

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Atribuição automática de estudantes universitários a turmas baseada em otimização multicritério

Gustavo Teixeira Nunes da Silva



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Dr. João Carlos Pascoal Faria

22 de julho de 2017

Atribuição automática de estudantes universitários a turmas baseada em otimização multicritério

Gustavo Teixeira Nunes da Silva

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Dr. João Carlos Pascoal Faria

22 de julho de 2017

Resumo

Na FEUP, como em qualquer instituição académica, existe a necessidade de dividir os estudantes por diferentes turmas de modo a reduzir a quantidade de alunos nas aulas práticas. Antes do início de cada ano letivo, o número de turmas e os seus horários de aula são definidos; depois, os alunos devem inscrever-se nelas de forma a compor o seu horário semanal. Porém, este processo apresenta vários desafios, visto que os alunos têm diferentes preferências relativamente às horas em que frequentam as aulas e aos docentes que as lecionam. O processo atual está dividido em várias fases, sendo que na primeira os estudantes são priorizados pelo seu desempenho académico e atribuídos às turmas que pretendem. No entanto, no fim, existe uma percentagem considerável de alunos que ficam sem horário, sendo preciso recorrer a uma segunda fase, considerada bastante problemática, em que os estudantes são colocados por ordem de chegada. Existe ainda uma terceira fase, na qual os estudantes têm de recorrer à secretaria.

Neste trabalho, para minorar os inconvenientes do atual sistema de inscrição nas turmas, foi concebido e implementado um novo método de inscrição baseado em otimização multicritério com programação linear. Este sistema tenta colocar estudantes em turmas com base nos seguintes objetivos: maximizar o número de colocações, satisfazer as preferências dos estudantes (com prioridade a estudantes com média académica mais elevada), maximizar a qualidade dos horários resultantes e garantir o equilíbrio das turmas no que respeita à sua ocupação.

O novo método foi implementado através de um programa em Java que recorre ao motor IBM ILOG CPLEX para a resolução do problema de otimização. Os dados de entrada são extraídos a partir do sistema de informação da FEUP, o SIGARRA, através de *scripts* produzidos pelo pessoal administrativo.

Comparando os dados da atribuição de turmas de unidades curriculares obrigatórias do MIEIC do 2.º semestre do ano letivo de 2016/2017 do processo atual com os do novo algoritmo concebido, conseguiu-se um aumento da taxa de colocação dos estudantes de 76% para 100%, sendo que 86% ficaram colocados numa das suas opções. No caso dos estudantes colocados fora das suas opções, o algoritmo tenta maximizar a qualidade do horário resultante da sua colocação. A taxa total de subocupação (desvio em relação a uma ocupação equitativa) das turmas foi ainda reduzida de 14% para 3%.

Abstract

At FEUP, as in any academic institution, there's a need to section students between different teaching groups in order to reduce the number of students in each practical class. Before the start of each school year, the number of groups and their class schedules are set. Then, students must enroll in them in order to make up their individual weekly schedule. This process poses several challenges, however, since students have different preferences regarding which times they want to go to class at and which professors they are assigned to. The current process is divided in multiple phases, the first of which has students being prioritized according to their academic performance and then assigned to the groups they want. Nonetheless, at the end there's still a considerable percentage of unassigned students, which leads into a second phase—which is considered to be quite problematic—where students are assigned on a first come, first served basis. There's yet a third phase in which students must go to the administration office.

With this work, in order to minimize the problems with the current group enrollment method, a new multicriteria optimization with linear programming-based enrollment system was developed and implemented. This system attempts to assign students to groups based on the following objectives: maximize the number of assignments, fulfill students' preferences (prioritizing students with a higher academic grade), maximize the quality of the resulting weekly schedules and keep groups balanced in regard to the number of assigned students.

The new method was implemented with a Java program which uses the IBM ILOG CPLEX engine to solve its optimization problem. The input data is extracted from FEUP's information system SIGARRA using scripts the administrative staff created.

Comparing data about the assignment of groups of MIEIC mandatory courses in the 2nd semester of the 2016/2017 school year from the current process with the one from the newly developed algorithm, the student assignment rate increased from 76% to 100%, of which 86% were assigned to one of the exact preferences they had chosen. For students who are not assigned to one of their preferences, the algorithm tries to maximize the quality of their resulting schedule. The total rate of group underutilization (deviation from uniform utilization) was also reduced from 14% to 3%.

Conteúdo

Introdução.....	1
1.1	Motivação 1
1.2	Objetivos e resultados esperados 2
1.3	Metodologia 3
1.4	Estrutura do documento 3
Análise do problema.....	5
2.1	Necessidade de separação de alunos por turmas 5
2.2	Processo de inscrição em unidades curriculares..... 6
2.3	Processo de escolha de turmas 7
2.3.1	Primeira fase..... 7
2.3.2	Segunda fase 11
2.3.3	Terceira fase 11
2.4	Limitações e problemas do sistema atual 12
2.4.1	Primeira fase..... 12
2.4.2	Segunda fase 12
Análise do estado da arte	15
3.1	Análise de complexidade computacional 15
3.2	UniTime 16
3.2.1	Seccionamento em massa 17
3.2.2	Seccionamento em tempo real 18
3.2.3	Avaliação da ferramenta 18
3.3	Formulação como problema de programação linear 18
3.4	<i>Software</i> de resolução de problemas de programação linear 19
3.5	Outras referências consultadas..... 20
Conceção e implementação do novo sistema	21
4.1	Conceptualização de um novo processo de colocação 21
4.2	Critérios de avaliação de resultados 22
4.3	Arquitetura e tecnologias 23
4.4	Dados de entrada..... 24

4.4.1	Escolhas introduzidas pelos estudantes no SIGARRA	24
4.4.2	Médias académicas dos estudantes	25
4.4.3	Lista de unidades curriculares	25
4.4.4	Turmas pertencentes a cada unidade curricular.....	26
4.4.5	Compostos de turmas e as turmas que os constituem.....	26
4.4.6	Horário semanal das turmas das unidades curriculares	26
4.5	Dados de saída.....	27
4.5.1	Colocações dos estudantes.....	27
4.5.2	Estatísticas de ocupação das turmas.....	27
4.5.3	Estatísticas por unidade curricular	28
4.6	Variáveis e restrições	28
4.6.1	Notação.....	28
4.6.2	Variáveis de decisão.....	29
4.6.3	Restrições.....	29
4.7	Critérios de otimização.....	32
4.7.1	Maximizar pares aluno-UC colocados	33
4.7.2	Maximizar alunos com colocação completa	33
4.7.3	Maximizar <i>timeslots</i> totais ocupados	34
4.7.4	Maximizar preferências satisfeitas.....	35
4.7.5	Minimizar blocos ocupados para alunos colocados fora das preferências	37
4.7.6	Minimizar blocos não pretendidos para alunos colocados fora das preferências	38
4.7.7	Minimizar subocupações de turmas.....	38
4.8	Detalhes da implementação.....	39
Validação experimental.....		43
5.1	Comparação de estatísticas entre ambos os processos.....	43
5.1.1	Colocação em unidades curriculares optativas.....	43
5.1.2	Colocação em unidades curriculares obrigatórias.....	45
5.2	Apresentação e validação presencial com <i>stakeholders</i>	49
Conclusão.....		51
6.1	Avaliação do trabalho desenvolvido	51
6.2	Trabalho futuro.....	51
6.2.1	Generalização do processo para diferentes cursos da FEUP	52
6.2.2	Recolha de preferências de turmas mais abrangentes	52
6.2.3	Integração direta com o SIGARRA	52
Referências		53

Lista de figuras

Figura 1. Processo faseado de escolha de turmas	7
Figura 2. Interface do SIGARRA para seleção das unidades curriculares para as quais o estudante pretende obter turma	8
Figura 3. Interface para a seleção de opções de inscrição em turmas	9
Figura 4. Introdução do número de vagas por turma por parte do administrador, antes da execução do processo de colocação	10
Figura 5. Lista apresentada durante a segunda fase de colocação, com indicação das turmas atribuídas e opção para alterar a colocação, indicando as vagas restantes	11
Figura 6. Conceptualização de alto nível do novo processo de colocação que seria desenvolvido	22
Figura 7. Esquema que ilustra o fluxo de dados entre todas as entidades relacionadas com o novo sistema	24
Figura 8. Excerto de código que representa a criação de uma variável de decisão com a biblioteca Java do CPLEX	39
Figura 9. Excerto de código que representa a definição de uma restrição com a biblioteca Java do CPLEX	39
Figura 10. Excerto de código que representa a criação de uma expressão linear e, posteriormente, a adição de um termo	40
Figura 11. Excerto de código que representa a definição de uma restrição condicional	40
Figura 12. Excerto de código que representa a definição da função-objetivo com a biblioteca Java do CPLEX	40
Figura 13. Excerto de código que representa a configuração do parâmetro de tempo-limite do CPLEX	41
Figura 14. Excerto de código que representa a chamada ao método de resolução do problema já formulado no CPLEX	41
Figura 15. Excerto de código que representa a forma de obtenção do valor final de uma variável	41

Figura 16. Satisfação de preferências dos estudantes no processo atual de atribuição de turmas de unidades curriculares optativas	44
Figura 17. Satisfação de preferências dos estudantes no novo processo de atribuição de turmas de unidades curriculares optativas	44
Figura 18. Satisfação de preferências dos estudantes no processo atual de atribuição de turmas de unidades curriculares obrigatórias	45
Figura 19. Satisfação de preferências dos estudantes no novo processo de atribuição de turmas de unidades curriculares obrigatórias	46
Figura 20. Taxas de subocupação por turma no processo atual de atribuição de turmas de unidades curriculares obrigatórias	47
Figura 21. Taxas de subocupação por turma no novo processo de atribuição de turmas de unidades curriculares obrigatórias	47

Lista de tabelas

Tabela 1. Estatísticas de colocação em turmas de estudantes do MIEIC na primeira fase do 2.º semestre do ano letivo de 2016/2017	12
Tabela 2. Indicadores de qualidade do novo processo de colocação e comparação com a abordagem do processo atual	23
Tabela 3. Critérios de otimização considerados na função-objetivo	32
Tabela 4. Estatísticas globais comparando o processo atual e o novo processo de atribuição de turmas de unidades curriculares obrigatórias	48

Abreviaturas e símbolos

API	Interface de programação de aplicações (do inglês <i>application programming interface</i>)
ECTS	Sistema Europeu de Transferência e Acumulação de Créditos (do inglês <i>European Credit Transfer and Accumulation System</i>). Esta sigla é habitualmente utilizada não com o sentido de “sistema de créditos”, mas para designar cada crédito individual
FEUP	Faculdade de Engenharia da Universidade do Porto
MIEIC	Mestrado Integrado em Engenharia Informática e Computação
SIGARRA	Sistema de Informação para a Gestão Agregada dos Recursos e dos Registos Académicos, o sistema de informação utilizado na Universidade do Porto
UC	Unidade curricular

Capítulo 1

2 Introdução

1.1 Motivação

4 O tema desta dissertação foi proposto pela própria Faculdade de Engenharia da
6 Universidade do Porto, visto que o problema abordado é de grande relevância para os
docentes e alunos que a frequentam.

8 Todos os semestres, existe a necessidade de separar os estudantes de uma dada
unidade curricular por diferentes turmas, visto que as salas de aula frequentemente não
10 possuem capacidade para todos os alunos inscritos e há o interesse em realizar as aulas
práticas em turmas de dimensão moderada. Mesmo antes do início de cada ano letivo,
define-se o número de turmas e os horários das aulas de cada unidade curricular, sendo
12 que, depois, no começo de cada semestre, os alunos devem inscrever-se em turmas de
forma a obterem um horário semanal. O processo atual de inscrição nas turmas
14 apresenta alguns obstáculos tanto para alunos como para docentes. Como não se
consegue fazer a atribuição de todos os estudantes numa única fase, recorre-se a uma
16 segunda fase que traz diversos inconvenientes.

18 Na perspetiva dos alunos, o facto de o único critério de colocação da segunda fase
ser a ordem de chegada traduz-se em três principais problemas: por um lado, os
estudantes são obrigados a estar em frente ao computador na hora exata em que a fase
20 se inicia, atualizando a página constantemente para conferir eventuais alterações nas
vagas; por outro, e como consequência do anterior, o sistema fica visivelmente mais lento
22 e inconstante, aumentando ainda mais o fator sorte que já é inerente a esta fase. Por
último, o critério da segunda fase de apenas respeitar a ordem de chegada dos pedidos
24 não é o mais desejável, pois abandona por completo a ideia da primeira fase de beneficiar
os estudantes com médias de classificação mais elevadas e menos unidades curriculares
26 em atraso, deixando o desfecho nas mãos de quem, por exemplo, possua uma ligação

Introdução

28 mais rápida ou a quem, por acaso, tenha a sorte de encontrar uma vaga que fora dada
instantes antes.

30 Por outro lado, o sistema atual implica que um administrador de sistema esteja
constantemente a monitorizar o processo e a adicionar novas vagas a turmas (desde que
32 não excedam a capacidade máxima previamente definida) quando já restarem poucas,
de modo a evitar que, caso as turmas disponibilizem imediatamente o número máximo
34 de vagas, exista uma grande afluência de estudantes para uma ou duas turmas e que, no
final, haja uma desigualdade considerável no balanço de alunos por turma. Porém,
apesar desta precaução, ainda continuam a existir casos de turmas pouco equilibradas,
36 o que causa desagrado aos docentes das unidades curriculares.

1.2 Objetivos e resultados esperados

38 Com este trabalho, pretendia-se desenvolver e validar um sistema informático que
conseguisse, numa única fase, colocar um número significativo de alunos (idealmente
40 todos) em turmas sem conflitos de horários e que, ao mesmo tempo, fosse capaz de
mitigar os problemas que se verificam na segunda fase do processo atual, como descrito
42 anteriormente. Mesmo que a solução ideal não fosse atingível, pretendia-se, acima de
tudo, que o novo sistema trouxesse melhorias e resolvesse vários dos problemas
44 evidenciados no processo atual:

- 46 • A atribuição de horários na primeira fase ao maior número possível de
estudantes, de forma a não se depender de uma segunda fase onde o critério de
colocação seja apenas a ordem de chegada e a não obrigar à existência de um
48 funcionário administrativo que acompanhe o processo e abra manualmente
mais vagas em turmas;
- 50 • O balanceamento automático de alunos por turma e a não excedência da
capacidade das mesmas;
- 52 • Uma satisfação global maior ou, pelo menos, similar por parte dos estudantes
com o seu horário final comparativamente ao que obtêm com o processo atual.

54 Tendo o novo sistema desenvolvido, seria necessário proceder à validação da nova
solução. No entanto, antes de se proceder à validação propriamente dita, foi preciso
56 definir os critérios com que avaliámos uma dada solução ou como efetuámos a avaliação
comparativa de duas soluções (uma antiga e uma nova), como, por exemplo, o grau de
58 satisfação das preferências de cada estudante (com prioridade para estudantes com
melhor classificação), a taxa de colocação dos estudantes, a uniformidade da distribuição
60 por turmas, etc.

Introdução

62 Por fim, o sistema final seria, numa visão ideal, integrado por API no SIGARRA,
64 permitindo a introdução de preferências por parte dos estudantes diretamente no
66 sistema de informação, a utilização automática destas por parte da aplicação
68 desenvolvida no âmbito desta dissertação e a posterior devolução dos resultados finais
70 de volta para o sistema da faculdade, por forma a serem utilizados diretamente pelos
seus estudantes e docentes. No entanto, visto que a questão da integração completa com
o SIGARRA teria de ser sujeita a diversos pareceres e não seria imediata, pretendia-se,
em segunda hipótese, dotar o novo sistema da capacidade de funcionamento *offline*,
através da importação/exportação manual dos ficheiros que contêm os dados de entrada
e a atribuição de turmas.

