

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL



Construção de Calendários de Exames

João Miguel dos Santos Pedro Alves Natário

Mestrado em Estatística e Investigação Operacional
Especialização em Investigação Operacional

Dissertação orientada por:
Professor Doutor Pedro Martins Pereira Serrão de Moura

Resumo

Todos os anos, os estabelecimentos de ensino têm de criar calendários de exames. Uma grande maioria destes estabelecimentos ainda opta pela sua criação manual, resultando numa tarefa demorada, e muitas vezes, pouco eficiente. A Faculdade de Ciências da Universidade de Lisboa não é uma exceção, utilizando um calendário de exames já criado em anos anteriores, e adaptando manualmente ao ano letivo corrente. O problema da calendarização de exames pertence à classe de problemas NP-Difícil e, como tal, para obter soluções de boa qualidade em tempos aceitáveis são necessários algoritmos heurísticos. Um dos tipos de algoritmos utilizados para este género de problemas são os algoritmos de coloração de grafos. Nesta dissertação são propostos três algoritmos, testados em dez instâncias diferentes, para a criação de um calendário de exames direcionado para a época especial, uma época de exames na qual apenas dias antes da sua realização se sabe ao certo o número de alunos para cada exame. Os primeiros algoritmos utilizados neste trabalho são de coloração de grafos e coloração equitativa, para a criação do calendário de exames, e têm como objetivo uma distribuição mais uniforme, e justa para os alunos, dos exames pelos dias da época especial. O terceiro algoritmo desenvolvido é baseado em algoritmos de caminho ótimo, para a reorganização dos dias do calendário criado para a melhoria de uma restrição *soft*, a garantia de que todos os alunos tenham pelo menos um dia de intervalo entre os dois exames. Os resultados obtidos foram favoráveis para todas instâncias, sendo possível concluir que a utilização de algoritmos de coloração de grafos permite atingir resultados promissores, principalmente se complementados com outro tipo de algoritmos direcionados para a melhoria das restrições *soft*.

Palavras chave: Calendarização de Exames, Coloração de Grafos, Coloração Equilibrada

Abstract

Every year, educational institutions have to create exam schedules. A large majority of these institutions still create them manually, resulting in a time-consuming and often not very efficient task. The Faculdade de Ciências da Universidade de Lisboa is no exception, using an already existing exam schedule already created in previous years, and adapting it manually to the current academic year. The exam scheduling problem belongs to the class of NP-hard problems and, as such, to obtain good quality solutions in acceptable times heuristic algorithms are required. Graph coloring algorithms are one of the types of algorithms used for this kind of problems. In this dissertation three algorithms are proposed, tested in ten different instances, for the creation of an exam schedule aimed to the special examination period, an exam period in which only a few days before its realization it is known for sure the number of students enrolled in each exam. The two first algorithms used in this work are graph coloring and equitable coloring, for the creation of the exam schedule, with the goal of a more uniform, and fairer distribution for students, of exams over the days of the special period. The third algorithm developed is based on optimal path algorithms, for the reorganization of the days of the created schedule for the improvement of the soft constraint, the guarantee that all students have at least one day between the two exams. The obtained results were favorable for all instances, and it is possible to bring to a conclusion that the use of graph coloring algorithms allows the achievement of promising results, especially if complemented with other types of algorithms, aimed at improving the soft constraints.

Keywords: Exam Scheduling, Graph Coloring, Equitable Coloring

Agradecimentos

Quero começar por agradecer à minha família, pelo apoio ao longo da minha vida escolar e académica, e por me terem proporcionado os meios necessários para perseguir e alcançar os meus objetivos, apesar de ter sido mais demorado que o suposto.

Quero também agradecer aos meus amigos, que têm estado ao meu lado ao longo destes anos e que me proporcionaram memórias inesquecíveis, principalmente àqueles que me deram palavras encorajadoras e me continuaram a motivar nesta fase mais complicada da minha vida académica. Ainda um especial obrigado aos amigos que fiz na faculdade, sem essas pessoas teriam sido tempos mais aborrecidos, e a vida teria sido mais complicada.

Por último, mas não de todo menos importante, um grande obrigado ao meu orientador Pedro Moura pela paciência, a sua enorme disponibilidade para reunir regularmente comigo e com quem foi um prazer ter trabalhado.

Índice

Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
1.1 Contextualização do problema no universo da Faculdade de Ciências	1
1.2 Objetivo	2
1.3 Estrutura da Dissertação	2
2 Revisão de Literatura	5
2.1 Problemas de Calendarização	5
2.2 Problema da calendarização de exames	6
2.3 Diferentes Abordagens	8
2.3.1 Programação linear inteira mista	8
2.3.2 Programação por restrições	9
2.3.3 Técnicas de pesquisa local	9
2.3.4 Algoritmos populacionais	12
2.3.5 Técnicas Multicritério	15
2.3.6 Hiper-heurísticas	16
2.3.7 Técnicas de decomposição/agregação	17
3 Coloração de grafos	21
3.1 O problema	21
3.1.1 Coloração de Nós	21
3.1.2 Coloração de Arestas	23
3.1.3 Coloração de faces (Mapas)	23
3.1.4 Coloração de Lista	23
3.1.5 Coloração de Caminhos	24
3.1.6 Coloração Total	24
3.2 Algoritmos	25
3.3 Coloração equitativa	26
3.4 Aplicações	27
3.5 Coloração de grafos em calendarização de exames	28
3.5.1 Abordagens com coloração de grafos aplicadas à calendarização de exames . . .	28

4	Metodologia Utilizada	33
4.1	Algoritmo de Coloração de Grafos	33
4.2	Algoritmo de Coloração Uniforme	36
4.3	Algoritmo de Melhoria da Restrição <i>Soft</i>	38
5	Resultados computacionais	41
5.1	Dados e Instâncias	41
5.2	Resultados	42
5.2.1	Número mínimo de Cores	43
5.2.2	Construção do Calendário de exames	43
5.2.3	Restrição <i>Soft</i>	45
6	Conclusões e Trabalho Futuro	47
6.1	Conclusões	47
6.2	Trabalho Futuro	48
	Bibliografia	51
	Anexos	61

Lista de Figuras

3.1	Uma coloração de nodos com três cores.	21
3.2	Uma coloração de arestas com três cores.	23
3.3	Um exemplo de coloração de mapas. <i>In</i> [5]	24
3.4	Aplicação de coloração de faces ao grafo da Fig.3.1	24
4.1	Exemplo da Criação da Matriz de Pesos e Matriz Ordenada de Pesos	34
4.2	Exemplo da coloração de grafos.	37
4.3	Exemplo de solução após aplicação do algoritmo 3.	39

