilidade de muta o inicial era elevada e que decrescia exponencialmente com o n mero de itera es. Back [3] tamb m

verificou o mesmo para problemas multimodais e definiu um limite de probabilidade de mutação óptima $P_m = 1/L$, em que L é o comprimento do cromossoma.

De um modo geral, é necessário um cuidado especial com a probabilidade de mutação, pois se for muito elevada, pode impedir a convergência para uma solução óptima, transformando-se a pesquisas numa pesquisa aleatória. Se for muito reduzida, pode promover uma convergência antecipada, estagnar num óptimo local e não ter capacidade de explorar outras zonas do espaço de pesquisa. Recentemente, Ochoa *et al.* [79] determinaram uma relação entre a probabilidade de mutação dos algoritmos evolutivos e a pressão selectiva, com base no conceito de limite erro, usado no estudo da evolução molecular. Este conceito representa a probabilidade de mutação crítica em que a estrutura obtida pelo processo evolutivo é destruída com maior frequência por mutação do que é reproduzida por selecção. Estes autores identificam, assim, uma relação entre a pressão selectiva e a probabilidade de mutação, e que será um conceito abordado neste estudo.

Em problemas em que o cromossoma é representado por uma permutação, é necessário um cuidado especial com o desenho do mecanismo de mutação para assegurar a admissibilidade da solução. É também necessário ter em consideração que uma simples mutação de um gene num cromossoma pode provocar o movimento da solução para um ponto distante do espaço de procura. Deste modo, tem sido natural que as abordagens aos operadores de mutação favoreçam a pesquisa local. Alguns operadores de mutação foram definidos para determinados problemas como o problema do caixeiro viajante. Neste problema, é possível tirar partido da adjacência entre os elementos, definindo um algoritmo genético com mutação baseada no algoritmo 2-opt, em que a ordem de uma sequência entre dois pontos no cromossoma é invertida. Syswerda [98] definiu vários operadores de mutação para problemas de escalonamento e afectação de recursos. Os operadores definidos baseiam-se na troca de posições entre dois genes e na reordenação aleatória de uma sequência da permutação.

Em Problemas de Geração de Horários têm surgido algumas propostas inovadoras de operadores de mutação. Burke *et al.* [9], na resolução de um PGHA, propõem um operador de mutação que altere a afectação do período e sala ao exame de acordo com determinadas prioridades. Ross *et al.* [87] implementaram um operador de mutação dinâmico em que a probabilidade de mutação iniciava a 0,003 e aumentava 0,0003 em cada geração até a um limite de 0,02. Mais tarde, os mesmos autores [88] apresentaram um novo tipo de operador de mutação que consiste num mecanismo que atribui uma maior probabilidade de mutação aos genes que apresentam maior violação das restrições. Este operador é constituído por dois aspectos distintos, respectivamente, a escolha da posição a mutar e a

escolha do valor a atribuir-lhe. Estes autores observaram que os métodos determinísticos da escolha do valor a atribuir, independente da posição escolhida, produziam melhores resultados. Corne *et al.* [28] generalizam este operador para Problemas de Geração de Horários com representações directas e distinguiram entre operadores de mutação *violation-directed* e *event-freeing*. Ambos assumem a escolha da posição para mutação com base naquele que apresentar maior violação nas restrições. A distinção reside no método de afectação do valor à posição escolhida. A mutação *violation-directed* atribui um intervalo de tempo aleatório e a mutação *event-freeing* atribui um intervalo de tempo que reduz, em maior grau, a violação das restrições. O operador de mutação proposto no Capítulo 4 segue uma abordagem na linha destes autores.

3.6 Recombinação

A recombinação ocorre entre pares de cromossomas para troca de fragmentos de informação. No caso da representação directa, este mecanismo é usualmente constituído pelo seguinte procedimento:

- Dois indivíduos são escolhidos da população;
- Uma ou mais localizações são determinadas para delimitar a sequência para troca;
- Os segmentos são trocados e combinados para criar um par de descendentes.

Associado a este operador existe uma probabilidade de recombinação que determina a frequência com que este operador é utilizado por cada geração. Estudos empíricos indicam que este valor deverá estar compreendido entre 0,75 e 0,95 [92]. Existem, contudo, outras técnicas que permitem adaptar este valor no decurso da execução do algoritmo genético [32].

O primeiro tipo de recombinação definido foi o cruzamento de um ponto. Consiste na selecção de um ponto de recombinação e troca de segmentos contíguos que começam ou terminam nesse ponto. Outra variante deste operador denominado recombinação de n pontos consiste em seleccionar n pontos para recombinação. Syswerda [97] definiu a recombinação uniforme em que a decisão de trocar cada posição do cromossoma é efectuada probabilisticamente em cada posição do cromossoma. Booker [7] implementou a recombinação *reduced-surrogate*, que restringe os pontos de recombinação aos locais onde os genes dos cromossomas dos indivíduos seleccionados apresentam valores diferentes, garantindo que os descendentes sejam diferentes dos pais. Eshelman *et al.* [37]

implementaram a recombinação *shuffle*, em que o cromossoma é previamente permutado aleatoriamente antes de ser submetido à recombinação de um ou n pontos. Depois de aplicar este operador, o cromossoma regressa à ordem original.

Embora a maioria das implementações de algoritmos evolutivos aplicados a Problemas de Geração de Horários apresentem operadores de recombinação convencionais, algumas variantes demonstraram sucesso em alguns casos. Corne *et al.* [28] apresentaram um método de recombinação que utiliza a informação do grau de violação das restrições. Dada a selecção de dois indivíduos, consideram-se os valores *maxv* e *minv* que representam o máximo e o mínimo do grau de violação verificados em cada um deles. Para cada evento i , calcula-se a probabilidade p_i em que

$$p_i = \frac{v_i - \text{minv}}{1 + \text{maxv} - \text{minv}}$$

e em que p_i representa o número de violações no evento i . O gene i do descendente é afecto ao gene i do pai com probabilidade $p_i - 1$.

Num PGHE, Burke *et al.* [11] estudaram vários tipos de operadores de recombinação para representações directas em que a admissibilidade dos descendentes é mantida por um algoritmo de selecção dos exames que irão ser trocados. Os autores implementaram vários métodos de selecção, respectivamente, selecção aleatória, baseada em heurísticas de coloração de grafos, heurísticas específicas e combinação de heurísticas. Foi observado que este último método apresentava melhor desempenho.

Terashima [99] definiu um operador de recombinação para PGHE baseado em cliques. Como é possível formalizar um PGHE como um Problema de Coloração de Grafos, a ideia deste operador é extrair as cliques resolvidas de dois progenitores para gerar um descendente. Uma clique é um sub-grafo com conexão máxima, ou seja, todos os nós desse sub-grafo estão ligados entre si. Segundo este autor, e transportando a analogia para PGHEs, uma clique é um agrupamento de exames que têm de ser marcados em tempos diferentes devido a uma determinada restrição. O objectivo do operador é agregar e combinar conjuntos de cliques resolvidas a partir de cada progenitor, assegurando a admissibilidade da solução.

3.7 População

A utilização de uma população de soluções representa uma das grandes vantagens dos algoritmos evolutivos, pois permite a exploração simultânea de vários pontos dispersos no espaço de pesquisa.

Contudo, o tamanho da população tem sido bastante questionado, pois certos autores apontam para uma população reduzida [92], outros para uma população de grandes dimensões [57] e outros para variação do tamanho da população durante a execução [75].

É comum a ideia que uma população com uma dimensão maior do que o necessário para a resolução do problema poderá fornecer diversidade, mas representará um custo computacional desnecessário. Por outro lado, uma população demasiado pequena favorecerá uma convergência antecipada em óptimos locais. É necessário ainda estudar convenientemente qual o tamanho da população ideal para determinados tipos de problemas, tendo em conta a utilização de certos operadores e da representação utilizada. Muhlenbein [77] observou que a mutação é ineficiente em grandes populações nos problemas de optimização combinatória, necessitando de uma pressão selectiva muito forte. Por outro lado, Allender [1] observou que o algoritmo evolutivo seria eficiente se a população fosse suficientemente grande para recombinar os possíveis blocos constructivos das soluções.

Contudo, o tamanho da população não tem sido alvo de grandes debates na aplicação da metodologia evolutiva a Problemas de Geração de Horários. Pelo contrário, a inicialização do algoritmo evolutivo, na geração da população, tem sido suficientemente bem discutida. Em geral, estratégias de reinicialização aleatória em problemas de optimização combinatória tem sido utilizada para algoritmos evolutivos [98]. Contudo, nas várias abordagens da metodologia evolutiva aplicada à resolução de Problemas de Geração de Horários, vários autores utilizam heurísticas constructivas iniciais para gerar indivíduos na população. Corne *et al.* [29] efectuaram várias inicializações do algoritmo evolutivo para resolução de um PGHE para comparar o desempenho de um algoritmo "guloso" com um algoritmo "esfomeado". Enquanto que o primeiro tem por objectivo verificar todos os intervalos de tempo disponíveis, seleccionando o que apresenta menos conflitos, o algoritmo "esfomeado" verifica só k intervalos de tempo. Estes autores verificaram a superioridade desta última abordagem.

As diferentes estruturas da população têm vindo a ser estudadas em problemas que se pretende obter mais do que uma solução. Problemas do domínio da classificação, simulação de sistemas complexos e adaptativos e funções de optimização multimodais e multiobjectivo requerem a localização e manutenção de múltiplas soluções. Os métodos de *ninching* são técnicas que promovem a pesquisa de vários óptimos no espaço de procura e dividem-se em dois tipos: partilha de aptidão [55] e *crowding* [72]. A partilha de aptidão é um mecanismo que reduz o valor de aptidão de cada

indivíduo em função do número de indivíduos semelhantes na população. Por outro lado, a técnica de *crowding* consiste na inserção de novos elementos na população, substituindo os indivíduos semelhantes. Estes métodos apresentam a desvantagem de permitir o cruzamento entre duas soluções que se encontram em dois pontos óptimos distintos, produzindo, porventura, indivíduos letais. Um modo de superar esta desvantagem consiste em só permitir o cruzamento de indivíduos semelhantes [56].

Outros modelos de população estruturada são utilizados para adaptar os algoritmos evolutivos ao ambiente de processamento paralelo, respectivamente o modelo de ilhas [60] e difusão [53]. O primeiro consiste na execução simultânea de vários algoritmos evolutivos em processadores diferentes. Periodicamente, fracções das populações são trocadas entre os algoritmos genéticos, implementando um mecanismo de migração. No modelo de difusão a população está distribuída numa grelha de processadores em que cada indivíduo interage unicamente com os seus vizinhos directos.

Da extensa literatura sobre Problemas de Geração de Horários, o estudo elaborado por Turner [103] realça as diferentes estruturas de população nos algoritmos evolutivos para estes problemas. Sem ter por objectivo a obtenção de múltiplas soluções para resolver o problema de uma perspectiva multiobjectivo, muitas das técnicas utilizadas são perfeitamente adaptadas para este tipo de optimização. No Capítulo 4 deste estudo, devido à sua ligação com a abordagem multiobjectivo, será alvo de uma descrição mais pormenorizada.

3.8 Restrições

A resolução de problemas com restrições através de algoritmos evolutivos não é trivial. Muitas das técnicas convencionais sobre metodologia evolutiva não permitem lidar directamente com as restrições no espaço de procura. Muitas abordagens têm sido testadas para resolver este problema, desde utilização de funções de penalidade, decodificação, algoritmos de reparação e desenho de operadores especiais.

Um das técnicas mais utilizadas são as funções de penalidade. O objectivo destas funções é penalizar, de algum modo, as soluções que violam as restrições do problema. No caso de restrições com diferentes graus de importância, é possível introduzir coeficientes na função objectivo. A desvantagem deste método é evidenciada nos problemas com vários objectivos distintos, em que é difícil determinar correctamente os coeficientes de ponderação na função objectivo.

A descodificação consiste num mapeamento do espaço de representação na região admissível do espaço de solução. Deste modo, o cromossoma é interpretado por um descodificador sobre o método de construir a solução. Este mecanismo apresenta a desvantagem de ser dependente do problema, dificultar o estudo do algoritmo devido à representação não corresponder directamente à solução e de nem todas as restrições serem possíveis de serem implementadas.

Os algoritmos de reparação consistem no mapeamento de uma solução não-admissível numa solução admissível. Estes algoritmos podem alterar aleatória ou deterministicamente a solução não-admissível. Michalewicz [75] observou que a reparação determinística apresentava melhor desempenho em problemas combinatórios. Contudo, este mecanismo também sofre da desvantagem de ser demasiado dependente do problema, dificultando generalizações.

Os operadores que preservam a admissibilidade das soluções são métodos alternativos para os algoritmos evolutivos lidarem com as restrições. A anterior definição de operadores de mutação e cruzamento refere alguns operadores aos quais foi adicionado conhecimento do problema em estudo. Mais uma vez, a desvantagem deste método é similar à verificada para algoritmos de reparação e descodificação.

Outros métodos mais recentes de lidar com as restrições consistem no modelo co-evolutivo [84] e algoritmos evolutivos multiobjectivo [41]. No primeiro, as restrições e as soluções interagem entre si, com base no modelo de predador e presa na Natureza. Os algoritmos evolutivos multiobjectivo serão descritos na próxima secção e serão a base deste estudo.

O modo de lidar com restrições com algoritmos evolutivos aplicados a Problemas de Geração de Horários tem sido abordado por funções de penalidade [87], descodificação [81], algoritmos de reparação [18] e desenho de operadores especiais [12]. Recentemente surgiram algumas abordagens [19] [17] que realçam a necessidade de alterar o modo de lidar com as restrições. Como estas abordagens estão relacionadas com perspectivas multi-critério e multiobjectivo ao PEHs, serão referidas no contexto da secção seguinte sobre Algoritmos Evolutivos Multiobjectivo.

3.9 Algoritmos Evolutivos Multiobjectivo

Muitos problemas de optimização da vida real apresentam múltiplos objectivos conflituosos e não-comensuráveis. O Decisor pode, porventura, pretender um conjunto de boas soluções equivalentes mas dispersas no espaço objectivo para escolher aquela que mais lhe interessa. Os algoritmos

evolutivos, porque se baseiam numa população de soluções e exploram paralelamente várias regiões do espaço de pesquisa, são um dos métodos mais promissores para este tipo de optimização.

3.9.1 Optimização Multiobjectivo

Formalização

Um problema de optimização multiobjectivo é um problema de minimização simultânea de n componentes f_k , $k = 1, \dots, n$, de uma função f não-linear de uma variável de decisão x num universo S onde

$$f(x) = (f_1(x), \dots, f_n(x))$$

Este tipo de problemas não apresenta, normalmente, uma solução única, mas um conjunto de soluções não-dominadas, denominado conjunto Ótimo de Pareto. Assumindo um problema de minimização, um vector $u = (u_1, \dots, u_n)$ domina $v = (v_1, \dots, v_n)$ sse é parcialmente inferior a v , ou seja,

$$\forall i \in \{1, \dots, n\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, n\} : u_i < v_i.$$

Uma solução $x_u \in S$ é um Ótimo de Pareto sse não existir $x_v \in S$ em que $v = f(x_v) = (v_1, \dots, v_n)$ domina $u = f(x_u) = (u_1, \dots, u_n)$.

Contudo, a escolha de uma solução de compromisso entre várias alternativas não-dominadas depende das preferências subjectivas do decisor. Deste modo, a solução final é o resultado tanto de um processo de optimização como de um processo de tomada de decisão [45].

Articulação de Preferências

As três seguintes classes de métodos de optimização multiobjectivo podem ser definidas com base na combinação do processo de optimização e de tomada de decisão na pesquisa de soluções de compromisso [45]:

- Articulação *a priori* de preferências, em que o decisor expressa as preferências através da combinação de diferentes objectivos numa função escalar de custo, transformando o problema multiobjectivo num problema com um só objectivo;

- Articulação *a posteriori* de preferências, em que o otimizador apresenta um conjunto de soluções candidatas não-dominadas e em que o decisor expressa as preferências e escolhe a solução de compromisso;
- Articulação progressiva de preferências, em que o otimizador fornece conjuntos de alternativas ao decisor durante o processo de otimização e este indica ao otimizador quais de entre as que lhe são apresentadas.

A articulação de preferências exprime o custo que discrimina as soluções candidatas. Seguidamente se apresentam algumas abordagens à formalização destas funções de custo:

- Coeficientes de Ponderação, que são valores reais que expressam a importância relativa dos objectivos. A soma ponderada é um exemplo clássico que pode ser definido como:

$$F_{sp}(x) = \sum_{i=1}^n c_i f_i(x)$$

em que n corresponde à soma dos objectivos e as ponderações c_i são constantes positivas. Este método apresenta a desvantagem de ser sensível à definição das ponderações e dependente do problema;

- Prioridades, que são valores inteiros que determinam por que ordem os objectivos devem ser optimizados, de acordo com a sua importância. Um exemplo é o método lexicográfico. Dado n objectivos f_1, \dots, f_n , a cada objectivo f_i , $i = 1, \dots, n$ é atribuído uma prioridade i , onde valores elevados de i correspondem a prioridades elevadas. A função de custo lexicográfica F_{lex} é definida por:

$$F_{lex}(x) < F_{lex}(y) \iff \exists i \in \{1, \dots, n\} : \forall j \in \{1, \dots, i\}, f_j(x) \leq f_j(y) \wedge f_i(x) < f_i(y).$$

O método lexicográfico assume que a prioridade dos objectivos é bastante clara, o que pode não ser o caso em vários tipos de problemas;

- Metas, que indicam valores desejados de desempenho em cada dimensão de objectivos. Esta definição é expressa por um vector $G = \{G_1, G_2, \dots, G_n\}$ que está associado ao vector de objectivos $f(x)$. As metas são mais fáceis de definir porque se aproximam mais do que se entende por solução final do problema [47].

Optimização com Restrições

A satisfação de restrições é claramente um problema multiobjectivo de minimização de funções até certas metas serem atingidas. Quando nem todas as metas podem ser atingidas simultaneamente, o conjunto de pontos não-dominados é a solução do problema.

Quando existe uma definição da importância das restrições no problema, ou uma clara distinção entre restrições *soft* e *hard*, a combinação das definições de prioridades e metas ajudam a uma clara definição da estratégia de classificação de múltiplas soluções. Fonseca e Fleming [47] estabeleceram operadores de comparação que incorporam e caracterizam a informação sobre as prioridades e metas. Segundo estes autores, e de um modo simples, dois vectores de soluções são comparados primeiro em relação aos componentes com prioridade mais elevada, deixando de lado os que já cumpriram a meta. Se ambos cumpriram todas as metas com esta prioridade, ou se violaram algumas ou todas, mas do mesmo modo, considera-se o nível de prioridade seguinte. O processo continua até atingir o nível de prioridade mais baixo, em que o resultado é decidido pela comparação dos componentes com esta prioridade na fronteira de Pareto.

3.9.2 Revisão dos Algoritmos Evolutivos Multiobjectivo

Schaffer [91] foi o primeiro a desenvolver um modelo de algoritmo evolutivo para resolução de problemas multiobjectivo. O *Vector Evaluated Genetic Algorithm* (VEGA) proposto por este autor apresenta um operador de selecção que selecciona os indivíduos separadamente para cada objectivo e em proporções iguais.

Fonseca e Fleming [41] implementaram abordagens de Pareto com a definição de dominância integrada na afectação de aptidão. Estes autores propõem um método de seriação em que a posição de um indivíduo é igual ao número de indivíduos da população que o dominam. A média dos valores de aptidão é calculada para os indivíduos com posições iguais. Este conceito foi extendido por estes autores para incluir a partilha adaptativa da aptidão [43] e introdução de imigrantes aleatórios [47].

Horn e Nafpliotis [66] combinaram a selecção por Torneio e o conceito de dominância para o desenho do *Niched Pareto Genetic Algorithm* (NPGA). Neste operador de selecção dois indivíduos são seleccionados aleatoriamente com base numa amostra da população. Se um dos indivíduos domina o outro, o primeiro é escolhido como vencedor do torneio. Se não existir nenhum caso de domínio de um indivíduo sobre o outro, o torneio é decidido pela partilha de aptidão.

Srinivas e Deb [95] implementaram o *Nondominated Sorting Genetic Algorithm* (NSGA) que utiliza o conceito de profundidade de dominância. A primeira frente de indivíduos não-dominados pertence ao nível um de não-dominância, a frente seguinte pertence ao nível dois de não-dominância, e assim por adiante. A aptidão de um indivíduo está relacionado com o nível de não-dominância a que pertence, em que o primeiro nível apresenta maior probabilidade de reprodução. Estes autores admitem que a aptidão é partilhada dentro de cada nível de não-dominância.

O *Strength Pareto Evolutionary Algorithm* (SPEA) desenvolvido por Zitzler e Thiele [104] admite a existência de uma segunda população externa que apresenta as soluções não-dominadas actualizadas continuamente. Se esta população exceder um determinado limite é reduzida por métodos de *clustering*. A aptidão é atribuída a indivíduos de ambas as populações de uma forma independente. A aptidão de um indivíduo na segunda população é inversamente proporcional ao número de indivíduos na população da corrente geração que ele próprio domina e vice-versa. Os indivíduos de ambas as populações são sujeitos ao mesmo operador de selecção.

A proposta apresentada por Fonseca e Fleming [41] foi a metodologia escolhida para este estudo.

3.9.3 Abordagens de Optimização Multiobjectivo para Problemas de Geração de Horários

As abordagens evolutivas multiobjectivo a Problemas de Afectação de Recursos, em particular no Problema de Geração de Horários, são recentes e em número muito reduzido. Até à data da conclusão deste estudo, só houve uma abordagem publicada de algoritmos evolutivos multiobjectivo por Carrasco e Pato [19] aplicado a um PGHA.

Turner [103], não de uma perspectiva multiobjectivo, aplicou várias estruturas de população para resolver Problemas de Geração de Horários, respectivamente, selecção espacial, ilhas, *crowding*, partilha e tribos, esta última proposta pelo autor. Foi verificado que, em várias execuções de um algoritmo evolutivo em problemas com um grande número de soluções, o tempo para obter soluções distintas foi mais reduzido com a aplicação de um algoritmo evolutivo simples do que com qualquer outro método de população estruturada. Porém, em problemas com número reduzido de soluções, a selecção espacial e tribos foram mais eficientes na obtenção de soluções distintas. Contudo, a intenção do autor era obter uma solução óptima e várias sub-óptimas, sem considerar a possibilidade de obter várias soluções não-dominadas no espaço de pesquisa.

Corne *et al.* [27] afirmaram que um futuro tema de investigação para Problemas de Geração de

Horários seria o emprego de otimização de Pareto. Estes assumem que seria um método eficiente para entender a interação entre diferentes restrições sem especificar ponderações na função objetivo. Paechter *et al.* [82] implementaram um *interface* visual multiobjectivo para um algoritmo evolutivo aplicado a um PGHA que permite ao utilizador interagir com as ponderações e metas atribuídas à função objetivo.