1.3 Metodologia

72 O desenvolvimento deste trabalho começou, nos primeiros meses, por uma análise
ao estado da arte. Foi realizada uma pesquisa sobre problemas semelhantes e a sua forma
74 de resolução.

Após estudar o problema colocado e avaliar o estado da arte, decidiu-se abordar
76 esta questão como um problema de programação linear com otimização multicritério.
Para isso, foi realizado um desenvolvimento incremental, começando com uma
78 formulação extremamente básica (apenas as variáveis de decisão principais), testando e
iterando, e construindo a função-objetivo e as diferentes restrições ao longo do tempo.

80 A cada iteração, foram calculadas algumas estatísticas que permitiram avaliar a
qualidade de cada nova solução encontrada. No final do desenvolvimento do novo
82 sistema, foi recolhido um conjunto ainda mais extenso de indicadores de qualidade
relativamente a ambos os sistemas – tanto o novo como o atual – e, com eles, foi realizada
84 uma reunião com os principais *stakeholders* envolvidos no processo de atribuição de
turmas com o propósito de lhes apresentar o trabalho desenvolvido.

1.4 Estrutura do documento

86 Este relatório encontra-se dividido em seis capítulos. No primeiro, é feita uma
88 introdução resumida do problema que motivou este trabalho e são apresentados os
objetivos de alto nível que se pretendiam atingir. No segundo capítulo, faz-se uma
90 análise do problema de separação de alunos por turmas e de que forma este é atualmente
resolvido na FEUP. São apresentadas todas as características deste processo, desde a
92 perspectiva dos estudantes até à configuração do processo de colocação por parte do
administrador do sistema. São, ainda, salientados os pormenores específicos que levam
94 às consequências negativas que são observadas hoje em dia durante este processo. No

Introdução

terceiro capítulo, as referências consultadas são mencionadas, explicando-se, de uma
96 forma resumida, as especificidades dos problemas abordados em cada uma e a
abordagem que os seus autores aplicam para os solucionar. É feita uma comparação
98 entre cada problema estudado e o problema atual da FEUP. No quarto capítulo, é feita
uma extensa descrição do novo sistema produzido, desde a conceptualização criada
100 antes mesmo do seu desenvolvimento, passando por uma breve explicação de alto nível
sobre todos os componentes que o constituem, passando pelos dados de entrada
102 necessários e os dados de saída produzidos, explicando as variáveis de decisão,
restrições e objetivos utilizados na modelação do problema, recorrendo à ajuda de
104 fórmulas matemáticas, e, por fim, mostrando alguns detalhes concretos sobre a
implementação deste *software*. No quinto capítulo, apresentam-se os resultados
106 experimentais obtidos através da aplicação do novo sistema aos dados históricos de um
processo de atribuição de turmas anterior. Por fim, no sexto e último capítulo, é feito um
108 rescaldo do trabalho produzido com esta dissertação e são identificados alguns pontos
que poderão vir a ser melhorados no futuro.

110 **Capítulo 2**

Análise do problema

112 **2.1 Necessidade de separação de alunos por turmas**

114 Na FEUP, existem vários tipos de aulas para as unidades curriculares oferecidas. As
116 aulas teóricas requerem uma interação mínima entre alunos e professores, pelo que são
118 habitualmente lecionadas em auditórios que, pela sua dimensão e configuração espacial,
120 são capazes de comportar um número elevado de estudantes. Por esta razão, as aulas
122 teóricas de uma determinada unidade curricular são ministradas simultaneamente para
124 todos os alunos nela inscritos, ou seja, todas as turmas partilham a mesma ocorrência de
126 uma aula teórica. As restantes aulas (teórico-práticas, práticas laboratoriais, entre outras;
128 neste documento referidas coletivamente como “aulas práticas”) requerem uma atenção
130 mais individualizada e uma interação próxima entre estudantes e docente, pelo que
132 obrigam a limitar consideravelmente o número de alunos presentes na sala de aula, que,
134 por sua vez, é normalmente uma sala “tradicional” com uma capacidade
significativamente mais limitada do que os auditórios: habitualmente cerca de 24 alunos.
Desta forma, é necessário separar os alunos em diferentes turmas. Visto ser habitual um
mesmo docente lecionar mais do que uma destas aulas práticas por semestre e, outras
vezes, por razões de disponibilidade dos outros docentes, é muito frequente estas aulas
não decorrerem ao mesmo tempo para todas as turmas, o que introduz possíveis
conflitos com outras unidades curriculares. Por exemplo, um estudante que fique
colocado numa turma com uma aula prática à segunda-feira das 09:00 às 13:00
automaticamente não pode ficar colocado numa turma de outra unidade curricular que
possua uma aula prática que decorra, mesmo que parcialmente, ao mesmo tempo, visto
que todas as aulas para além das teóricas são de frequência obrigatória.

Assim, constata-se que não só se afigura necessário separar os alunos por diferentes turmas, como também este processo levanta alguns desafios. Um estudante precisa de

Análise do problema

136 ficar colocado em uma e só uma turma de cada unidade curricular que vai frequentar
138 num dado semestre, mas a escolha de uma turma específica vai, muito provavelmente,
140 limitar de alguma forma as turmas que poderá selecionar para as restantes unidades
curriculares. Isto aumenta consideravelmente a dificuldade da procura da “solução
ótima” para cada estudante, visto que uma má escolha inicial pode fazer toda a diferença
mais tarde.

142 **2.2 Processo de inscrição em unidades curriculares**

No início de cada ano letivo, os estudantes que não ingressam no 1.º ano pela
144 primeira vez (estes são colocados automática e aleatoriamente pela secretaria) são
chamados a inscrever-se nas unidades curriculares que pretendem frequentar. Pode
146 separar-se estas unidades em dois grupos principais: obrigatórias e optativas. As
unidades curriculares obrigatórias são comuns para todos os alunos desse ano
148 académico desse curso e, como o seu nome indica, os estudantes necessitam de obter
aprovação em todas elas para concluir o curso. As unidades optativas são descritas como
150 um dado número de ECTS que os estudantes precisam de obter. Para isso, estes
inscrevem-se em certas unidades curriculares, dentro de um leque de opções, que lhes
152 confirmam o número necessário de ECTS. Interessa realçar que, na FEUP, a atribuição
destas unidades optativas é realizada como um processo de atribuição de turmas: cada
154 optativa possui apenas uma turma e, se um aluno ficar colocado numa destas turmas,
significa que ficou também inscrito na optativa correspondente. Como os alunos
156 geralmente ainda não ponderaram as unidades curriculares optativas que pretendem
frequentar no momento de inscrição (início do ano letivo), estes escolhem não as
158 optativas em concreto, mas sim “unidades curriculares não discriminadas” com um
determinado número de ECTS. Este passo dá-se antes do processo de escolha de turmas
160 em si, cuja descrição será realizada nas secções seguintes.

2.3 Processo de escolha de turmas

162 Desde o início do processo de atribuição de turmas até à obtenção de um horário
completo por parte de todos os estudantes, existem alguns momentos diferentes,
164 designados de “fases”. Uma representação gráfica deste processo, incluindo o momento
anterior de inscrição nas unidades curriculares, é ilustrado na figura 1.



Figura 1. Processo faseado de escolha de turmas

166 Estas fases possuem características muito díspares umas das outras, como
poderemos constatar de seguida.

168 2.3.1 Primeira fase

No início da primeira fase, o estudante depara-se com uma lista de todas as
170 unidades curriculares abrangidas pelo processo de escolha de turmas. Esta escolha dá-
se em momentos diferentes consoante estejamos a considerar unidades curriculares
172 obrigatórias ou optativas, embora o processo se desenrole da mesma forma para ambas,
portanto as unidades que são mostradas nesta lista podem variar. O estudante seleciona,
174 então, aquelas nas quais pretende candidatar-se a uma turma: no caso das obrigatórias,
todos os estudantes selecionam as mesmas; no caso das optativas, os estudantes
176 escolhem todas aquelas em que estão interessados, embora só pretendam terminar o
processo com um subconjunto das mesmas. Na figura 2, pode ver-se como este passo é
178 representado na interface do utilizador do SIGARRA.

Análise do problema

Você está em: Início > Inscrições nas Turmas > Escolher Unidades Curriculares

Boas vindas
Órgãos de Gestão
Departamentos
Serviços
Estudantes
Pessoal
Legislação
Cursos
I&D e Inovação
Cooperação
Candidatos
Alumni
Empresas
Notícias
Pesquisa

Autenticação
Utilizador:
Gustavo Silva
Terminar sessão

Mapa das Instalações

Exposição TRABALHOS DA OFICINA DE PINTURA DA FEUP 13 DE JANEIRO A 3 DE FEVEREIRO DE 2017

ATALHOS
Ver Lista
Adicionar Página

Inscrições nas turmas 2º Semestre - 2016/2017: MIEIC - Mestrado Integrado em Engenharia Informática e Computação

Escolha de unidades curriculares de Gustavo Teixeira Nunes da Silva (201208205)

Selecione as unidades curriculares a que se quer inscrever nas turmas

Unidade Curricular			Selecionar
Ano	Nome	Código	
1	Complementos de Matemática	EIC0009	<input type="checkbox"/>
1	Física I	EIC0010	<input type="checkbox"/>
1	Métodos Estatísticos	EIC0105	<input type="checkbox"/>
1	Microprocessadores e Computadores Pessoais	EIC0016	<input type="checkbox"/>
1	Programação	EIC0012	<input type="checkbox"/>
4	Arquitectura de Sistemas de Software	EIC0048	<input type="checkbox"/>
4	Arquitectura e Gestão de Redes e Sistemas	EIC0095	<input type="checkbox"/>
4	Computação Paralela	EIC0089	<input type="checkbox"/>
4	Comunicações Móveis	EIC0082	<input type="checkbox"/>
4	Desenvolvimento de Jogos de Computador	EIC0090	<input type="checkbox"/>
4	Engenharia de Requisitos de Sistemas de Software	EIC0053	<input type="checkbox"/>
4	Gestão de Operações e Logística	EIC0091	<input type="checkbox"/>
4	Linguagens de Anotação e Processamento de Documentos	EIC0107	<input type="checkbox"/>
4	Marketing	EIC0061	<input type="checkbox"/>
4	Modelação e Simulação de Sistemas	EIC0098	<input checked="" type="checkbox"/>
4	Multimédia e Novos Serviços	EIC0064	<input type="checkbox"/>
4	Segurança em Sistemas Informáticos	EIC0072	<input checked="" type="checkbox"/>
4	Sistemas Embutidos e de Tempo Real	EIC0102	<input checked="" type="checkbox"/>
4	Tecnologias de Bases de Dados	EIC0076	<input checked="" type="checkbox"/>
4	Tecnologias de Distribuição e Integração	EIC0077	<input checked="" type="checkbox"/>

Continuar

Página gerada em: 2017-01-26 às 18:07:31

Figura 2. Interface do SIGARRA para seleção das unidades curriculares para as quais o estudante pretende obter turma

180 Seguidamente, o aluno é direcionado para uma nova página, onde este pode
consultar os critérios de prioridade de alocação, algumas informações sobre o processo
182 e uma lista em branco, que deverá ser preenchida com as diferentes combinações de
turmas que o estudante deseja. Uma dada combinação representa um horário. Na
184 primeira opção, o estudante indica o seu horário ideal; na seguinte, este introduz um
horário alternativo que pretende que o sistema considere caso não seja possível satisfazer
a sua primeira opção e assim sucessivamente, até um máximo de 10 opções. Na parte
186 inferior da página, é mostrada uma representação visual do horário semanal que
corresponde às unidades curriculares selecionadas para a opção que está a ser
188 preenchida no momento, na qual conflitos que impeçam a presença numa aula teórica
são assinalados a amarelo e aqueles que impeçam a frequência de uma aula prática são
190 destacados a vermelho. As aulas sem conflitos são representadas a verde. Na figura 3,
demonstra-se a seleção de algumas unidades curriculares, na primeira opção, que não
192 resultam em qualquer conflito de horário.

Análise do problema

Você está em: Início > Cursos/CE > MIEIC > Inscrições nas turmas > Escolha de turmas

Boas vindas
Órgãos de Gestão
Departamentos
Serviços
Estudantes
Pessoal
Legislação
Cursos
I&D e Inovação
Cooperação
Candidatos
Alumni
Empresas
Notícias
Pesquisa

Autenticação
Utilizador: Gustavo Silva
Terminar sessão

Mapa das Instalações

Inscrições nas turmas 2º Semestre - 2016/2017: MIEIC - Mestrado Integrado em Engenharia Informática e Computação

Critérios definidos para o processo de alocação:

Ordem	Critério
1º	Média normal
2º	data/hora da primeira inscrição em cada opção
3º	Numero da opção

Se já estiver colocado numa turma a determinada unidade curricular e quiser manter esta colocação não deve alterar a opção. Colocar uma opção nula (-) implica a eliminação da atribuição de turma a essa unidade curricular (no caso de já ter sido colocado).

Os valores à frente de cada turma representam as vagas existentes no momento em que esta página foi carregada.

Cada opção equivale a um horário.

Selecione as turmas a que se pretende inscrever

Ano	Unidade Curricular	Código	Atribuída	Opção 1
4	Tecnologias de Bases de Dados	EIC0076		4MIEIC01
4	Tecnologias de Distribuição e Integração	EIC0077		4MIEIC01
4	Segurança em Sistemas Informáticos	EIC0072		4MIEIC02
4	Modelação e Simulação de Sistemas	EIC0098		4MIEIC01

Concluir Reset Seguinte >>

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
08:00						
08:30						
09:00			SSIN (8:00 - 11:00) 4MIEIC02 (TP) B012		TBDA (9:00 - 12:00) 4MIEIC01 (TP) B305	GTD
09:30						
10:00						
10:30						
11:00						
11:30						
12:00						
12:30						
13:00						
13:30						
14:00						
14:30						
15:00	TDIN (14:00 - 17:00) 4MIEIC01 (TP) B313					
15:30		APM				
16:00						
16:30						
17:00						
17:30						
18:00				MSSI (16:30 - 19:30) 4MIEIC01 (TP) B313		RR
18:30						
19:00						
19:30						
20:00						

Página gerada em: 2017-01-26 às 18:05:55

Figura 3. Interface para a seleção de opções de inscrição em turmas

194 Como já referido, é ainda possível observar-se os critérios de colocação e a sua
196 ordem. Estes critérios destinam-se a criar uma lista ordenada de estudantes que será
percorrida sequencialmente pelo algoritmo quando esta fase estiver concluída e as
turmas estiverem a ser geradas.

198 Depois de introduzidas todas as opções desejadas, o estudante conclui este passo,
envia a sua inscrição em turmas e é-lhe apresentado um ecrã de confirmação de
submissão com as suas escolhas. O aluno pode sempre visitar a página anterior e fazer
200 as alterações que pretender às suas escolhas até à hora de fecho desta fase. As datas e
horas de início e conclusão de cada fase são previamente comunicadas aos estudantes

Análise do problema

202 por correio eletrónico e são indicadas também no SIGARRA, muito embora sejam
 204 meramente indicativas e não aplicadas na prática pelo sistema, pois a abertura e o fecho
 206 são despoletados manualmente por um administrador. Este fica encarregado, ainda, de
 definir o número de vagas existentes por turma para colocação de alunos, num painel
 como se pode verificar na figura 4.