Lista de Tabelas

2.1	Restrições <i>hard</i> mais comuns em problemas de calendarização de exames	6
2.2	Restrições <i>soft</i> mais comuns em problemas de calendarização de exames	7
2.3	Resumo das diferentes abordagens à calendarização de exames	18
3.1	Resumo das diferentes abordagens com coloração de grafos à calendarização de exames	32
5.1	Descrição das Instâncias	42
5.2	Tempo de CPU e número mínimo de cores necessárias	43
5.3	Distribuição de exames pelos diferentes dias	44
5.4	Número de alunos com d dias de intervalo entre exames antes e depois da aplicação do Algoritmo 3	46
A.1	Calendário de exames da instância 1	61
A.2	Calendário de exames da instância 2	62
A.3	Calendário de exames da instância 3	62
A.4	Calendário de exames da instância 4	63
A.5	Calendário de exames da instância 5	63
A.6	Calendário de exames da instância 6	64
A.7	Calendário de exames da instância 7	64
A.8	Calendário de exames da instância 8	65
A.9	Calendário de exames da instância 9	65
A.10	Calendário de exames da instância 10	66
A.11	Alunos e respectivos exames (instância 1).	67
A.12	Alunos e respectivos exames (instância 2).	68
A.13	Alunos e respectivos exames (instância 3).	69
A.14	Alunos e respectivos exames (instância 4).	70
A.15	Alunos e respectivos exames (instância 5).	71
A.16	Alunos e respectivos exames (instância 6).	72
A.17	Alunos e respectivos exames (instância 7).	73
A.18	Alunos e respectivos exames (instância 8).	74
A.19	Alunos e respectivos exames (instância 9).	75
A.20	Alunos e respectivos exames (instância 10).	76

Capítulo 1

Introdução

O problema da calendarização, no geral, consiste na atribuição de meios tais como salas, pessoas ou máquinas a um período de tempo [73]. Este tipo de problemas pode ser encontrado em diferentes tipos de instituições de diferentes áreas. Por exemplo um tipo de instituições que todos os anos enfrenta este problema são as universidades, nas quais a calendarização se divide em calendarização de aulas e calendarização de exames. Nesta dissertação a calendarização de exames é aprofundada. Este tipo de calendarização consiste na afectação de exames a um número limitado de dias, períodos por dia (definidos por cada instituição) e a salas ou auditórios, e tem como objetivo a eliminação de conflitos (por exemplo, alunos com mais de um exame por dia).

O processo de calendarizar exames pode ser complicado e muito dispendioso a nível de tempo. O que dificulta este processo é a quantidade de restrições diferentes que se podem encontrar, sendo que algumas destas não devem ser violadas de todo enquanto outras chegam mesmo a ser contraditórias. Os problemas de calendarização de exames, assim como o da calendarização das aulas, têm sido bastante estudado nos últimos 50 anos, [18, 40, 73, 110, 122].

Apesar de já existirem inúmeras abordagens ao problema, as universidades ainda não parecem adeptas à troca dos seus métodos tradicionais manuais por novos métodos mais automatizados. No entanto, devido ao aumento de alunos matriculados no ensino superior ao longo das últimas décadas [124, 108], é fundamental dar o próximo passo e adotar técnicas mais eficientes e que permitam uma maior equidade para com os alunos [117]. Uma das abordagens estudadas no decorrer dos anos, e explorada nesta dissertação, é a coloração de grafos, onde o objectivo é atribuir diferentes cores aos nodos de um grafo de modo a que nenhum nodo tenha a mesma cor que o seu adjacente.

1.1 Contextualização do problema no universo da Faculdade de Ciências

Assim como nos demais estabelecimentos de ensino, todos os anos a Faculdade de Ciências da Universidade de Lisboa tem de lidar com a tarefa da construção do seu calendário de exames. Tal como muitas outras faculdades, a Faculdade de Ciências, mantém um método manual para a construção do seu calendário de exames. Ano após ano é utilizado um calendário de exames criado, uma única vez, através da colaboração entre os coordenadores de cada curso, presidentes de departamento e com o Gabinete de Organização Pedagógica (GOP) que, no início de cada ano letivo, após a sugestão das datas de cada coordenador de curso, caso existam alterações nos planos, o GOP insere as datas num ficheiro Microsoft Excel que, com a utilização dos filtros disponíveis, permite avaliar as restrições que constam

do Regulamento do Regime de Avaliação de Conhecimentos ¹ dentro de cada plano curricular e até mesmo a distribuição dos exames pelas salas.

Podendo, o método em uso, funcionar bem para a duas primeiras épocas de exames da Faculdade o mesmo não pode ser declarado para a época especial. Sendo uma época de exames com restrições adicionais tais como: apenas alunos finalistas e sob algumas exceções² a podem frequentar e no máximo cada aluno pode realizar apenas dois exames (no caso dos alunos finalistas, se estes acabarem o ano lectivo com mais de duas cadeiras sem aprovação perdem o direito à época especial). Devido a tais restrições, muitos alunos inscritos em exames desta época acabam por não poder comparecer ao(s) seus(s) exame(s), podendo mesmo haver exames que não tenham alunos para os realizarem na data marcada, estando assim a ocupar uma vaga que podia ser útil a outro exame.

Dado as inúmeras abordagens ao problema da calendarização de exames, exploradas nos capítulos subsequentes, faz sentido manter o método atual numa época de exames tão mutável? Por exemplo, se um departamento da Faculdade de Ciências tiver no total 100 cadeiras mas apenas 40 destas cadeiras têm alunos elegíveis para realizar os seus respetivos exames, com o método atual ficariam 60 vagas disponíveis. Seria possível, com o auxílio de um método aproximativo com um tempo de execução reduzido, distribuir os exames que serão realizados de uma maneira mais justa para com os alunos.

1.2 Objetivo

Nesta dissertação realizou-se um estudo sobre o problema da calendarização de exames e algumas das diferentes abordagens ao problema. Uma vez que este tipo de problemas pertence à classe NP-difícil [60], é proposto um algoritmo de aproximação de coloração de grafos, ao invés de um método exato, com o intuito de o aplicar ao caso específico da construção do calendário de exames da época especial da Faculdade de Ciências da Universidade de Lisboa. Os objetivos do algoritmo proposto são os seguintes:

- Encurtar o tempo da calendarização para uma época de exames, na qual apenas dias antes se sabe o número de alunos que potencialmente poderão realizar exames na época;
- Automatizar o processo da criação do calendário de exames;
- Evitar, se possível, que um aluno realize dois exames no mesmo dia e garantir que nunca tenha dois exames agendados para o mesmo período ou períodos seguidos;
- Tendo em consideração os pontos citados acima, garantir uma distribuição homogénea de exames pela semana da época especial.