Só recentemente surgiram algumas inclusões da teoria multi-critério para lidar com este tipo de problemas. Burke *et al.* [17] propõem uma abordagem multi-critério ao PGHE. A metodologia apresentada por estes autores sugere que o processo de otimização seja realizado em duas fases. O objectivo da primeira fase é encontrar horários de qualidade elevada em relação a cada critério. Na segunda fase é realizada uma negociação entre os valores dos critérios para encontrar uma solução de compromisso.

Por sua vez, Carrasco e Pato [19] desenvolveram uma aplicação para resolver PGHAs de uma perspectiva multiobjectivo. Os objectivos considerados por estes autores são a procura de soluções óptimas de Pareto que satisfaça os docentes e estudantes. Utilizando o *Nondominated Sorted Genetic Algorithm* com partilha de aptidão e selecção elitista, com população secundária e com operadores específicos de cruzamento e mutação, estes autores reportam sucesso em relação à versão manual. Contudo, este excessivo número de parâmetros e combinação de metodologias tornam a proposta algo complexa. No entanto é de realçar que foi a primeira proposta até à corrente data a empregar metodologias consagradas na optimização evolutiva multiobjectivo.

3.10 Conclusões

Muitos problemas podem ser formulados através de um único objectivo que se pretende minimizar ou maximizar. Os algoritmos evolutivos têm demonstrado bom desempenho na resolução destes problemas. Contudo, muitos dos problemas da vida real necessitam de uma formalização que considere vários objectivos separadamente. O problema consiste em encontrar soluções de compromisso dando igual importância aos objectivos presentes. Neste sentido, alguns conceitos típicos dos algoritmos evolutivos deverão ser alterados.

Os algoritmos evolutivos multiobjectivo descritos na secção anterior têm lidado com a resolução deste tipo de problemas. Com a excepção do proposto em [91], todas as abordagens visam, de uma maneira ou de outra, a obtenção de uma fronteira de soluções não-dominadas denominada

fronteira de Ótimo de Pareto. A definição de metas e prioridades [41] disponíveis da formalização do problema adicionam as valências necessárias ao algoritmo evolutivo multiobjectivo para lidar com vários objectivos de acordo com as preferências definidas pelo decisor. Contudo, é necessário encontrar um equilíbrio entre a concentração e a dispersão do processo de pesquisa. O algoritmo deve ser capaz de se concentrar na satisfação das metas propostas, mas sem perder a perspectiva multiobjectivo da satisfação simultânea dos objectivos presentes na formalização.

O próximo Capítulo demonstra a aplicação da metodologia proposta por Fonseca e Fleming [41] na resolução de Problemas de Afectação de Recursos em que é possível localizar as violações das restrições existentes, como é o exemplo do Problema de Geração de Horários.

Capítulo 4

Um Algoritmo Evolutivo Multiobjectivo para Problemas de Afectação de Recursos

4.1 Introdução

A abordagem expressa neste Capítulo lida com Problemas de Afectação de Recursos em que o processo de decisão contempla objectivos múltiplos e em que é possível localizar as afectações que dão origem a violações das restrições. O Problema de Geração de Horários, por consistir num problema que envolve objectivos múltiplos, muitas vezes conflituosos, e onde é possível obter a informação sobre a violação das restrições, é um exemplo típico que pode ser resolvido por esta abordagem. Esta abordagem tem por princípio a não utilização de técnicas constructivas que podem enviesar implicitamente o processo de pesquisa e não corresponder ao processo de decisão definido com base em metas e prioridades. Contudo, assume-se a necessidade de enviesar, de um modo explícito, o processo de pesquisa com base na satisfação das restrições, actuando sobre a violação das restrições, no sentido de acelerar a convergência para o óptimo.

Neste Capítulo serão apresentadas as técnicas evolutivas propostas para abordar os problemas no âmbito deste estudo. A secção 4.2 apresenta o tipo de representação que deverá ser utilizado. A qualidade da solução é expressa na secção 4.3 e a atribuição de aptidão aos indivíduos será referida na secção 4.4. O operador de mutação, recombinação e selecção serão referidos, respectivamente,

nas secções 4.5, 4.6 e 4.7. Por fim, a secção 4.8 apresentará conclusões sobre esta abordagem.

4.2 Representação

Num problema de afectação de recursos, segundo uma formalização multiobjectivo, a representação directa parece ser a codificação mais apropriada. Além de mais simples, facilmente se contabiliza o nível de satisfação das restrições. As representações indirectas e com base em permutações [81] ao processarem sempre soluções admissíveis com base em heurísticas para satisfação total das restrições *hard*, dão maior importância aos objectivos para os quais as heurísticas foram desenvolvidas, enviesando implicitamente o processo de pesquisa. Uma representação directa em que cada gene corresponde a um item e que apresenta um conjunto de recursos disponíveis, permite que o enviesamento da pesquisa no sentido dos objectivos seja fornecido pelas próprias funções objectivo e respectivas prioridades e metas [41]. A adição de operadores evolutivos no processo de pesquisa que utilizem alguma informação do problema podem ser igualmente um factor de enviesamento no processo de pesquisa. Este enviesamento deve, contudo, ser suficientemente explícito.

A codificação de um indivíduo num problema de afectação de recursos pode ser representada por um vector r em que

$$r = (r_1, r_2, \dots, r_n)$$

e cada r_i é um vector que representa os índices dos recursos afectos ao item i . A desvantagem deste tipo de representação em relação a uma representação indirecta que codifica o modo como a solução será construída é o facto de facilitar a excessiva produção de soluções inadmissíveis. Contudo, o desenho de operadores diferentes dos convencionais que aproveitam o conhecimento do problema pode minorar este obstáculo.

4.3 Qualidade da Solução

O conceito de restrições está intimamente ligado aos Problemas de Afectação de Recursos, pois são elas que limitam a afectação. O grau de violação das restrições representa a qualidade da solução, ou seja, posiciona a solução no espaço de pesquisa do problema. A posição da solução em relação à população indica a sua aptidão, tornando-a independente da escala dos objectivos.

O processo de tomada de decisão multiobjectivo proposto em [47] permite a representação de um Problema de Afecção de Recursos através da combinação de prioridades e metas que estão disponíveis na formalização do problema. A distinção simples entre restrições *hard* e *soft* permite definir claramente os níveis de prioridade em cada objectivo. De acordo com a vontade do decisor, é possível ir mais longe e segmentar os objectivos com vários níveis de prioridade de acordo com a sua importância, ou agregar restrições, visto que a presença de um número elevado de objectivos conflituosos pode produzir um grande número de soluções não-dominadas em que é impossível tomar uma decisão. Num caso de Problema de Geração de Horários, é possível relacionar determinadas classes de objectivos com diferentes níveis de prioridades. Muitas das restrições de aresta e justaposição podem ser representadas como objectivos com prioridades elevadas. Outras restrições, como dispersão de eventos ou de agentes podem ser definidas como objectivos de prioridades mais reduzidas.

As metas são definidas de acordo com o desempenho que se pretende obter com o processo de optimização. Na realidade, definem a fronteira da admissibilidade do espaço de pesquisa. Uma solução que cumpre todas as metas apresenta a satisfação total dos objectivos. Contudo, uma solução que cumpre unicamente as metas associadas aos objectivos associados a restrições *hard* é possível, mas insatisfatória. Uma solução que não atinge pelo menos uma meta associada a esses objectivos é considerada inadmissível.

Neste tipo de problemas é possível localizar os genes envolvidos em violações das restrições. Esta informação pode ser utilizada pelos operadores evolutivos para melhorar a qualidade da solução. Deste modo, assume-se a existência de um padrão de aptidão para cada solução que indica quais os genes no cromossoma mais susceptíveis a serem modificados na aplicação dos operadores. Um padrão de aptidão é representado por um vector *apt* em que

$$apt = (apt_1, apt_2, \dots, apt_n)$$

e em que apt_i é um valor que representa o número das violações de cada tipo de restrição associado ao item i . No caso de afectação de vários recursos distintos a cada item, é necessário segmentar as violações das restrições para cada recurso com ele relacionado, em que

$$apt_i = (apt_{i1}, apt_{i2}, \dots, apt_{im})$$

e apt_{ij} é um vector que representa a quantificação das violações de uma restrição que está relacionada com um determinado recurso j .

4.4 Atribuição de Aptidão

Ao contrário do caso de um só objectivo, a seriação da população em problemas multiobjectivo não é único. Considerando um indivíduo x_u na geração t com um vector u correspondente à função objectivo e admitindo que $r_u^{(t)}$ seja o número de indivíduos na população corrente que lhe são preferíveis, a posição de x_u na seriação dos indivíduos pode ser dada por

$$posição(x_u, t) = r_u^{(t)}$$

o que assegura que todos os indivíduos preferíveis na população sejam afectos a uma posição zero [47].

A atribuição de aptidão deve corresponder ao número de descendentes que um indivíduo espera produzir através do operador de selecção. O processo de selecção determina quais os indivíduos que influenciam a produção da próxima geração e, conseqüentemente, devem fazer parte da estratégia de pesquisa.

Deste modo a afectação de aptidão apresenta os seguintes passos:

1. Seriação da população;
2. Atribuição de aptidão por interpolação linear ou exponencial entre o melhor e o pior indivíduo;
3. Atribuição da média da aptidão a indivíduos com a mesma posição na seriação.

Estes procedimentos não fazem suposições acerca da natureza das variáveis de decisão envolvidas no processo de pesquisa, sendo independentes do domínio, aplicáveis a qualquer tipo de problemas e possíveis de serem adaptados a outras abordagens evolutivas.

4.5 Mutação

Em qualquer abordagem evolutiva é necessário obter um equilíbrio na probabilidade de mutação, porque se esta for demasiado elevada, a população irá divergir e se for demasiado reduzida poderá convergir prematuramente para um óptimo local. Ochoa *et al.* [79] reconhecem que a probabilidade óptima de mutação P_m depende da pressão selectiva μ e do comprimento do cromossoma L . Como referido na secção 3.5, estes autores demonstraram uma relação entre P_m e μ , em que o limite

superior de P_m é deduzido através da fórmula de limite de erro para populações infinitas definida por

$$\frac{\ln(\mu)}{L} \quad (4.1)$$

A abordagem considerada neste estudo utiliza as mesmas considerações sobre a relação entre pressão selectiva e probabilidade óptima de mutação.

Para manter a sobrevivência de pelo menos uma cópia do melhor indivíduo em cada geração numa reinserção geracional, P_m necessita de ter em conta o valor da pressão selectiva μ [48]. A probabilidade de sobrevivência P_s de um cromossoma de comprimento L com uma mutação independente P_m pode ser expressa por

$$P_s = (1 - P_m)^L$$

Sem recombinação, P_s não deverá ser inferior ao inverso da pressão selectiva μ , ou seja, do número esperado de descendentes do melhor indivíduo como proposto em [48]. Assim,

$$P_m \leq 1 - \mu^{-1/L} \quad (4.2)$$

Uma probabilidade de sobrevivência $P_s = 0,5$, e assumindo, conseqüentemente, uma pressão selectiva $\mu = 2$, é possível garantir a sobrevivência do melhor indivíduo para a próxima geração.

Num Problema de Afectação de Recursos, a cada item pode ser atribuído um conjunto de recursos de tamanho variável. Por este motivo, aos itens que apresentam maior número de recursos alternativos, deverá ser atribuída uma probabilidade de mutação mais elevada. De acordo com as considerações anteriores sobre probabilidade de sobrevivência, e assumindo a possibilidade de identificar quais os recursos que podem ser afectos a cada item, defina-se o comprimento efectivo Ω do cromossoma como sendo

$$\Omega = \sum_{i=1}^t \Omega_i$$

em que

$$\Omega_i = (a_i - 1)$$

e onde i corresponde ao i -ésimo item e a_i ao número de alternativas em i . A probabilidade de mutação individual P_{m_i} pode ser calculada por

$$P_{m_i} = 1 - P^{\Omega_i} \quad (4.3)$$

em que

$$P = \mu^{-1/\Omega}$$

Nesta classe de problemas, é também possível localizar as afectações que derão origem a violações das restrições. Ao manter um valor de P_s constante e ao fornecer maior valor de P_m nos genes que apresentam mais violação das restrições e/ou alternativas, o algoritmo enviesará explicitamente o processo de pesquisa em direcção da satisfação das restrições, tendo em conta as alternativas existentes na afectação de recursos. Assim, fazendo

$$\Omega_i = (a_i - 1)(c_i + 1) \quad (4.4)$$

em que i corresponde ao i -ésimo item, c_i ao número de violações de restrições em i e a_i ao número de alternativas em i , assegura-se que a mutação afecta, com diferentes probabilidades, os genes que apresentam um número diferente de violações de restrições e/ou um número diferente de alternativas, mas mantendo a probabilidade de sobrevivência constante. Estas expressões também asseguram que a probabilidade de mutação seja nula no caso da existência de um só recurso a afectar, independente do número de violações de restrições. Por outro lado, se o número de violações numa posição for nulo, mas se esta apresentar vários recursos alternativos, existe uma determinada probabilidade de ocorrer mutação. Este mecanismo pode ser útil para evitar a convergência em óptimos locais. Supondo um caso típico de PGHE de uma Faculdade que consiste na afectação de 100 exames a 30 intervalos de tempo, e supondo igualmente que $\mu = 2$ e que a solução corrente só apresenta um exame envolvido numa só violação, a razão entre probabilidade de mutação deste gene e de qualquer outro é de 1,99, ou seja, aproximadamente o dobro. Este cenário permite que, perto de um óptimo, os genes que cubram maior espaço de Ω , tenham aproximadamente o dobro da probabilidade de mutação do que os genes que não apresentam conflitos, assumindo o mesmo número de alternativas e mantendo P_s constante.

No caso de afectação de vários recursos distintos é natural distinguir entre as violações de restrições relacionadas com determinados recursos. Num PGHA, por exemplo, é necessário contabilizar o número de salas e intervalos disponíveis para cada aula. Neste caso,

$$\Omega_i = \left[\sum_{j \in R_i} (a_{ij} - 1) \right] \left[1 + \sum_{j \in R_i} (c_{ij}) \right] \quad (4.5)$$

onde R_i representa o conjunto de recursos afectos ao i -ésimo item e j representa o j -ésimo recurso afecto ao i -ésimo item. Neste cenário de afectação de vários recursos distintos, é possível ainda considerar que não se deverá alterar uma afectação de um dos recursos que não apresenta violações de restrições. O facto de uma aula apresentar sobreposição temporal numa turma não significa que seja necessário alterar a afectação da sala.

Assumindo que parte da qualidade da solução é expressa pelo número de restrições violadas, será natural enviesar ainda mais a mutação tendo em conta este número. Assim, é possível redefinir (4.5) para

$$\Omega_i = \left[\sum_{j \in R_i} (a_{ij} - 1) \right] \left[1 + \beta \sum_{j \in R_i} (c_{ij}) \right] \quad (4.6)$$

em que se assume um valor de viés β para o número de restrições violadas. Num extremo, em que $\beta = 0$, P_m só contabilizará as alternativas. Se $\beta = \infty$, P_{mi} só será diferente de 0 se o item estiver envolvido em violações de restrições. É de esperar que o valor de β esteja relacionado com a complexidade do problema de optimização. Em princípio, quanto mais difícil for o problema, menor deverá ser o β , de modo a fornecer um enviesamento reduzido necessário para pesquisar outras regiões do espaço de pesquisa. Por outro lado, um problema com reduzida complexidade poderá ser resolvido com $\beta \rightarrow \infty$, ou até sem contabilizar a presença de alternativas. Estas considerações foram já observadas em [42]. Um valor de $\beta = L$, de acordo com o problema anterior, assegura que P_m do exame que apresenta um conflito seja cerca de 87 vezes maior que P_m de qualquer outro exame representado no cromossoma. Este valor de β também garante que a P_s do exame em conflito seja da mesma ordem de grandeza a P_s total de todos os outros exames sem violações associadas.

Visto que a mutação independente necessita de gerar L números aleatórios por cada cromossoma, este procedimento pode tornar-se computacionalmente exigente. Outra formalização mais simplificada pode ser efectuada através de um operador estocástico de selecção para escolher o gene que irá ser submetido ao operador de mutação. Este operador deverá ter em conta o número de violações de restrições, alternativas para cada gene e o parâmetro β . Por questões de simplificação, o primeiro operador denomina-se por mutação independente e o segundo por mutação de uma posição.

4.6 Recombinação

A escolha do operador de recombinação deve ser efectuada de acordo com o tipo de problema que se deve abordar. Cada operador de cruzamento produz um efeito diferente na exploração do espaço de pesquisa e de acordo com a representação do problema.

Eshelman *et al.* [37] caracterizaram o viés de posição e de distribuição de vários operadores de recombinação. Um operador de recombinação apresenta um viés de posição se a criação de um novo *schemata* depende da localização dos genes do cromossoma. O viés de distribuição está relacionado com a quantidade de informação que se espera que venha a ser trocada na recombinação.

Num Problema de Afectação de Recursos como o Problema de Geração de Horários, o viés de posição poderá prejudicar o desempenho do algoritmo, pois aparentemente não existe uma interligação entre genes que estão próximos no cromossoma. Um bom bloco constructivo nesta classe de problemas não é representado necessariamente por um bloco contíguo de genes. Este facto deixa de lado as versões de recombinação com um ou mais pontos, visto que apresentam um forte viés de posição. Pelo contrário, a recombinação *shuffle* [37] evita a ruptura de possíveis *schemata* representados por genes em posições disjuntas, ou seja, não apresenta viés de posição. O viés de distribuição da recombinação *shuffle* depende do modo de partição da informação do cromossoma. Quando combinado com o procedimento *reduced-surrogate* [7], este operador impõe um reduzido viés de distribuição e apresenta a vantagem de produzir sempre descendentes diferentes.

A implementação do operador de recombinação deverá ter em conta as considerações anteriores sobre o papel da mutação na secção anterior. Na presença da recombinação, a P_m óptima deverá estar abaixo do limite em (4.2), ou seja,

$$P_m = 1 - (\delta\mu)^{-1/L}, \quad \delta \leq 1$$

onde δ é um parâmetro que compensará a acção destrutiva da recombinação. Deste modo, a probabilidade de mutação deverá decrescer, tendo em consideração que a recombinação teve alguma probabilidade de destruir os melhores indivíduos. Contudo, os resultados iniciais obtidos no âmbito deste estudo com um $\delta = 0,9$ como sugerido em [45] e probabilidade de recombinação de 0,75 e 0,8, não demonstraram bom desempenho na implementação deste operador, pelo que foi decidido não utilizar este operador. Um estudo mais aprofundado deverá ser efectuada para estudar possíveis valores óptimos de δ em conjunto com a probabilidade de recombinação.

Uma alternativa aos operadores de recombinação referidos é a geração de descendentes com base nos blocos constructivos existentes no cromossoma. Num Problema de Geração de Horários de Exames, por exemplo, é possível admitir que esses blocos estarão relacionados com as restrições. Os exames que pertencem à mesma turma ou os exames que são comuns em várias turmas podem ser talvez encarados como blocos constructivos. Contudo, algumas experiências efectuadas no âmbito deste estudo não demonstraram grande sucesso com esta definição. É possível que os blocos constructivos não se resumem unicamente aos exames de cada turma, visto que duas ou mais turmas podem ter exames em comum.

Lobo [70] apresenta uma metodologia que considera a população como um conjunto de pontos que sumariam os movimentos passados do algoritmo. As frequências dos valores afectos a cada gene no decurso do algoritmo servem, então, de padrão para gerar populações que são semelhantes entre si com base nos melhores indivíduos. Através de medidas de entropia, o algoritmo consegue detectar correlações entre genes e detectar blocos constructivos de outro modo imperceptíveis. Esta abordagem parece ser uma futura área promissora na aplicação da metodologia evolutiva a problemas referidos neste estudo.

4.7 Selecção e Reinserção

O operador de selecção mais apropriado, pelas razões descritas na secção 3.4, é o *SUS* que impede a existência sistemática de erros de selecção. De modo a tirar o máximo partido deste operador, a reinserção da nova população na geração seguinte deverá ser geracional. Uma reinserção incremental, para além de provocar implicitamente um aumento na pressão selectiva, resulta na selecção de um número reduzido de indivíduos de cada vez, o que torna o *SUS* semelhante ao *RWS*.

4.8 Conclusões

A abordagem apresentada neste Capítulo considera a utilização de uma representação directa de modo a permitir que a pesquisa seja guiada, tanto quanto possível, pelos objectivos formulados no problema. Contudo, um certo enviesamento é explicitamente fornecido pelo operador de mutação, que orienta o processo de pesquisa em direcção da redução do número de conflitos, tendo em conta o número de recursos alternativos afectos a cada item. Este operador de mutação foi definido com

base na relação entre a probabilidade de sobrevivência e pressão selectiva, que permite deduzir uma probabilidade óptima de mutação. Um operador de recombinação foi igualmente definido com base nos mesmos conceitos anteriores e propôs-se uma nova abordagem baseada no trabalho efectuado por Lobo [70], para detectar correlações entre genes.

Seguidamente se descreve a aplicação desta abordagem no Problema de Geração de Horários da Universidade do Algarve.

Capítulo 5

Abordagem Multiobjectivo ao Problema de Elaboração de Horários na Universidade do Algarve

5.1 Introdução

A Universidade do Algarve lecciona actualmente um total de 22 cursos de mestrado, 28 cursos de licenciatura, 12 cursos de licenciatura bietápica e 2 cursos de bacharelato a um total de cerca de 10000 estudantes através das suas Unidades Orgânicas de Ensino Universitário (Faculdades) e Politécnico (Escolas Superiores). Até ao ano lectivo de 1999/2000 existiam quatro Unidades de Ensino Superior distintas no Campus de Gambelas, então designadas Unidade de Ciências Económicas e Empresariais (UCEE), Unidade de Ciências Exactas e Humanas (UCEH), Unidade de Ciências e Tecnologias Agrárias (UCTA) e Unidade de Ciências e Tecnologias dos Recursos Aquáticos (UCTRA). As duas primeiras Unidades foram as escolhidas como casos de teste para implementar a metodologia proposta por serem aquelas cujos os dados estavam mais facilmente disponíveis. Este estudo foi realizado num momento de transição decorrente da revisão dos estatutos da Universidade do Algarve e da criação de novas Unidades Orgânicas que a precedeu. Neste sentido, o PGHE apresentado neste capítulo refere-se ao exames de Setembro do ano lectivo 1999/2000 na UCEE e na UCEH. Por outro lado, o PGHA refere-se aos horários do ano lectivo de 2000/01 na Faculdade de Ciências Humanas e Sociais (FCHS) que resultou da agregação das Áreas Departamentais de

Ciências da Educação e Psicologia, Línguas e Literaturas e História e Arqueologia da anterior UCEH.