Ano	Unidade curricular		Turma			Turma			Turma			Turma			Turma			Total de Vagas	Desfazer	Gravar				
	Nome	Código	Nome	Util.	Vagas Disp.				Util.	Disp.														
1	Complementos de Matemática	EIC0009	IMEIC01	18	999	IMEIC02	20	999	IMEIC03	20	999	IMEIC04	21	999	IMEIC05	19	999	IMEIC06	17	999	115	5994		
1	Física I	EIC0010	IMEIC01	19	999	IMEIC02	20	999	IMEIC03	20	999	IMEIC04	21	999	IMEIC05	19	999	IMEIC06	17	999	116	5994		
1	Métodos Estatísticos	EIC0105	IMEIC01	18	999	IMEIC02	20	999	IMEIC03	20	999	IMEIC04	21	999	IMEIC05	19	999	IMEIC06	16	999	114	5994		
1	Microprocessadores e Computadores Pessoais	EIC0016	IMEIC01	19	999	IMEIC02	20	999	IMEIC03	20	999	IMEIC04	21	999	IMEIC05	19	999	IMEIC06	16	999	115	5994		
1	Programação	EIC0012	IMEIC01	19	999	IMEIC02	20	999	IMEIC03	20	999	IMEIC04	21	999	IMEIC05	19	999	IMEIC06	17	999	116	5994		
4	Arquitetura de Sistemas de Software	EIC0048	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Arquitetura e Gestão de Redes e Sistemas	EIC0095	AMEIC02	0	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	20		
4	Computação Paralela	EIC0089	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Comunicações Móveis	EIC0082	AMEIC02	0	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	12		
4	Desenvolvimento de Jogos de Computador	EIC0090	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Engenharia de Requisitos de Sistemas de Software	EIC0053	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Gestão de Operações e Logística	EIC0091	AMEIC02	0	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Linguagens de Anotação e Processamento de Documentos	EIC0107	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Marketing	EIC0061	AMEIC02	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Modelação e Simulação de Sistemas	EIC0098	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Multimédia e Novos Serviços	EIC0064	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Segurança em Sistemas Informáticos	EIC0072	AMEIC02	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Sistemas Embebidos e de Tempo Real	EIC0102	AMEIC02	0	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	8		
4	Tecnologias de Bases de Dados	EIC0076	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		
4	Tecnologias de Distribuição e Integração	EIC0077	AMEIC01	0	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0		

Figura 4. Introdução do número de vagas por turma por parte do administrador, antes da execução do processo de colocação

208 Nesta fase, tal como mencionado no parágrafo anterior, o algoritmo percorre a lista
 210 de estudantes, já ordenados pelos critérios pré-determinados, e verifica, começando pela
 212 sua primeira opção, se ainda existem vagas nas turmas que cada aluno escolheu. Em
 214 caso afirmativo, este fica com essa atribuição; caso contrário, é verificada a sua segunda
 opção e assim sucessivamente. Se nenhuma das suas opções for válida, o estudante fica
 sem atribuição e terá de recorrer às fases seguintes para obter um horário. Considera-se
 válida uma opção unicamente caso existam vagas em todas as turmas nela selecionadas.

216 Mesmo que apenas uma das turmas não possua vagas remanescentes, toda a opção é
 descartada e é considerada a opção seguinte.
 218 Após a execução do algoritmo, o administrador analisa os resultados, geralmente
 verificando a quantidade de vagas totais vs. vagas preenchidas em cada turma. Se
 220 considerar necessário, para tentar aumentar o número de estudantes colocados, pode
 ajustar as vagas e refazer o processo. Senão, o processo é dado como concluído e os
 222 resultados são oficialmente publicados. Os estudantes devem, então, consultar os seus
 horários e verificar se obtiveram uma colocação. Aqueles que não a tenham e os que
 pretendam fazer ajustes devem recorrer à segunda fase.

2.3.2 Segunda fase

224 De um ponto de vista de interface, a segunda fase é muito semelhante à primeira.
 226 Normalmente, como todos os alunos que participam nesta fase já concorreram na
 primeira, o SIGARRA não volta a questionar as unidades curriculares a que estes se
 228 pretendem candidatar, passando logo para o passo seguinte. No entanto, é sempre
 possível alterar esta escolha através de uma opção no menu lateral direito da página. A
 principal diferença é na lista de atribuições, que, desta vez, em vez de apresentar até 10
 230 opções de turmas, apresenta uma coluna com a turma que se encontra atualmente
 atribuída e uma outra coluna com as turmas e o número de vagas ainda restantes em
 232 cada uma no momento em que a página foi carregada. Esta lista é apresentada na figura
 5.

Unidade Curricular			-	
Ano	Nome	Código	Atribuída	Nova (Vagas)
3	Inteligência Artificial	EIC0029		3MIEIC02 (0) ▾
3	Proficiência Pessoal e Interpessoal	EIC0031		3MIEIC06 (0) ▾
3	Sistemas Distribuídos	EIC0036		3MIEIC01 (1) ▾
3	Laboratório de Bases de Dados e Aplicações Web	EIC0085		3MIEIC01 (1) ▾

Figura 5. Lista apresentada durante a segunda fase de colocação, com indicação das turmas atribuídas e opção para alterar a colocação, indicando as vagas restantes

234 Os estudantes podem realizar os ajustes que pretenderem e, após enviarem o
 formulário, ficam imediatamente colocados nas turmas selecionadas, desde que ainda
 236 existam vagas em todas elas no momento da submissão.

238 Porém, há ainda alunos que, por algum motivo especial ou apenas por razões de
 conflitos de horário, não conseguem obter um horário completo no final desta fase. Estes
 casos são pontuais, mas necessitam de avaliação manual durante a terceira fase.

240 2.3.3 Terceira fase

A terceira fase é destinada apenas aos poucos casos de alunos que ainda precisam
 242 de fazer ajustes ao seu horário devido a razões especiais e, deste modo, dirigem-se à
 secretaria e expõem a sua situação a um funcionário, que tratará de ponderar a melhor
 244 opção e atribuir ao estudante manualmente as turmas de que necessite.

246 Finda esta fase, todos os estudantes possuem horários completos e podem começar
 a frequentar as aulas normalmente.

2.4 Limitações e problemas do sistema atual

2.4.1 Primeira fase

O principal problema que se verifica no processo atual de atribuição de turmas é, de facto, uma taxa de colocação de estudantes no final da primeira fase mais baixa do que seria ideal. Esta é a fase que se pode considerar mais “justa”, pois dá prioridade aos estudantes com maior sucesso académico, mas não é difícil perceber que esta é uma abordagem “ingénua” se pretendermos colocar o maior número de estudantes de uma só vez: o que acontece é que os alunos com maior prioridade no processo acabam por ocupar vagas que vão, mais tarde, impedir outros estudantes de obter o seu horário. Como referência, podem consultar-se, na tabela 1, estatísticas relativas à primeira fase no 2.º semestre do presente ano letivo.

Tabela 1. Estatísticas de colocação em turmas de estudantes do MIEIC na primeira fase do 2.º semestre do ano letivo de 2016/2017

Estudantes colocados	321	~76%
Estudantes sem colocação	104	~24%
Total de candidatos nesta fase	425	100%

258

Os estudantes que acabam sem colocação ficam sujeitos a um processo consideravelmente mais injusto, como poderá ser verificado na subsecção seguinte.

260

2.4.2 Segunda fase

A partir do momento em que se inicia a segunda fase, os alunos podem colocar-se a si próprios nas turmas que pretenderem, desde que exista pelo menos uma vaga disponível. Sendo que o único critério é a ordem de chegada, põe-se por completo de parte o interesse que existe na fase anterior de recompensar o sucesso académico dos estudantes, o que pode tornar-se mais injusto para os alunos intermédios que tenham ficado sem horário “por pouco” e que se veem obrigados a competir com os alunos com médias inferiores por vagas nesta fase.

Além disso, como outra consequência do critério de ordem de chegada, acontece que os estudantes tentam todos ser os primeiros para conseguir as melhores vagas. Como consequência, na hora de abertura da segunda fase, é notável um abrandamento significativo do SIGARRA, o que faz com que as páginas demorem vários segundos a carregar, prejudicando a experiência de todos os utilizadores.

272

Análise do problema

274 Por outro lado, de modo a garantir que os estudantes não preenchem rapidamente
todas as vagas para uma dada turma, possivelmente causando um desequilíbrio entre
276 turmas e/ou impedindo, mais tarde, certos alunos de conseguirem um horário livre de
conflitos, esta segunda fase obriga a intervenção manual por parte de um administrador,
278 que necessita de ir aumentando as vagas à medida que estas vão diminuindo e a
composição das turmas vai estabilizando, para permitir que mais estudantes possam
280 entrar nelas. Isto significa, como é óbvio, que, mesmo no início desta fase, as turmas não
são configuradas com a sua capacidade máxima absoluta.

282 Capítulo 3

Análise do estado da arte

284 3.1 Análise de complexidade computacional

286 Em Dostert *et al.* [1], é feito um estudo de complexidade à separação de estudantes
por turmas dados horários já existentes. Por “horários já existentes”, neste artigo, os
autores referem-se ao facto de não ser necessário efetuar nenhum escalonamento de
288 tarefas, ou seja, os tempos de aulas já se encontram pré-definidos e só é preciso atribuir
estudantes a estas aulas. É este o problema que se adequa à atual realidade da FEUP e
290 que se pretende estudar, pois aqui não se contempla a hipótese de fazer alterações aos
tempos de aulas.

292 Os autores formulam este problema como um problema de emparelhamento em
grafos bipartidos e iniciam descrevendo os seus modelos em mais detalhe: em primeiro
294 lugar, é apresentado o que estes chamam de “problema básico de divisão de alunos em
turmas” (*basic student sectioning problem*, ou BSS), no qual todos os estudantes estão
296 inscritos em todas as unidades curriculares consideradas. Cada unidade é composta por
algumas turmas e cada turma possui uma capacidade fixa e um tempo de aula pré-
298 atribuído. Pretende-se, então, determinar, o número máximo de estudantes que podem
ser atribuídos a uma turma de cada unidade curricular de tal forma que:

- 300 a) Cada estudante tenha, no máximo, uma aula por período de tempo; e
- b) Não seja excedida a capacidade máxima de nenhuma turma.

302 Os autores consideram que estas são as duas restrições mais importantes, pois
refletem o principal conflito de interesses neste problema: os estudantes quererem um
304 horário sem conflitos e as características físicas dos espaços académicos obrigam a
separar o elevado número de alunos por diferentes turmas.

306 Os autores admitem, no entanto, que, muito embora este modelo se aplique em
alguns casos, na prática, podem surgir mais restrições. Assim, estudam ainda modelos

308 de BSS com mais três características, sendo que as duas seguintes são importantes para
o problema desta dissertação:

310 a) Os estudantes escolhem as suas unidades curriculares a partir de uma lista
maior de unidades oferecidas, tendo esta escolha de ser respeitada pelo
312 algoritmo;

314 b) As turmas podem ter mais do que uma aula por período de tempo, sendo
que o horário final resultante da colocação de cada aluno deve estar livre de
conflitos entre todas as aulas de todas as turmas.

316 A primeira afigura-se relevante, em primeiro lugar, dado que os alunos nem sempre
se encontram a frequentar o conjunto completo de unidades curriculares de um dado
318 ano do curso (devido a reprovações, melhorias de classificação e/ou mudanças de curso)
e, depois, caso queiramos expandir o algoritmo para incluir unidades curriculares
320 optativas. A segunda restrição também é fulcral, na medida em que, na sua grande
maioria, as unidades curriculares possuem pelo menos uma aula teórica e uma aula
322 prática.

Nas secções seguintes do documento, os autores concluem que o problema básico
324 de BSS é computacionalmente eficiente (ou, mais concretamente, pode ser resolvido em
tempo polinomial). No entanto, explorando qualquer uma das restrições adicionais (que
326 são fulcrais para o problema da FEUP), os autores determinam que o problema se torna
NP-completo. Isto significa que não há garantias de que seja possível determinar a
328 solução ótima em tempo polinomial e que, com grande certeza, não será possível
alcançar a melhor colocação possível na FEUP, visto que o nosso problema introduz
330 ainda mais restrições, sendo talvez a principal o desejo de solicitar e respeitar as
preferências dos estudantes.

332 **3.2 UniTime**

Tomáš Müller possui um vasto trabalho de investigação na área de geração de
334 horários e seccionamento de estudantes por turmas. É o autor do *software* de código
aberto UniTime [2], que consiste numa aplicação cliente-servidor com uma interface web
336 que facilita diversos processos de agendamento de tarefas comuns em ambientes
académicos, seja a geração dos horários, a atribuição de turmas a estudantes ou o
338 planeamento de datas de testes ou outros eventos. É uma aplicação distribuída, pelo que
a tarefa de processamento pode ser relegada para diferentes clientes de forma a tornar o
340 processo mais rápido. É, também, utilizada na prática na Universidade de Purdue, onde
gere milhares de estudantes, unidades curriculares e aulas. O UniTime recorre a uma
342 biblioteca de Java para resolução de problemas de otimização com restrições criada pelo

mesmo autor. No estudo realizado no contexto desta dissertação, vamos focar apenas a
344 componente de atribuição de turmas a alunos.

Em Müller *et al.* [3], detalha-se a abordagem tomada por este *software* ao
346 seccionamento de estudantes. Existem três fases de processamento: o seccionamento
durante a fase de geração de horários para as unidades curriculares, um seccionamento
348 em massa realizado após os horários estarem definidos e uma fase de atribuição de
turmas em tempo real para facilitar a realização de pequenos ajustes nas turmas
350 atribuídas. A primeira destas fases não será alvo de estudo deste trabalho, visto que o
processo de geração de horários na FEUP é um problema já resolvido tradicionalmente
352 de uma forma específica e não se espera vir a ser alterado em tão curto prazo. Assim, nas
subsecções seguintes, descreve-se a forma como cada uma das restantes duas fases é
354 processada.

3.2.1 Seccionamento em massa

356 Após a definição do horário de todas as turmas de todas as unidades curriculares,
executa-se um processo de seccionamento em massa. Um detalhe que é importante
358 sublinhar relativamente à relevância deste *software* para resolver o problema da FEUP é
que este não é tão focado nas turmas em si, mas em unidades curriculares: os estudantes
360 escolhem as unidades que gostariam de frequentar, mas não selecionam turmas
individuais, sendo que para cada unidade curricular, os alunos podem indicar
362 alternativas que gostariam que o algoritmo considerasse caso não fiquem colocados. Esta
abordagem é especialmente útil para a atribuição de unidades curriculares optativas. As
364 preferências dos estudantes são tidas em conta pelo algoritmo e estes são atribuídos a
turmas específicas de modo a minimizar conflitos. Os estudantes que não fiquem
366 colocados numa dada unidade curricular nem nas suas alternativas ficam colocados nas
respetivas listas de espera, que o *software* utiliza caso venham a surgir novas vagas nas
368 unidades pretendidas. Este processo também toma em conta o número esperado de
estudantes que desejam cada unidade curricular de forma a reservar algumas vagas nas
370 turmas, para que, na fase seguinte, ainda seja possível colocar estudantes que precisem
de uma aula a uma certa hora, caso contrário ficariam em risco de não ficar colocados
372 em nenhuma turma.

Este seccionamento em massa é implementado utilizando um algoritmo de *iterative*
374 *forward search* [4], que se assemelha a métodos de pesquisa local, mas que opera sobre
soluções válidas mas possivelmente incompletas, onde algumas variáveis podem ainda
376 não ter valor atribuído, embora todas as restrições fortes tenham de ser cumpridas.

3.2.2 Seccionamento em tempo real

378 Após a fase anterior, os estudantes podem ainda fazer alterações aos seus horários.
Esta fase processa-se em tempo real e novas soluções são computadas mal o aluno
380 submeta o seu pedido de alteração. Este pode alterar turmas ou unidades curriculares
inteiras, sendo que os estudantes em lista de espera são automaticamente colocados
382 assim que haja vagas compatíveis com o resto do seu horário. Quando um estudante faz
uma alteração, o algoritmo emprega um processo de *backtracking* [5] que averigua
384 eventuais conflitos entre a nova escolha e as anteriores, começando pela opção que o
aluno indicara como sendo de prioridade mais elevada e assim sucessivamente.