1.3 Estrutura da Dissertação

A dissertação é composta por um total de seis capítulos. No Capítulo 2, é introduzido o tipo de problemas de calendarização, e além disso, o tema central da dissertação: o problema da calendarização de exames. Ainda no mesmo capítulo, são expostas algumas das diferentes técnicas para abordar este tipo de problema. No Capítulo 3 é apresentada a abordagem que será utilizada nesta dissertação: a Coloração de Grafos; assim como a sua origem e o diferente tipo de problemas nos quais esta técnica é geralmente utilizada. No Capítulo 4, são apresentados e explicados, passo a passo, os três algoritmos

¹https://ciencias.ulisboa.pt/sites/default/files/fcul/institucional/cpedagogico/RegulamentodeAvalicao_FCUL.pdf

²<https://ciencias.ulisboa.pt/estatutos-especiais>

desenvolvidos nesta dissertação. Os resultados computacionais, comentados, poderão ser consultados no Capítulo 5. Finalmente, o Capítulo 6 resume as principais conclusões obtidas através dos resultados obtidos no Capítulo precedente, terminando com algumas indicações para um possível trabalho futuro, a partir do trabalho realizado nesta dissertação.

Capítulo 2

Revisão de Literatura

2.1 Problemas de Calendarização

Desde os anos 60 que os problemas de calendarização têm trazido um enorme desafio tanto na área da Investigação Operacional como em Inteligência Artificial. Nos últimos anos, tem-se observado um aumento significativo na investigação deste tipo de problemas, inclusive mesmo o surgimento de uma conferência internacional, realizada semestralmente desde 1995, sobre a Prática e Teoria da Programação Automática (PATAT). Estas conferências têm servido como um fórum tanto para investigadores como trabalhadores na área para trocas de ideias e metodologias, assim como a criação de um grupo de trabalho em calendarização automatizada, em 1996, dentro da Associação Europeia das Sociedades de Investigação Operacional(EURO, <https://www.euro-online.org>).[110]

Os problemas de calendarização surgem em inúmeras instituições de grande dimensão de variadas áreas como, por exemplo, em Hospitais na criação de escalas para enfermeiros e médicos, calendários de eventos desportivos, horários para empresas de transporte, calendarização de exames em instituições de ensino, entre outros variados exemplos. No entanto, é um tipo de problema de enorme dificuldade e no qual a obtenção de soluções ótimas consome muito tempo, porque além de envolver muitas restrições e um extenso espaço de soluções que, apesar de finito, cresce exponencialmente com a dimensão do problema, existe também uma enorme ausência, em muitos casos, de ferramentas capazes de os resolver eficazmente [90]. O problema da calendarização é ao mesmo tempo um problema de afetação e um problema de escalonamento pois consiste em calendarizar um conjunto de eventos (aulas, exames, eventos desportivos, cirurgias, viagens) e afetar a um conjunto de recursos (professores, alunos, médicos, veículos) num espaço (salas de aula, auditórios, estádios, consultórios) dentro de um tempo estabelecido [46]. Estes problemas são, não só teoricamente como também computacionalmente, intratáveis. [47]

Desde 1961, começando com Appleby *et al.*[12], Bush *et al.*[39] e Gotlieb [75] que alguns algoritmos foram desenvolvidos e usados para problemas de calendarização automatizados. No entanto devido à magnitude das instâncias reais do problema, quase todos os métodos para obtenção de soluções são heurísticas para aplicações no mundo real, não garantindo a otimalidade das soluções. Em 1985, Dewerra [59] reviu alguns elementos básicos e modelos para problemas de calendarização baseados em métodos de grafos e redes. Carter em 1986 [40] apresentou um estudo, em ordem cronológica, sobre aplicações de sucesso à calendarização de exames que usam várias heurísticas de coloração de grafos, que eram populares na altura.

Em 2004 Burke *et al.*[31] definiram de uma forma geral os problemas de calendarização: “Um problema de calendarização é um problema com quatro parâmetros: um conjunto finito de intervalos de tempo (períodos) T , um conjunto finito de recursos R , um conjunto finito de reuniões M e um conjunto

finito de restrições C. O problema é a atribuição dos períodos e recursos para as sessões, de modo a satisfazer as restrições até onde for possível.”

Entre a vasta variedade de problemas de calendarização, calendarização na educação é um dos mais estudados. Todos os anos em todas as instituições de ensino enfrenta-se esse problema seja anualmente, semestralmente ou noutros períodos definidos pela instituição. Sendo a qualidade importante para todos os envolvidos, professores, alunos, etc. Dentro da calendarização na educação existem algumas variantes como, por exemplo, a criação de horários de aulas, horários para professores e turmas, escalonamento de seguranças (se os tiverem), calendários de exames entre outros. Existe uma enorme proximidade entre calendarização de exames e a calendarização de aulas [116], não existindo muitas diferenças significativas [95]. Nesta dissertação o foco será em calendarização de exames.

Após a publicação do artigo de Carter, acima referido, seguido por um artigo, de forma esclarecedora, publicado por Carter e Laporte em 1996 [41], inúmeros artigos nesta área têm sido publicados [110].

2.2 Problema da calendarização de exames

A resolução de problemas de calendarização de exames é uma tarefa recorrente e que consome bastante tempo [110]. Estes problemas consistem na atribuição de um conjunto de exames a um número limitado de intervalos de tempo (períodos) e salas com as suas respetivas capacidades, sujeitos a um dado conjunto de restrições. A complexidade deste tipo de problemas aumenta consoante a larga variedade de restrições, algumas das quais podendo ser algo contraditórias, têm de ser satisfeitas nas diferentes instituições. Devido não só ao fator mencionado anteriormente, como também ao facto de muitas vezes não ser possível obter uma solução ótima, o problema da calendarização de exames pertence à classe de problemas NP-difícil [60]. Na bibliografia dos problemas de calendarização, as restrições estão divididas em dois grupos diferentes: restrições *hard* e restrições *soft*.

As restrições *hard* (ver Tabela 2.1) não podem ser violadas em circunstância alguma. Por exemplo no problema em estudo os alunos não podem ter dois exames no mesmo período, nenhuma sala pode exceder a sua capacidade, entre outros. Uma solução de um problema de calendarização que cumpra todas as restrições deste tipo é dita admissível.

Tabela 2.1: Restrições *hard* mais comuns em problemas de calendarização de exames

Restrições <i>hard</i> mais comuns
1. Nenhum exame pode ter os mesmos alunos, professores ou salas atribuídas ao mesmo tempo.
2. O recursos dos exames têm de ser suficientes (i.e. o número de inscritos num exame tem de ser menor ou igual à capacidade da sala).