No sentido de reforçar o apoio na gestão de espaços, foi criado na Universidade um Gabinete de Gestão de Informação que desenvolveu uma base de dados que facilita o apoio à elaboração manual e promove a divulgação de informação sobre planos curriculares dos cursos e horários de aulas pela Internet.

Com a excepção de alguns exames e aulas de laboratório, os exames e as aulas leccionadas no Campus de Gambelas são efectuadas num espaço lectivo comum denominado Complexo Pedagógico. A gestão deste espaço é efectuada pelo Sector Pedagógico da Assessoria do Planeamento. A elaboração manual de horários de exames segue normalmente os seguintes procedimentos:

1. Cada Unidade faz a marcação das datas e horas dos exames de acordo com a separação desejada entre exames e a disponibilidade e preferências de docentes;
2. A versão final do horário é enviada à Assessoria do Planeamento que efectua a afectação das salas disponíveis aos exames tendo em consideração as restrições de capacidade das salas;
3. Quaiquer alterações pontuais são comunicadas à Assessoria do Planeamento que efectua uma reafectação dos espaços se necessário.

Por sua vez, a elaboração manual dos horários de aulas processou-se do seguinte modo (em 1999/2000):

1. À UCEE, UCTA e UCTRA foram atribuídos determinados conjuntos de salas, de acordo com uma estimativa das respectivas necessidades;
2. As três Unidades elaboram os horários dos seus cursos de acordo com restrições de disponibilidades e preferências de docentes, separação desejada entre aulas teóricas e práticas, intervalos requeridos e capacidades das salas;
3. A versão final dos horários foi enviada à Assessoria do Planeamento, juntamente com a informação sobre os espaços comuns deixados livres;
4. Depois de todos os horários entregues, a UCEH iniciou o processo de elaboração dos seus horários tendo em conta as suas próprias restrições e aproveitando os tempos livres deixados pelas outras Unidades;

5. O Gabinete de Gestão da Informação colocou os dados dos horários na base de dados e tornou-os disponíveis na Internet.

Seguidamente, nas secções 5.2 e 5.3 serão apresentadas, respectivamente, a caracterização dos Problemas de Geração de Horários de Exames e de Aulas que servem de base para este estudo, assim como o processo de decisão adoptado para abordar estes problemas de uma perspectiva multiobjectivo. Algumas considerações computacionais serão igualmente apresentadas na secção 5.4.

5.2 Problema de Geração de Horários de Exames

5.2.1 Caracterização do Problema

A UCEH leccionava 14 cursos de licenciaturas, respectivamente, Bioquímica (BQ), Engenharia Física e Tecnológica (EFT), Engenharia de Sistemas e Computação Plano Normal (ESC) e Plano Específico BEIA / CFNCV (ESCe), Estudos Portugueses - Ramos Educacional e Científico (EP), Física e Química (FQ), Ensino de Informática, Informática (Ramo de Gestão) (IG), Línguas e Literaturas Modernas variante de Estudos Portugueses e Franceses - Ramos Educacional e Científico (LLM-EPF), Línguas e Literaturas Modernas variante de Estudos Portugueses e Ingleses - Ramo Educacional (LLM-EPI), Matemática (M), Património Cultural (PC) e Psicologia (P) e Química. A tabela 5.1 apresenta o número de exames da UCEH. O número de exames considerados na UCEH foi de 249. O número de exames considerados é inferior ao total de exames apresentados na tabela, porque alguns destes são comuns a vários cursos. Os intervalos de tempo considerados foram 30, distribuídos por dois intervalos por dia, respectivamente às dez horas e às quatorze horas e trinta minutos, durante três semanas de segunda a sexta-feira. Para todos os exames foi reservado um intervalo de três horas.

A UCEE lecciona 2 cursos de licenciatura, respectivamente Gestão de Empresas (GE) e Economia (E). A tabela 5.2 apresenta a caracterização dos dados da UCEE.

O número de exames considerados na UCEE foi de 64. Tal como no problema anterior, o número de exames é inferior ao total de exames porque alguns destes são comuns nos dois cursos. Os intervalos de tempo considerados foram 39, distribuídos por três intervalos por dia, respectivamente às nove horas e trinta minutos, às quatorze horas e trinta minutos e às dezasseis horas, durante

Cursos	Anos					Total
	1 ^o	2 ^o	3 ^o	4 ^o	5 ^o	
BQ	8	8	10	-	-	26
EFT	9	10	10	5	-	34
ESC	10	8	8	12	5	43
ESCe	9	-	-	-	-	9
EP	9	8	9	4(C) 9(E)	1(E)	30(C) 36(E)
FQ	10	10	10	10	1	41
IE	9	8	8	10	-	35
IG	8	8	9	8	-	33
LLM-EPF	8	7	7	1(C) 10(E)	1(E)	24(C) 33(E)
LLM-EPI	7	8	8	10	1	34
M	8	8	10	10	1	37
PC	14	-	-	-	-	14
P	10	-	-	-	-	10
Q	8	9	8	-	-	25

Tabela 5.1: Número de exames na UCEH

duas semanas de sábado a sábado. Devido a um processo de reestruturação das licenciaturas, algumas disciplinas ocorrem simultaneamente no mesmo curso em anos diferentes. Com exceção dos exames de informática que duravam oito horas por necessitarem de turnos, os exames ocupavam um intervalo de três horas.

Os casos considerados neste estudo apresentaram dois tipos diferentes de problemas. A UCEE apresentou um espaço de pesquisa inferior ao da UCEH. O número disponível de intervalos de tempo por exame foi relativamente superior na UCEE. A interação entre turmas na UCEE e o número de exames por curso foi também superior ao verificado na UCEH.

Embora aparentemente diferentes, estas características apresentadas pouco informam sobre a complexidade dos dois casos em estudo. Um espaço de pesquisa inferior e uma maior disponibilidade de marcação de exames poderiam indicar que a UCEE poderia ser mais fácil de resolver. Este panorama é completamente alterado com a definição das restrições. Nesta Unidade, as restrições

Cursos	N ^o exames				Total
	1 ^o	2 ^o	3 ^o	4 ^o	
GE	14	11	10	16	51
E	14	11	11	16	52

Tabela 5.2: Número de exames na UCEE

foram definidas pelos Presidente do Conselho Directivo e Representante dos Estudantes no Conselho Directivo. Seguidamente se apresentam as restrições consideradas:

1. Separação de exames do mesmo ano. O tempo mínimo de separação considerado foram 12 horas;
2. Não sobreposição de exames de anos consecutivos;
3. Preferências dos docentes responsáveis dos exames por determinados períodos de tempo. Esta Unidade, devido à especificidade do conteúdo programático dos cursos leccionados, necessita de docentes convidados que exercem outras funções exteriormente à Universidade, mas que fornecem um importante contributo na componente lectiva. Aos exames sob a responsabilidade destes docentes foram afectos a nove intervalos temporais correspondentes aos Sábados;
4. Separação máxima permitida entre determinados exames. Esta restrição ocorreu entre um par de disciplinas e o intervalo de tempo considerado foi de 5 horas;
5. Antecedência de um exame em relação a outro de um determinado período de tempo. Esta restrição ocorreu entre dois pares de disciplinas e o tempo considerado foram dez dias de antecedência;
6. Antecedência de um exame em relação a exames de determinados anos num determinado período de tempo. Nove disciplinas foram sujeitas a esta restrição por serem consideradas particularmente difíceis para os alunos pelos decisores. O tempo considerado foi três dias; Considerou-se, adicionalmente, o seguinte objectivo:
7. Manter o número de estudantes homogéneo entre intervalos de tempo. Este valor foi calculado com base na variância do número de estudantes previstos para cada intervalo de tempo considerado;

Na UCEH, a definição das restrições foi elaborada pelo Presidente do Conselho Pedagógico, depois de uma auscultação das restrições dos docentes que foram consideradas como restrições no domínio temporal dos exames. Seguidamente se apresentam as restrições consideradas:

1. Separação de exames do mesmo ano. O tempo mínimo de separação considerado foi 12 horas;
2. Não sobreposição de exames de anos consecutivos.

Como é observável, só a definição das restrições implica que o PEHE da UCEE é um problema mais exigente que o da UCEH.

5.2.2 Definição de Prioridades e Metas

De acordo com os decidores, as restrições (1) e (2) do problema da UCEE foram consideradas como restrições *hard*. Neste sentido, estas restrições foram encaradas como dois objectivos com um nível de prioridade um. As restrições de (3) a (6) foram consideradas como restrições *soft*, estando afectas a um nível de prioridade nulo. Para evitar a produção de um número excessivo de soluções finais, decidiu-se agregar estas restrições num só objectivo. O objectivo (7) foi normalizado e afecto à mesma prioridade nula. As metas dos dois primeiros objectivos foram afectas a valores nulos, visto que a admissibilidade da solução é definida pela obtenção de custo nulo nestes objectivos. Para os dois objectivos de prioridade nula foram definidas metas com base na qualidade das soluções passadas.

No problema da UCEH não foi necessário distinguir níveis de prioridades, visto que os dois objectivos têm de ser cumpridos de igual modo. As metas foram igualmente afectas a valores nulos como foi utilizado no problema da UCEE.

5.2.3 Definição da Qualidade da Solução

A qualidade da solução é, neste caso, definida pelo número de violações de cada tipo de restrições em cada exame, no caso da UCEH, e também pela minimização da variância do número de alunos presentes em cada intervalo de tempo, no caso da UCEE. Assim, os exames que estão presentes em sobreposições nas restrições dos problemas seriam assinalados com um conflito. Esta informação localizada das violações das restrições (1) e (2) da UCEH e da (1) à (6) da UCEE destina-se a vir a ser utilizada pelo operador de mutação.

5.3 Problema de Geração de Horários de Aulas

5.3.1 Caracterização do Problema

O PGHA escolhido foi a geração de horários do 1º semestre do ano lectivo 2000/2001 da FCHS. Esta nova Faculdade lecciona 5 cursos de licenciatura, respectivamente, Línguas e Literaturas Modernas variante de Estudos Portugueses e Ingleses (Ramos Educacional e Científico), Línguas e Literaturas Modernas variante de Estudos Portugueses e Franceses (Ramos Educacional e Científico), Estudos Portugueses (Ramos Educacional e Científico), Património Cultural e Psicologia. A tabela 5.3 apresenta a estrutura dos cursos da Faculdade.

Cursos	Anos				
	1	2	3	4	5
LLM-EPF	13	14	13	4(C) 12(E)	3(E)
LLM-EPI	28	33	26	6(C) 12(E)	3(E)
EP	36	28	22	5(C) 20(E)	3(E)
PC	28	22	-	-	-
P	28	15	15	-	-

Tabela 5.3: Número de aulas na FCHS

O número de aulas consideradas neste problema foi de 265. Tal como nos PGHEs anteriores, o número de aulas é inferior ao total expresso no quadro porque algumas destas ocorrem em simultâneo em várias turmas. O intervalo de tempo considerado foi de segunda-feira a sexta-feira, das oito horas e trinta minutos às dezanove horas e trinta minutos, com intervalos de trinta minutos, ou seja, 105 intervalos de tempo. Os eventos foram classificados em aulas teóricas, com duração compreendida entre uma a duas horas, aulas teórico-práticas, com duração compreendida entre uma e três horas, e aulas práticas, com duração entre uma hora e trinta minutos e três horas. 12 salas estiveram totalmente disponíveis, um anfiteatro estava disponível só de manhã, e as aulas de Desenho e Introdução à Informática só poderiam ser marcadas em salas próprias. Este perfil diferenciou-se do que na realidade foi efectuado em termos manuais, visto que neste processo se contou com 39 salas. Estas salas não estavam afectas no início da elaboração manual de horários, mas visto que não foi possível resolver o problema com o número de salas inicialmente atribuído, foi

necessário ocupar espaços livres deixados por outras faculdades. O número de docentes envolvidos na leccionação das aulas foi de 61. Para efeitos deste estudo, foram definidas as seguintes restrições:

1. Não sobreposição temporal de aulas da mesma turma;
2. Não sobreposição temporal de aulas na mesma sala;
3. Não sobreposição temporal de aulas do mesmo docente;
4. Afectação de sala no domínio temporal definido;
5. Intervalo de uma hora para almoço para as turmas, compreendido entre as 12:00 e as 14:00;
6. Intervalo de uma hora para almoço para os docentes, compreendido entre as 12:00 e as 14:00;
7. Não sobreposição de aulas teórico-práticas e práticas com aulas do ano anterior por cada curso;
8. Permanência da turma na Universidade não superior a 9 horas por dia;
9. Permanência do docente na Universidade não superior a 9 horas por dia;
10. Número de horas de aulas efectivas da turma não superior a 7 horas por dia;
11. Número de horas de aulas efectivas do docente não superior a 7 horas por dia;
12. Número de horas de aulas seguidas da turma não superior a 4 horas por dia;
13. Número de horas de aulas seguidas do docente não superior a 4 horas por dia.

Considerou-se, adicionalmente, os seguintes objectivos:

14. Minimização do número de tardes e de manhãs com aulas das turmas;
15. Minimização dos dias com aulas dos docentes;

5.3.2 Definição de Prioridades e Metas

As restrições de (1) a (6) foram consideradas neste problema como restrições *hard* e as restantes como restrições *soft*. Devido ao número elevado de objectivos, foi necessário efectuar um determinado nível de agregação. Embora seja admissível outro tipo de procedimento, decidiu-se agregar

as restrições que afectavam em comum determinados recursos. Assim, dentro das restrições consideradas como *hard*, agregou-se (1) e (5) por estarem ligadas às turmas, (2) e (4) por estarem ligadas às salas e (3) e (6) por estarem relacionadas com os docentes, totalizando três objectivos com prioridade um. Nas restrições *soft* agregou-se as restrições (7), (10) e (12) por estarem ligadas às turmas e as restrições (9), (11) e (13) por envolverem os docentes, totalizando dois objectivos com prioridades nulas. Finalmente, e visto que (14) e (15) são objectivos que não apresentam as mesmas características que os anteriores, foi decidido que cada um corresponderia a um objectivo a minimizar. As metas foram definidas com um vector nulo, visto que não há um conhecimento prévio do nível de desempenho pretendido com os objectivos.

Numa segunda formulação do problema, as restrições definidas anteriormente por *hard* foram agregadas também num só objectivo.

5.3.3 Definição da Qualidade das Soluções

A qualidade da solução é definida pelo número de violações das restrições em cada evento e pelo número de tardes e de manhãs com aulas das turmas e pelo número de dias com aulas dos docentes, representados pelos objectivos (14) e (15). Como este problema se refere a uma dupla afectação de salas e intervalos de tempos, e devido às características distintas das restrições disponíveis, foi a afectação de conflitos aos exames de modo diferente da utilizada nos PGHE. Deste modo foi considerado que as violações das restrições (1), (3) e (7) implicariam um conflito temporal por cada aula que apresentasse sobreposição temporal com outra, respectivamente, na mesma turma, no mesmo docente, e em turmas de anos consecutivos. A violação das restrições (2) e (4) implicariam, por sua vez, um conflito temporal e espacial por cada aula que estivesse marcada, respectivamente, em sobreposição com outra na mesma sala, e num intervalo de tempo considerado fora do domínio temporal da sala a que foi afecto. As violações das restrições (5) e (6) seriam assinaladas como conflitos temporais nas aulas que estivessem presentes no horário definido. As violações das restrições (8) e (9) corresponderiam a conflitos temporais nas aulas que estivessem marcadas nos primeiros e últimos (*tempo total de permanência - tempo admissível de permanência*) intervalos de tempo. As violações das restrições de (10) a (13) seriam assinaladas como conflitos temporais nas aulas que estivessem marcadas nos dias em que se verificou a violação. Finalmente, a informação dos objectivos (14) e (15) não foi disponibilizada ao operador de mutação.

Seguidamente se descreve a implementação computacional da aplicação desta metodologia.

5.4 Implementação Computacional

5.4.1 Introdução

Os dados utilizados neste estudo foram obtidos de uma base de dados existente na Universidade do Algarve, desenvolvida em PostgreSQL [76]. Algumas tabelas para representação das restrições e outras adicionais foram implementadas nesta base de dados. O código desenvolvido foi maioritariamente implementado em Python [71] e a linguagem C foi utilizada para o desenvolvimento da função de avaliação, por questões de desempenho.

5.4.2 Pré-Processamento

Definição de Domínios e Restrições

A base de dados implementada pelo Gabinete de Gestão de Informação está acessível através de uma Intranet com base em *scripts* implementados em Python e PHP, permitindo a consulta de uma grande quantidade de informação através de um *web browser*. No âmbito deste estudo foram adicionadas várias tabelas à base de dados existente para definição do domínio temporal e espacial e das restrições do problema. O domínio temporal dos exames nos PGHEs foi definido nas seguintes tabelas:

- `gr_incl`: `<exame><início><fim>`(Domínio temporal do exame `<exame>`) ;

Os domínios temporais e espaciais das várias entidades no PGHA foram definidos pelas seguintes tabelas:

- `gr_incl`: `<aula><início><fim>`(Domínio temporal da aula `<aula>`) ;
- `gr_esp`: `<aula><sala>`(Domínio espacial da aula `<aula>`);
- `sala_incl`: `<sala><início><fim>`(Domínio temporal da sala `<sala>`);
- `doc_incl`: `<docente><início><fim>`(Domínio temporal do docente `<docente>`).

Seguidamente estão representadas as tabelas implementadas para as restrições dos PGHEs de acordo com a definição das restrições na secção 5.2.1:

- **gr_dist** : <grupo><tempo>- Distância mínima de <tempo> entre exames do grupo <grupo>(restrição 1 do PGHE);
- **gr_dist_g**: <grupo_A><grupo_B><tempo>- Distância mínima de <tempo>entre exames do grupo <grupo_A>e do grupo <grupo_B>(restrição 2 do PGHE);
- **pref**: <exame><início><fim>- Definição das preferência temporais associadas ao exame <exame>entre <início>e <fim>(restrição 3 do PGHE);
- **dist_max**: <exame_A><exame_B><tempo>- Distância máxima de <tempo>entre exames <exame_A>e exame <exame_B>(restrição 4 do PGHE);
- **antes**: <exame_A><exame_B><tempo>- Antecedência no mínimo de <tempo>do exame <exame_A>do exame <exame_B>(restrição 5 do PGHE);
- **antes_gr**: <exame><grupo><tempo>- Antecedência no mínimo de <tempo>do exame <exame>de qualquer exame do grupo <grupo>(restrição 6 do PGHE);

No PGHA, e de acordo com as definições das restrições expressas na secção 5.3.1, foram definidas as seguintes tabelas:

- **gr_dist** : <grupo><tempo>- Distância mínima de <tempo> entre aulas do grupo <grupo>(restrição 1 do PGHA);
- **sala_dist**: <sala><tempo>- Distância mínima de <tempo>entre aulas na sala <sala>(restrição 2 do PGHA);
- **doc_dist**: <docente><tempo>- Distância mínima de <tempo> entre aulas do docente <docente>(restrição 3 do PGHA);
- **gr_temp_livre**: <grupo><início><fim><tempo>- Máximo de <tempo>como tempo livre por dia do grupo <grupo>entre <início>e <fim>(restrição 5 do PGHA);
- **doc_temp_livre**: <docente><início><fim><tempo>- Máximo de <tempo>como tempo livre por dia do docente <docente>entre <início>e <fim>(restrição 6 do PGHA);
- **gr_dist_g**: <grupo_A><grupo_B><tempo>- Distância mínima de <tempo>entre aulas do grupo <grupo_A>e do grupo <grupo_B>(restrição 7 do PGHA);

- `gr_perm`: `<grupo><tempo>`- Permanência máxima de `<tempo>` por dia na Universidade do grupo `<grupo>`(restrição 8 do PGHA);
- `doc_perm`: `<docente><tempo>`- Permanência máxima de `<tempo>` por dia na Universidade do docente `<docente>`(restrição 9 do PGHA);
- `gr_efect`: `<grupo><tempo>`- Máximo de `<tempo>` por dia de aulas do grupo `<grupo>`(restrição 10 do PGHA);
- `doc_efect`: `<docente><tempo>`- Máximo de `<tempo>` por dia de aulas do docente `<docente>`(restrição 11 do PGHA);
- `gr_seg`: `<grupo><tempo>`- Máximo de `<tempo>` por dia de aulas seguidas do grupo `<grupo>`(restrição 12 do PGHA);
- `doc_seg`: `<docente><tempo>`- Máximo de `<tempo>` por dia de aulas seguidas do docente `<docente>`(restrição 13 do PGHA).

A restrição (4) consiste na verificação do intervalo de tempo afecto a cada aula no domínio temporal da sala a que está afecta. Deste modo, não é uma restrição que esteja representada na base de dados, mas implícita no código. Na secção seguinte serão descritas as considerações sobre os domínios das entidades dos problemas em estudo.

Redução do Espaço de Pesquisa

A informação obtida das restrições e dos domínios é posteriormente manipulada por um módulo escrito em Python [71] que tem por objectivo efectuar o pré-processamento dos dados para o algoritmo evolutivo. Alguma redução do espaço de pesquisa é efectuada para cada tipo de problema com base nas restrições definidas na base de dados. A abordagem escolhida é semelhante às implementações mais simples utilizadas no âmbito dos métodos de Programação de Restrições [102].

No PGHE, o domínio temporal dos exames é reduzido com base na restrição definida na tabela `gr_antes`. Assumindo a existência do exame A que deve ocorrer antes do exame B num máximo de T unidades temporais, o algoritmo efectua a seguinte transformação $R_{A,B}$:

$$R_{A,B}[Dt(A)] = Dt(A) \cup \{x_{B-T}, x_{B-T+1}, \dots, x_n\}$$

$$R_{A,B}[Dt(B)] = Dt(B) \cup \{y_1, y_2, \dots, y_{A+T}\}$$

em que $Dt(A) = \{x_1, x_2, \dots, x_n\}$ representa o domínio temporal do exame A , $Dt(B) = \{y_1, y_2, \dots, y_m\}$ é o domínio temporal do exame B , x_B é o elemento de A que é igual ao último elemento de B e y_A é o elemento de B que é igual ao primeiro elemento de A . O algoritmo reinicia a redução do espaço de pesquisa cada vez que ocorre uma redução para permitir a propagação das modificações do domínio temporal de outros exames. Se um dos domínios é igual a um conjunto vazio, então o problema não tem solução.

No PGHA, o domínio temporal final de uma aula A é expressa pela intercepção do domínio temporal da aula A , com o domínio temporal do docente que lecciona a aula A e com a reunião do domínio temporal das salas atribuídas à aula A . Tal como no caso anterior, se o domínio temporal de uma aula corresponder a um intervalo vazio, o problema não tem solução.