386 Durante esta fase, o algoritmo continua a reservar algumas vagas em turmas para
os estudantes que se preveja que apenas vão fazer a sua escolha neste momento, sendo
388 que, para isso, faz uso de dados históricos de inscrições para calcular uma previsão da
procura de uma determinada unidade curricular ou turma.

390 3.2.3 Avaliação da ferramenta

No início do desenvolvimento deste trabalho, o UniTime foi instalado e
392 experimentado, tendo sido concluído que não seria a ferramenta ideal, pois, como já
referido, não opera ao nível de turmas individuais. Os estudantes colocam preferências
394 apenas ao nível de unidades curriculares e o *software* distribui-os por turmas à sua
discrição.

396 Considerou-se, também, a possibilidade de modificação deste programa para ir ao
encontro das necessidades sentidas pela FEUP, mas concluiu-se que seria uma tarefa
398 demasiado árdua e que possivelmente existiria uma forma mais eficiente de abordar o
problema enfrentado.

400 3.3 Formulação como problema de programação linear

Em Heitmann *et al.* [6], é estudada uma forma de modelação do problema de
402 atribuição de alunos a turmas com base em programação linear inteira mista (*mixed
integer linear programming*). A utilização da abordagem baseada em programação linear
404 traz diversas vantagens, tais como o facto de ser flexível, permitindo combinar critérios
e restrições de uma forma simples e fácil de visualizar mentalmente, comparativamente
406 à forma como pensamos nas questões do mundo real, e é suportada por diversas
ferramentas. Como principal desafio constata-se a necessidade de nos cingirmos a
408 expressões lineares, que nem sempre são “poderosas” o suficiente para modelar certo
tipo de restrições de alto nível mais complexas.

410 Concretamente, este artigo sugere representar a atribuição de um dado aluno a uma
412 dada turma de uma dada curricular como uma variável binária. Define, também, as
restrições necessárias para garantir o cumprimento de requisitos tais como não exceder
as capacidades das turmas e não permitir que um aluno acabe com mais do que uma
414 aula simultânea.

Este artigo também apresenta outras formulações relevantes para o problema da
416 FEUP: por exemplo, uma restrição fraca que tenta equilibrar o número de estudantes
colocados em cada turma com base num número-objetivo calculado a partir da
418 capacidade dessa turma, das capacidades das outras turmas da mesma unidade
curricular e do número de estudantes inscritos a essa UC. São ainda utilizadas variáveis
420 auxiliares para representar consequências da atribuição principal, como, por exemplo,
para indicar se um dado aluno tem uma aula num dado *timeslot*.

422 Dado que esta abordagem se afigurou pertinente para muitas das necessidades
identificadas no problema da FEUP, esta serviu de base para a formulação do sistema
424 desenvolvido. No entanto, existem outras especificidades do nosso problema que
necessitam de especial atenção se as pretendemos abordar. São elas:

- 426 • Existência de unidades curriculares obrigatórias e optativas, que devem ser
tratadas independentemente, visto que a atribuição de um dado conjunto de
428 optativas vai influenciar as escolhas possíveis na fase das obrigatórias;
- 430 • Possibilidade de colocação de alunos de forma incompleta ou fora das suas
preferências, isto é, não nos limitarmos unicamente a considerar as preferências
432 dos estudantes, mas abrir a hipótese de não os colocar em todas as unidades
curriculares pretendidas ou coloca-los em turmas que estes não tenham
indicado nas suas preferências;
- 434 • Formato de preferências já existente na FEUP com uma configuração bastante
específica e que, por questões de retrocompatibilidade com o sistema atual, se
436 pretende manter da forma em que se encontram hoje em dia.

3.4 *Software* de resolução de problemas de programação 438 linear

Na publicação mencionada na subsecção anterior, era feita referência ao *solver* IBM
440 ILOG CPLEX, portanto começou a considerar-se a sua utilização para resolução do
problema de programação linear.

442 Após mais alguma análise, em Jablonský [7], é feita uma comparação recente entre
os *solvers* mais potentes da atualidade, entre os quais o CPLEX se encontra no topo das
444 duas ferramentas com melhor desempenho – a par da ferramenta Gurobi –, podendo-se,
assim, confirmar a eficiência daquele. É ainda de destacar que este *software* pode ser

446 adquirido, de forma gratuita e sem quaisquer restrições ao tamanho do problema, para
fins académicos, visto que a Universidade do Porto já se encontra registada na rede de
448 parceiros académicos da IBM. Desta forma, optou-se pelo CPLEX como motor de
resolução do problema de programação linear.

450 **3.5 Outras referências consultadas**

Em Alvarez-Valdes *et al.* [8], é estudada uma abordagem ao problema de
452 seccionamento de estudantes por turmas utilizando uma estratégia de pesquisa tabu
(*tabu search*). No entanto, não só não leva em conta as preferências dos estudantes como
454 também não permite a inscrição em unidades curriculares optativas, já que o algoritmo
considera que cada estudante deve ficar colocado em uma turma por cada unidade
456 curricular que escolhe.

Schindl [9] aborda também o problema de separação de alunos por diferentes
458 turmas, mas, de forma semelhante à referência anterior, a lista de unidades curriculares
dos estudantes é fixa, sendo que todos eles se inscrevem às mesmas UCs. Mais uma vez,
460 as preferências dos alunos não entram em momento algum em consideração pelo
algoritmo.

Por outro lado, Busam [10] apresenta uma solução para atribuição de turmas que
462 tenta respeitar preferências, mas o algoritmo proposto está assente na premissa de que
os estudantes simplesmente colocam as turmas de cada unidade curricular por uma
464 determinada ordem, sem fazer diferentes opções com turmas específicas. Ou seja, a
atribuição de uma turma que o estudante colocou como 3.^a opção pode impedir a criação
466 de um horário sem conflitos caso este estudante venha a ser colocado noutra unidade
curricular na turma da sua 1.^a opção. O algoritmo tem em conta e tenta evitar estes
468 conflitos; no entanto, na FEUP, pretende-se dar ao estudante um maior controlo da sua
mancha de horário pedindo-lhe para criar um horário completo, e não saber as suas
470 preferências de turmas por cada unidade curricular individual.

472 **Capítulo 4**

474 **Conceção e implementação do novo sistema**

476 Neste capítulo, descrever-se-á, nas duas primeiras secções, a conceptualização que
478 foi realizada do novo sistema antes mesmo do início do seu desenvolvimento. De
480 seguido, é descrito todo o funcionamento do mesmo, começando por uma visão geral de
482 todos os componentes, passando por uma explicação dos dados de entrada necessários
484 e os dados de saída produzidos, seguindo-se a descrição das variáveis de decisão,
486 restrições e objetivos do problema modelado, e concluindo com alguns pormenores de
488 cariz mais técnico sobre a implementação do sistema.

482 **4.1 Conceptualização de um novo processo de colocação**

484 Tal como referido num capítulo anterior, a ideia deste trabalho foi desenvolver um
486 sistema que resultasse numa maior taxa de colocação de alunos no final da primeira fase
488 do que a que se verifica hoje em dia na FEUP. Com esse fim, o componente mais
490 importante deste sistema seria o processo que efetivamente atribui os estudantes às
492 turmas. Este processo necessitou de alguns dados de entrada, sendo estes: as
capacidades de cada turma, os critérios que definem a qualidade de uma dada solução
(incluindo os dados relacionados com estes, tais como as médias académicas dos
estudantes), as preferências de turmas destes mesmos estudantes e os horários das
turmas pré-elaborados. No fim do processamento, pretendia-se obter uma lista de
colocações que fosse facilmente exportável para um formato de ficheiro capaz de ser
importado no SIGARRA.

Conceção e implementação do novo sistema

494 Um outro requisito que se pretendeu cumprir foi a retrocompatibilidade do novo
496 sistema com o sistema atual; mais especificamente, na forma como as preferências dos
498 estudantes eram utilizadas. Durante o processo de desenvolvimento, considerou-se a
500 hipótese de pedir aos estudantes preferências com um formato diferente, de mais alto
502 nível (como, por exemplo, perguntar que manchas de horário seriam as mais
504 interessantes, ao invés de pedir para indicar turmas individuais); no entanto, optou-se
506 por não o fazer por duas razões. Em primeiro lugar, porque, desta forma, seria possível
utilizar os mesmos dados de entrada de processos anteriores para testar o novo processo.
Depois, porque a transição entre ambos os sistemas pode, assim, ser feita de forma mais
segura: se algo correr mal com o novo processo, é trivial utilizar-se o processo antigo
visto que os dados a recolher dos estudantes são os mesmos.

A figura 6 representa, de uma maneira informal, a conceptualização deste novo processo.

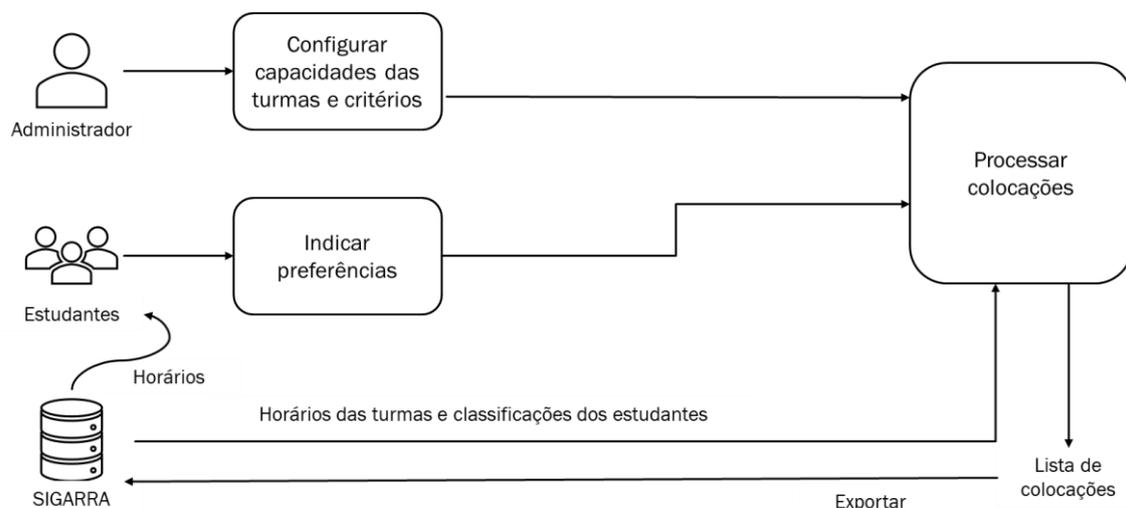


Figura 6. Conceptualização de alto nível do novo processo de colocação que seria desenvolvido

4.2 Critérios de avaliação de resultados

508 De forma a não só proceder à validação experimental dos resultados obtidos com o
510 novo processo, como também guiar o algoritmo de colocação durante a sua execução,
512 foi necessário definir os critérios de otimização deste problema e o seu peso individual.
514 Um dos principais fatores a ter em conta foi a taxa de colocação de estudantes, visto ser
este o principal motivador desta dissertação. No entanto, também era de grande
interesse continuar a satisfazer, em grande parte, as preferências de turmas dos
estudantes, tendo em conta a prioridade que cada estudante tem perante o algoritmo,
cujos critérios são definidos individualmente para cada processo de atribuição de

516 turmas. Por fim, interessava assegurar que não existiam grandes desvios entre o número
 de estudantes de cada turma. Na tabela 2, indicam-se de forma sucinta os critérios que
 518 tiveram de ser conjugados e de que forma o sistema atual os considera.

Tabela 2. Indicadores de qualidade do novo processo de colocação e
 comparação com a abordagem do processo atual

Critério	Sistema atual
Taxa de colocação	Ignora
Equilíbrio de turmas	Requer intervenção manual
Satisfação de prioridades entre estudantes	1.º critério
Satisfação das preferências dos estudantes	2.º critério

4.3 Arquitetura e tecnologias

520 O *software* desenvolvido é composto por dois componentes: o principal, em Java, e
 um secundário, em Python. O primeiro lê os dados de entrada, formula o problema
 522 (variáveis, restrições e objetivos) recorrendo ao motor *IBM ILOG CPLEX*, obtém deste
 uma solução e guarda-a sob a forma de dados de saída. O segundo recebe tanto os dados
 524 de entrada como os de saída do componente anterior e compila uma série de estatísticas
 que são, por sua vez, novamente guardadas como dados de saída.

526 Os dados de entrada dividem-se em dois conjuntos: os dados relativos ao processo
 de inscrição em turmas – extraídos do SIGARRA pelo pessoal administrativo através de
 528 *scripts* criados para o efeito – e dados específicos da aplicação, tais como os pesos de cada
 critério e a indicação do tipo de atribuição que se pretende fazer (unidades curriculares
 530 optativas ou obrigatórias).

Nos dados extraídos do SIGARRA, incluem-se as opções dos estudantes, que estes
 532 previamente introduziram através de uma interface apropriada para o efeito na página
 web da FEUP, bem como todas as restantes informações pertinentes sobre unidades
 534 curriculares, as suas turmas, horários de aulas e ainda médias dos estudantes.

Nos dados de saída, encontram-se as colocações finais dos estudantes e estatísticas
 536 de número de alunos colocados tanto ao nível de turmas individuais como ao nível de
 unidades curriculares. A partir destes dados, um administrador pode avaliar casos em
 538 que, por exemplo, o número de vagas disponíveis esteja a reduzir consideravelmente o
 resultado da colocação para, deste modo, realizar ajustes e iterar novamente o processo.

540 Na figura 7, são ilustrados todos estes fluxos de dados.

Conceção e implementação do novo sistema

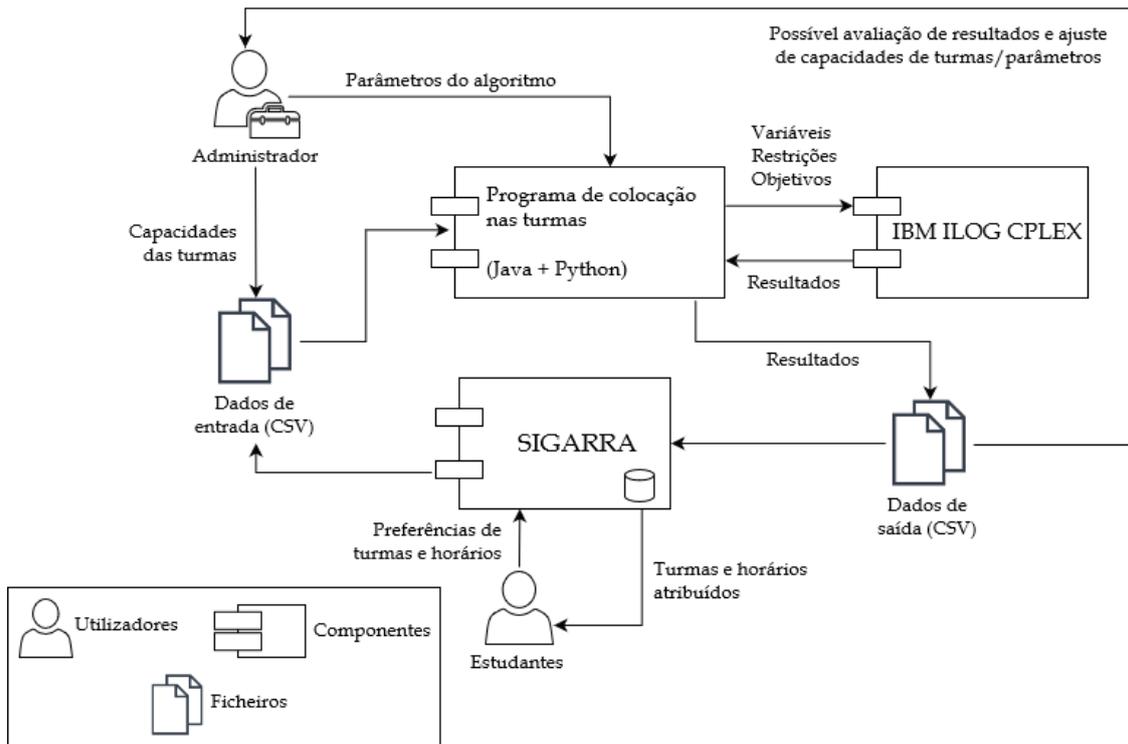


Figura 7. Esquema que ilustra o fluxo de dados entre todas as entidades relacionadas com o novo sistema

4.4 Dados de entrada

542 O sistema desenvolvido lê a maioria dos dados de que necessita a partir de ficheiros
543 CSV (dados representados em forma de tabela, cujas colunas são separadas por
544 vírgulas). Os dados de entrada necessários são explicados nas subsecções seguintes.