As restrições *soft* (ver Tabela 2.2) são restrições que idealmente se querem cumpridas, mas não são indispensáveis, sendo geralmente atribuída uma penalização na função objetivo dependendo da importância de cada restrição. Normalmente, na prática é impossível encontrar soluções admissíveis que respeitem todas as restrições *soft*. Estas restrições costumam variar de instituição para instituição no que diz respeito às restrições em si e à sua importância. Tentar calendarizar os exames que tenham alunos em comum no maior intervalo de tempo possível, de maneira que possibilite a revisão das matérias, é uma das restrições *soft* mais utilizadas na bibliografia dos problemas de calendarização de exames. Um exemplo de restrição que possa contradizer o exemplo anterior poderia ser a necessidade de marcar os exames com um maior número de alunos inscritos, o mais cedo possível para que haja tempo sufici-

ente para serem corrigidos. A qualidade dos horários geralmente é medida verificando a quantidade de restrições *soft* que são violadas nas soluções geradas [110].

Tabela 2.2: Restrições *soft* mais comuns em problemas de calendarização de exames

Restrições <i>soft</i> mais comuns
1. Distribuir o mais uniformemente possível exames em conflito (com alunos em comum).
2. Conjuntos de exames têm de ser agendados no mesmo local, no mesmo dia ou na mesma localização
3. Exames que, por alguma razão, tenham de ser consecutivos ou seguir uma determinada ordem.
4. Agendar todos os exames, ou os exames com maior número de inscritos, o mais cedo possível na época.
5. As precedências dos exames (caso existam) têm de ser satisfeitas.
6. Número limitado de alunos e/ou exames em qualquer período.
7. Exigências temporais (p.ex. alguns exames têm de ser num determinado período).
8. Exames com alunos em comum, agendados no mesmo dia, a serem realizados perto uns dos outros, para evitar que os alunos demorem muito a deslocar-se entre exames.
9. O mesmo exame pode ser dividido em várias salas, de preferência no mesmo piso ou edifício.
10. Apenas exames com a mesma duração podem ser encaixados na mesma sala.
11. Necessidades de recursos adicionais (p.ex. computadores).

Os problemas de calendarização de exames podem ser divididos em duas categorias: não capacitados ou capacitados. Nos primeiros não são consideradas as capacidades das salas de aula. Já nos segundos, estas são consideradas como uma restrição *hard* (sendo estas mais aproximadas das situações reais), juntamente com as outras restrições *hard* habitualmente utilizadas. A maioria da pesquisa vista na bibliografia tem-se focado nos problemas não capacitados, sendo o conjunto de dados de Toronto [43] um bom exemplo. A investigação neste conjunto de dados tem-se concentrado maioritariamente no desempenho dos algoritmos para produzir rapidamente soluções perto do ótimo. No entanto, McCollum [95], Qu *et al.* [110] e Carter e Laporte [41] consideraram que os investigadores não estavam a considerar todos os aspetos do problema, isto é, meramente trabalharam com uma versão simplificada do problema dos exames, na qual apenas abordam algumas restrições *hard* e *soft* mais comuns.

Os problemas capacitados aproximam-se mais dos problemas do mundo real uma vez que incluem as restrições de capacidade de salas, além de que, uma pesquisa realizada por Burke *et al.* [33] mostra que 73% das universidades concordam que dividir os exames por diferentes salas é um problema difícil por si só. As dificuldades são causadas, por exemplo, pela falta de salas devido a estas estarem em uso para aulas e por vezes os exames terem de ser divididos entre duas ou mais salas. Apenas a quantidade de alunos inscritos num exame é razão suficiente para dividir o exame por diferentes salas, adicionando restrições ao problema tais como dividir o exame por edifícios diferentes e/ou a distância entre salas.

Contudo, o problema capacitado tem recebido menos atenção da comunidade de investigação, provavelmente devido à falta de conjuntos de dados de referência. Além do mais, requer dados mais completos sendo que necessita de incluir as capacidades das salas, e essa informação pode ser difícil de obter. Alguns exemplos de conjuntos de dados disponíveis na bibliografia para os problemas capacitados são os conjuntos de dados de Nottingham [37], de Melbourne [99], da competição internacional de calendarização (ITC2007)[96] e de Toronto, o qual inclui a capacidade total de lugares sentados, introduzido em [37] [82].

Pode-se dizer que a cronologia dos estudos de calendarização de exames acompanha a evolução dos

algoritmos de otimização e equipamento informático, uma vez que, ao longo dos anos praticamente todas as técnicas de otimização foram aplicadas ao problema em discussão. Notando-se pela diferença entre os primeiros estudos de 1964, os quais eram mais direcionados para vários métodos construtivos como se pode ver em [48], onde são utilizadas tabelas ordenadas e [25], onde a "minimização" é alcançada através da avaliação de um conjunto de equações não lineares; até técnicas mais avançadas como em [13], onde se usa uma abordagem em três fases envolvendo Afetação Quadrática e [42], com o auxílio de um sistema robusto e flexível para a calendarização de exames, o EXAMINE.

Devido às diferenças entre requerimentos e restrições para cada instituição de ensino, não existe nenhuma solução ou formulação que possa ser obtida para a generalidade deste tipo de problemas. Por essa mesma razão, este problema é um dos tópicos mais trabalhados pelos investigadores. Com esta diversidade nascem inúmeras abordagens diferentes para a resolução do problema, conforme poderá ser consultado no tópico seguinte.

2.3 Diferentes Abordagens

Nos últimos anos tem havido uma quantidade significativa de investigação sobre a calendarização de exames criando novas e diferentes abordagens ao problema, sendo algumas delas métodos exactos, enquanto outras são métodos aproximativos. Muitas das metodologias de sucesso que têm aparecido representam hibridizações de algumas técnicas. Nesta secção serão apresentadas algumas dessas diferentes abordagens, que podem também ser consultadas na tabela resumo 2.3.