5.4.3 Descrição da Aplicação

O Python [71] foi escolhido por ser uma linguagem que combina uma grande capacidade de declaração com a interactividade com outras aplicações, requisitos importantes para os propósitos deste estudo. É uma nova linguagem de programação interpretada que é bastante amigável ao programador devido à sua sintaxe, com vantagens comparáveis a outras linguagens interpretadas. A existência de módulos disponíveis para manipulação de listas e matrizes torna-o numa ferramenta apropriada para desenvolvimento rápido de algoritmos e heurísticas. A interacção com bases de dados e outras aplicações é simples e o facto de ser uma linguagem orientada por objectos, adiciona-lhe mais essa vantagem. A desvantagem da utilização do Python, tal como noutras linguagens de interpretadas, reside na sua velocidade de execução. Por este motivo, a função de avaliação, que demonstrou ser a fase mais demorada no conjunto de procedimentos iterativos da aplicação, foi implementada em C, com o recurso à *interface* de programação Python/C interligação de variáveis entre as duas linguagens.

O algoritmo foi desenhado segundo os princípios inerentes a qualquer algoritmo evolutivo com população de soluções, ou seja, uma fase de inicialização e um conjunto iterativo de procedimentos como a avaliação, classificação e selecção das soluções, aplicação de operadores de mutação e subs-

tuição da população. A aplicação foi implementada em vários módulos que estão seguidamente descritos:

`pea_data`, *interface* com a base de dados e pré-processamento dos dados;

`pea_eval_C`, declara a função de avaliação em C;

`pea_genome`, responsável pela afectação inicial de valores ao cromossoma, avaliação e conversão do cromossoma em variáveis de decisão;

`pea_individual`, declara as funções necessárias para instanciação do objecto indivíduo, geração e avaliação do indivíduo;

`pea_op_muteCI`, implementação do operador de mutação independente de acordo com as definições na secção 4.5;

`pea_op_muteCR`, implementação do operador de mutação de uma posição de acordo com as definições na secção 4.5;

`pea_op_reGen`, implementação da reinserção geracional de acordo com as definições na secção 4.7;

`pea_op_selSUS`, implementação do método de selecção *Stochastic Universal Sampling* de acordo com as definições na secção 4.7;

`pea_rankGA`, seriação da população para o algoritmo evolutivo uniobjectivo;

`pea_rankMOGA`, seriação da população para o algoritmo evolutivo multiobjectivo de acordo com as definições na secção 4.4;

`pea_run`, módulo de configuração dos parâmetros e métodos do algoritmo;

`pea_showGA`, módulo de visualização e armazenamento de dados do desempenho do algoritmo evolutivo;

`pea_showMOGA`, módulo de visualização e armazenamento de dados do desempenho do algoritmo multiobjectivo;

`peas`, declara as funções para inicialização da população e os vários procedimentos dentro de cada iteração;

peas_main, inicializa as iterações e define o critério de paragem;

O algoritmo produz vários resultados, respectivamente, um ficheiro com o prefixo *time* que descreve o horário produzido pelo algoritmo, um ficheiro com o prefixo *run_best* que guarda, em cada iteração, a soma dos valores das funções objectivo de acordo com as prioridades definidas, um ficheiro com o prefixo *run_pop* que guarda a iteração e os valores das funções objectivo das melhores soluções encontradas até então, e um ficheiro com o prefixo *run_sol* que guarda os valores das funções objectivo das melhores soluções encontradas na última iteração. Estes ficheiros constituem os dados necessários para a realização de estatísticas através de aplicações implementadas em Python e C e de acordo com as considerações definidas no Capítulo seguinte.

5.4.4 Tempo de Execução

O tempo de execução dos algoritmos propostos para a resolução dos PGHE num só processador de computador Dual-Pentium III com 128 MB de RAM e 5000 gerações, variou entre 1 hora e trinta minutos, no caso da aplicação de mutação de uma posição e método de selecção uniobjectivo no PGHE da UCEE, e oito horas e trinta minutos, no caso da aplicação de mutação independente e método de selecção multiobjectivo no PGHE da UCEH. Estes valores estão dentro do aceitável para ter uma, ou várias, marcações alternativas de exames dentro de tempo útil, visto que o processo manual demora alguns dias.

O algoritmo para resolução do PGHA demora aproximadamente, e em média, trinta e seis horas para 10000 gerações. As dimensões do problema e o número de objectivos justificam este tempo. Contudo, fica abaixo do número de dias que são necessários no processo manual.

É de salientar que o método multiobjectivo é mais demorado que o método de selecção uniobjectivo. Contudo, a ordem de grandeza depende do número de soluções não-dominadas existentes na população. Quanto maior o número de soluções não-dominadas, mais tempo demora o algoritmo a seriar os indivíduos. Contudo, este factor é compensado pelo maior número de horários alternativos gerados. Em termos de operadores de mutação, a mutação de uma posição é mais rápida que a mutação independente e a diferença temporal aproxima-se de um segundo com uma população de 100 indivíduos.

Estes tempos registados podem ser bastante mais reduzidos se o código for convertidos para linguagem C. A paralelização dos algoritmos evolutivos também pode reduzir o tempo computaci-

onal. Admitindo que o algoritmo seja executado num dos processadores disponíveis, cada geração pode simplesmente ser avaliada noutros processadores, visto que a componente de avaliação é a componente mais demorada nestas abordagens a estas classes de problemas. Outros modelos baseados em modelo de ilhas [60] e difusão [53], como referidos na secção 3.7 podem ser utilizados para reduzir o tempo computacional.

No próximo Capítulo serão apresentados e discutidos os resultados experimentais utilizados neste estudo

Capítulo 6

Resultados Experimentais

6.1 Introdução

Como é evidenciado por Hooker [64], o modo de comparar algoritmos pode ser efectuado através do desenho de experiências controladas para evidenciar se a presença ou ausência de certas características do algoritmo podem contribuir para o bom ou mau desempenho deste. O tratamento estatístico neste estudo assume a utilização da metodologia experimental para obter alguma compreensão sobre os efeitos dos métodos e parâmetros descritos no Capítulo 4 no comportamento global do algoritmo. Este tipo de comparação é efectuada com base na aplicação repetida de execuções independentes dos algoritmos propostos.

O algoritmo evolutivo é na realidade um algoritmo de natureza estocástica que utiliza componentes aleatórias na sua execução. A avaliação do desempenho dos algoritmos estocásticos pode ser efectuado em dois cenários [49]. A análise pode basear-se no custo das soluções produzidas num determinado tempo de execução ou no tempo necessário para produzir uma solução até um determinado custo. Dado que nesta classe de algoritmos o tempo de execução e o custo das soluções são aleatórios, a avaliação do desempenho dos algoritmos reduz-se à análise das distribuições correspondentes.

Contudo, deve haver um cuidado especial na análise das distribuições. Como observado por Hoos e Stutzle [65], o cálculo da média sobre amostras aleatórias de instâncias distintas fornece uma medida imprecisa do comportamento do algoritmo. Em relação ao primeiro cenário, a análise das funções de distribuição empíricas dos resultados finais pode ser considerada como uma medida

eficiente para fornecer alguma luz sobre a qualidade das soluções obtidas. As mesmas considerações podem ser utilizadas para o segundo cenário. Em [65] é definida uma metodologia estatística para análise de algoritmos estocásticos com base na análise das distribuições do tempo de execução de instâncias individuais de problemas. Distribuições estatísticas do tempo de execução podem ser identificadas, tornando possível a utilização de testes estatísticos para determinar diferenças no desempenho dos algoritmos.

Neste estudo ambos cenários são considerados simultaneamente na análise do comportamento dos algoritmos propostos. Ao contrário do descrito em [65], a avaliação do desempenho de optimizadores é aqui considerada também como um problema multiobjectivo, em que tanto o custo da solução como o próprio tempo de execução constituem, na realidade, objectivos distintos. Esta abordagem foi inicialmente sugerida em [49] de modo a permitir estudar o comportamento de qualquer algoritmo estocástico, multiobjectivo ou não, sob uma plataforma comum.

Neste Capítulo são apresentados os resultados obtidos nos PGHEs e PGHA descritos anteriormente. A avaliação do desempenho é discutida na secção 6.2 e a metodologia experimental é apresentada na secção 6.3. Os resultados obtidos para cada problema são apresentados na secção 6.4 e discutidos na secção 6.5. Finalmente, são apresentadas algumas conclusões finais na secção 6.6.

6.2 Avaliação de Desempenho de Algoritmos Evolutivos Multiobjectivo

6.2.1 Funções de Aproveitamento

Numa perspectiva multiobjectivo, e de acordo com a definição de meta enquanto valor desejável a atingir, um algoritmo atinge uma determinada meta se produzir pelo menos uma solução igual ou melhor que o ponto do espaço dos objectivos representado por essa meta. Fonseca e Fleming [46] interpretam o desempenho de um algoritmo estocástico multiobjectivo pela probabilidade do algoritmo atingir tal meta numa única execução. O desempenho do algoritmo pode, então ser entendido como uma função dessa meta, ou seja, a chamada função de aproveitamento.

A função de aproveitamento é definida pela função $\gamma_{\chi}(\cdot) : R^d \rightarrow [0, 1]$ em que

$$\begin{aligned}\gamma_{\chi}(z) &= P(X_1 \leq z \vee X_2 \leq z \vee \dots \vee X_M \leq z) \\ &= P(\chi \preceq z).\end{aligned}$$

em que o conjunto $\chi = \{X_j \in R^d, j = 1, \dots, M\}$ representa um conjunto de pontos aleatórios, em que cada elemento X_j é não-dominado relativamente a todas as restantes, e M é um número aleatório [49]. A expressão $\gamma_{\chi}(z)$ corresponde à probabilidade do algoritmo encontrar pelo menos uma solução que atinge um vector-meta z numa execução única. Se o algoritmo multiobjectivo produzir um só vector objectivo por execução, então $M = 1$ e a função de aproveitamento reduz-se à função de distribuição multivariada.

Esta probabilidade pode ser estimada directamente a partir dos dados produzidos por várias execuções, de modo análogo à estimação de funções de distribuição multivariadas, servindo de base para comparação entre desempenho de algoritmos. Recentemente [49] foi estabelecida uma ligação entre a teoria desenvolvida para comparação de optimizadores multiobjectivo e a teoria dos conjuntos fechados aleatórios, que demonstra que a função de aproveitamento é, num dado sentido, uma medida de média para os resultados de um optimizador multiobjectivo.

As funções de aproveitamento empíricas assim obtidas a partir dos resultados dos algoritmos multiobjectivo, quer estes produzam um ou mais vectores de pontos por execução, podem ser comparadas através de procedimentos de testes estatísticos não-paramétricos baseados em testes de permutações [93]. Inclusive, a evolução da qualidade das soluções pode ser avaliada considerando o tempo como um objectivo a minimizar.

O cálculo das funções de aproveitamento para vários objectivos não é um problema trivial. Os resultados computacionais aqui apresentados foram obtidos usando código ainda em desenvolvimento por Fonseca. Uma breve discussão das questões computacionais envolvidas podem ser encontradas em [49].

6.2.2 Testes Estatísticos

A natureza estocástica das soluções obtidas por dois algoritmos evolutivos multiobjectivo A e B permite a formalização de testes de hipóteses. Dado que as várias execuções dos algoritmos são independentes, este teste pode ser formalizado como um problema de duas amostras independentes. As hipóteses podem ser enunciadas do seguinte modo:

H_1 : Não existe diferença de desempenho entre A e B em termos de funções de aproveitamento para um determinado problema de optimização, *versus*

K_1 : Existe diferença de desempenho entre A e B em termos de funções de aproveitamento para um determinado problema de optimização,

em que H_1 e K_1 são, respectivamente, a hipótese nula e hipótese alternativa.

Dada a independência entre as execuções de ambos algoritmos, uma estatística de teste T adequada para testar as hipóteses formalizadas é a máxima distância absoluta entre as funções de aproveitamento empíricas, em analogia ao teste de Smirnov [25]. Sendo $S_1(x)$ uma função de aproveitamento empírica baseada na amostra $\chi_{11}, \chi_{12}, \dots, \chi_{1n}$ e $S_2(x)$ uma função de aproveitamento empírica baseada noutra amostra $\chi_{21}, \chi_{22}, \dots, \chi_{2n}$, a estatística de teste considerada consiste na maior diferença absoluta entre as funções de aproveitamento empíricas, ou seja,

$$T = \sup_x |S_1(x) - S_2(x)|$$

Para um dado nível de significância α , escolhido à *priori*, e normalmente compreendido entre 0,01 e 0,05, a hipótese nula deve/pode ser rejeitada se $T \geq c_{1-\alpha}$ [69]. O valor crítico $c_{1-\alpha}$ corresponde ao $(1 - \alpha)$ -quantil da distribuição de T nas condições da hipótese nula.

Quando se pretender comparar mais do que dois algoritmos, é necessário abordar o teste como um problema de k amostras independentes. Neste caso, a máxima distância absoluta entre todas as combinações de pares de funções de aproveitamento empíricas pode ser utilizada como estatística de teste, por analogia com o teste Birnbaum-Hall [25]. Sendo $S_k(x)$ k funções de aproveitamento empíricas baseadas nas amostras $\chi_{k1}, \chi_{k2}, \dots, \chi_{kn}$, a estatística de teste considerada utiliza a maior diferença absoluta entre $S_i(x)$ e $S_j(x)$,

$$T = \sup_{x,i,j} |S_i(x) - S_j(x)|$$

As mesmas regras de rejeição referidas na estatística de teste anterior são aplicadas neste teste. Contudo, a rejeição da hipótese nula significa, agora, que as distribuições de amostras terão funções de aproveitamento diferentes.

Embora, mesmo nas condições da hipótese nula, a distribuição da estatística de teste T seja dependente da distribuição da função de aproveitamento que descreve os dados e, por isso, desconhecida, o cálculo da distribuição exacta de T pode ser efectuada através de testes de permutações

[61]. Como é possível admitir que um dado conjunto de soluções tanto pode ser proveniente de um algoritmo como de outro, ou seja, admitir que os conjuntos de soluções são permutáveis, as hipóteses formalizadas cumprem os requisitos necessários para a abordagem do teste de permutações [61]. O teste apresenta os seguintes procedimentos:

1. Rotular as observações;
2. Calcular T dados os rótulos originais das observações;
3. Permutar as rótulos e recalculer T para os rótulos permutados. Repetir até se obter a distribuição de T para todas as permutações possíveis;
4. Aceitar ou rejeitar a hipótese com base na distribuição obtida.

A decisão do procedimento de teste pode ser efectuada através da comparação do valor de T baseado nas duas amostras originais com $c_{1-\alpha}$. Do mesmo modo, o valor de p que indica a probabilidade de obter um valor da estatística de teste tão extremo como o observado, nas condições da hipótese nula, pode ser determinado. Nesse caso, rejeita-se a hipótese nula se $p < \alpha$. Quando o número de permutações é excessivamente grande para ser enumerado, é possível recorrer a procedimentos Monte Carlo para obter uma distribuição aproximada [61].

O resultado da estatística de teste num conjunto de duas ou mais amostras pode ser associado a pontos ou regiões do espaço dos objectivos onde a distância excede o valor crítico da estatística de teste [46], permitindo a identificação das regiões no espaço dos objectivos onde existem diferenças estatísticas relevantes. O sinal das diferenças cujo o valor absoluto excede esse valor crítico pode, inclusive, ser indicativo do algoritmo que apresentou melhor desempenho nessa região.

6.3 Metodologia Experimental

A metodologia proposta na secção anterior tem por objectivo comparar os resultados obtidos por várias combinações dos métodos e parâmetros descritos no Capítulo 4, através do valor mínimo da agregação dos objectivos e de funções de aproveitamento.

Este estudo considera três factores relevantes, respectivamente, operador de selecção, tipo de mutação e nível de parâmetro β . O operador de selecção apresenta duas alternativas: o método de selecção multiobjectivo, em que cada objectivo é tratado separadamente, e o método de selecção

uniobjectivo que trata os objectivos de uma forma agregada. É de salientar que um algoritmo que utilize o método de selecção uniobjectivo também produzirá, em geral, diversos pontos não-dominados, visto que só a seriação dos indivíduos é que tem em conta a agregação dos objectivos. Duas alternativas são também consideradas para o operador de mutação, respectivamente, a mutação independente e mutação de uma posição. Por fim, cinco níveis são considerados para o parâmetro β , respectivamente, 0, 1, L , $2L$ e ∞^1 em que L representa o comprimento do cromossoma.

Um conjunto de execuções é determinado por cada combinação possível dos três factores apresentados. Neste estudo será necessário contar com vinte conjuntos de execuções, em que os resultados para cada execução são armazenados. A análise estatística poderá verificar se algum dos factores apresentou algum efeito significativo no desempenho do algoritmo, assumindo que os restantes factores são mantidos constantes a qualquer nível. Também seria possível, através de uma estatística adequada, verificar as interacções entre os factores. Contudo, este estudo considerou unicamente a análise dos efeitos principais de cada factor presente do problema. Um estudo futuro deverá alargar a metodologia para a análise das interacções entre factores.

A comparação é efectuada em termos da análise de custo solução em tempo fixo para ambos os PGHEs e do tempo de execução/custo da solução no PGHE da UCEH. A análise de tempo de execução/custo da solução não foi realizada para o PGHE da UCEE, porque não foi possível processar o volume de dados gerados para este problema em tempo útil, devido à necessidade de calcular a função de aproveitamento empírica em quatro dimensões.

Tal como sugerido na secção anterior, o teste de permutações é efectuada para comparar o desempenho dos algoritmos. Contudo, para reduzir a variância, o desenho de blocos foi implementado, tal como proposto em [61]. No passo 3 do procedimento de permutações, os elementos do conjunto de soluções são permutados entre os níveis diferentes do factor que se pretende investigar, mas dentro dos mesmos níveis dos outros factores que possam existir, reduzindo a variância da estatística de teste, e simultaneamente o número de permutações necessárias.

Outro aspecto a salientar é a necessidade de detectar a origem de diferenças no caso da rejeição da hipótese formalizada para testar os vários parâmetros β . O método de testes fechados através de múltiplas comparações, como descrito por Hsu [67], permite detectar essas diferenças, controlando

¹ $\beta = \infty$ significa que $\Omega_i = \sum_{j \in R_i} c_{ij} (a_{ij} - 1)$ de acordo com a secção 4.5

a probabilidade de erro máximo Tipo I, que corresponde à máxima probabilidade de uma ou mais hipóteses serem rejeitadas incorrectamente. Sendo $\{H_i, i = 1, \dots, L\}$ uma família de hipóteses, o seu *fecho* é obtido pelas conjunções não-vazias $H_V = \bigcap_{i \in V} H_i$ em que $V \subseteq \{1, \dots, L\}$. O método de testes fechados considera a rejeição de H_V sse cada H_U em que $U \supseteq V$ é rejeitado pelo nível de α considerado. O autor realça que o teste sobre as conjunções do género $H_{ij,kl}$ ($H'_{ij} : X_i = X_j$ e $H'_{kl} : X_k = X_l$) seja efectuado a um nível de significância a $1 - (1 - \alpha)^{1/2}$. O teste também podem ser representado pelo valor de p em que a hipótese nula H_i é rejeitada se o valor máximo de máximo p onde estiver presente essa hipótese for rejeitada a α [106]. Nas conjunções $H_{ij,kl}$, o valor de p é corrigido para $1 - (1 - \min(p(H'_{ij}), p(H'_{kl})))^2$ e testado a nível α .

6.4 Resultados

6.4.1 PGHE da UCEH - Análise de Custo de Solução em Tempo Fixo

Configuração das Execuções

Os algoritmos evolutivos propostos utilizaram uma população de 200 indivíduos para o caso da UCEH. A pressão selectiva foi 2,0 e a probabilidade de sobrevivência foi 0,5, de acordo com as especificações na secção 4.5. O operador de selecção foi o *Stochastic Universal Sampling* e a reinserção populacional geracional foi utilizada para manter a pressão selectiva. O critério de paragem foi 5000 gerações. De acordo com a secção anterior, 10 execuções foram realizadas para cada combinação de operador de selecção, tipo de mutação e nível de parâmetro β , totalizando 200 execuções.

Neste problema, cada um dos algoritmos registou os valores mínimos da agregação dos objectivos e os vectores de valores objectivo de pontos não-dominados acumulados até à última geração. Todas as restrições no problema da UCEH foram agregadas no método de selecção uniobjectivo. Na selecção multiobjectivo, o processo de decisão expresso na secção 5.2.2 foi aplicado.

Definição de Hipóteses

Para comparação entre métodos de selecção e tipos de mutação, segundo a análise de custo de solução em tempo fixo, o teste é formalizado como um problema de duas amostras independentes. As hipóteses estão seguidamente definidas:

H_1 : Não existe diferença de desempenho em termos de valor mínimo da agregação de objectivos

entre os dois métodos de selecção, *versus*

K_1 : Existe diferença de desempenho em termos de valor mínimo da agregação de objectivos entre os dois métodos de selecção,

em que H_1 e K_1 são, respectivamente, a hipótese nula e hipótese alternativa;

H_2 : Não existe diferença de desempenho em termos de funções de aproveitamento entre os dois métodos de selecção, *versus*

K_2 : Existe diferença de desempenho em termos de funções de aproveitamento entre os dois métodos de selecção,

em que H_2 e K_2 são, respectivamente, a hipótese nula e hipótese alternativa;

H_3 : Não existe diferença de desempenho em termos de valor mínimo da agregação de objectivos entre os dois tipos de mutação, *versus*

K_3 : Existe diferença de desempenho em termos de valor mínimo da agregação de objectivos entre os dois tipos de mutação,

em que H_3 e K_3 são, respectivamente, a hipótese nula e hipótese alternativa;

H_4 : Não existe diferença de desempenho em termos de funções de aproveitamento entre os dois tipos de mutação, *versus*

K_4 : Existe diferença de desempenho em termos de funções de aproveitamento entre os dois tipos de mutação,

em que H_4 e K_4 são, respectivamente, a hipótese nula e hipótese alternativa;

A comparação entre os vários níveis do parâmetro β do operador de mutação pode ser formalizado como um problema de k amostras independentes. As hipóteses definidas são as seguintes:

H_5 : Não existe diferença de desempenho em termos de valor mínimo da agregação de objectivos entre os cinco níveis do parâmetro β , *versus*

K_5 : Pelo menos dois dos níveis de β apresentam diferença de desempenho em termos de valor mínimo da agregação de objectivos,

em que H_5 e K_5 são, respectivamente, a hipótese nula e hipótese alternativa;

H_6 : Não existe diferença de desempenho em termos de funções de aproveitamento entre os cinco valores de β do operador de mutação, *versus*

K_6 : Pelo menos dois dos níveis de β apresentam diferença de desempenho em termos de funções de aproveitamento,

em que H_6 e K_6 são, respectivamente, a hipótese nula e hipótese alternativa.