4.4.1 Escolhas introduzidas pelos estudantes no SIGARRA

546 Este ficheiro contém uma linha por cada escolha (par UC-turma) por cada opção de
547 cada aluno, e é através destes dados que o programa cria internamente uma estrutura
548 de dados com as informações dos estudantes.

549 Sendo este o ficheiro mais importante no processo, decidiu-se manter todas as
550 colunas que já vinham incluídas com a exportação realizada pelo pessoal administrativo
551 a partir do SIGARRA. Deste modo, reduz-se a probabilidade de surgirem eventuais
552 problemas quando o novo sistema for utilizado em ambiente real, embora o programa
553 não faça uso de todos estes campos. Assim, este ficheiro deve conter as seguintes
554 colunas:

Conceção e implementação do novo sistema

- Número de versão do processo
- 556 • Código do estudante
- Nome do estudante
- 558 • Número de identificação interno do estudante
- Ano letivo
- 560 • Número do semestre
- Número de ordem da preferência
- 562 • Código da unidade curricular
- Código da turma

564 O número de versão do processo é simplesmente um identificador que o SIGARRA
atribui a cada uma das ocorrências dos processos: 1.^a fase de UCs optativas, 2.^a fase de
566 UCs optativas, 1.^a fase de UCs obrigatórias, e assim em diante.

4.4.2 Médias académicas dos estudantes

568 Este conjunto de dados relaciona o código de cada estudante com a sua média
académica. Esta informação afigura-se importante, pois a média é um fator que possui
570 um peso elevado na atribuição de turmas aos estudantes de acordo com as suas
preferências.

572 Este ficheiro deve conter as seguintes colunas:

- Código do estudante
- 574 • Média do estudante

4.4.3 Lista de unidades curriculares

576 Através destes dados, o programa armazena informação sobre cada unidade
curricular, de forma a atribuir-lhes, de seguida, as turmas pertencentes.

578 Este ficheiro deve conter:

- Código da unidade curricular
- 580 • Sigla da unidade curricular
- Nome da unidade curricular
- 582 • Número total de horas de aula semanais que um estudante desta unidade
curricular deve frequentar
- 584 • Indicação de unidade curricular obrigatória (valor 0) ou optativa (valor 1)

4.4.4 Turmas pertencentes a cada unidade curricular

586 Este ficheiro contém a informação de cada turma de cada unidade curricular, bem
588 como a sua capacidade. O algoritmo nunca irá colocar mais alunos numa turma do que
o valor de capacidade aqui especificado.

Este ficheiro deve conter as seguintes colunas:

- 590 • Código da unidade curricular
- Código da turma
- 592 • Capacidade da turma

4.4.5 Compostos de turmas e as turmas que os constituem

594 Certas aulas (maioritariamente aulas teóricas) são lecionadas não para uma turma
individual, mas para um conjunto de turmas. Estes conjuntos (designados por
596 “compostos de turmas”) são definidos *a priori* contendo um dado grupo de turmas,
definidas pelo seu código. Embora turmas de unidades curriculares diferentes que
598 possuam o mesmo código sejam tecnicamente consideradas turmas diferentes (pois a
sua composição provavelmente não será a mesma), um dado composto contém sempre
600 as turmas com os mesmos códigos. Por exemplo, imagine-se que o composto *COMP_44*
é constituído pelas turmas *2MIEIC05* e *2MIEIC07*. Na prática, embora existam diferentes
602 turmas chamadas *2MIEIC05* e *2MIEIC07* em cada UC diferente, o composto *COMP_44*
contém sempre as turmas chamadas *2MIEIC05* e *2MIEIC07* de uma dada UC.

604 Este ficheiro deve conter as seguintes colunas:

- Código do composto
- 606 • Turmas pertencentes ao composto (uma por cada coluna)

4.4.6 Horário semanal das turmas das unidades curriculares

608 Este ficheiro contém informação sobre o horário semanal de todas as turmas do
semestre considerado. Para cada aula, é indicada a UC e turma à qual pertence, bem
610 como o dia da semana, a sua hora de início, a sua duração e o seu tipo (teórica, teórico-
prática, etc.).

612 Este ficheiro deve conter as seguintes colunas:

- Código da turma a que a aula pertence
- 614 • Código da unidade curricular a que essa turma pertence
- Hora de início da aula
- 616 • Duração da aula em horas

- Tipo de aula

618 **4.5 Dados de saída**

620 De forma semelhante aos dados de entrada, os de saída são escritos pelo programa no final da sua execução em ficheiros CSV. Estes dados indicam-se nas subsecções seguintes.

622 **4.5.1 Colocações dos estudantes**

624 Neste ficheiro, é apresentada uma linha por cada colocação (par UC-turma) de cada aluno. Juntamente com o código, nome e média do estudante, é indicada a opção em que este foi colocado (entre 1 e 10¹, ou -1 caso o estudante tenha ficado colocado fora das suas opções) e os códigos da UC e turma da sua colocação.

626 Este ficheiro contém as seguintes colunas:

- 628 • Código do estudante
- Nome do estudante
- 630 • Média do estudante
- Número de ordem da preferência
- 632 • Código da unidade curricular
- Código da turma

634 **4.5.2 Estatísticas de ocupação das turmas**

636 Este ficheiro contém uma linha para cada turma de cada UC, juntamente com o número de estudantes nela colocados e a sua capacidade total. É ainda indicado se se trata de uma UC obrigatória ou optativa, para efeitos informativos.

638 Este ficheiro contém as seguintes colunas:

- Código da unidade curricular
- 640 • Código da turma
- Número de estudantes colocados
- 642 • Capacidade da turma
- Indicação de unidade curricular obrigatória (valor 0) ou optativa (valor 1)

¹ Assume-se que cada estudante pode indicar até 10 opções de horário, mas este limite é configurável.

644 4.5.3 Estatísticas por unidade curricular

646 Por fim, este ficheiro de saída contém a informação agregada de todas as turmas de
cada unidade curricular. Para cada UC, é mostrada a informação sobre se se trata de uma
648 obrigatória ou optativa, o número total de vagas (soma das vagas de todas as turmas), o
número de estudantes que a escolheram em qualquer opção, o número de estudantes
650 que a escolheram como 1.^a opção (no caso das optativas, este número poderá dar uma
informação mais relevante acerca da procura efetiva da UC) e a proporção, sob a forma
652 de percentagem, entre o número de estudantes que escolheram a UC e o número total
de vagas. Este valor – especialmente se se encontrar acima dos 100% – ajuda os
654 funcionários administrativos a averiguarem os casos em que poderão não existir vagas
suficientes para satisfazer a procura de uma determinada unidade curricular.

Este ficheiro contém as seguintes colunas:

- 656 • Código da unidade curricular
- Nome da unidade curricular
- 658 • Indicação de unidade curricular obrigatória (valor 0) ou optativa (valor 1)
- Número de vagas disponíveis em todas as turmas
- 660 • Número de estudantes que colocaram esta unidade curricular em qualquer
opção
- 662 • Número de estudantes que colocaram esta unidade curricular em 1.^a opção
- Proporção entre o número de estudantes que escolheram esta unidade
664 curricular em qualquer opção e o número de vagas disponíveis em todas as
turmas

666 4.6 Variáveis e restrições

4.6.1 Notação

668 Nesta secção, são apresentadas as variáveis mais frequentemente utilizadas nas
equações que figurarão nas secções seguintes. Outras variáveis mais pontuais serão
670 apresentadas quando forem utilizadas, por razões de conveniência de leitura.

- J – conjunto de todas as unidades curriculares
- 672 • J_{ob} – conjunto de todas as unidades curriculares obrigatórias
- J_{op} – conjunto de todas as unidades curricula optativas
- 674 • K – conjunto de todas as turmas
- K_j – conjunto das turmas pertencentes à unidade curricular $j \in J$
- 676 • I – conjunto de todos os estudantes

- 678 • m_i - média académica do estudante $i \in I$. Teoricamente, este valor pode encontrar-se no intervalo $[0, 20]$, mas, na prática, visto que os alunos apenas obtêm aprovação a unidades curriculares com um mínimo de 10 valores, encontra-se no intervalo $[10, 20]$
- 680 • J_i - conjunto das unidades curriculares escolhidas pelo estudante $i \in I$ (com base na informação das suas preferências)
- 682 • T - conjunto de todos os *timeslots* (períodos de 30 minutos nos quais o horário semanal dos estudantes está subdividido)
- 684

4.6.2 Variáveis de decisão

686 A solução que se pretende obter é simplesmente um conjunto de colocações de alunos às turmas das unidades curriculares que estes pretendem frequentar. Deste modo, para cada estudante $i \in I$, para cada unidade curricular j incluída nas suas preferências J_i e para cada turma k dessa unidade curricular ($k \in K_j$), é criada uma variável de decisão X_{ijk} . Esta é uma variável binária, no sentido em que apenas pode tomar o valor de 1 ou 0, consoante o estudante i fique colocado na turma k da UC j ou não, respetivamente.

692 Internamente, são criadas outras variáveis de decisão que auxiliam na definição das restrições e da função-objetivo. No entanto, o seu valor é sempre dependente do valor das variáveis descritas no parágrafo anterior e dos dados de entrada. Assim, estas variáveis auxiliares serão descritas mais tarde, junto das equações que delas fazem uso.

4.6.3 Restrições

698 O problema de atribuição de turmas é relativamente simples quanto às restrições que têm de ser implementadas. Em primeiro lugar, sabendo que um dado estudante possui variáveis de decisão para todas as turmas de uma dada unidade curricular, é necessário implementar uma restrição para garantir que cada aluno é apenas atribuído a uma turma por UC, ou seja, que a soma de todas as variáveis de decisão X_{ijk} para um dado i e um dado j é inferior ou igual a 1. Formalmente:

$$704 \quad \sum_{k \in K_j} X_{ijk} \leq 1, \quad \forall i \in I, j \in J_i \quad (1)$$

Uma outra restrição que foi necessário definir tem a ver com a obrigatoriedade de frequência das aulas práticas na FEUP. Por esta razão, os alunos são impedidos de se inscreverem em turmas com aulas práticas coincidentes. Assim, é implementada uma restrição que garante que, em cada *timeslot*, a soma das variáveis de decisão X_{ijk} para um dado i e para todos os k que possuam uma aula prática nesse *timeslot* seja inferior ou

Conceção e implementação do novo sistema

710 igual a 1. Isto impede que o horário resultante da colocação de um estudante contenha
sobreposições de aulas práticas, independentemente de o estudante ter escolhido um
712 horário assim nas suas preferências ou de o algoritmo lhe ter feito uma colocação fora
das suas preferências. Formalmente:

$$714 \quad \sum_{j \in J_i} \sum_{k \in K'_{jt}} X_{ijk} \leq 1, \quad \forall i \in I, t \in T \quad (2)$$

Em que:

- 716 • K'_{jt} – conjunto das turmas pertencentes à unidade curricular $j \in J$ que possuem
uma aula prática no *timeslot* $t \in T$

718 De seguida, é preciso garantir que o número de alunos que ficam atribuídos a uma
dada turma não ultrapasse a capacidade máxima previamente definida para esta. Desta
720 forma, para um dado j e k , garante-se que a soma das variáveis decisão X_{ijk} para todos
os i é inferior ou igual à capacidade de k . Formalmente:

$$722 \quad \sum_{i \in I_j} X_{ijk} \leq q_{jk}, \quad \forall j \in J, k \in K_j \quad (3)$$

Em que:

- 724 • I_j – conjunto dos estudantes que incluíram a unidade curricular $j \in J$ numa das
suas preferências
- 726 • q_{jk} – capacidade da turma $k \in K_j$ pertencente à UC $j \in J$

Importa ainda realçar que esta restrição não é aplicada quando estamos a lidar com
728 turmas de unidades curriculares optativas e o programa se encontra no modo de
atribuição de unidades curriculares obrigatórias. Visto que a atribuição de turmas de
730 UCs obrigatórias é sempre realizada após a atribuição de UCs optativas, os estudantes
vão incluir, por predefinição, a sua anterior atribuição de optativas juntamente com as
732 suas preferências de obrigatórias. Sendo assim, o programa assume que os alunos já se
vão encontrar distribuídos pelas optativas de tal forma que não ultrapassem a sua
734 capacidade máxima ou, caso contrário, que alguma alteração pontual que viole este
requisito tenha sido sancionada pelos serviços administrativos. Desta forma, a
736 capacidade máxima das UCs optativas não é verificada durante a fase de UCs
obrigatórias.

738 Por fim, é ainda definida uma restrição fraca que está relacionada com o objetivo de
garantir um equilíbrio entre a quantidade de estudantes atribuídos a todas a turmas de
740 uma dada unidade curricular. Este objetivo será descrito em mais detalhe na secção
seguinte. Sucintamente, cada turma possui um número mínimo ideal de estudantes que
742 gostaríamos de colocar de forma a que todas as turmas dessa mesma UC ficassem com

Conceção e implementação do novo sistema

uma taxa de ocupação igual. Sabendo que nem sempre será possível alcançar esse ideal,
744 quanto maior for a diferença entre o número de estudantes efetivamente colocados e o
número ideal, maior será a penalização da função-objetivo. Sendo assim, constrói-se uma
746 restrição que, para cada turma, garante que o número de estudantes efetivamente
colocados somado com a diferença anteriormente mencionada seja superior ou igual ao
748 número ideal de estudantes. Formalmente:

$$\left(\sum_{i \in I_j} X_{ijk} \right) + S_{jk} \geq obj_{jk}, \quad \forall j \in J_{ob}, k \in K_j \quad (4)$$

$$750 \quad obj_{jk} = \frac{q_{jk}}{\sum_{k' \in K_j} q_{jk'}} \times |I_j|, \quad \forall j \in J_{ob}, k \in K_j \quad (5)$$

$$0 \leq S_{jk} \leq obj_{jk} \quad (6)$$

752 Em que:

- obj_{jk} - número-objetivo de estudantes que idealmente seriam colocados na
754 turma $k \in K_j$ da UC $j \in J$, de forma a conseguir-se uma taxa de ocupação
uniforme em todas as turmas dessa unidade curricular. Este valor é calculado
756 recorrendo apenas aos dados de entrada, previamente à geração das equações
- S_{jk} - variável de “folga” que é somada ao número de estudantes efetivamente
758 colocados na turma $k \in K_j$ da UC $j \in J$ para que o valor total resultante seja
igual ou superior ao número-objetivo. No final, esta variável acaba por
760 representar a diferença entre o número-objetivo e o número efetivo de
estudantes, visto que é acrescentada uma parcela à função-objetivo que tenta
762 minimizar o seu valor, como poderá ser constatado na secção seguinte

Esta restrição só é aplicada quando o programa se encontra no modo de atribuição
764 de turmas de unidades curriculares obrigatórias. Isto deve-se a dois principais motivos:
em primeiro lugar, as UCs optativas geralmente não possuem mais do que uma turma,
766 o que já reduz a importância da restrição; e, em segundo lugar, não é possível prever *a priori*
quantos estudantes vão ficar colocados em cada UC optativa, sendo, por isso,
768 difícil calcular o número-objetivo de alunos colocados. Refira-se que, no caso das UCs
obrigatórias, basta averiguar o número de estudantes que colocam cada UC em pelo
770 menos uma das suas preferências ($|I_j|$).