2.3.1 Programação linear inteira mista

Desde que Dantz [9] desenvolveu um algoritmo para resolver modelos de Programação linear inteira mista em 1946, tem existido bastante desenvolvimento e foco em Programação linear inteira mista no decorrer da história. A Programação linear inteira mista tem sido utilizada por vários investigadores para resolver problemas da calendarização. As técnicas para a resolução destes problemas que usam Programação linear inteira mista derivam do seu objetivo principal de otimizar uma função objetivo linear, maximizando ou minimizando, sujeita às suas respectivas restrições. Uma série de investigações em otimização de problemas de calendarização têm sido desenvolvidas usando modelos de programação inteira, mista e modelos binários que se enquadram dentro da Programação linear inteira mista e apenas diferem na formulação da função objetivo e restrições. As soluções para o problema da calendarização que utilizam este método variam entre as instituições que o usam, uma vez que os parâmetros diferem entre si, e até mesmo o conjunto de restrições *soft* pode variar. Geralmente este método é aplicado sem ser combinado com outras abordagens, no entanto algumas heurísticas de construção podem ser usadas juntamente com o modelo de Programação linear inteira mista, facilitando a análise das restrições. Quanto maior for o número de parâmetros e o número de variáveis e restrições maior será o tempo para chegar a uma solução ótima, numa relação exponencial, sendo uma desvantagem para o uso de modelos de Programação linear inteira mista [73].

Em 2004 Daskalaki, Birbas e Housos [57] apresentaram um método de Programação linear inteira mista para resolver os problemas de calendarização de exames nas universidades no qual o objetivo é atribuir um conjunto de cadeiras entre os professores e grupos de alunos e ainda um conjunto de pares de períodos diários e semanais.

Em 2005, Daskalaki e Birbas [56] apresentaram um método de relaxação em duas etapas baseado em Programação linear inteira mista para gerar soluções eficientes de calendarizações em duas etapas.

Durante a primeira etapa, as sessões de cadeiras que deviam ser realizadas consecutivamente foram agendadas através da atribuição de cadeiras a dias e tempos especiais. Durante a segunda etapa, devia ser assegurado que as cadeiras (apresentadas ao mesmo grupo de alunos) que requerem mais de um período fossem agendadas consecutivamente.

Em 2008 Bakir e Aksop [20] apresentaram uma abordagem de Programação linear inteira mista binária para organizar cadeiras e professores, de acordo com os períodos e salas disponíveis, resultando numa insatisfação dos alunos e professores durante a implementação de regras finitas por um conjunto de restrições em simultâneo.

Em 2017 Aslan *et al.*[14] desenvolveram um modelo de Programação linear inteira mista binária com o objetivo de minimizar o número de salas de aula e, por consequência, o número de vigilantes.

2.3.2 Programação por restrições

Programação lógica com restrições e técnicas de satisfação de restrições originaram-se na investigação de inteligência artificial. Estes dois métodos têm atraído a atenção de investigadores na área da calendarização devido à facilidade e flexibilidade com que podem ser usadas neste tipo de problemas. Os exames são modelados enquanto variáveis com domínios finitos. Os valores dentro dos domínios (representam os períodos e salas) são atribuídos de forma sequencial às variáveis para construir soluções para os problemas.

Normalmente as técnicas de satisfação de restrições são computacionalmente dispendiosas devido ao facto do número de possíveis atribuições aumentarem exponencialmente com o número de variáveis. Sozinhas, estas técnicas, geralmente são incapazes de gerar soluções de alta qualidade comparando com abordagens de última geração em problemas de optimização, por isso, são integradas diferentes heurísticas e técnicas com o intuito de reduzir a complexidade temporal para a resolução de problemas práticos.[110]

Em 1998 David P.[58] aplicou técnicas de satisfação de restrições nas quais soluções parciais eram obtidas inicialmente, e baseando-se nessas soluções parciais estratégias de reparação local específicas eram aplicadas sucessivamente para obter soluções completas e melhorias.

Em 1999 Reis e Oliveira [111] desenvolveram um sistema de calendarização baseado em ECLiPSe, no qual um conjunto de restrições *soft* e *hard* de um problema era introduzido em um modelo de satisfação de restrições, onde as variáveis definidas eram tratadas pelas bibliotecas do ECLiPSe.

Em 2003 Merlot *et al.*[99] usaram uma linguagem de programação de optimização chamada OPL para produzir soluções iniciais, aplicando *simulated annealing* e *hill climbing* para melhorar as soluções.

Em 2006 Le Huédé *et al.*[80] realizaram uma pesquisa interessante, integrando optimização Multi-critério em programação de restrições. Diferentes estratégias de pesquisa foram definidas para critérios diferentes e foram escolhidas dinamicamente durante a pesquisa. As variáveis foram ordenadas pelo grau maior, com estratégias de marcação definidas para cada critério.

2.3.3 Técnicas de pesquisa local

Métodos de pesquisa local são uma família de técnicas gerais que resolvem problemas através da pesquisa de uma solução selecionada para a sua vizinhança. Estruturas de vizinhanças diferentes e operadores de mudança dentro do local de pesquisa distinguem as diferentes técnicas de pesquisa local. A pesquisa é guiada por uma função objectivo definida, que por sua vez é utilizada para avaliar a qualidade dos horários gerados.

Estas técnicas representam uma grande parte do trabalho que tem aparecido no final dos anos 90 e início da década de 2000. Além do mais, têm sido aplicadas a uma variedade de problemas de calendarização, devido à relativa facilidade que diferentes restrições conseguem ser tratadas. O desempenho e eficácia destas técnicas são altamente dependentes dos parâmetros e propriedades do espaço de pesquisa.[110]

Em 2019 Cheiraitia, Haddadi e Salhi [47] sugeriram um novo algoritmo híbrido que combina o algoritmo da propagação de plantas e pesquisa local. Uma testagem extensiva no conjunto de dados padrão da universidade de Toronto, e em comparação com um número grande de não só métodos novos como também já bem estabelecidos, mostra que esta nova meta heurística é competitiva e representa uma adição substancial ao arsenal de ferramentas para resolver o problema da calendarização de exames.

Pesquisa Tabu

Na pesquisa Tabu, influenciada pelo trabalho proposto por Fred Glover [74], é explorado um espaço de pesquisa não revisitando uma lista de movimentos recentes, mantidos numa lista tabu. No entanto estes elementos podem ser selecionados no caso de gerarem a melhor solução obtida até ao momento, através do uso de uma técnica de aspiração. Caso contrário a pesquisa continua a encontrar outras soluções mesmo que sejam piores que as soluções selecionadas, com o objectivo de escapar de um ótimo local.

Em 2001 Di Gaspero e Schaerf [61] investigaram uma família de técnicas inspiradas em pesquisa tabu cujas vizinhanças diziam respeito às que contribuíam para a violação das restrições. Os comprimentos das listas eram dinâmicos e a função de custo era definida adaptativamente durante a pesquisa.

Também em 2001 White e Xie [130] desenvolveram uma pesquisa tabu em quatro fases chamada OTTABU, na qual as soluções eram melhoradas gradualmente por serem consideradas mais restrições em cada fase. Além da memória a curto prazo foi também usada uma memória a longo prazo para registar os passos mais ativos no histórico de pesquisa.