Comparação entre Métodos de Selecção

No PGHE da UCEH, a menor soma dos valores objectivo obtida foi 4. Verificaram-se três ocorrências no método de selecção multiobjectivo e uma no método uniobjectivo. A maior soma foi 34, obtidas pela versão multiobjectivo. A seguinte tabela apresenta os valores de p obtidos através de 10000 execuções Monte Carlo para PGHE da UCEH na comparação do dois métodos de selecção.

	valor de p
Agregação de Objectivos	0,0080
Funções de Aproveitamento	0,0000

Tabela 6.1: Valores de p para comparação de métodos de selecção na UCEH na análise do custo de solução em tempo fixo

Em termos de soma mínima da agregação de objectivos, foi encontrada uma diferença significativa entre os métodos de selecção, indicando a rejeição de H_1 . Uma análise do sinal das diferenças indica que o método de selecção multiobjectivo obteve melhor desempenho. Em termos de funções de aproveitamento, o valor de p implica a rejeição de H_2 . O sinal das diferenças indica que a versão multiobjectivo apresentou igualmente melhor desempenho.

Comparação entre Tipos de Mutação

A menor soma de conflitos foi obtida tanto pela versão de mutação independente com uma ocorrência, como pela versão de mutação de uma posição com três ocorrências. A maior soma foi obtida pela versão de mutação independente com uma ocorrência.

A tabela 6.2 apresenta os valores de p obtidas através de 10000 execuções Monte Carlo da UCEH na comparação do tipo de mutação.

	valor de p
Agregação de Objectivos	0,7238
Funções de Aproveitamento	0,4478

Tabela 6.2: Valores de p para comparação de tipos de mutação na UCEH na análise do custo de solução em tempo fixo

Como é possível observar, H_3 e H_4 não são rejeitados com $\alpha = 0,05$.

Comparação do Parâmetro β de Mutação

A menor soma dos valores de conflitos foi obtida com $\beta = L$ e $\beta = 2L$ com duas ocorrências em cada. A maior soma foi obtida com $\beta = 0$. A tabela 6.3 apresenta os valores de p obtidos através de 10000 execuções Monte Carlo da UCEH na comparação dos parâmetros β .

	valor de p
Agregação de Objectivos	0,0000
Funções de Aproveitamento	0,0000

Tabela 6.3: Valores de p para comparação de parâmetros β na UCEH na análise do custo de solução em tempo fixo

Tanto H_5 como H_6 são claramente rejeitadas em ambos os problemas com $\alpha = 0,05$. Deste modo é necessário detectar as diferenças, de acordo com os procedimentos descritos na secção anterior. A figura 6.1 apresenta um esquema das comparações efectuadas e dos valores de p obtidos para testar a hipótese H_5 no problema da UCEH. As zonas em cinzento indicam a aceitação da hipótese nula com $\alpha = 0,05$.

É possível observar que existem diferenças significativas entre $\beta = 0$, $\beta = 1$ e $\beta = \{L, 2L, \infty\}$ com $\alpha = 0,05$. Os sinais obtidos das diferenças indicam que este último conjunto apresenta o melhor desempenho.

A figura 6.2 apresenta um esquema dos valores de p obtidos para testar a hipótese H_6 no problema da UCEH.

Os resultados obtidos são semelhantes aos dados da figura 6.1, ou seja, $\beta = 0$, $\beta = 1$ e $\beta = \{L, 2L, \infty\}$ apresentam diferenças estatísticas com $\alpha = 0,05$.

6.4.2 PGHE da UCEE - Análise de Custo de Solução em Tempo Fixo

Configuração das Execuções

Os algoritmos evolutivos propostos para a resolução do PGHE da UCEE utilizaram uma população de 100 indivíduos. Os restantes parâmetros foram definidos de igual modo à configuração das execuções do problema anterior.

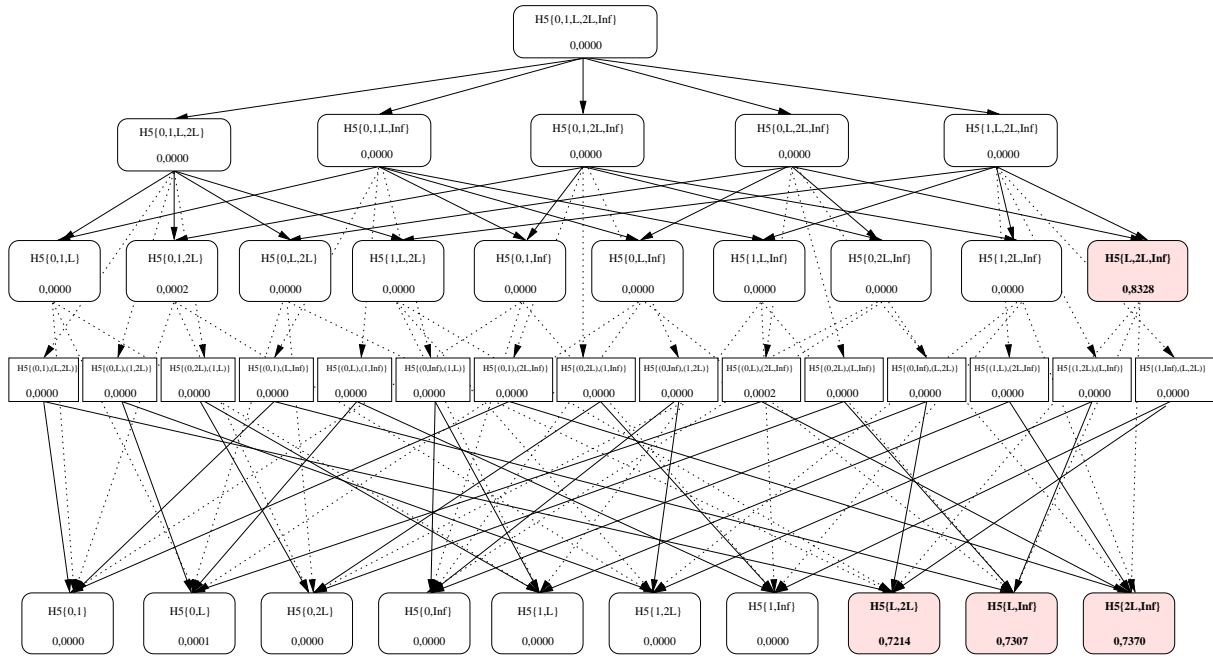


Figura 6.1: Valores de p para comparações múltiplas de parâmetros β em termos de agregação de objectivos na UCEH na análise do custo de solução em tempo fixo

Cada um dos algoritmos produziu, igualmente, os valores mínimos da agregação dos objectivos e os vectores de valores objectivo dos pontos não-dominados na última geração. Todas as restrições neste problema foram agregadas num só objectivo no método de selecção uniobjectivo. Devido à existência de diferentes prioridades entre objectivos na versão uniobjectivo, a melhor solução é escolhida por ordem lexicográfica. Deste modo, a melhor solução é aquela que satisfaz em primeiro lugar os objectivos de maior prioridade. Se existir mais do que uma solução com o mesmo valor no objectivo com prioridade elevada, a solução que apresentar a soma mínima dos valores do objectivo com prioridade mais reduzida é a escolhida. No método de selecção multiobjectivo, o objectivo associado à restrição (7) foi normalizado e não foi utilizado pelo operador de mutação. O processo de decisão expresso na secção 5.2.2 foi utilizado.

Definição de Hipóteses

Para comparação entre métodos de selecção, entre tipos de mutação e parâmetros de β , segundo a análise de custo de solução em tempo fixo, a definição de hipóteses e do teste estatístico é semelhante

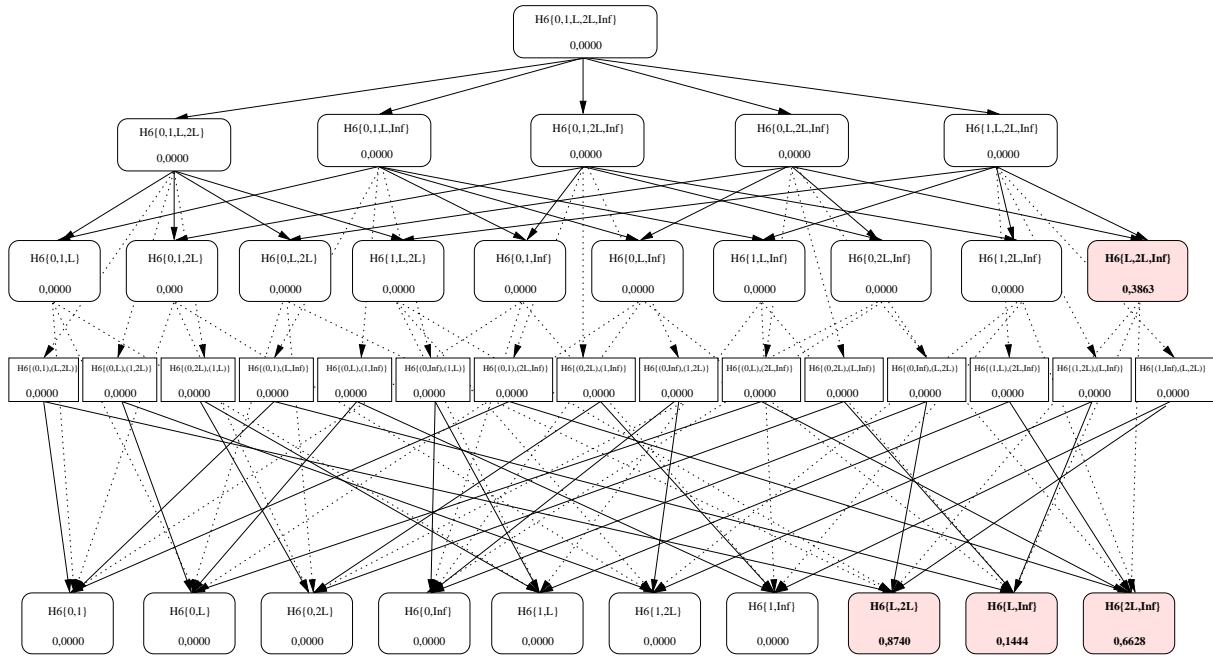


Figura 6.2: Valores de p para comparações múltiplas de parâmetros β em termos de funções de aproveitamento na UCEH na análise do custo de solução em tempo fixo

ao anterior para o PGHE da UCEH.

Comparação entre Métodos de Selecção

A menor soma de valores objectivo foi obtida na selecção uniobjectivo com 16 conflitos nos objectivos de elevada prioridade e 38,001614 nos objectivos de menor prioridade. A maior soma foi obtida também pela selecção uniobjectivo com 22 conflitos nos objectivos de elevada prioridade e 104,00183 nos objectivos de menor prioridade. A tabela 6.4 apresenta os valores de p obtidos através de 10000 execuções Monte Carlo na comparação dos dois métodos de selecção.

	valor de p
Agregação de Objectivos	0,1085
Funções de Aproveitamento	0,0000

Tabela 6.4: Valores de p para comparação de métodos de selecção na UCEE na análise do custo de solução em tempo fixo

Os testes estatísticos em termos de valor mínimo da agregação dos objectivos indicam que a diferença entre métodos de selecção não é significativa com $\alpha = 0,05$. Em termos de funções de aproveitamento, o valor de p implicou a rejeição de H_2 . O sinal das diferenças indica que a versão multiobjectivo apresentou melhor desempenho.

Comparação entre Tipos de Mutação

A menor soma dos valores objectivo foi obtida tanto pela versão de mutação independente, com uma ocorrência, como pela versão de mutação de uma posição com três ocorrências. A maior soma foi obtida pela versão de mutação independente com uma ocorrência.

A tabela 6.5 apresenta os valores de p obtidos através de 10000 execuções Monte Carlo na comparação do tipo de mutação.

	valor de p
Agregação de Objectivos	0,6778
Funções de Aproveitamento	0,0000

Tabela 6.5: Valores de p para comparação de tipos de mutação na UCEE na análise do custo de solução em tempo fixo

Como se observa, é possível rejeitar H_4 com $\alpha = 0,05$ em termos de funções de aproveitamento. O sinal indica que a versão independente forneceu um melhor desempenho.

Comparação do Parâmetro β de Mutação

Neste problema, a menor soma foi verificada com $\beta = \infty$ e a maior com $\beta = 1$. A tabela 6.6 apresenta os valores de p obtidos através de 10000 execuções Monte Carlo na comparação dos parâmetros β .

As hipóteses H_5 e H_6 são rejeitadas em ambos com $\alpha = 0,05$. Deste modo é necessário detectar as diferenças, de acordo com os procedimentos descritos na secção anterior. A figura 6.3 apresenta um esquema das comparações efectuadas e dos valores de p obtidos para testar a hipótese H_5 no problema da UCEE

É possível observar que existem diferenças significativas entre $\beta = 0$ e $\beta = \{1, L, 2L, \infty\}$ com $\alpha = 0,05$. Os sinais obtidos das diferenças indicam que este último conjunto apresenta melhor

	valor de p
Agregação de Objectivos	0.0001
Funções de Aproveitamento	0.0000

Tabela 6.6: Valores de p para comparação de parâmetros β na UCEE na análise do custo de solução em tempo fixo

desempenho.

A figura 6.4 apresenta um esquema dos valores de p obtidos para testar a hipótese H_6 no problema da UCEE em termos de funções de aproveitamento.

Os resultados obtidos são semelhantes aos dados da figura 6.3, ou seja, $\beta = 0$ e $\beta = \{1, L, 2L, \infty\}$ apresentam diferenças estatísticas com $\alpha = 0,05$.

6.4.3 PGHE da UCEH - Análise de Custo de Solução / Tempo de Execução

Configuração das Execuções

Os algoritmos executados, e referidos na secção 6.4.1, registaram também o valor mínimo da agregação dos objectivos e os vectores de valores objectivo dos pontos não-dominados durante a execução. Os valores aqui apresentados foram obtidos com os mesmos parâmetros e com a mesma definição de objectivos. O tempo de execução considerado foi o número de gerações.

Visto que neste tipo de análise é acrescentado o tempo aos objectivos existentes, é sempre necessário analisar os vectores de pontos não-dominados obtidos durante a execução através de funções de aproveitamento. Por exemplo, os dados produzidos por um algoritmo uniobjectivo durante a execução corresponde, na realidade, a dois objectivos, dos quais um é o tempo.

Definição de Hipóteses

Para comparação entre métodos de selecção e tipos de mutação, segundo a análise de custo de solução / tempo de execução, o teste é formalizado como um problema de duas amostras independentes. As hipóteses estão seguidamente formalizadas:

H_1 : Não existe diferença de desempenho em termos de funções de aproveitamento do valor mínimo da agregação de objectivos *versus* tempo de execução entre os dois métodos de selecção, *versus*

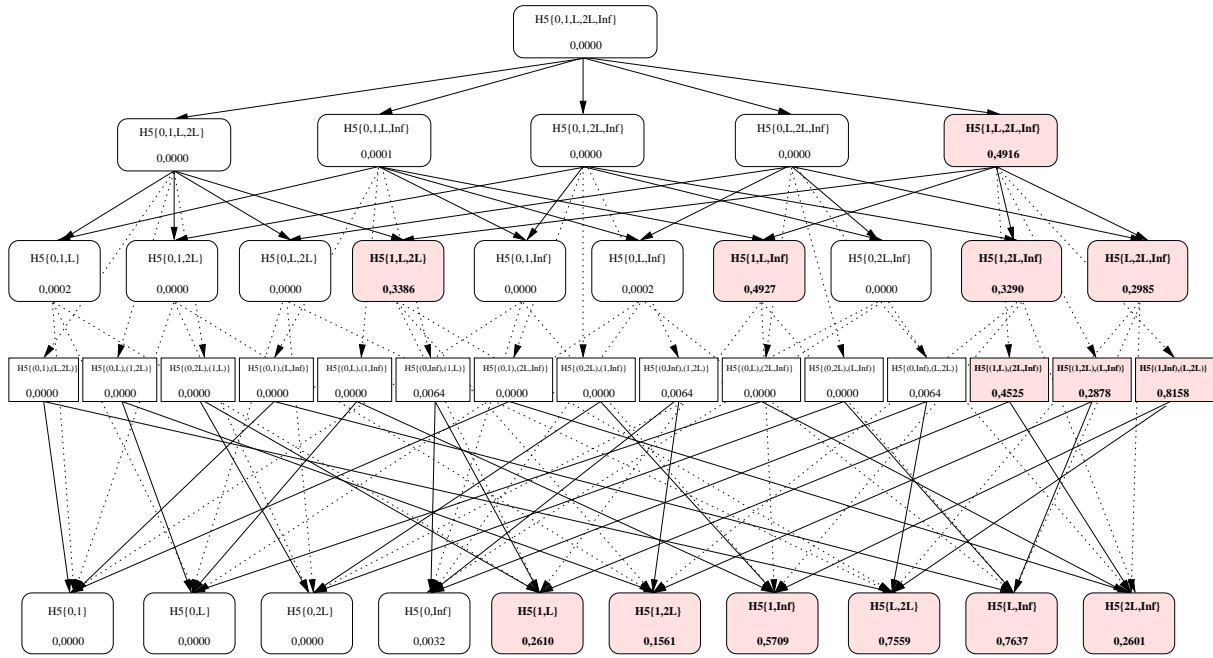


Figura 6.3: Valores de p para comparações múltiplas de parâmetros β em termos de agregação de objectivos na UCEE na análise do custo de solução em tempo fixo

K_1 : Existe diferença de desempenho em termos de funções de aproveitamento do valor mínimo da agregação de objectivos *versus* tempo de execução entre os dois métodos de selecção,

em que H_1 e K_1 são, respectivamente, a hipótese nula e hipótese alternativa;

H_2 : Não existe diferença de desempenho em termos de funções de aproveitamento dos vectores de valores objectivo não-dominados do custo de solução *versus* tempo de execução entre os dois métodos de selecção, *versus*

K_2 : Existe diferença de desempenho em termos de funções de aproveitamento dos vectores de valores objectivo não-dominados do custo de solução *versus* tempo de execução entre os dois métodos de selecção,

em que H_2 e K_2 são, respectivamente, a hipótese nula e hipótese alternativa;

H_3 : Não existe diferença de desempenho em termos de funções de aproveitamento do valor mínimo da agregação de objectivos *versus* tempo de execução entre os dois tipos de mutação, *versus*

K_3 : Existe diferença de desempenho em termos de funções de aproveitamento do valor mínimo

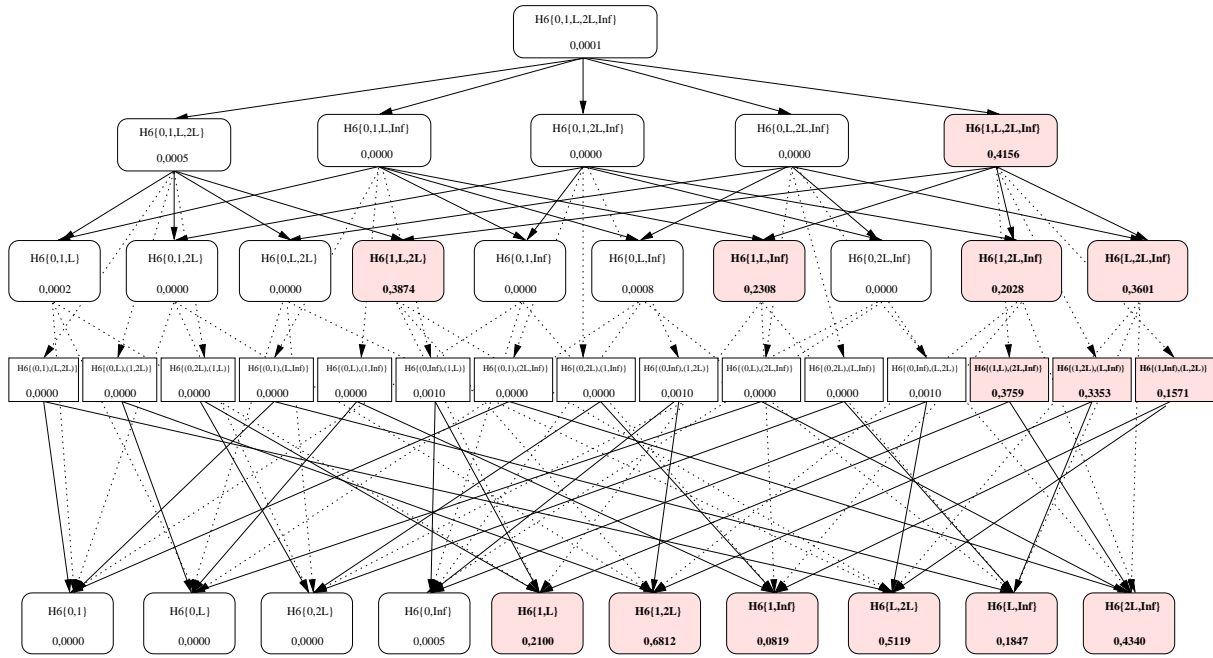


Figura 6.4: Valores de p para comparações múltiplas de parâmetros β em termos de funções de aproveitamento na UCEE na análise do custo de solução em tempo fixo

da agregação de objectivos *versus* tempo de execução entre os dois tipos de mutação,

em que H_3 e K_3 são, respectivamente, a hipótese nula e hipótese alternativa;

H_4 : Não existe diferença de desempenho em termos de funções de aproveitamento dos vectores de valores objectivo não-dominados do custo de solução *versus* tempo de execução entre os dois tipos de mutação, *versus*

K_4 : Existe diferença de desempenho em termos de funções de aproveitamento dos vectores de valores objectivo não-dominados do custo de solução *versus* tempo de execução entre os dois tipos de mutação,

em que H_4 e K_4 são, respectivamente, a hipótese nula e hipótese alternativa;

A comparação entre os vários níveis do parâmetro β do operador de mutação só foi formulada em termos de funções de aproveitamento dos valores mínimos da agregação de objectivos *versus* tempo de execução. O cálculo das funções de aproveitamento dos vectores de valores objectivo não-dominados do custo de solução *versus* tempo de execução demonstrou ser computacionalmente exigente. Deste modo, as hipóteses formalizadas foram as seguintes:

H_5 : Não existe diferença de desempenho em termos de funções de aproveitamento do valor mínimo da agregação de objectivos *versus* tempo de execução entre os cinco níveis do parâmetro β , *versus*

K_5 : Pelo menos, dois dos níveis de β apresentam diferença de desempenho em termos de funções de aproveitamento do valor mínimo da agregação de objectivos *versus* tempo de execução, em que H_5 e K_5 são, respectivamente, a hipótese nula e hipótese alternativa;

Comparação entre Métodos de Selecção

A tabela 6.7 apresenta os valores de p obtidos através de 5000 execuções Monte Carlo para o PGHE da UCEH na comparação do método de selecção.

	valor de p
Agregação de Objectivos	0,0000
Funções de Aproveitamento	0,0000

Tabela 6.7: Valores de p para comparação de métodos de selecção na UCEH na análise do custo de solução/tempo de execução

Ambos os valores de p indicam a rejeição de H_1 e H_2 com $\alpha = 0,05$. A figura 6.5 apresenta as superfícies de aproveitamento nº1,50,51,100,101,150,151 e 200 obtidas através dos valores mínimos da agregação de objectivos *versus* tempo de execução obtidas na comparação de método de selecção. As zonas mais escuras representam a localização das diferenças.