No entanto, neste caso (modo de atribuição de unidades curriculares optativas), se
772 os administradores ainda o pretenderem, estes podem definir uma taxa mínima de
estudantes a colocar em todas as turmas, relativa apenas à capacidade máxima de cada
774 uma. Por exemplo, é possível configurar o algoritmo para utilizar sempre pelo menos

20% da capacidade máxima de cada turma. Esta restrição é aplicada pela seguinte
776 fórmula:

$$\sum_{i \in I_j} X_{ijk} \geq t_{\min} \times q_{jk}, \quad \forall j \in J_{op}, k \in K_j \quad (7)$$

778 Em que:

- t_{\min} - taxa mínima (global) de ocupação das turmas

780 4.7 Critérios de otimização

Tendo as variáveis de decisão definidas no CPLEX, este encarrega-se de lhes
782 atribuir valores respeitando as restrições e tentando aproximar a solução final do valor
ideal. A avaliação da qualidade de uma solução é realizada através do cálculo de uma
784 função-objetivo que, no caso deste programa, pretendemos maximizar (isto é, obter o
valor mais elevado possível).

786 No decorrer do desenvolvimento deste sistema, foram identificados alguns critérios
(alguns parcialmente dependentes de outros) que se consideraram importantes para se
788 obter uma colocação que aumente a satisfação de todos os intervenientes, quando
comparado com o sistema que é utilizado hoje em dia na FEUP.

790 Sabendo que nos interessa alcançar mais do que um objetivo, este problema torna-
se, então, num problema de otimização multicritério (ou multiobjetivo), em que nem
792 sempre é fácil comparar duas soluções e identificar qual é a melhor, pois cada um dos
critérios pode traduzir-se numa parcela da função-objetivo com uma ordem de grandeza
794 diferente da de outro critério. Desta forma, todos as parcelas são normalizadas; isto é, o
seu valor é sempre limitado entre 0 e 1, em que 0 representa uma satisfação nula do
796 critério e 1 uma satisfação total do mesmo, embora nem sempre seja possível alcançar
um ou ambos os extremos, dependendo da natureza do critério subjacente.

798 Os critérios utilizados no novo sistema encontram-se listados na tabela 3 e serão
explicados com maior detalhe de seguida.

Tabela 3. Critérios de otimização considerados na função-objetivo

Objetivo	Critério	Peso usado
Maximizar colocações	Maximizar pares aluno-UC colocados	30%
	Maximizar alunos com colocação completa	10%
	Maximizar <i>timeslots</i> totais ocupados	10%

Conceção e implementação do novo sistema

Satisfazer preferências dos estudantes	Maximizar preferências satisfeitas	10%
Garantir qualidade do horário	Minimizar blocos ocupados para alunos colocados fora das preferências	10%
	Minimizar blocos não pretendidos para alunos colocados fora das preferências	10%
Garantir equilíbrio das turmas	Minimizar subocupações de turmas	20%

800 4.7.1 Maximizar pares aluno-UC colocados

802 Este critério procura obter uma taxa total de colocação de estudantes mais elevada,
 803 independentemente das preferências dos alunos, visto que esta é a estatística mais
 804 importante que levou ao desenvolvimento do novo sistema. Cada variável que
 805 representa a colocação de um estudante é pesada com a sua média académica, de modo
 806 a conferir maior importância a alunos com médias superiores. Isto obtém-se através da
 seguinte fórmula:

$$\max. \sum_{i \in I} \sum_{j \in J_i} \sum_{k \in K_j} (m_i \times X_{ijk}) \quad (8)$$

808 De forma a normalizar esta parcela da função-objetivo para o intervalo [0,1], divide-se
 a mesma pelo valor máximo teórico que esta pode tomar, ou seja, por:

$$810 \sum_{i \in I} (m_i \times |J_i|) \quad (9)$$

4.7.2 Maximizar alunos com colocação completa

812 Foi acrescentada uma outra parcela que tenta garantir que os estudantes não fiquem
 813 com colocações parciais. Estes casos são indesejáveis, pois um aluno que fique, por
 814 exemplo, com uma única unidade curricular sem turma atribuída pode não conseguir
 815 obter mais tarde nenhuma turma que seja compatível com o horário das suas outras UCs,
 816 fazendo com que tenha de proceder à alteração de outras turmas e potencialmente
 libertando vagas que poderiam ter sido úteis para outro estudante. Deste modo, a
 818 seguinte parcela é adicionada à função-objetivo:

$$\max. \sum_{i \in I} (m_i \times C_i) \quad (10)$$

820

Em que:

- 822 • C_i – variável auxiliar que toma o valor de 1 caso o estudante $i \in I$ tenha obtido
 824 colocação em todas as unidades curriculares a que se candidatava e 0 na situação
 contrária

$$C_i = \begin{cases} 0, & \text{se } \sum_{j \in J_i} \sum_{k \in K_j} X_{ijk} < |U_i| \\ 1, & \text{se } \sum_{j \in J_i} \sum_{k \in K_j} X_{ijk} = |U_i| \end{cases} \quad (11)$$

826 Na implementação com CPLEX, o cálculo de cada C_i é realizado através de duas
 restrições condicionais auxiliares, de acordo com a definição anterior.

828 A multiplicação pela média do estudante (m_i), dentro desta parcela da função-
 objetivo, pretende dar prioridade a estudantes com média mais elevada. Além disso, de
 830 forma a normalizá-la para o intervalo $[0,1]$, divide-se a mesma pelo valor máximo
 teórico que esta pode tomar, ou seja, por:

832
$$\sum_{i \in I} m_i \quad (12)$$

4.7.3 Maximizar *timeslots* totais ocupados

834 O algoritmo garante que um estudante não fica colocado em turmas com
 sobreposição de aulas práticas, visto que estas são de frequência obrigatória. Idealmente,
 836 também se colocaria uma restrição semelhante no caso das aulas teóricas. No entanto,
 existem casos, como, por exemplo, os de estudantes que frequentam unidades
 838 curriculares de diferentes anos, em que não é possível encontrar um horário em que não
 exista sobreposição de aulas teóricas. Sendo assim, embora este tipo de sobreposição seja
 840 aceite, o algoritmo tenta maximizar o número de *timeslots* ocupados pelos alunos em
 geral. Isto tem a consequência de minimizar, tanto quanto possível, estes casos de
 842 sobreposição. Este critério é traduzido pela seguinte parcela:

$$\max. \sum_{i \in I} \sum_{t \in T} O_{it} \quad (13)$$

844 Em que:

- O_{it} – variável auxiliar que toma o valor de 1 caso o estudante $i \in I$ obtenha uma
 846 colocação em que o *timeslot* $t \in T$ fica ocupado e 0 na situação contrária

$$O_{it} = \begin{cases} 0, & \text{se } \sum_{j \in J_i} \sum_{k \in K_{jt}} X_{ijk} = 0 \\ 1, & \text{se } \sum_{j \in J_i} \sum_{k \in K_{jt}} X_{ijk} \neq 0 \end{cases} \quad (14)$$

848 Em que:

- 850 • K_{jt} - conjunto de turmas pertencentes à unidade curricular $j \in J$ que possuem uma aula no *timeslot* $t \in T$

852 Na implementação com CPLEX, o cálculo de cada O_{it} é realizado através de duas restrições condicionais auxiliares, de acordo com a definição anterior.

854 De forma a normalizar esta parcela da função-objetivo para o intervalo $[0,1]$, divide-se a mesma pelo valor máximo teórico que esta pode tomar, ou seja, por:

$$\sum_{i \in I} \sum_{j \in J_i} t_j \quad (15)$$

856 Em que:

- 858 • t_j - número total de *timeslots* que um estudante desta unidade curricular deve frequentar

4.7.4 Maximizar preferências satisfeitas

860 Para os estudantes, o cumprimento das suas preferências individuais é o critério mais importante. Este critério faz com que o algoritmo não coloque sempre os alunos em
862 turmas arbitrárias, mas tente respeitar as suas preferências. Neste caso, através da observação dos resultados, determinou-se que seria útil colocar o parâmetro da média
864 académica dos estudantes com um peso exponencial de forma a minimizar a diferença, em comparação com o resultado obtido pelo processo atual, percecionada pelos
866 estudantes no que diz respeito às médias que habitualmente garantem a colocação na sua primeira opção. Utilizando a média por si só, é mais provável um aluno com, por
868 exemplo, 15 valores ficar colocado na sua segunda opção, sendo que não é o que habitualmente se constata no processo atual. A parcela correspondente na função-
870 objetivo é a seguinte:

$$\max. \sum_{i \in I} \sum_{p \in P_i} \left(2^{m_i} \times \left(10 - (o_{ip} - 1) \right) \times A_{ip} \right) \quad (16)$$

872

Conceção e implementação do novo sistema

Em que:

- 874 • P_i - conjunto das preferências do estudante $i \in I$
 876 • o_{ip} - n.º de ordem (entre 1 e 10) da preferência $p \in P_i$ do estudante $i \in I$
 878 • A_{ip} - variável auxiliar que toma o valor de 1 caso a preferência $p \in P_i$ tenha sido atribuída ao estudante (todas as turmas que a compõem) e 0 na situação contrária

$$A_{ip} = \begin{cases} 0, & \text{se } \left(\sum_{j \in J_p} \sum_{k \in K_{jp}} X_{ijk} < |p| \right) \vee \left(\sum_{j \in J_p} \sum_{k \in K_{jp}} X_{ijk} < \sum_{j \in J_i} \sum_{k \in K_j} X_{ijk} \right) \\ 1, & \text{caso contrário}^2 \end{cases} \quad (17)$$

880 Em que:

- 882 • J_p - conjunto de unidades curriculares presentes na preferência $p \in P_i$
 884 • K_{jp} - conjunto (unitário) das turmas da unidade curricular $j \in J_p$ presentes na preferência $p \in P_i$
 884 • $|p|$ - número de unidades curriculares presentes na preferência $p \in P_i$

886 Na implementação com CPLEX, o cálculo de cada A_{ip} é realizado através de duas restrições condicionais auxiliares, de acordo com a definição anterior.

888 As condições da equação indicam que se considera um estudante i colocado na opção p (isto é, $A_{ip} = 1$) se e só se ficou colocado em todas as turmas indicadas nessa opção (negação do lado esquerdo da disjunção) e não ficou colocado em mais turmas (negação do lado direito da disjunção).

892 Tal como referido numa secção anterior, o valor de m_i pode encontrar-se no intervalo $[10, 20]$, pelo que o peso 2^{m_i} se encontra no intervalo $[1024, 1\,048\,576]$. O peso $10 - (o_{ip} - 1)$, por sua vez, varia dentro do intervalo $[1, 10]$, dependendo da ordem da preferência do estudante. Desta forma, constata-se que o algoritmo faz questão de tratar estudantes com médias diferentes de forma consideravelmente diferente, mas não distingue tão significativamente duas opções de um mesmo estudante.

898 De forma a normalizar esta parcela da função-objetivo para o intervalo $[0,1]$, divide-se a mesma pelo valor máximo teórico que esta pode tomar, ou seja, por:

$$\sum_{i \in I} (2^{m_i} \times 10) \quad (18)$$

² O valor de 1 não é explicitamente atribuído neste caso, mas a variável toma este valor visto que, tal como se pode verificar na equação acima, a função-objetivo tenta maximizá-la.

900 **4.7.5 Minimizar blocos ocupados para alunos colocados fora das**
preferências

902 No caso dos estudantes que ficam colocados fora das suas opções, tentou-se
 melhorar a qualidade do seu horário resultante. Assim, para estes alunos, gostaríamos
 904 de reduzir o número total de blocos (sendo que um “bloco” é uma manhã ou tarde)
 ocupados no seu horário, o que acontece com a seguinte parcela:

906
$$\min. \sum_{i \in I} \sum_{b \in B} O_{ib} \quad (19)$$

Em que:

- 908 • B – conjunto de blocos de horário possíveis (manhãs/tardes)
 • O_{ib} – variável auxiliar que toma o valor de 1 se o estudante $i \in I$, não tendo
 910 ficado colocado em nenhuma das suas opções, tiver aulas atribuídas no bloco
 $b \in B$ e 0 na situação contrária

912
$$O_{ib} = \begin{cases} 1, & \text{se } \left(\sum_{p \in P_i} A_{ip} = 0 \right) \wedge \left(\sum_{t \in T_b} O_{it} \geq 1 \right) \\ 0, & \text{caso contrário}^3 \end{cases} \quad (20)$$

Em que:

- 914 • T_b – conjunto dos *timeslots* que pertencem ao bloco $b \in B$

Na implementação com CPLEX, o cálculo de cada O_{ib} é realizado através de duas
 916 restrições condicionais auxiliares, de acordo com a definição anterior.

De forma a normalizar esta parcela da função-objetivo para o intervalo [0,1], divide-
 918 se a mesma pelo valor máximo teórico que esta pode tomar, ou seja, por:

$$|I| \times 12 \quad (21)$$

920 Sendo que 12 é o número total de blocos num horário da FEUP (6 dias \times 2 blocos).
 O resultado da divisão é subtraído a 1 antes de ser inserido na função-objetivo, uma vez
 922 que esta é uma função a maximizar.

³ O valor de 0 não é explicitamente atribuído neste caso, mas a variável toma este valor visto que, tal como se pode verificar na equação acima, a função-objetivo tenta minimizá-la.

4.7.6 Minimizar blocos não pretendidos para alunos colocados fora das preferências

924

De forma semelhante ao critério anterior, este critério debruça-se sobre os blocos que um dado aluno nunca colocou em nenhuma das suas preferências. A ideia é: se um estudante nunca escolheu aulas, por exemplo, a uma quinta-feira de tarde, talvez queira a todo o custo evitar esse bloco, possivelmente não podendo sequer estar presente. Deste modo, a seguinte fórmula é adicionada à função-objetivo:

926

928

930

$$\min. \sum_{i \in I} \sum_{b \in B \setminus B_i} O_{ib} \quad (22)$$

Em que:

932

- B_i - conjunto de blocos de horário que constam das preferências do estudante $i \in I$

934

De forma a normalizar esta parcela da função-objetivo para o intervalo $[0,1]$, divide-se a mesma pelo valor máximo teórico que esta pode tomar, ou seja, por:

936

$$|I| \times 12 \quad (23)$$

938

Sendo que 12 é o número total de blocos num horário da FEUP (6 dias \times 2 blocos). O resultado da divisão é subtraído a 1 antes de ser inserido na função-objetivo, uma vez que esta é uma função a maximizar.

940

4.7.7 Minimizar subocupações de turmas

942

944

946

Por fim, e dando seguimento àquilo que foi apresentado numa das secções anteriores sobre restrições, existe uma parcela na função-objetivo que tenta reduzir ao máximo a subocupação das turmas (isto é, a diferença entre o número de estudantes efetivamente colocados nessa turma e o número ideal de estudantes que gostaríamos de colocar de forma a garantir uma taxa de ocupação uniforme para todas as turmas dessa unidade curricular). A fórmula resultante é esta:

$$\min. \sum_{j \in J} \sum_{k \in K_j} S_{jk} \quad (24)$$

948

De forma a normalizar esta parcela da função-objetivo para o intervalo $[0,1]$, divide-se a mesma pelo valor máximo teórico que esta pode tomar, ou seja, por:

950

$$\sum_{j \in J} \sum_{k \in K_j} obj_{jk} \quad (25)$$

O resultado da divisão é subtraído a 1 antes de ser inserido na função-objetivo, uma
952 vez que esta é uma função a maximizar.