Em 2002 Paquete e Stüzle [105] desenvolveram uma metodologia de pesquisa tabu na qual prioridades ordenadas foram dadas para as restrições. As restrições eram consideradas de duas maneiras: uma restrição de cada vez por ordem decrescente de prioridades, onde empates eram decididos considerando as restrições de prioridade mais baixa; todas as restrições ao mesmo tempo por ordem decrescente de prioridades.

Em 2002 Alvarez *et al.*[10] criaram um método de pesquisa tabu com duas fases: a primeira fase está dividida em três passos, no primeiro passo gera-se um conjunto de soluções para cada aluno, no segundo passo há uma combinação entre um conjunto de soluções e a aplicação da pesquisa tabu com estratégias locais para obter horários de alta qualidade sem considerar as piores soluções para cada aluno, por último no terceiro passo atribui-se as salas e melhora-se a atribuição sem mudar a atribuição inicial das cadeiras aos períodos. Na segunda fase é utilizada a pesquisa tabu para melhorar a qualidade do horário inicial, além disso são usados também alguns movimentos: movimento simples, troca e múltiplas trocas.

Simulated Annealing

Outra técnica de pesquisa local é a *simulated annealing*, trata-se de uma pesquisa local probabilística que encontra uma solução ótima local para funções objetivo que possam ter vários ótimos locais. Esta técnica foi proposta por Kirkpatrick et al em 1983 [88]. Para resolver problemas de otimização combinatória, o algoritmo, traz conceitos da área da mecânica estatística. Motivado pelo processo de recozimento natural, a ideia é de pesquisar uma área de pesquisa maior no início do processo, aceitando

iterações piores com uma maior probabilidade, que reduz gradualmente com o decorrer da pesquisa. É usada uma temperatura dentro de um esquema de arrefecimento para controlar a probabilidade de aceitação das iterações piores na pesquisa. Vários parâmetros devem ser afinados nesta técnica incluindo as temperaturas iniciais e finais e o factor de arrefecimento, uma vez que afectam o desempenho e sucesso desta abordagem.

Em 1998, Thompson e Dowsland [125] desenvolveram uma abordagem de duas fases onde as soluções possíveis da primeira fase são introduzidas num processo de *simulated annealing* na segunda fase para melhorar a satisfação de restrições *soft*.

Em 2016, Cheraitia e Haddadi [46] propuseram uma nova abordagem para resolver o problema da calendarização de exames não capacitado. A sua heurística pode ser vista como um método de duas fases: na primeira fase, utilizando uma heurística simples de coloração de grafos, constroem um horário viável; na segunda fase, é aplicado *simulated annealing* para melhorar a qualidade do horário (minimizando a violação da restrição *soft*)

Em 2017, Leite, Melício e Rosa [90] propuseram uma nova variante do algoritmo de *simulated annealing* chamado FastSA para resolver o problema da calendarização de exames. Começaram por utilizar um algoritmo de construção, que utiliza a heurística de saturação de grau, para criar soluções iniciais válidas. O algoritmo FastSA executa um número inferior de análises comparado ao algoritmo normal de *simulated annealing*. Este algoritmo usa um critério de aceitação modificado nos quais os valores da temperatura são divididos em intervalos de temperatura, ou contentores de temperatura, onde um número igual de análises são computadas em cada contentor. Cada exame selecionado para agendamento é apenas mudado se esse exame tiver movimentos aceites no contentor de temperatura imediatamente precedente, é habitual existirem poucos ou até mesmo zero movimentos aceites no futuro, uma vez que se torna cristalizado. Por isso, os movimentos de todos os exames que foram fixados no decorrer do algoritmo já não são novamente avaliados, resultando num número inferior de análises em comparação com o algoritmo de referência.

Algoritmo do *Great Deluge*

Em 1993, Dueck [63] introduziu o algoritmo do grande dilúvio que opera de uma maneira semelhante ao *simulated annealing*. No entanto o *Great Deluge* utiliza um limite superior (ou o nível de água, como é referido comumente) enquanto limite de aceitação em vez de uma temperatura. O algoritmo inicia com um limite igual à solução inicial. Aceita soluções piores se o custo for inferior ao limite, o qual é reduzido em cada iteração de acordo com um valor pré-determinado (conhecida como a taxa de decaimento). Este algoritmo envolve apenas um parâmetro (a taxa de decaimento), o que é uma vantagem em relação ao *simulated annealing* uma vez que a eficiência de uma técnica meta heurística depende frequentemente da afinação de parâmetros.

Em 2003 Burke e Newall [36] implementaram o algoritmo do grande dilúvio de forma a resolver problemas da calendarização de exames. A taxa de decaimento é calculada multiplicando a solução inicial por um valor fornecido pelo utilizador e dividido pelo número de iterações. O algoritmo foi executado até duzentas milhões iterações e a pesquisa terminava se não houvesse melhorias no último milhão de iterações. Compararam o desempenho do algoritmo com *simulated annealing* e *hill climbing*, e reportaram que este algoritmo era superior a ambos.

Em 2004 Burke *et al.*[32] implementaram um algoritmo do *Great Deluge* com tempo predefinido para o problema da calendarização de exames, o algoritmo inclui dois parâmetros definidos pelo utilizador: tempo permitido para o algoritmo correr e a solução desejada (uma estimação). A taxa de decai-

mento é calculada como a diferença entre a solução inicial e a solução desejada dividida pelo tempo de computação (ou o número de iterações).

Em 2013 Kahar e Kendall [82] introduziram um algoritmo do *Great Deluge* modificado e expandido para o problema da calendarização de exames num caso real, que utiliza um parâmetro simples e fácil de entender. Investigando diferentes soluções iniciais, que são usadas enquanto um ponto de partida para o algoritmo do *Great Deluge*, assim com também alterando o número de iterações. Além do descrito, correram análises estatísticas para comprar os resultados usando os diferentes parâmetros. A metodologia proposta produziu soluções de boa qualidade quando comparada com soluções do algoritmo do *Great Deluge* original.

Juntamente com o estudo de maneiras diferentes de escapar a ótimos locais em técnicas de pesquisa local, alguns investigadores viraram-se para a investigação do efeito de formar vizinhanças diferentes, na qual têm obtido algum sucesso em problemas de calendarização, demonstrando, que não só o método de pesquisa, como também a estrutura da vizinhança pode ter um impacto significativo nos algoritmos de pesquisa.