É possível verificar que as diferenças residem entre as 100 e as 500 iterações. O sinal observado revela que o método de selecção uniobjectivo é o método preferido nestas regiões, o que significa que este método converge mais rapidamente em termos de agregação dos objectivos. Contudo, a diferença deixa de ser aparente nas iterações seguintes.

A figura 6.6 apresenta as zonas onde foram identificadas diferenças em termos de vectores de valores objectivo não-dominados do custo de solução *versus* tempo de execução. As diferenças a favor do método uniobjectivo estão representadas a negro, as diferenças a favor do método multiobjectivo estão representadas a vermelho. Para efeitos de referência, a primeira superfície de aproveitamento está representada a verde. Os objectivos F(1) e F(2) correspondem aos objectivos (1) e (2) da formalização do problema.

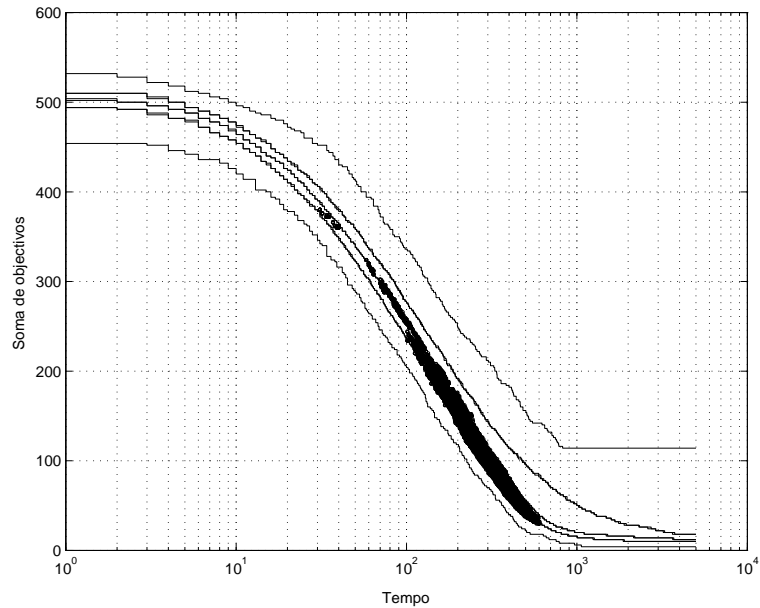


Figura 6.5: Comparação de métodos de selecção na UCEH em termos de agregação de objectivos na análise do custo de solução/tempo de execução

É possível identificar uma região que favorece o método uniobjectivo no início das iterações em que ambos os valores dos objectivos são relativamente baixos. Contudo, a maior área pertence claramente ao método multiobjectivo, pois consegue cobrir desde o início das iterações até quase ao fim da execução. Também é de realçar que a área de cobertura do método multiobjectivo parece indicar que o objectivo 1 é melhor resolvido do que o objectivo 2, ao contrário do método uniobjectivo que encontra um equilíbrio entre estes.

Comparação entre Tipos de Mutação

A tabela 6.8 apresenta os valores de p obtidos através de 5000 execuções Monte Carlo no problema da UCEH na comparação de tipos de mutação.

Ambos os valores de p indicam claramente a rejeição de H_1 e H_2 com $\alpha = 0,05$. A figura 6.7 apresenta as superfícies de aproveitamento n^o 1,50,51,100,101,150,151 e 200 obtidas através dos valores mínimos de agregação de objectivos *versus* tempo de execução obtidas na comparação dos tipos de mutação. As zonas mais escuras representam a localização das diferenças.

É possível verificar que as diferenças residem entre as 10 e as 300 iterações. O sinal observado

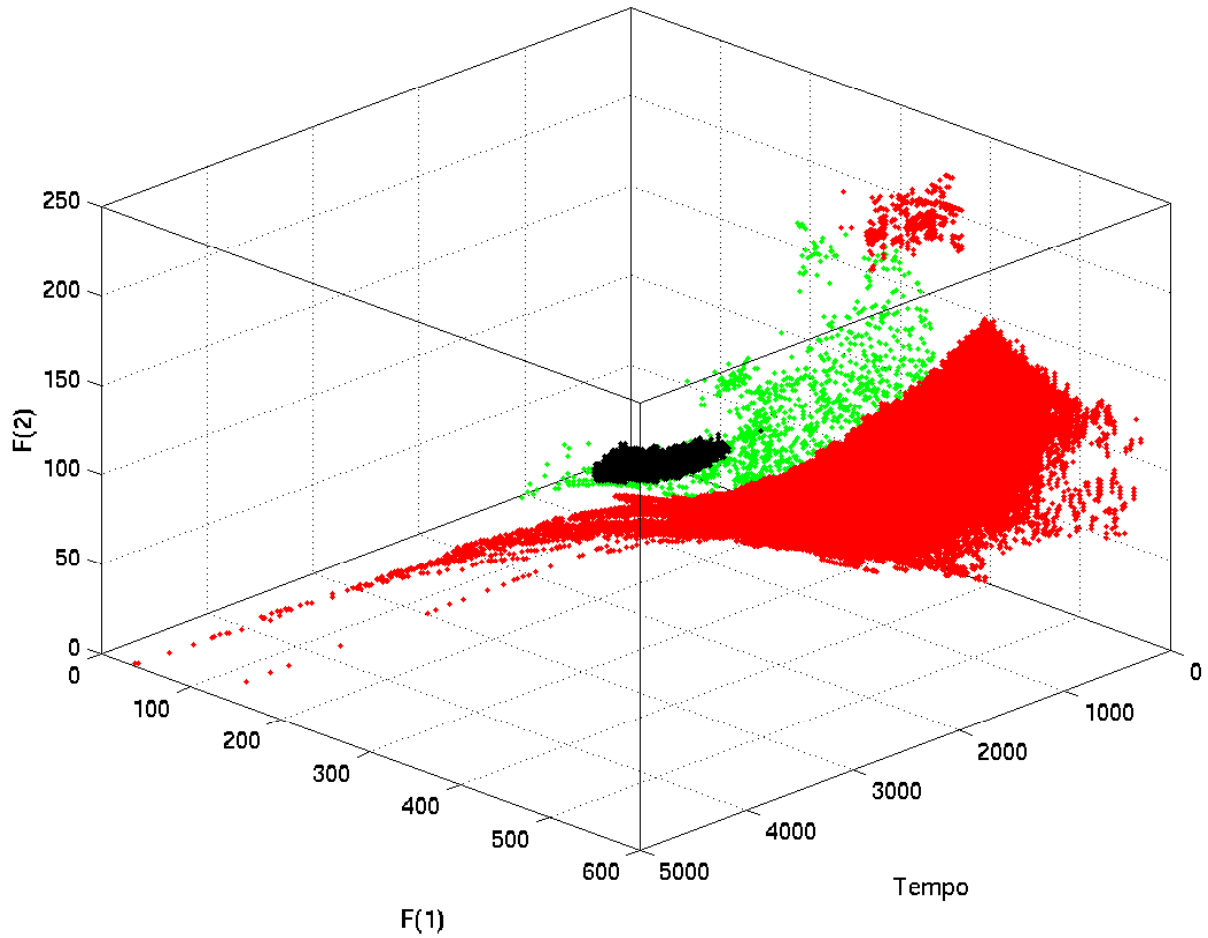


Figura 6.6: Comparação de métodos de selecção na UCEH em termos de separação de objetivos na análise do custo de solução/tempo de execução

	valor de p
Agregação de Objectivos	0,0000
Funções de Aproveitamento	0,0000

Tabela 6.8: Valores- p para comparação de tipos de mutação na UCEH na análise do custo de solução/tempo de execução

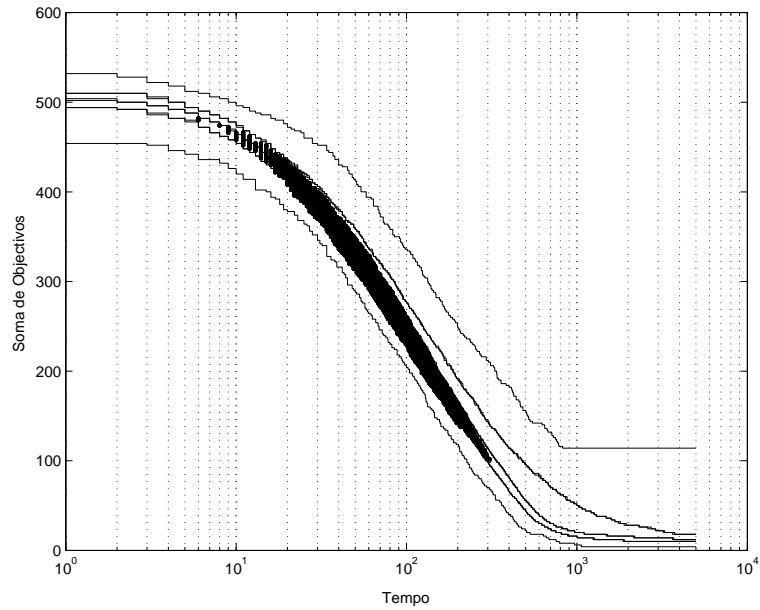


Figura 6.7: Comparação de métodos de selecção na UCEH em termos de agregação de objectivos na análise do custo de solução/tempo de execução

revela que o método de mutação independente é o método preferido nestas regiões, o que significa que este método converge mais rapidamente em termos de agregação dos objectivos. Contudo, a diferença deixa de ser notória nas iterações seguintes.

A figura 6.8 apresenta as zonas onde foram identificadas diferenças em termos de vectores de valores objectivo não-dominados do custo de solução *versus* tempo de execução. As diferenças a favor do método de mutação independente estão representadas a negro e a primeira superfície de aproveitamento está representada a verde. Os objectivos $F(1)$ e $F(2)$ correspondem aos objectivos (1) e (2) da formalização do problema.

Facilmente se verifica que o tipo de mutação independente é mais favorável nas primeiras iterações, confirmando as ilações anteriores.

Comparação do Parâmetro β de Mutação

O valor de p obtido através de 5000 execuções Monte Carlo na comparação dos parâmetros β foi 0,000, indicando claramente a existência de diferenças entre, pelo menos, dois parâmetros. De acordo com os procedimentos descritos na secção 6.3 foi efectuado o teste estatístico fechado. A

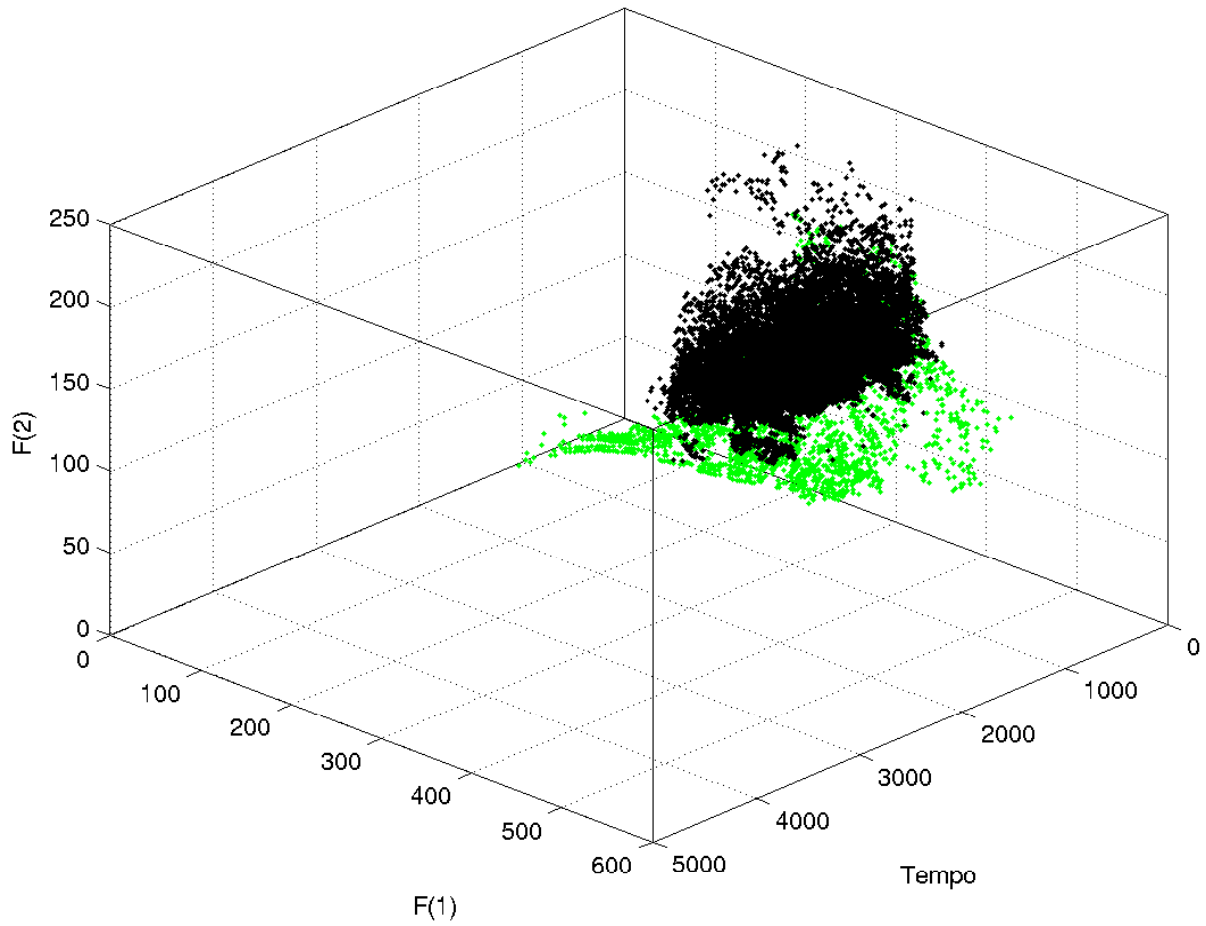


Figura 6.8: Comparação de tipos de mutação na UCEH em termos de separação de objectivos na análise do custo de solução/tempo de execução

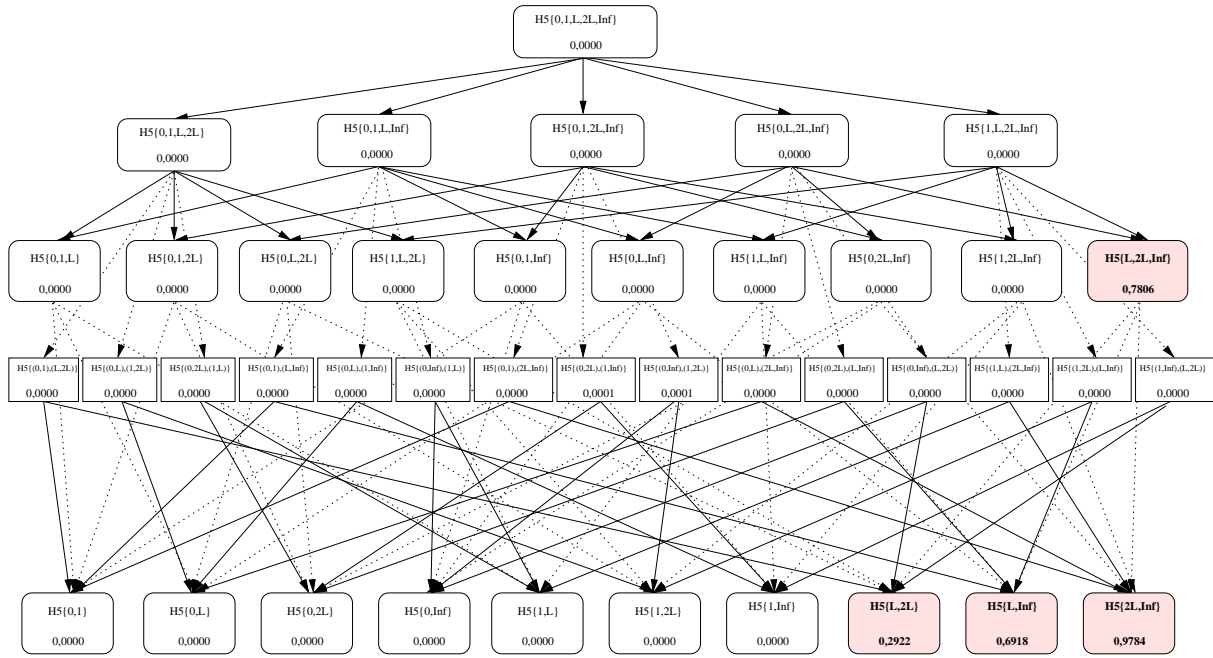


Figura 6.9: Valores de p para comparações múltiplas de parâmetros β em termos de separação de objectivos na UCEH na análise do custo de solução/tempo de execução

figura 6.9 apresenta um esquema das comparações efectuadas e dos valores de p obtidos para testar a hipótese H_4 .

É possível observar que existem diferenças significativas entre $\beta = 0, \beta = 1$ e $\beta = \{L, 2L, \infty\}$ com $\alpha = 0,05$. Os sinais obtidos das diferenças revelam que este último conjunto apresenta melhor desempenho, tal como nas valores de p verificados na secção 6.4.1.

6.4.4 PGHA da UCEH

A análise anterior sobre os PGHEs serviu de base para a escolha de parâmetros do algoritmo para resolver o PHGA da FCHS. Deste modo, foi utilizado um algoritmo com método de selecção multiobjectivo com uma população de 250 indivíduos com 256 genes, pressão selectiva de 2,0 e probabilidade de sobrevivência de 0,5. O método de reinserção foi geracional e o operador de selecção foi o *Stochastic Universal Sampling*. O operador de mutação seleccionado foi a mutação independente com $\beta = 2L$. O critério de paragem foi 10000 gerações.

Duas abordagens foram testadas com três execuções cada uma. A +agregação corresponde à

abordagem que considerou a agregação total dos objectivos com prioridade elevada e –agregação corresponde à agregação dos objectivos de acordo com os recursos envolvidos, como referidos na secção 5.3.2. A tabela 6.9 descreve a solução manual e os melhores resultados obtidos pelas duas abordagens. Como não foi possível satisfazer o objectivo de maior prioridade na abordagem de +agregação, os resultados deste algoritmo sempre consistiram numa só solução. Os valores apresentados pelo –agregação correspondem à solução das três execuções que violou menos as restrições *hard* do problema.

Restrições	Manual	–Agregação	+Agregação
(1)	0	12	0
(2)	0	20	8
(3)	0	2	0
(4)	-	6	3
(5)	11	2	0
(6)	14	0	0
(7)	180	122	130
(8)	19	51	41
(9)	10	12	7
(10)	24	10	5
(11)	4	0	0
(12)	108	8	2
(13)	32	0	0
(14)	168	238	236
(15)	155	195	188

Tabela 6.9: Número de Conflitos no PGHA

É possível verificar que a solução manual produziu melhores resultados. Contudo, é de realçar que esta solução manual utilizou mais salas do que estavam inicialmente afectas. Parte das 39 salas correspondem às que estavam inicialmente afectas a outras faculdades. Os tempos livres deixados pelas outras faculdades permitiram a ocupação das aulas da FCHS sem violar nenhuma restrição *hard* de sobreposição ((1), (2), (3)). É de salientar que a restrição (4) não é contabilizada na

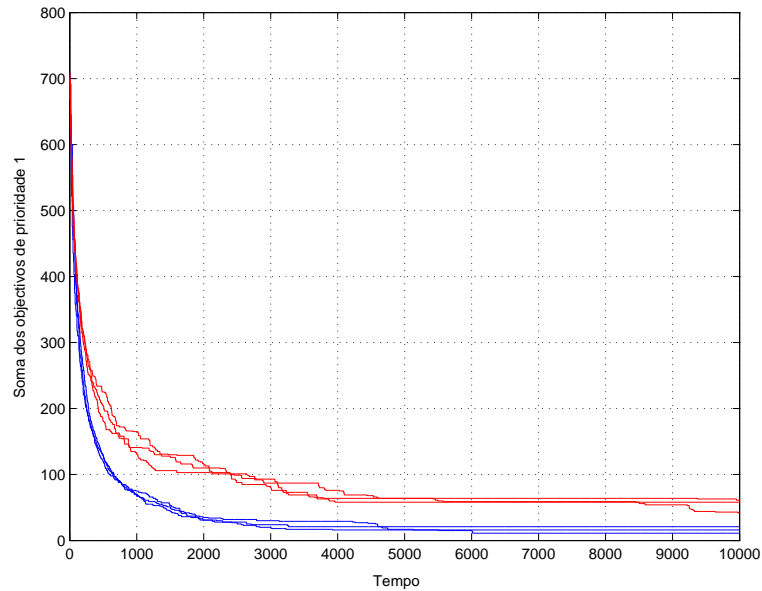


Figura 6.10: Convergência do algoritmo nas abordagens +agregação e -agregação

abordagem manual porque se admite que as marcações efectuadas não consideraram a ocupação de salas em intervalos de tempo exteriores aos seus domínios. É possível que a disponibilidade de mais uma ou duas salas fosse suficiente para gerar horários através da abordagem +agregação. Por outro lado, as restrições *soft* apresentaram melhores resultados em ambas as abordagens computacionais, com excepção das restrições (8), (14) e (15).

A abordagem +agregação apresentou melhores resultados do que a abordagem -agregação. Estes resultados podem ser consequência de uma certa dispersão no processo de pesquisa utilizado pelo algoritmo multiobjectivo quando lida com muitos objectivos. A figura 6.10 apresenta a convergência do algoritmo nas duas abordagens em relação à soma dos objectivos com prioridade mais elevada. As linhas azuis referem-se à abordagem +agregação e as linhas vermelhas à -agregação.

É possível verificar que a convergência é mais acentuada na +agregação do que na -agregação. Contudo, nota-se que a abordagem +agregação estagna mais rapidamente do que a abordagem -agregação. A actividade realçada nas últimas gerações parece indicar que esta última abordagem poderia convergir ainda mais em direcção ao óptimo.