4.8 Detalhes da implementação

954 Este programa foi desenvolvido recorrendo à linguagem de programação Java,
visto ser uma na qual possuo alguma experiência e ser uma das linguagens suportadas
956 pelo *software* de resolução de problemas escolhido. De facto, esta rapidamente revelou
ser uma enorme vantagem, pois o projeto arrancou e foi progredindo ao longo das
958 semanas de uma forma bastante célere.

Para a modelação e posterior resolução do problema de otimização, recorreu-se ao
960 IBM ILOG CPLEX. Este *software* vem incluído com bibliotecas que permitem a integração
do seu motor de resolução de problemas numa variedade de linguagens de
962 programação. Estas bibliotecas são de muito fácil e rápida aprendizagem. Por exemplo,
na figura 8, demonstra-se como é criada a variável binária que representa a atribuição
964 de um aluno a uma turma de uma unidade curricular, bastando definir a variável como
booleana e passando-lhe, opcionalmente, uma cadeia de caracteres para a rápida
966 identificação da variável em caso de necessidade de depuração.

```
IloIntVar studentGroupAssignment = cplex.boolVar("(" + student.getCode() + " :  
" + course.getCode() + "-" + group.getCode() + ")");
```

Figura 8. Excerto de código que representa a criação de uma variável de decisão com a biblioteca Java do CPLEX

Definir uma restrição também é muito simples, como pode ser constatado na figura
968 9. Aqui, para um dado aluno e dado *timeslot*, temos uma expressão que soma todas as
variáveis binárias que representam a sua atribuição a turmas com aulas práticas nessa
970 hora. Indicamos simplesmente que estamos a representar uma restrição “menor ou
igual” (“less than or equal”, ou “Le”, em inglês) entre essa expressão e a constante 1, ou
972 seja, um estudante não pode ser atribuído a mais do que uma aula prática simultânea.

```
cplex.addLe(sumAllPracticalClasses, 1);
```

Figura 9. Excerto de código que representa a definição de uma restrição com a biblioteca Java do CPLEX

As expressões são somas de variáveis multiplicadas com coeficientes, podendo
974 estas ser utilizadas para agregar diferentes variáveis e colocá-las numa restrição ou na

Conceção e implementação do novo sistema

função-objetivo. Na figura 10, ilustra-se como criar uma expressão linear inteira e, mais
976 tarde, como adicionar-lhe um termo (no exemplo, o coeficiente é 1).

```
IloLinearIntExpr sumAllPracticalClasses = cplex.linearIntExpr();  
(...)  
sumAllPracticalClasses.addTerm(1, assignmentVariable);
```

Figura 10. Excerto de código que representa a criação de uma expressão linear e, posteriormente, a adição de um termo

Por vezes, são ainda definidas restrições condicionais para atribuir valores a
978 variáveis auxiliares. Por exemplo, pretende-se exigir que uma dada variável tome o valor
de 1 apenas caso um conjunto de condições seja cumprido, caso contrário toma um valor
980 diferente. Deste modo, na figura 11 é apresentado um exemplo de uma destas restrições.

```
cplex.add(  
    cplex.ifThen(  
        cplex.and(  
            cplex.eq(sumStudentFulfilledPreferences, 0),  
            cplex.ge(currentSumOccupiedTimeslots, 1)),  
        cplex.eq(occupiedPeriod, 1)  
    ));
```

Figura 11. Excerto de código que representa a definição de uma restrição condicional

Por fim, definir a função-objetivo não é muito diferente, como indica a figura 12.
982 Basta indicar que se pretende maximizar a expressão que se segue e passar ao CPLEX
essa expressão. Neste caso, estamos a somar várias parcelas, cada uma com um peso já
984 definido.

```
cplex.addMaximize(cplex.sum(  
    cplex.prod(.3, objMaximizeSumAllAssignments),  
    cplex.prod(.1, objMaximizeCompleteStudents),  
    cplex.prod(.1, objMaximizeOccupiedTimeslots),  
    cplex.prod(.1, objMaximizeFulfilledPreferences),  
    cplex.prod(.2, objMinimizeGroupUtilizationSlacks),  
    cplex.prod(.1, objMinimizeOccupiedPeriodsWithNoPreferenceAssigned),  
    cplex.prod(.1, objMinimizeUnwantedOccupiedPeriods)  
));
```

Figura 12. Excerto de código que representa a definição da função-objetivo com a biblioteca Java do CPLEX

O CPLEX permite também configurar alguns parâmetros relativos à forma como
986 pretendemos que o problema seja resolvido. No exemplo da figura 13, demonstra-se de
que forma, no sistema desenvolvido, é possível definir um tempo-limite máximo de 600
988 segundos (10 minutos) para a resolução do problema.

Conceção e implementação do novo sistema

```
cplex.setParam(IloCplex.DoubleParam.TiLim, 600);
```

Figura 13. Excerto de código que representa a configuração do parâmetro de tempo-limite do CPLEX

990 Para instruir ao CPLEX que o problema deverá ser resolvido, faz-se uma simples
chamada ao método correspondente, como indicado na figura 14. Para obter o valor final
de uma determinada variável, existe também um método disponível para esse efeito, tal
992 como se ilustra na figura 15.

```
cplex.solve();
```

Figura 14. Excerto de código que representa a chamada ao método de resolução do problema já formulado no CPLEX

```
cplex.getValue(assignmentVar);
```

Figura 15. Excerto de código que representa a forma de obtenção do valor final de uma variável

994 Foi ainda desenvolvido um *script* em Python que, dado um conjunto de dados de
entrada e de saída de um processo, compila estatísticas sobre o processo de colocação.
Estas estatísticas serviram, posteriormente, para comparar o desempenho do novo
996 sistema em relação ao sistema atual, como poderá ser constatado no capítulo seguinte.

998 Devido à grande quantidade de variáveis, expressões e restrições necessárias para
a modelação do problema, quando o algoritmo é executado, este necessita de uma
quantidade considerável de memória RAM para funcionar. Em todas as execuções
1000 realizadas durante o desenvolvimento do sistema numa máquina com 8 GB de RAM,
nunca se constatou a utilização de mais de 2 GB; no entanto, a utilização de memória por
1002 parte do CPLEX é regulada de acordo com a quantidade de memória total disponível.

Capítulo 5

1004 Validação experimental

5.1 Comparação de estatísticas entre ambos os processos

1006 Nesta secção, será feita uma comparação entre os dados estatísticos extraídos da
1008 colocação de estudantes realizada pelo processo atual na 1.^a fase e aqueles extraídos da
1010 colocação obtida com o novo processo. Estes dados dizem respeito ao 2.^o semestre do
ano letivo de 2016/2017, no Mestrado Integrado em Engenharia Informática e
Computação (MIEIC) da FEUP.

5.1.1 Colocação em unidades curriculares optativas

1012 No processo de atribuição de UCs optativas, existe uma quantidade
significativamente menor de dados para processar. Sabendo que os dados analisados
1014 dizem respeito ao 2.^o semestre, a maioria dos estudantes envolvidos neste processo serão
alunos do 4.^o ano, sendo que, neste ano letivo, corresponde a 135 pessoas, que
1016 necessitam, em média, de colocação em 3 unidades curriculares optativas.

Para avaliar o novo sistema quanto à atribuição destas UCs optativas, as
1018 capacidades das turmas utilizadas foram as capacidades finais definidas pelos
administradores aquando da execução do algoritmo atual. Interessa também referir que
1020 vamos comparar o novo processo com a 1.^a fase do processo atual – aquela que é
completamente automatizada e na qual os alunos não vão alterando manualmente as
1022 suas vagas ao longo do processo.

1024 Sendo que o processo de atribuição das optativas é o mais simples, pois o algoritmo
tem obrigatoriamente de se cingir às preferências dos alunos, as únicas estatísticas
relevantes a extrair são os dados relativos à ordem da preferência em que cada estudante

Validação experimental

1026 ficou colocado, agrupando-os por média académica. Nas figuras 16 e 17, é possível observar-se a comparação destes dados entre o sistema atual e o novo sistema.

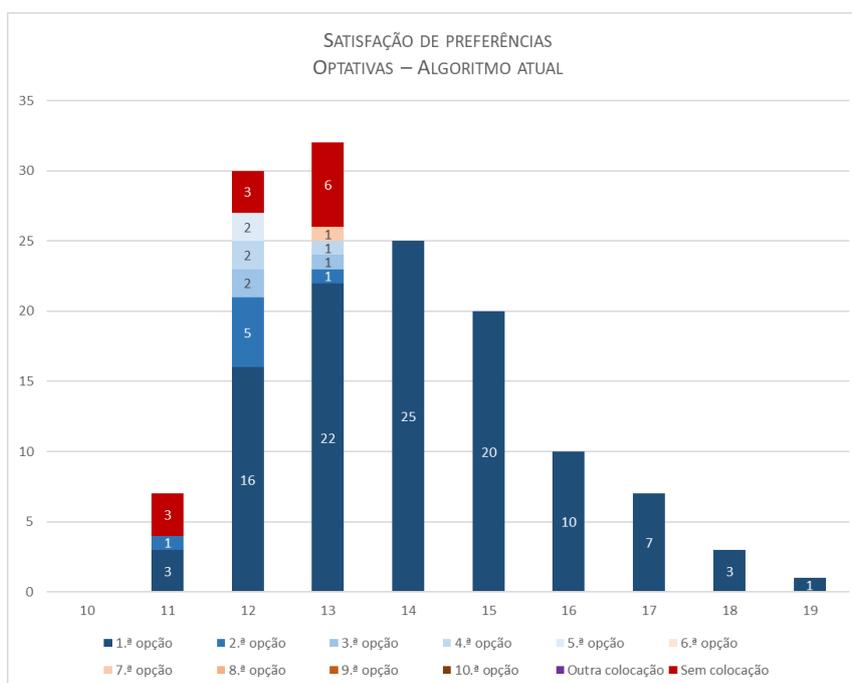


Figura 16. Satisfação de preferências dos estudantes no processo atual de atribuição de turmas de unidades curriculares optativas

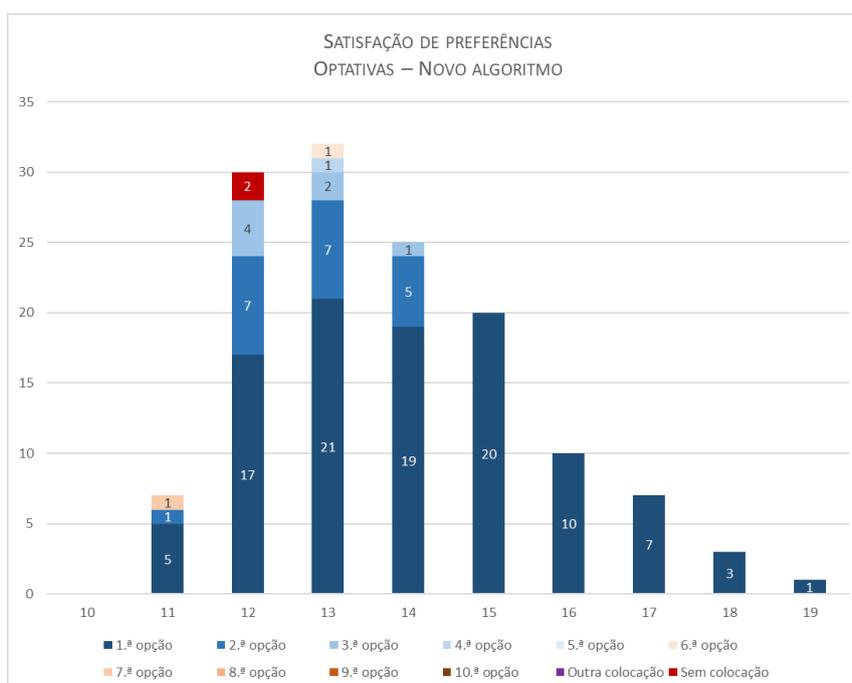


Figura 17. Satisfação de preferências dos estudantes no novo processo de atribuição de turmas de unidades curriculares optativas

Validação experimental

1028 Como é possível constatar-se, a taxa de colocação aumentou de forma considerável,
1030 tendo-se verificado uma alteração de 12 estudantes não colocados (~9%) para apenas 2
1032 (~1%). Continua a observar-se uma predominância de estudantes colocados em 1.^a
opção, sendo que todos os alunos com média igual ou superior a 15 valores obtiveram a
sua colocação mais desejada.

5.1.2 Colocação em unidades curriculares obrigatórias

1034 Já no caso das UCs obrigatórias, embora o algoritmo continue a tentar respeitar as
preferências dos estudantes, há casos em que isto não acontece. Assim sendo, é possível,
1036 para além de uma comparação semelhante à do processo das UCs optativas (satisfação
de preferências), analisar ainda o equilíbrio de alunos colocados entre as várias turmas
1038 da mesma unidade curricular e alguns indicadores que dizem respeito à qualidade de
horário daqueles estudantes que são colocados pelo algoritmo fora das suas preferências.

1040 Nas figuras 18 e 19, de forma idêntica às unidades curriculares optativas, faz-se uma
comparação entre os dois processos no que diz respeito ao número da opção em que os
1042 estudantes ficam colocados.

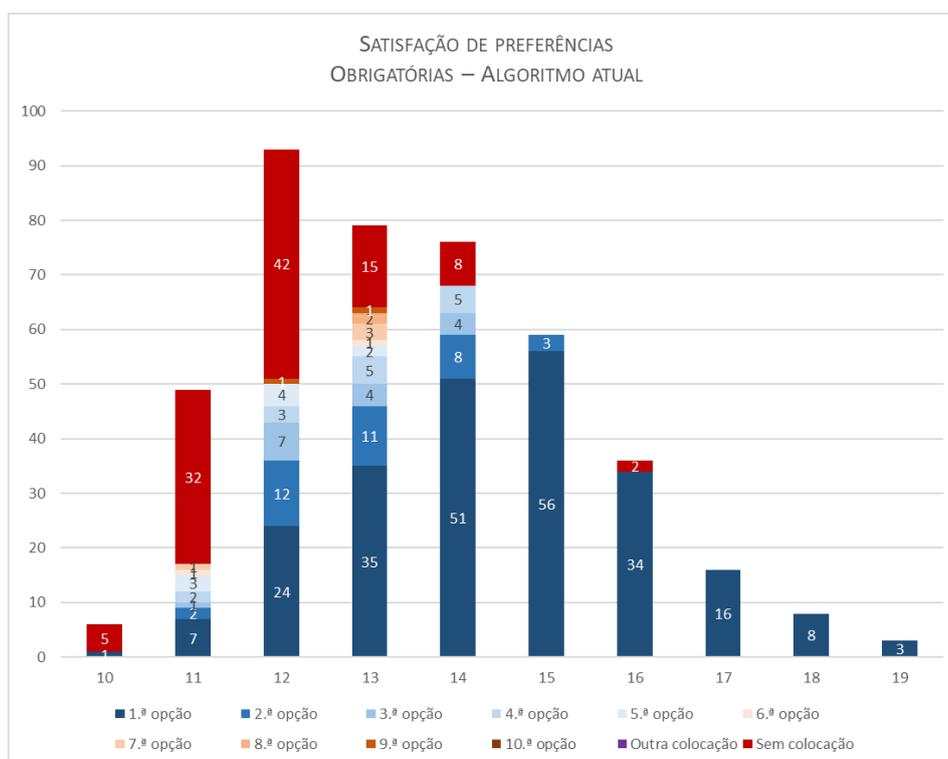


Figura 18. Satisfação de preferências dos estudantes no processo atual de atribuição de turmas de unidades curriculares obrigatórias