2.3.4 Algoritmos populacionais

No início de um método populacional tem-se um número inicial de soluções, que será chamado de população inicial. A cada iteração um mecanismo de selecção é usado para seleccionar a(s) melhor(es) solução(/ões) dentro da população presente, depois são aplicadas mudanças, dependendo do método utilizado, nas soluções selecionadas de maneira a obter melhorias na solução, as soluções melhoradas são substituídas com a população danificada. Este procedimento continua até se alcançar uma solução desejável.

Algoritmos evolucionários

Dentro dos algoritmos populacionais existem os algoritmos evolucionários, sendo os algoritmos genéticos os algoritmos evolucionários mais estudados na investigação em calendarização de exames. Os algoritmos genéticos representam uma analogia com o processo de evolução na natureza, manipulando e evoluindo populações de soluções dentro do espaço de pesquisa.

Criado por John Holland no início dos anos 70 tem como objetivo encontrar a solução que encaixe num critério, neste caso as restrições. O primeiro passo é começar com um conjunto de diferentes soluções possíveis para problemas de calendarização. O grupo de soluções representará uma população e uma única solução é geralmente chamada de cromossoma. Cada solução tem características diferentes que podem ser mudadas de forma aleatória, resultando numa nova população [50]. Após as mutações ocorrerem, é selecionada a solução mais adequada.

A estratégia nos algoritmos genéticos difere principalmente das técnicas de pesquisa local, mencionadas acima, no sentido que se lida com várias soluções (população) duma só vez, ao invés de apenas uma solução ser melhorada ao longo de uma série de iterações. Várias faculdades têm aplicado esta técnica.

Em 1994 Corne *et al.*[51] apresentaram um breve estudo sobre o uso de algoritmos genéticos em calendarização educacional e abordou alguns problemas e perspectivas futuras. Uma contribuição do trabalho mostrou que o uso direto da representação em algoritmos genéticos é incapaz de lidar com determinadas estruturas de problemas em alguns problemas de coloração de grafos especialmente gerados.

Em 1998 Ross *et al.*[112] forneceram novas provas em relação à deficiência da utilização da codificação direta em algoritmos genéticos. Os autores observaram a falha de um número de abordagens, evolu-

cionárias e não evolucionárias, para resolver classes especiais de problemas de coloração de grafos e sugeriram que os algoritmos genéticos deviam antes pesquisar por algoritmos em vez de soluções.

Em 1999 Terashima-Marin *et al.* [123] investigaram a codificação não direta em algoritmos genéticos, onde estratégias de construção de soluções e heurísticas eram codificadas e não as próprias soluções. Os resultados promissores desta abordagem nos conjuntos de dados de Toronto mostram potencial da representação não direta dos algoritmos genéticos.

Em 2001 Erben [65] desenvolveu um algoritmo genético de agrupamento no qual codificação adequada, operadores de cruzamento e mutação especialmente criados, e funções *fitness* foram estudados. Os genes foram agrupados para cada cor em problemas de coloração de grafos.

Em 2002 Sheibani [118] construiu um modelo matemático especial e desenvolveu um algoritmo genético padrão para resolver problemas de calendarização de exames em centros de formação com o objetivo de maximizar os intervalos entre exames. Uma rede com atividades nos eixos foi utilizada para estimar a proximidade entre exames, sendo usada na função *fitness* no algoritmo genético.

Em 2005 Côté *et al.* [55] investigaram um algoritmo evolucionário biobjetivo com os objetivos de minimizar o comprimento do horário e de espaçar exames em conflito. Dois operadores de pesquisa local em vez de operadores combinatórios foram aplicados para lidar com as restrições *hard* e *soft*.

Em 2007 Ulker *et al.* [127] desenvolveram um algoritmo genético que usou o código de ligação linear como método representativo. Diferentes operadores de cruzamento foram investigados em conjunto com esta representação em coloração de grafos e problemas de calendarização de exames padrão (variante a de Toronto). As existências de resultados promissores indicam que esta codificação com operadores genéticos adequados era uma metodologia de pesquisa viável.

Em 2008 Cheong *et al.* [45] desenvolveram um algoritmo evolucionário multiobjetivo com o objetivo de minimizar não só o número de alunos a realizarem exames em períodos consecutivos como também o comprimento do calendário de exames e a satisfação de restrições *hard* tal como a capacidade de lugares e não mais que um exame em cada período para cada aluno. Este algoritmo utiliza uma representação de cromossomas com a dimensão das variáveis e incorpora um algoritmo micro-genético e um *hill climber* para exploração local e também um esquema de classificação de Pareto com base nos objetivos para atribuir a força relativa das soluções.

Em 2009 Khonggamnerd e Innet [87] usaram um algoritmo genético para ordenar horários de uma universidade onde a taxa de cruzamento era de 70%, e apesar de nenhuma restrição *hard* ter sido violada, as restrições utilizadas resultaram em uma maior ocupação de salas e capacidade das salas.

Em 2017 Shatnawi, Fraiwan e Al-Qahtani [117] apresentaram uma abordagem com duas etapas: na primeira é utilizado um algoritmo *greedy* e na segunda um algoritmo genético. Os dois algoritmos trabalham em conjunto para gerar o melhor calendário de exames. A automatização deste processo reduziu significativamente o número de conflitos, dias de exames e locais.

Em 2018 Dener e Calp [60] desenvolveram um algoritmo genético de duas fases: na primeira fase foram atribuídas as sessões a cada exame, na segunda os estudantes foram atribuídos a salas de exame. O objetivo deste algoritmo é a utilização do menor número possível dos edifícios utilizados em cada sessão, dos supervisores e das salas.

Algoritmos meméticos

Introduzido por Pablo Moscato [102] em 1989, os algoritmos meméticos são uma extensão dos algoritmos genéticos, combinados com o algoritmo de *hill climbing*, que têm em conta a ideia básica de que os indivíduos de uma população podem ser melhorados ao longo da sua vida. Num algoritmo

memético o conjunto de soluções possíveis é reduzido à otimização local de sub-espacos, sendo esta uma das melhores vantagens destes algoritmos.

Em 1996, Burke *et al.*[37] desenvolveram um algoritmo memético onde operadores de mutações leves e fortes foram empregues para reatribuir exames e conjuntos de exames, respetivamente. Por si só nenhuma destas mutações apresentava um melhoramento significativo na qualidade da solução. Foi usado *hill climbing* para melhorar os indivíduos de a qualidade dos horários, apesar de ser necessário um maior tempo computacional.

Algoritmo *Ant Colony*

Outro algoritmo populacional utilizado é o algoritmo de *Ant Colony*. Este algoritmo simula a maneira como as formigas encontram o caminho mais curto para a comida deixando um rasto de feromonas. Os caminhos mais curtos geram níveis de feromonas mais fortes ao longo de um certo período de tempo. No algoritmo, cada formiga é usada para construir uma solução e a informação ganha durante a pesquisa é guardada enquanto feromona, a qual é usada para ajudar a gerar soluções na fase seguinte do algoritmo.