Os dois últimos objectivos são importantes para determinar a boa ou má qualidade de um horário. Um factor considerado pouco agradável para quem é docente ou aluno é a marcação de aulas em dias separados. Assume-se que estes intervenientes preferem um horário mais concentrado

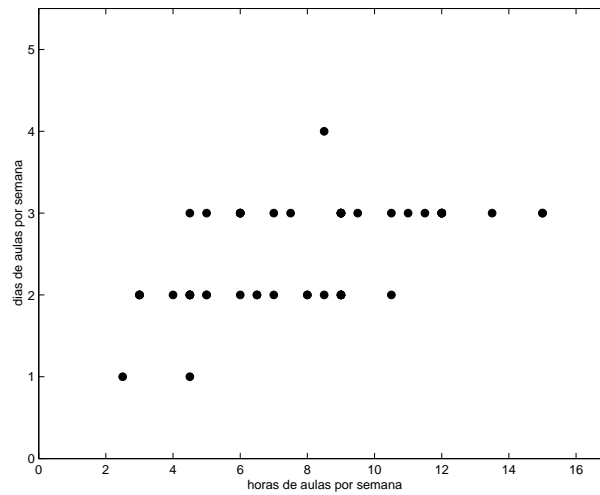


Figura 6.11: N^o de dias de aulas por semana *versus* n^o de horas de aulas por semana na abordagem manual (docentes)

em poucos dias do que disperso em vários dias. O processo de pesquisa tentou minimizar estes dois objectivos. As figuras 6.11- 6.16 representam o número de horas de aulas por semana *versus* o número de manhãs e tardes com aulas por semana, no caso das turmas, e o número de dias de aulas, no caso do docente. É possível verificar que a abordagem manual conseguiu, de facto, satisfazer, com maior sucesso, os objectivos relacionados com a concentração das aulas das turmas e dos docentes.

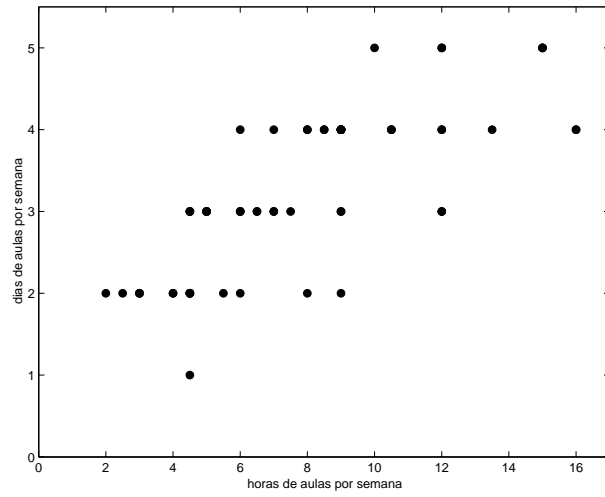


Figura 6.12: N^o de dias de aulas por semana *versus* n^o de horas de aulas por semana na abordagem –agregação (docentes)

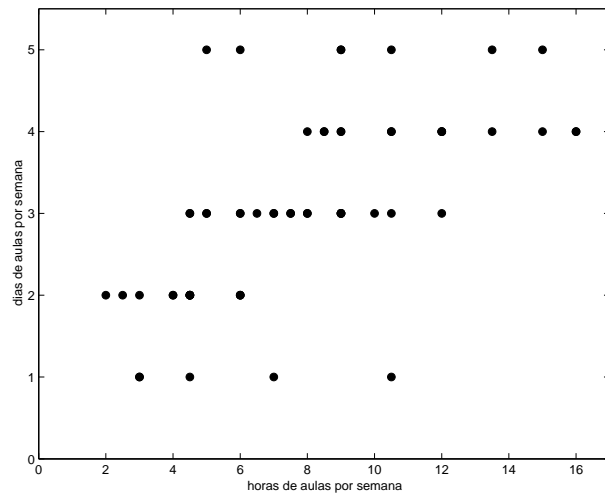


Figura 6.13: N^o de dias de aulas por semana *versus* n^o de horas de aulas por semana na abordagem +agregação (docentes)

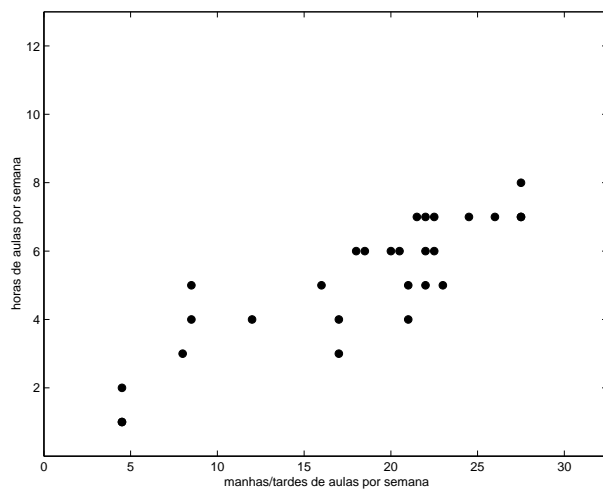


Figura 6.14: N^o de manhãs e tardes de aulas por semana *versus* n^o de horas de aulas por semana na abordagem manual (turmas)

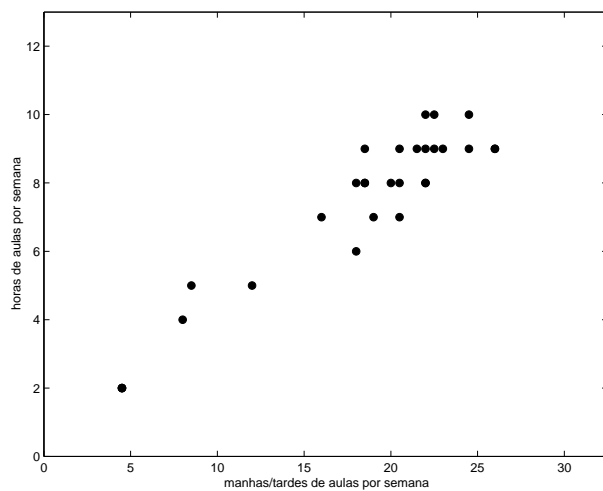


Figura 6.15: N^o de manhãs e tardes de aulas por semana *versus* n^o de horas de aulas por semana na abordagem –agregação (turmas)

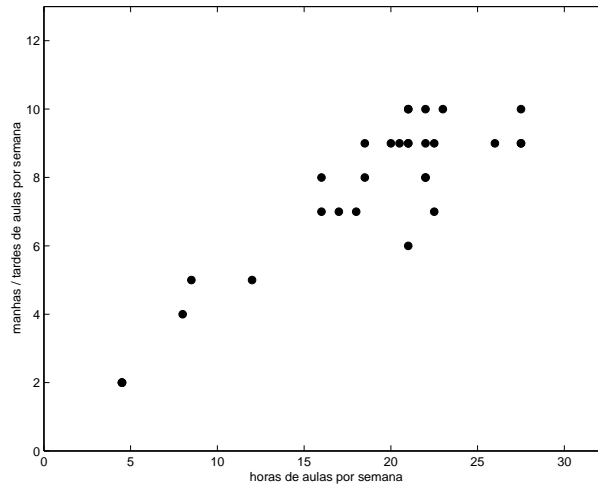


Figura 6.16: N^o de manhãs e tardes de aulas por semana *versus* n^o de horas de aulas por semana na abordagem +agregação (turmas)

6.5 Discussão

Os PGHEs abordados neste estudo apresentam dimensões diferentes. O número de restrições envolvidas em ambos os problemas é suficiente para distingui-los.

A resolução do PGHE da UCEH aproximou-se do valor nulo nos dois objectivos formulados. O método de selecção multiobjectivo foi o método que apresentou melhor desempenho, com a excepção na análise de custo de solução / tempo de execução em termos de soma mínima de agregação de objectivos. Contudo, nesta análise, as diferenças entre métodos foram identificadas nas primeiras gerações, ainda longe da total satisfação dos objectivos. Estes resultados revelam que o método uniobjectivo converge mais depressa, mas que deve estagnar, pois o algoritmo multiobjectivo consegue apresentar melhores resultado no fim das execuções, como é realçado nas análises de custo da solução em tempo fixo. Este facto também é evidenciado na análise de custo de solução / tempo de execução em termos de tempo de execução *versus* separação de objectivos, onde se identifica que o algoritmo que utiliza o método de selecção multiobjectivo é mais robusto durante quase todo o processo de pesquisa.

Os operadores de mutação foram considerados indistinguíveis na análise de custo da solução no tempo fixo. Contudo, na análise da evolução do custo da solução durante o tempo de execução, verificou-se uma vantagem da mutação independente nas primeiras gerações, indicando que este

método leva a uma convergência mais rápida. Contudo, esta vantagem desaparece nas gerações seguintes, apresentando um comportamento semelhante com a mutação de uma posição.

Por fim, os valores do intervalo $[L, 2L, \infty[$ demonstraram ser um intervalo de valores a considerar na análise do parâmetro β do operador de mutação proposto neste estudo. Estes resultados evidenciam a vantagem em utilizar a informação disponível no problema para fornecer um enviesamento explícito no processo de pesquisa.

O problema da UCEE apresenta determinadas características que o distinguem do problema anterior. O PGHE da UCEE apresenta duas restrições *hard*, quatro restrições *soft* e um objectivo a minimizar. Embora falte uma medida que meça a complexidade deste problema, poder-se-á afirmar, pelos resultados obtidos, que o problema deverá estar excessivamente restringido. Neste problema, embora tenha sido possível satisfazer totalmente as restrições de não sobreposição de exames em anos consecutivos, não foi possível satisfazer a restrição de separação de exames no mesmo curso. Contudo, uma inspecção ao número de disciplinas na UCEE permitiu identificar que os quartos anos de Gestão de Empresas e Economia apresentavam 16 disciplinas cada um, que é mais do que o número de dias para marcação de exames. Assim, perante as restrições definidas, o problema da UCEE não tem solução. Esta situação foi o resultado de uma época de transição de currículos de curso em que alunos do quarto ano de ambos os cursos teriam, na realidade, menos disciplinas. Cada aluno, de acordo com as disciplinas que teria deixado atrasadas e actuais estaria inscrito num novo perfil de disciplinas. Poder-se-á supor que uma reformulação do problema com base no currículo das disciplinas de cada aluno tornaria possível a obtenção de uma solução mais aceitável.

Os resultados obtidos no problema da UCEE apresentam algumas ténues diferenças dos observados na UCEH, muito provavelmente derivados da inadmissibilidade do primeiro. Embora o método de selecção multiobjectivo não se distinga do método de selecção uniobjectivo em termos de agregação dos objectivos, o mesmo não ocorreu em termos de funções de aproveitamento. Na comparação do tipo de mutação, também a mutação independente obteve vantagem em termos de funções de aproveitamento. Por sua vez, os valores do parâmetro β que evidenciaram melhor desempenho corresponderam a $\{1, L, 2L, \infty\}$. Este largo intervalo de valores poderá estar relacionado com as características do problema da UCEE que permite que a quantidade de enviesamento seja indiferente.

A geração de horários de aulas pelas abordagens referidas neste estudo indicam resultados pro-

missores que merecem um maior investimento futuro. É de realçar que o PGHA é um problema muito complexo e que demora cerca de duas semanas a ser resolvido manualmente. Os resultados obtidos pela abordagem da agregação dos objectivos com maior prioridade quase se aproximaram da versão manual com um número muito inferior de recursos. Este facto leva a pensar se o processo proposto neste estudo conseguiria apresentar melhor desempenho se existisse mais uma sala disponível.

Das duas abordagens propostas, a que teve em conta a desagregação dos objectivos de elevada prioridade apresentou o pior desempenho. Contudo, e observando o tipo de convergência verificada por esta abordagem nas últimas iterações, levanta a questão se o seu desempenho não acabaria por ultrapassar a abordagem da agregação dos objectivos se o número de iterações fosse maior.

Finalmente, a qualidade dos horários produzida por ambas as abordagens, em termos de concentração de aulas das turmas e docentes, não é comparável aos resultados obtidos pelo processo manual. Contudo é necessário lembrar que os resultados obtidos pelas outras abordagens não conseguiram satisfazer totalmente os objectivos de maior prioridade. Tal como referido anteriormente, é bem possível que a existência de mais uma sala disponível resolvesse estes objectivos de elevada prioridade e que permitissem uma maior satisfação dos objectivos com prioridade mais reduzida.

6.6 Conclusões

O método experimental combinado com princípios de inferência estatística, além de permitir avaliar o desempenho dos algoritmos propostos neste estudo em termos quantitativos, também forneceu alguma luz sobre os efeitos dos métodos e parâmetros presentes nos algoritmos. O conceito de funções de aproveitamento foi de grande utilidade para caracterizar os algoritmos evolutivos multiobjectivo. A adição do tempo de execução como mais um objectivo a minimizar permitiu uma reformulação inovadora do processo de avaliação do desempenho dos algoritmos.

Os resultados obtidos pelos algoritmos propostos indicam que a formalização multiobjectivo com base em metas e prioridades faz sentido nos problemas abordados. Do mesmo modo, um certo enviesamento explícito fornecido pelo operador de mutação demonstrou que favorecia o processo de pesquisa. A desagregação dos objectivos foi igualmente vantajosa nos PGHE. Pelo contrário, no PGHA abordado, os resultados indicam que a agregação poderá favorecer mais o processo de

pesquisa. Contudo, uma questão pertinente se levanta sobre se um maior número de iterações não irá beneficiar a desagregação dos objectivos.

Capítulo 7

Conclusões

7.1 Algoritmos Evolutivos Multiobjectivo para Afectação de Recursos

Este estudo consistiu numa formalização multiobjectivo de problemas de afectação de recursos, com incidência nos problemas de geração de horários, e aplicação da metodologia evolutiva proposta por Fonseca e Fleming [46] para resolver problemas multiobjectivo em dois PGHEs e num PGHA na Universidade do Algarve.

É possível constatar que a definição de metas e prioridades, como formalizadas neste estudo, apoiam o processo de decisão num problema típico de elaboração de horários. Muitas das abordagens efectuadas nesta classe de problemas baseiam-se na quantificação de ponderações que não são transparentes para o decisor. A especificação dos níveis de desempenho desejados por cada objectivo e a definição da ordem pela qual os objectivos devem ser satisfeitos permitem apoiar o decisor na articulação das suas preferências e, como consequência, definir naturalmente o processo de pesquisa. Contudo, este processo, definido por um método claramente explícito do que se pretende com cada objectivo, não deve ter por base mecanismos de optimização que podem enviesar de modo implícito, o processo de decisão. As escolhas efectuadas neste estudo têm em conta estas considerações.

Os algoritmos evolutivos, que exploram simultaneamente várias regiões do espaço de pesquisa através da interacção entre várias soluções candidatas, demonstram serem métodos que facilmente se adaptam ao conceito multiobjectivo. A utilização da representação directa é também uma escolha

relacionada com a natural representação do problema. Neste estudo, cada cromossoma representa um horário e cada gene desse cromossoma representa um evento ao qual é atribuído um valor que se refere aos recursos disponíveis para esse evento. Embora se admita que a representação directa origina muitas violações das restrições, como consequência da inexistência de um enviesamento implícito, este facto pode ser minorado com o apoio de operadores que utilizem informação do problema. Mas, mais uma vez, o enviesamento fornecido por estes operadores deverá ser explícito de acordo com o processo de decisão formalizado inicialmente. Assim, o operador de mutação proposto neste estudo tem em conta a satisfação dos objectivos através da intervenção sobre aquilo que impede que sejam satisfeitos, ou seja, na violação das restrições que compõem esses objectivos.

Esta acção da mutação não deverá ser inocente ao ponto de mudar apenas e de forma determinística as posições do cromossoma que apresentam mais conflitos. Este facto é perfeitamente compreensível se se considerar o verdadeiro espaço de pesquisa dos problemas em estudo como uma paisagem bastante irregular, repleto de óptimos locais e de "armadilhas" para operadores determinísticos. Tal como outras meta-heurísticas, os algoritmos evolutivos propostos utilizam mecanismos estocásticos para explorar outras regiões do espaço objectivo e para escapar de possíveis óptimos locais. Contudo, uma questão se levanta sobre quanta aleatoriedade deverá ser utilizada no processo de pesquisa sem que este se torne numa pesquisa aleatória. A componente estocástica, expressa pela relação entre a probabilidade de mutação óptima e a pressão selectiva, e o enviesamento explícito do processo de pesquisa no sentido da satisfação das restrições, fornecido pelo parâmetro β , fornecem o equilíbrio necessário. Os resultados obtidos, inclusive, sugerem que o valor de β deverá estar compreendido algures entre o comprimento do cromossoma e infinito.

A avaliação de desempenho dos algoritmos propostos consistiu numa abordagem baseada em inferência estatística, com metodologia experimental semelhante à utilizada noutras ciências empíricas. A metodologia experimental foi delineada para análise dos factores principais existentes, respectivamente, os métodos de selecção, tipos de mutação e valores do parâmetro β . A análise estatística apresentada permitiu determinar, através de estatística não-paramétrica, se um determinado factor apresenta algum efeito no desempenho do algoritmo, assumindo que os outros factores permaneceriam constantes a qualquer nível. É de realçar que os resultados obtidos por um algoritmo multiobjectivo, que produz várias soluções não-dominadas, são diferentes dos obtidos por um algoritmo que lide só com um objectivo. O seu tratamento não é trivial e é necessário um cuidado especial nas metodologias estatísticas a utilizar.

A função de aproveitamento lida com a análise de conjuntos de pontos não-dominados. Este método assenta numa ideia simples em que as distribuições dos resultados de várias execuções de um algoritmo multiobjectivo podem ser caracterizadas pela probabilidade do algoritmo atingir uma meta arbitrária numa só execução. Este conceito foi combinado com a metodologia experimental para comparar o desempenho dos algoritmos propostos. A utilização dos resultados obtidos pelos algoritmos propostos através da estimação de funções de aproveitamento empíricas e da aplicação de testes estatísticos não-paramétricos nelas baseadas permitiram não só identificar as diferenças estatísticas entre grupos de algoritmos, mas também observar graficamente as diferenças mais notáveis, para uma melhor compreensão dos resultados.

Este estudo também sugeriu e implementou a utilização de testes fechados para detectar a origem de diferenças, mantendo controlada a probabilidade de uma ou mais hipóteses serem rejeitadas incorrectamente. Finalmente, a análise do custo de solução *versus* tempo de execução, implementada pela primeira vez, permitiu aliar a interpretação de desempenho mais comum, que assenta sobre um compromisso entre o tempo de execução e a qualidade das soluções obtidas, à noção de qualidade das soluções na presença de vários objectivos, integrando ambos num só formalismo.

7.2 Perspectivas Futuras

7.2.1 Processo de Decisão Multiobjectivo

Como foi referido anteriormente, a definição de metas e prioridades permite, de uma forma transparente, apoiar o processo de decisão. Embora a escolha dos objectivos efectuada neste estudo tentasse reflectir o que os intervenientes no processo julgaram ser um bom horário de exames ou de aulas, é necessário, agora, formalizar um conjunto de objectivos para esta classe de problemas que englobe as preferências do decisor de uma forma mais global. Paralelamente, um estudo sobre a quantificação de cada nível de desempenho pretendido por cada objectivo deve ser efectuado.

É indiscutível que a representação gráfica facilita a tomada de decisão. Foi possível verificar que a presença de muitos objectivos dificultam a apresentação gráfica dos resultados ao decisor. A apresentação dos resultados finais deve ser convenientemente explorada para facilitar a escolha da solução final.

Este estudo também assumiu uma articulação das preferências do decisor *à priori*. Contudo, é possível incluir, interactivamente, o processo de decisão durante o processo de pesquisa. Este facto

permitirá que o decisor tenha mais intervenção no processo e que escolha as regiões do espaço objectivo mais interessantes para o algoritmo explorar.

7.2.2 Algoritmos Evolutivos Multiobjectivo

A aplicação de algoritmos evolutivos na resolução de problemas de optimização multiobjectivo é uma área de estudo relativamente recente. A existência de muitos problemas reais multiobjectivo têm motivado a continuação da exploração destas técnicas. Contudo, algumas questões estão ainda em aberto sobre o modo como o processo de decisão influencia o processo de pesquisa. Neste estudo observou-se que a desagregação dos objectivos foi favorável no PGHEs, mas foi aparentemente mais desfavorável nos PGHA. Num problema sobre optimização de sistemas térmicos, Wright e Loosemore [110] observaram que a redução do número de objectivos por agregação não só facilitava a interpretação dos resultados como reduzia o tempo de convergência do algoritmo. É necessário aprofundar até que ponto a agregação ou não dos objectivos afecta o comportamento do algoritmo multiobjectivo, e até que ponto esta agregação não afecta o processo de decisão. Algumas técnicas utilizadas pelos algoritmos evolutivos multiobjectivo tais como partilha de aptidão [55] e *crowding* [72] deverão ser testadas e comparadas para explorar até que ponto algoritmos que as incluam podem resolver mais rapidamente um problema com objectivos desagregados.

Outro aspecto importante é o tempo computacional dos algoritmos multiobjectivo. O algoritmo multiobjectivo necessita de comparar os pontos não-dominados da geração corrente com os pontos obtidos nas gerações anteriores para manter uma representação fiel das soluções encontradas para o problema. Este factor causa algum atraso no algoritmo se o número de pontos não-dominados em cada geração for elevado. É necessário, então, encontrar técnicas que reduzem os tempos computacionais. Modelos de população estruturada para ambientes de processamento paralelo, como o modelo de ilhas [60] e difusão [53], que além de reduzir o tempo computacional também afectam o processo de pesquisa, devem ser igualmente testados e comparados.

7.2.3 Aplicação em Problemas de Afectação de Recursos

Será necessário verificar a aplicação da abordagem efectuada neste estudo noutros problemas de afectação de recursos, tais como no âmbito de afectação de empregados a turnos, marcações de reuniões de trabalho entre agentes com diferentes preferências e afectação de frequências de rádio

a canais de comunicações, por exemplo. Esta metodologia também deverá ser estendida a outros problemas de optimização combinatória onde se possa localizar a violação de restrições.

Uma ligação com métodos de programação com restrições beneficia a pesquisa do algoritmo evolutivo proposto neste estudo, visto que a redução do espaço de pesquisa liberta o algoritmo de explorar regiões que são à partida inadmissíveis. A possibilidade de reforçar a aplicação destes métodos durante o processo de pesquisa deverá ser alvo de um estudo futuro. Marchioiri [73] e Barnier [6] apresentaram algumas propostas de hibridização destas duas técnicas em problemas teóricos que lidam com esta linha de investigação.

7.2.4 Operadores de Mutação e Cruzamento

Neste estudo não foi concretizada nenhuma tentativa de ajustar quaisquer parâmetros de operadores de recombinação. Deverá ser efectuada uma análise mais detalhada sobre a interacção destes operadores com os restantes parâmetros descritos neste estudo e se a mesma informação que é disponibilizada para o operador de mutação poderá, de facto, ser utilizada pela recombinação. Em complemento, um estudo sobre quais os melhores blocos constructivos para esta classe de problemas poderia fornecer alguma clarificação sobre o modo de partição do cromossoma no processo de recombinação. Em particular, a aplicação aos problemas em estudo da metodologia proposta por Lobo [70] poderá ser uma área de trabalho promissora.