Validação experimental

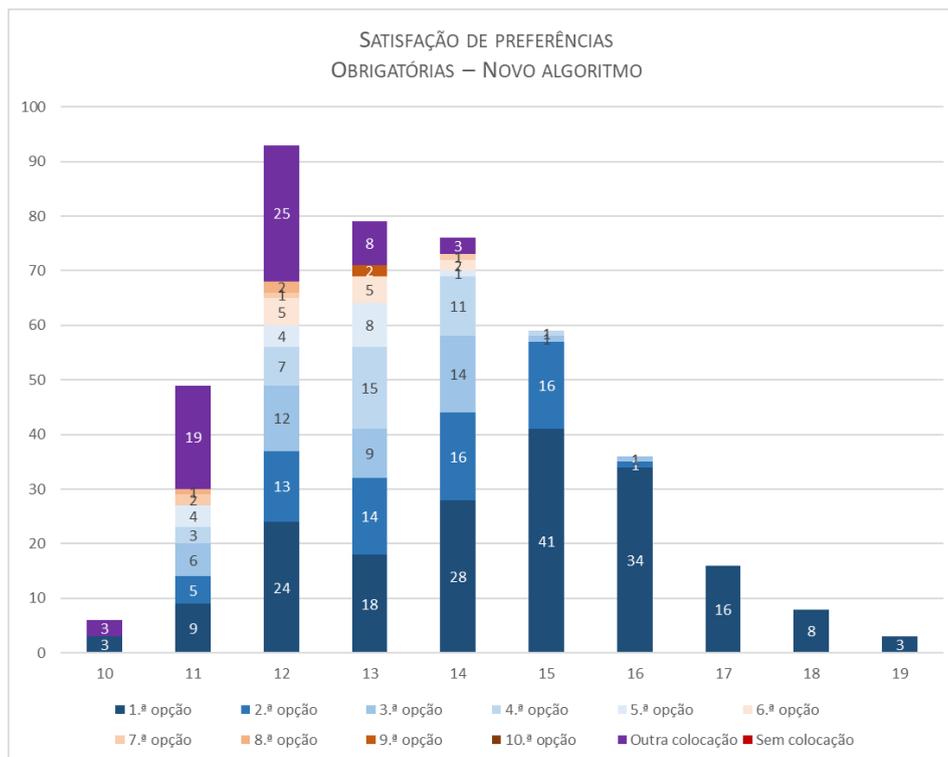


Figura 19. Satisfação de preferências dos estudantes no novo processo de atribuição de turmas de unidades curriculares obrigatórias

1044 No contexto das unidades curriculares obrigatórias, visto que tanto o número de
 1046 estudantes candidatos como de turmas disponíveis é significativamente maior,
 observam-se resultados ainda melhores. A taxa de colocação do processo atual (321 em
 425 estudantes, ~76%) foi maximizada para a totalidade da população (100%) no novo
 processo, sendo que, nalguns casos reduzidos (58 estudantes: ~14%) que possuíam
 1048 médias mais baixas, o algoritmo os colocou numa combinação de turmas que não
 haviam indicado em nenhuma das suas preferências. Apesar de tudo, este número é
 1050 inferior ao número de estudantes que ficaram sem colocação no processo atual (104
 estudantes: ~24%). Em qualquer caso, ainda existem muitos estudantes que ficaram
 1052 colocados na sua 1.ª opção, sendo que, se expandirmos esta estatística para incluir os que
 conseguiram colocação na sua 2.ª opção, obtemos 271 estudantes (~64% da população).

1054 Seguidamente, nas figuras 20 e 21, analisam-se as taxas de subocupação (número
 de estudantes a menos do que o número que seria ideal para se observar uma taxa de
 1056 ocupação uniforme entre todas as turmas).

Validação experimental

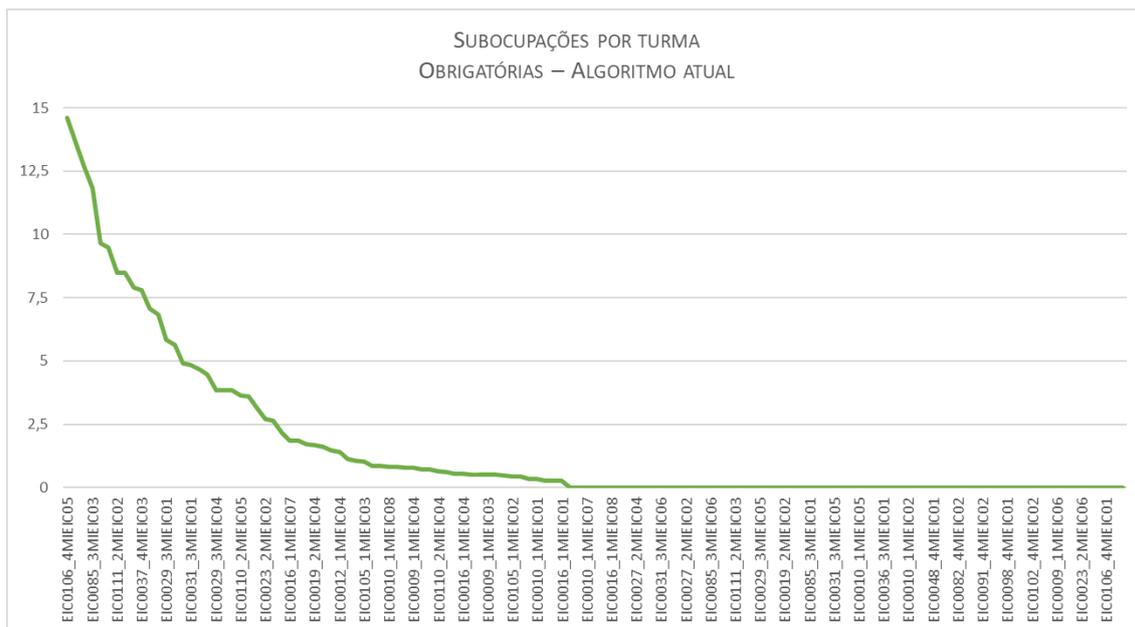


Figura 20. Taxas de subocupação por turma no processo atual de atribuição de turmas de unidades curriculares obrigatórias



Figura 21. Taxas de subocupação por turma no novo processo de atribuição de turmas de unidades curriculares obrigatórias

Observa-se que as turmas ficam, no geral, muito bem equilibradas com o novo processo. A turma onde se verifica uma taxa maior de subocupação (~5 alunos) faz, aliás, parte de uma unidade curricular onde os estudantes são obrigatoriamente atribuídos por via administrativa e, deste modo, muitos deles não chegam a indicar nenhuma preferência para esta UC aquando do processo de escolha de turmas.

Validação experimental

1062 Por fim, na tabela 4, é realizada uma síntese de indicadores que possibilitam
1066 averiguar cada um dos critérios que o novo algoritmo considera na sua função-objetivo:

Tabela 4. Estatísticas globais comparando o processo atual e o novo processo de atribuição de turmas de unidades curriculares obrigatórias

Variável	Algoritmo atual	Novo algoritmo
Número de estudantes com preferências	428	
Estudantes colocados	323 (76%)	428 (100%)
Colocados em todas as UCs	281 (66%)	425 (99,3%)
Colocados em parte das UCs	42 ⁴ (10%)	3 (0,7%)
Colocados numa das suas opções	323 (76%)	369 (86%)
Colocados em 1.^a opção	235 (55%)	184 (43%)
Colocados fora das suas opções	0 (0%)	59 (14%)
N.º médio de pares UC-turma fora das opções por estudante	N/A	3,7 ⁵
N.º médio de blocos ocupados por estudante	N/A	5,8
N.º médio de blocos ocupados fora das opções por estudante	N/A	0,1
Desvio de colocação nas turmas em relação a ocupação uniforme	207 (14%)	52 (2,5%)

1064 À parte daquilo que já se pôde observar nas imagens anteriores, podem destacar-se
1066 dois pormenores: em primeiro lugar, os 3 estudantes que, no novo processo, ficaram
1068 colocados apenas em parte das unidades curriculares em que se inscreveram
apresentaram opções inválidas, nas quais uma ou duas UCs optativas entravam em
conflito de horário com a aula prática de outra UC obrigatória que, devido às restrições
implementadas no algoritmo, é impossível acontecer.

1070 Por outro lado, verifica-se que, dentro dos estudantes que ficaram colocados fora
das suas opções, o algoritmo faz um bom trabalho a atribuir-lhes turmas, visto que, em

⁴ Estudantes que ficaram colocados numa opção que não incluía a totalidade das unidades curriculares que estes pretendiam frequentar.

⁵ Valor não incluído como critério de otimização, mas que seria interessante fazer como trabalho futuro.

Validação experimental

1072 média, cada estudante terá de frequentar aulas em ~6 blocos (manhãs/tardes), o que
parece ser um valor dentro do habitual, e praticamente ninguém terá de frequentar aulas
1074 em blocos que nunca escolhera numa das suas opções (observa-se apenas uma média de
0,1 blocos não pretendidos por cada aluno).

1076 **5.2 Apresentação e validação presencial com *stakeholders***

No dia 7 de junho de 2017, foi ainda realizada uma reunião nas instalações da FEUP
1078 entre o autor, Gustavo Silva, o orientador da dissertação, Dr. João Pascoal Faria, e os
seguintes elementos:

- 1080 • Susana Gaió – responsável do centro de informática da FEUP e encarregada de
gerir os processos de importação e exportação de dados no SIGARRA;
- 1082 • Dr.^a Ana Paula Rocha – responsável no MIEIC pelo processo de inscrições de
alunos em turmas;
- 1084 • Estudantes:
 - 1086 ○ Jorge Teixeira – 5.^o ano, anterior membro da comissão de
acompanhamento do MIEIC;
 - 1088 ○ Luís Duarte – 4.^o ano, membro da comissão de acompanhamento do
MIEIC;
 - 1090 ○ Marina Camilo – 4.^o ano, membro da comissão de acompanhamento do
MIEIC;
 - André Reis – 3.^o ano, representante do ano.

1092 O objetivo da reunião foi, por um lado, apresentar o novo sistema aos principais
stakeholders (interessados) no processo de inscrição em turmas e, por outro, pedir-lhes
1094 opiniões e esclarecer-lhes dúvidas.

Todos os presentes mostraram-se recetivos ao novo processo, tendo ainda
1096 apresentado sugestões, que serão mais tarde detalhadas como objetivos de melhoria do
trabalho. Tendo em conta os resultados apresentados, os participantes deram o seu
1098 parecer positivo à utilização, de forma experimental, do novo algoritmo aquando do
processo de inscrição em turmas do 1.^o semestre do ano letivo de 2017/2018. Visto que
1100 os dados de entrada de ambos os processos são os mesmos, é trivial recorrer-se ao
sistema anterior caso se verifique algum problema de maior.

1102 **Capítulo 6**

Conclusão

1104 **6.1 Avaliação do trabalho desenvolvido**

1106 O trabalho desenvolvido foi considerado bastante positivo em relação ao objetivo
1107 que se pretendia alcançar. O sistema ainda se encontra numa fase experimental, até
1108 porque ainda não foi realizado um número considerável de experiências com dados
1109 reais, mas será testado num ambiente real já no próximo semestre.

1110 Os resultados obtidos até ao momento são, no entanto, bastante encorajadores.
1111 Passar do sistema atual, que apenas percorre a lista completa de alunos um a um e não
1112 se preocupa em reservar algumas vagas para estudantes que, de outra forma, ficarão
1113 sem qualquer colocação, para o novo sistema, que, embora continuando a dar
1114 preferência aos estudantes com média mais elevada, se foca também na taxa geral de
1115 colocação, permitiu mitigar completamente o problema dos estudantes que chegam ao
1116 fim da primeira fase sem qualquer colocação e que se veem obrigados a construir um
1117 horário completo apenas com vagas sobrantes. Mesmo nos casos em que os estudantes
1118 ficam atribuídos a turmas que não escolheram nas suas preferências, o algoritmo tenta
1119 aumentar a qualidade do horário resultante, pois é esse, presumivelmente, o principal
1120 critério por detrás da escolha de turmas individuais.

1120 **6.2 Trabalho futuro**

1122 Como perspetiva de trabalho futuro, ainda é possível identificar diversos aspetos
1123 que podem ser melhorados. Seguem-se os principais pontos que poderão servir como os
1124 próximos passos no desenvolvimento deste sistema.

1124 **6.2.1 Generalização do processo para diferentes cursos da FEUP**

1126 Antes da criação de qualquer processo de inscrição em turmas da FEUP, os
1127 administradores podem definir a lista de critérios que pretendem que sejam tidos em
1128 conta pelo algoritmo durante a sua execução. O sistema desenvolvido contempla apenas
1129 o critério utilizado no MIEIC – a média académica dos estudantes –, mas noutros cursos
1130 são utilizados critérios diferentes, pelo que o sistema terá de ser modificado no futuro
para os considerar.

6.2.2 Recolha de preferências de turmas mais abrangentes

1132 Por outro lado, existe outro componente que pode estar, neste momento, a
1133 condicionar o número de soluções que se obtêm com o algoritmo e que poderia ser
1134 modificado: nomeadamente, a forma como os estudantes indicam as suas preferências.
De forma a propiciar uma maior “margem de manobra” ao algoritmo de colocação, seria
1136 útil que os estudantes pudessem indicar preferências mais alargadas/expandidas.

Uma das metodologias para se obter preferências expandidas seria permitir que os
1138 estudantes não indicassem turmas específicas, mas sim outros critérios mais latos, tais
como escolher determinados períodos de manhã/tarde que pretendessem que ficassem
1140 livres ou ocupados, preferir aulas lecionadas por um docente específico ou, por exemplo,
não indicar nenhuma preferência específica para uma determinada unidade curricular.

1142 **6.2.3 Integração direta com o SIGARRA**

Desde o início do desenvolvimento do novo sistema, sabia-se que a integração
1144 direta do mesmo no SIGARRA não poderia ser realizada em tempo útil desta
dissertação. No entanto, se o sistema demonstrar bons resultados nas fases
1146 experimentais que se seguirão (que, no início, serão conduzidas de forma manual), existe
a possibilidade de se investigar de que forma o sistema poderá ser integrado no
1148 SIGARRA para que o processo seja completamente automatizado.

Referências

1. Dostert, M., A. Politz e H. Schmitz, *A complexity analysis and an algorithmic approach to student sectioning in existing timetables*. *Journal of Scheduling*, 2015. **19**(3): p. 285-293.
2. Müller, T. *UniTime | University Timetabling*. [acedido a 26 de janeiro de 2017]; Disponível em: <http://www.unitime.org/>.
3. Müller, T. e K. Murray, *Comprehensive approach to student sectioning*. *Annals of Operations Research*, 2010. **181**(1): p. 249-269.
4. Müller, T., *Constraint-based Timetabling*. 2005.
5. Murray, K. e T. Müller. *Real-time student sectioning*. in *Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Application (MISTA 2007), Paris, France*. 2007.
6. Heitmann, H. e W. Brüggemann, *Preference-based assignment of university students to multiple teaching groups*. *OR spectrum*, 2014. **36**(3): p. 607-629.
7. Jablonský, J., *Benchmarks for Current Linear and Mixed Integer Optimization Solvers*. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, 2015. **63**(6): p. 1923-1928.
8. Alvarez-Valdes, R., E. Crespo e J.M. Tamarit, *Assigning students to course sections using tabu search*. *Annals of Operations Research*, 2000. **96**(1): p. 1-16.

9. Schindl, D., *Student sectioning for minimizing potential conflicts on multi-section courses*. 2016.
10. Busam, V.A., *An Algorithm for Class Scheduling with Section Preference*. *Communications of the Acm*, 1967. **10**(9): p. 567-&.