É possível constatar que a informação disponibilizada sobre o grau de violações das restrições e do número de alternativas é útil para o processo de pesquisa. Contudo, algo fica em aberto sobre uma possível relação entre a dificuldade dos problemas e o parâmetro β do operador de mutação. Como foi referido na secção 4.5, um problema com elevado grau de complexidade deverá necessitar de um parâmetro β que privilegie a capacidade de distender mais a pesquisa a outros locais do espaço dos objectivos. Pelo contrário, um problema com reduzido grau de complexidade poderá ser resolvido mais rapidamente com $\beta = \infty$. Nos problemas em estudo, observou-se que os melhores resultados eram obtidos com $\beta \in \{L, 2L, \infty\}$. Este facto indica que não será necessário enviar totalmente o algoritmo no sentido da tentativa de satisfação das restrições para se obter um bom desempenho. Contudo, será necessário efectuar um estudo mais aprofundado sobre este assunto, tendo em conta medidas de dificuldade dos problemas, como já efectuado por [89].

7.2.5 Avaliação Estatística de Algoritmos Estocásticos

A quantidade de dados resultantes, quer das funções de aproveitamento como do próprio algoritmo multiobjectivo, apresenta grandes dimensões e torna difícil a sua visualização. Deverá ser possível ultrapassar esta dificuldade através de uma redução do volume de dados gerados pelo algoritmo sem perder a uniformidade. Técnicas semelhantes às utilizadas em computação gráfica poderão ser utilizadas de uma forma análoga para remover pontos não-dominados redundantes sem afectar a estrutura principal.

Foi demonstrada a utilidade da aplicação de funções de aproveitamento para avaliar o desempenho de algoritmos multiobjectivo, mas também para avaliar, pela primeira vez, o comportamento de tais algoritmos durante o tempo de execução. A análise de desempenho em termos do custo da solução/tempo de execução deverá ser mais aprofundada e igualmente utilizada para comparar e analisar as propriedades de convergência de outros algoritmos estocásticos além dos algoritmos evolutivos, como algoritmos de pesquisa tabu, arrefecimento simulado e pesquisa local iterativa, em problemas tanto multiobjectivo como uniobjectivo.

Bibliografia

- [1] Allander, J., Population Size, Building Blocks, Fitness Landscape and Genetic Algorithm Search Efficiency in Combinatorial Optimization: An Empirical Study, in Chambers, L. (ed.), Practical Handbook of Genetic Algorithms - Complex Coding Systems, Vol 3, 1999, pp. 487 - 499;
- [2] Amaral, J., Tumer, K., Ghosh, J., Designing Genetic Algorithms for the State Assignment Problem, in IEEE Transactions on System, Man and Cybernetics, Vol. XX, 1999, pp. 100 - 108;
- [3] Bäck, T., Optimal Mutation Rates in Genetic Search, in Forrest. S. (ed), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufman, 1993, pp. 2 - 9;
- [4] Bäck, T., Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms, in Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE Press, Piscataway NJ, 1994, pp. 57 - 62;
- [5] Baker, J., Reducing Bias and Inefficiency in the Selection Algorithm, in Proceedings of the Second International Conference on Genetic Algorithms, 1987, pp. 14 - 21;
- [6] Barnier, N., Brisset, P., Optimization By Hybridization of a Genetic Algorithm with Constraint Satisfaction Techniques, in Proceedings of the Fifth International Conference of Evolutionary Computation, 1998, pp. 645 - 649;
- [7] Booker, L., Improving Search in Genetic Algorithm, Davis, L, (ed.), Genetic Algorithms and Simulated Annealing, Morgan Kaufmann, 1987;
- [8] Bullnheimer, B., An Examination Scheduling Model to Maximize Student's Study Time, in Burke, E., Carter, M., eds., Praticce and Theory of Automated Timetabling, Second Inter-

- national Conference, Edinburgh, 1997: Selected Papers, Lecture Notes in Computer Science n°1408, Springer, New York, 1998, pp. 78-91;
- [9] Burke, E., Elliman, D., Ford, P., Weare, R., A Genetic Algorithm Based University Timetabling System, in the Proceedings of the Second East-West International Conference on Computer Technologies in Education, Crimea, Ukraine, Vol 1, 1994, pp. 35-40;
- [10] Burke, E., Elliman, D., Weare, R., A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems, in the Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, 1995, pp. 605-610;
- [11] Burke, E., Elliman, D., Weare, R., Specialised Recombinative Operators for Timetabling Problems, in the Proceedings of the AISB Workshop in Evolutionary Computing, Lecture Notes in Computer Science, Springer, 1995, pp. 75-85;
- [12] Burke, E., Newall, J., Weare, R., A Memetic Algorithm for University Exam Timetabling, in Burke, E., Ross, P., (eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 241 - 250;
- [13] Burke, E., Ross, P. (eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996;
- [14] Burke, E., Elliman, D., Ford, P., Weare, R., Examination Timetabling in British Universities, in Burke, E., Ross, P., (eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 76 - 90;
- [15] Burke, E., Carter, M., (eds.), Second International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1408, Springer, 1998;
- [16] Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, Fachhochschule Konstanz - University of Applied Sciences, Germany, 2000;
- [17] Burke, E., Bykov, Y., Petrovic, S., Multicriteria Approach to Timetabling Problems, in Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, 2000, p. 180;

- [18] Caldeira, J., Rosa, A., School Timetabling using Genetic Search, Burke, E., Carter, M., (eds.), Proceedings of the Second International Conference in Practice and Theory of Automated Timetabling, 1998;
- [19] Carrasco, M., Pato, M., Automated Timetabling Using Multiobjective Genetic Algorithm, in Burke, E., Erben, W., (eds.), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, 2000, p. 127;
- [20] Carter, M., Recent Developments in Practical Examination Timetabling, in Burke, E., Ross, P.,(eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n^o1153, Springer, 1996, pp. 3-21;
- [21] Carter, M., Laporte, G., Lee, S., Examination Timetabling: Algorithms Strategies and Applications, Journal of Operational Research Society, 47, 1996, pp. 373-383;
- [22] Castelino, D., Hurley, S., Stephens, N., A Tabu Search Algorithm for Frequency Assignment, in Laporte, G., Osman, I. (eds.), Metaheuristics in Combinatorial Optimization, Annals of Operation Research, Vol. 63, 1996, pp. 301-320;
- [23] Cheeseman, P., Kanefsky, B., Taylor, W., Where the Really Hard Problems Are, in Proceedings IJCAI-91, Vol 1, 1991, pp. 331-337;
- [24] Cheng, C., Kang, L., Leung, N., White G., Investigations of a Constraint Logic Programming Approach to University Timetabling, in Burke, E., Ross, P.,(eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n^o1153, Springer, 1996, pp. 112-129;
- [25] Conover, J., Practical Nonparametric Statistics, John Wiley & Sons, New York, 1980;
- [26] Cooper, T., Kingston, J., The Complexity of Timetable Construction Problems, in Burke, E., Ross, P.,(eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n^o1153, Springer, 1996, pp. 283 - 295;
- [27] Corne, D., Fang, H.-L., & Mellish, C., Solving the modular exam scheduling problem with genetic algorithms. Technical Report 622, Department of Artificial Intelligence, University of Edinburgh, 1993;

- [28] Corne, D., Ross, P., Fang, H.-L., Evolving Timetables, in Chambers, L. (ed.), Practical Handbook of Genetic Algorithms - Applications, Vol 1, 1995, pp. 219 - 276;
- [29] Corne, D., Ross, P., Peckish Initialisation Strategies for Evolutionary Timetabling, Burke, E., Ross, P.,(eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 227-240;
- [30] Corne, D., Ogden, J. Evolutionary Optimisation of Methodist Preaching Timetables, in Burke, E., Carter, M., (eds.), Second International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1408, Springer, 1998, pp. 142 - 155;
- [31] Dakic, T., Adhikary, J., Gaur, D., Jackson, W., Backtrack-Free Search for Resource Allocation Problems, Technical Report , CMPT TR 95-2, School of Computing Science, Simon Fraser University, 1995;
- [32] Davis, L., Applying Operator Probabilities in Genetic Algorithms, Schaffer, J. (ed), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufman, 1989, pp. 61 - 69;
- [33] Werra, D., An Introduction to Timetabling, European Journal of Operational Research, Elsevier, 1985, pp. 151 - 162;
- [34] Dowsland, K., Simulated Annealing Solutions for Multi-objective Scheduling and Timetabling, in Rayward, V., Osman, I., Reeves, C., Smith G. (eds.), Modern Heuristic Search Methods, 1996, John Wiley & Son's, pp. 155 - 166.
- [35] Erben, W., A Grouping Genetic Algorithm for Graph Colouring and Exam Timetabling, in Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, 2000, pp. 397 - 421;
- [36] Erben, W., Keppler, J., A Genetic Algorithm Solving a Weekly Course Timetabling Problem, Burke, E., Ross, P.,(eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 198 - 211;

- [37] Eshelman, L., Caruana, R., Schaffer, J., Biases in the Crossover Landscape, in Schaffer, J., (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufman, 1989, 10 - 19;
- [38] Ferland, J., Generalized Assignment-Type Problems - A Powerful Modeling Scheme, in Burke, E., Carter, M., (eds.), Second International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1408, Springer, 1998, pp. 53 - 77;
- [39] Fogarty, T., Varying the Probability of Mutation in the Genetic Algorithm, Schaffer, J. (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufman, 1989, 104 -109;
- [40] Fogel, D., Introduction, Back, T., Fogel, D., Michalewicz (eds), Handbook of Evolutionary Computation, Oxford University Press, 1997, pp. A.1 - A.1.2;
- [41] Fonseca, C., Fleming, P., Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalization, in Forrest, S. (ed.), Proceedings of the Fifth International Conference in Genetic Algorithms, Morgan Kaufmann, 1993, pp.141-153;
- [42] Fonseca, C., Mendes, E., Fleming, P., Billings, S., Non-linear Model Term Selection with Genetic Algorithms," in IEE/IEEE Workshop on Natural Algorithms in Signal Processing, vol. 2, 1993, pp. 27/1 - 27/8;
- [43] Fonseca, C., Fleming, P., Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restrictions, in First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 95), 1995, pp. 45 - 52;
- [44] Fonseca, C., Fleming, P., An Overview of Evolutionary Algorithms in Multiobjective Optimization, Evolutionary Computation, 3(1), 1995, pp. 1-16;
- [45] Fonseca, C., Multiobjective Genetic Algorithm with Application to Control Engineering Problems, Tese de Doutoramento, University of Sheffield, 1995;
- [46] Fonseca, C., Fleming, P., On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers, in Voigt, H.-M., Ebeling, W., Rechenberg, I., Schwefel, H.- P.(eds.),

- Proceedings of the Fourth Conference on Parallel Problems Solving from Nature, number 1141, Springer Verlag, 1996, pp. 584 - 593;
- [47] Fonseca, C., Fleming, P., Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms - Part I: A Unified Formulation, IEEE Transactions on Systems, Man and Cybernetics, 28 (1), 1998, pp. 26-37;
- [48] Fonseca, C., Fleming, P., Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms - Part II: Application Example, IEEE Transactions on Systems, Man and Cybernetics, 28 (1), 1998, pp. 38 - 47;
- [49] Fonseca, V., Fonseca, C., Hall, A., Inferential Performance Assessment of Stochastic Optimizers and the Attainment Function, in Zitzler, E., Deb, K., Thiele, L., Coello, C., Corne, D., First International Conference in Evolutionary Multi-criterion Optimization, Lecture Notes 1993, Springer, 2001, pp. 213 - 225;
- [50] Garey, M., Johnson, D., Computers and Intractability, W. F. Freeman, 1979;
- [51] Gaspero, L., Schaerf, A., Tabu Search Techniques for Examination Timetabling, in Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, 2000, pp. 176 - 179;
- [52] Gent, I., Phase Transitions from Real Computational Problems, in Proceedings of the Eighth International Symposium on Artificial Intelligence, 1995, pp. 356 - 364;
- [53] George-Schleuter, M., ASPARAGOS: An Asynchronous Parallel Genetic Algorithm Optimization Strategy, in Schaffer, J. (ed.) Proceedings of the Third International Conference on Genetic Algorithm, 1985, pp. 422 - 427;
- [54] Gueret, C., Jussien, N., Boizumault, P., Prins, C., Building University Timetables Using Constraint Logic Programming, Burke, E., Ross, P. (eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 130-145;
- [55] Goldberg, D., Richardson, J., Genetic Algorithms with Sharing for Multimodal Function Optimization, in Grefenstette, J.(eds), Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum, 1992, pp. 41- 49;

- [56] Goldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989;
- [57] Goldberg, D., Sizing Populations for Serial and Parallel Genetic Algorithms, in Schaffer, J., (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufman, 1989, pp. 70-79;
- [58] Goldberg, D., Deb, K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, Rawlins, G., (ed.) , Foundations of Genetic Algorithms, Morgan Kaufmann, 1991, pp. 69 - 93;
- [59] Gomes, C., Structure, Duality, and Randomization: Common Themes in AI and OR, in the Proceedings of AAAI00, 2000;
- [60] Gordon, V., Whitley, D., Bohn, A., Dataflow Paralellism in Genetic Algorithms, in Manner, R., Manderick, B. (eds.), Proceedings of the Second International Conference on Parallel Problem Solving from Nature, Elsevier Science, 1992, pp. 533 - 542;
- [61] Good, P. Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses. Springer-Verlag, New York, 1994;
- [62] Hart, E., Corne, D., The State of Art in Evolutionary Approaches to Timetabling and Scheduling, EvoStim Report,1993;
- [63] Henz, M., Wurtz, J., Using Oz for College Timetabling, Burke, E., Ross, P. (eds.), First International Conference in Praticce and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 162-177;
- [64] Hooker, J., Testing Heuristics: We Have it All Wrong, Journal of Heuristics, 1996, pp. 33 - 42;
- [65] Hoos, H., Stutzle, T., Evaluating Las Vegas Algorithms - Pitfalls and Remedies, in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence,1998, pp. 238 - 245;
- [66] Horn, J., Nafpliotis, N., Multiobjective Optimization Using the Niched Pareto Genetic Algorithm, IlliGAL Report 93005, Illinois Genetic Algorithm Laboratory, University of Illinois, Champaign, 1993;

- [67] Hsu, J., Multiple Comparisons - Theory and Methods - Chapman & Hall / CRC, 1996;
- [68] Kiaer, L., Yellen, J., Weighted Graphs and University Course Timetabling, Computers Operations Research, Vol 1, 1992, pp. 363-375;
- [69] Lehmann, E. L., Testing Statistical Hypothesis, Springer Verlag, 1997;
- [70] Lobo, F., The Parameter-Less Genetic Algorithm: Rational and Automated Parameter Selection for Simplified Genetic Algorithm Operation, Tese de Doutorado, Universidade Nova de Lisboa, 2000;
- [71] Lutz, M., Programming Python, O'Reilly, 1996;
- [72] Mahfoud, S., Crowding and Preselection Revisited, in Manner, R., Manderick, B. (eds.), Proceedings of the Second Conference on Parallel Problem Solving from Nature, Elsevier Science, 1992, pp. 27 - 36;
- [73] Marchiori, E., Combining Constraint Processing and Genetic Algorithms for Constraint Satisfaction Problems, in Back, T., (ed), Proceedings of the Seventh International Conference on Genetic Algorithms, 1997, pp. 330 - 337
- [74] Meisels, A., Lusternik, N., Experiments on Network of Employee Timetabling Problems, Burke, E., Carter, M., (eds.), Second International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n^o1408, Springer, 1998, pp.130 - 141;
- [75] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1996;
- [76] Momjiam, B., Postgresql, Introduction and Concepts, Addison-Wesley, 2000;
- [77] Mühlenbein, H., Genetic Algorithms, in Aarts, E., Lenstra, J., (eds.), Local Search in Combinatorial Optimization, John Wiley & Sons, 1997;
- [78] Muslija, N., Gartner, J., Slany, W., Efficient Generation of Rotating Workforce Schedules, in Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, 2000, pp. 314 - 333;

- [79] Ochoa, G., Harvey, I., Buxton, H., Error Thresholds and Their Relation to Optimal Mutation Rates, (eds) in Floreano, D., Nicoud, J.-D., Mondada, Proceedings of the Fifth European Conference on Artificial Life, Springer-Verlag, 1999;
- [80] Oliveira, E., Smith, B., Solving Single-Track Railway Scheduling Problems With Constraint Programming, in Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, 2000, pp. 372-373;
- [81] Paechter, B., Luchian, H., Petruic, M., Two Solutions to the General Timetable Problem Using Evolutionary Methods, in Michalewicz, Z., Schaffer, D., Schwefel, H.-P., Fogel, D., Kitano, H. (eds), Proceedings of the First IEEE International Conference on Evolutionary Computation, Volume I, Orlando, June, 1994, pp. 300 - 305;
- [82] Paechter, B., Rankin, R., Cumming, A., Fogarty, T., Timetabling the Classes of an Entire University with an Evolutionary Algorithm, in the Proceedings of the Fifth Conference in Parallel Problem Solving from Nature, 1998;
- [83] Papadimitriou, C., Steiglitz, K., Combinatorial Optimization, Dover, 1982;
- [84] Paredis, J., Co-Evolutionary Constraint Satisfaction, in Davidor, Y., Schwefel, H.-P., Manner, R. (eds.), Third Conference on Parallel Problem Solving from Nature, Springer Verlag, 1994, pp. 46 - 55;
- [85] Pato, M., Carrasco, M., A Mathematical Formulation for a Secondary School Timetabling Problem, *Estudos de Economia*, Vol 18, n^o. 2, 1998, pp. 145-160;
- [86] Ross, P., Fang, H., Mellish, C., Solving the Modular Exam - Scheduling Problem with Genetic Algorithms, in Lovegrove, Chung, Ali, (eds.), Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Gordon and Breach Science Publishers, 1993, pp. 370 - 373;
- [87] Ross, P., Fang, H., Corne, D., Successful Lecture Timetabling with Evolutionary Algorithms, Technical Report, University of Edinburgh, 1994;
- [88] Ross, P., Corne, D., Fang, H., Improving Evolutionary Timetabling with Delta Evaluation and Directed Mutation, in Davidor, Y., Schwefel, H.-P., Manner, R. (eds.), Third Conference on Parallel Problem Solving from Nature, Springer Verlag, 1994, pp. 556 - 565;

- [89] Ross, P., Corne, D., Terashima, H., The Phase Transition Niche for Evolutionary Algorithms in Timetabling, in Burke, E., Ross, P. (eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 309-324;
- [90] Ross, P., Hart, E., Corne, D., Some Observations About GA Based Timetabling, in Burke, E., Carter, M., (eds.), Second International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1408, Springer, 1998, pp. 115-130;
- [91] Schaffer, J., Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, in Grefenstette (ed.), Proceeding of the International Conference on Genetic Algorithms and Their Applications, pp. 93-100, 1985;
- [92] Schaffer, J., Caruana, R., Eshelman, L., Das, R., A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization, in Schaffer, J., (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufman, 1989, pp. 51 - 60;
- [93] Shaw, K., Nortcliffe, A., Thompson, A., Fonseca, C., Love, J., Fleming, P, Assessing the Performance of Multiobjective Genetic Algorithms for Optimization of a Batch Process Scheduling Problem, in Proceedings of the Congress of Evolutionary Computation, 1999;
- [94] Smith, B., A Tutorial on Constraint Programming, Report 95.14, School of Computer Studies, University of Leeds, 1995;
- [95] Srinivas, N., Deb, K., Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, 2(3), 1994, pp. 221 - 248;
- [96] Stamatopoulos, P., Viglas, E., Karaboyas, S., Nearly Optimum Timetable Construction Through CLP and Intelligent Search, *International Journal of Artificial Intelligence Tools*, Vol 7, N°4, 1998, pp. 415 - 442;
- [97] Syswerda, G., Uniform Crossover in Genetic Algorithms, in Schaffer, J., (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufman, 1989, pp. 2 - 9;

- [98] Syswerda, G., Schedule Optimization Using Genetic Algorithms, in Davis, L., (ed.), Handbook of Genetic Algorithms, International Thompson Computer Press, 1996;
- [99] Terashima, H., Combinations of GAs and CSP Strategies for Solving the Examination Timetabling Problem, Tese de Doutorado, 1998;
- [100] Thompson, J., Dowsland, K., General Cooling Schedules for a Simulated Annealing Based Timetabling System, Burke, E., Ross, P. (eds.), First International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n^o1153, Springer, 1996, pp.345 - 363;
- [101] Trick, A., A Schedule-then-Break Approach to Sports Timetabling, in Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, 2000, pp. 340 - 352;
- [102] Tsang, E., Foundations of Constraint Satisfaction, Academic Press, 1993;
- [103] Turner, P., Genetic Algorithms and Multiple Distinct Solutions, Tese de Mestrado, University of Edinburgh, 1994;
- [104] Zitzler, E., Thiele, L., Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Study, in Eiben, A., Back, T., Schoenauer, M., Schwefel, H.-P. (eds), Fifth International Conference on Parallel Problem from Nature, 1998, pp 292 - 301;
- [105] Weert, A., Schreuder, J., Construction of Basic Schedules for Sports Competitions by Using Graph Theory, Burke, E., Carter, M., (eds.), Second International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n^o1408, Springer, 1998, pp. 201 - 210;
- [106] Westfall, P., Wolfinger, R., Closed Multiple Testing Procedures and PROC MULTTEST, White Paper, SAS, 1996;
- [107] White, G., Zhng, J., Generating Complete University Timetables by Combining Tabu Search with Constraint Logic, in Burke, E., Carter, M., (eds.), Second International Conference in Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science n^o1408, Springer, 1998 pp. 187 - 198;

- [108] White, G., Xie, B., Examination Timetables and Tabu Search With Longer Term Memory, in Burke, E., Erben, W., (eds), Proceedings of the Third International Conference of the Practice and Theory of Automated Timetabling, Germany, 2000, pp. 184-201;
- [109] Wren, A., Scheduling, Timetabling and Rostering - A Special Relationship, Burke, E., Ross, P. (eds.), First International Conference in Praticce and Theory of Automated Timetabling, Lecture Notes in Computer Science n°1153, Springer, 1996, pp. 46 - 75;
- [110] Wright, J., Loosemore, H., An Infeasibility Objective for Use in Constrained Pareto Optimization, in Zitzler, E., Deb, K., Thiele, L., Coello, C., Corne, D., First International Conference in Evolutionary Multi-criterion Optimization, Lecture Notes 1993, Springer, 2001, pp. 257-268;