Em 2004, Naji Azimi [17] implementou um sistema de *Ant Colony* e comparou-o com *simulated annealing*, pesquisa tabu e um algoritmo genético sob uma estrutura unida para resolver problemas de calendarização de exames sistematicamente concebidos. Soluções iniciais para o sistema de *Ant Colony* foram geradas heurísticamente e melhoradas através de uma pesquisa local. O resultados analisados ao longo do tempo de execução indicaram que a abordagem da colónia de formigas teve o melhor desempenho, apesar de não ter sido em todos os problemas, e a pesquisa tabu teve o melhor nível de melhoria sobre as soluções iniciais geradas aleatoriamente. Três variantes da hibridização da pesquisa tabu e método da colónia de formigas foram estudadas mais tarde em 2005 também por Naji Azimi [104]. Foi observado que as abordagens híbridas funcionavam melhor que cada algoritmo sozinho, sendo que o uso do sistema de *Ant Colony* seguido por uma pesquisa tabu obteve melhores resultados. No entanto, apenas dados gerados aleatoriamente foram usados para testar estes algoritmos.

Em 2005 Dowsland e Thompson [62] desenvolveram algoritmos de *Ant Colony* baseados no modelo de coloração de grafos estudado em [53]. Foram realizadas experiências extensivas para medir o desempenho do algoritmo com configurações diferentes. Os resultados obtidos foram competitivos com os resultados já existentes na mesma base de dados. Foi também observado que os métodos de inicialização tinham uma influência significativa na qualidade da solução.

Em 2007 Eley [64] comparou dois algoritmos de formigas modificados baseados nos sistemas de *Ant Max-Min* para calendarização de aulas [120], e o algoritmo de *Ant Colony* para coloração de grafos [53]. Foi observado que o sistema simples de *Ant Colony* superou o sistema de *Ant Max-Min* quando ambos os algoritmos foram hibridizados com um *hill climber*. O autor concluiu também que ajustando parâmetros pode melhorar consideravelmente o desempenho destes sistemas.

Algoritmo de otimização *Bee Mating*

Abbass propôs em 2001 [1] o algoritmo de otimização *bee mating*. Uma colónia de abelhas é composta por uma rainha (melhor solução), zangões (soluções existentes), abelhas-operárias (heurísticas) e por ninhadas (soluções experimentais). Este algoritmo simula o comportamento natural de acasalamento da rainha quando esta abandona a colmeia para acasalar com zangões. Após cada acasalamento bem-sucedido, o esperma do zangão é adicionado à espermateca da rainha. Antes do voo de acasalamento começar, a rainha é inicializada com alguma energia e apenas acaba o voo quando o seu nível de energia desce abaixo de um limite inferior (perto de zero). No final a rainha volta à colmeia, entretanto

começa a procriar escolhendo aleatoriamente o esperma de um zangão e efetua o cruzamento para produzir uma ninhada, sendo posteriormente alimentada por uma abelha-operária para melhorar as ninhadas. O número de abelhas-operárias utilizadas no algoritmo representa o número de heurísticas. Se a ninhada mais apta for superior à rainha, esta é substituída. Todas as outras ninhadas assim como a antiga rainha são destruídas e o processo volta a ser iniciado com uma nova rainha e os mesmos zangões.

Em 2012 Sabar *et al.*[115] propuseram uma variação do algoritmo de otimização *bee mating* para resolver problemas de calendarização na educação, testando-o em problemas padrão de calendarização de exames e disciplinas. Para gerar uma população de soluções possíveis iniciais são usadas três heurísticas híbridas de coloração de grafos. A melhor solução dentro da população inicial torna-se a rainha, as restantes tornam-se nos zangões. Após a determinação de qual o zangão escolhido, se o acasalamento for bem-sucedido o esperma do zangão é adicionado à espermateca. A seguir a rainha reproduz-se e as ninhadas formam-se. A abelha-operária, os autores utilizam um algoritmo de descida simples, é então aplicada às ninhadas resultantes. As ninhadas são melhoradas com a possibilidade de substituírem a rainha. O número de voos de acasalamento é pré-definido.

2.3.5 Técnicas Multicritério

Na maioria das abordagens em calendarização, penalizações em violações das diferentes restrições são somadas e usadas para indicar a qualidade das soluções. No entanto, em circunstâncias do mundo real, as restrições são consideradas de pontos de vista diferentes pelas diferentes entidades envolvidas nos processos de calendarização. A simples soma dos custos das diferentes restrições nem sempre resolvem a situação nesses casos. As técnicas de Multicritério têm sido estudadas em calendarização recentemente com o intuito de lidar com diferentes restrições facilmente considerando um vetor de restrições ao invés de uma soma de penalizações. Nas técnicas de Multicritério, cada critério pode ser considerado para corresponder a uma restrição, tendo um certo nível de importância e sendo lidado individualmente. Em algumas abordagens foram utilizadas múltiplas fases para lidar com diferentes objetivos.

Em 1995 Colijn e Layfield [49] aplicaram uma abordagem de múltiplas fases ao problema de calendarização de exames na Universidade de Calgary. Na primeira fase, exames individuais e conjuntos inteiros de exames em períodos de tempo foram movidos para reduzir o número de estudantes a realizarem dois exames de seguida. Na segunda fase, estudantes a realizarem três e quatro exames de seguida foram considerados usando um processo semelhante. Os autores consideraram também os casos nos quais os horários têm de ser modificados em circunstâncias imprevistas na segunda fase da abordagem, tendo um processo altamente interativo dentro de uma interface visual onde os exames podem ser movidos, adicionados ou removidos.

Em 2001 Burke *et al.*[26] desenvolveram uma abordagem Multicritério de duas fases a lidar com nove critérios em problemas de calendarização. Na primeira fase, foi usada saturação de grau para gerar um conjunto de soluções possíveis, nas quais cada critério era tratado individualmente. Na segunda fase as soluções eram melhoradas heurísticamente em simultâneo. Um método de Multicritério chamado *Compromise Programming* [132] foi utilizado onde a qualidade das soluções era avaliada pela distância entre estas e um ponto ideal representando soluções ótimas relativamente a todos os critérios. Esta técnica foi depois estudada em [107] e integrada no algoritmo do *Great Deluge* [32]. Estas abordagens forneceram a flexibilidade aos programadores de horários para obterem as soluções desejadas através da gestão dos pesos das diferentes restrições.

Já mencionado anteriormente, em 2006 Le Huédé [80] integrou otimização Multicritério em programação de restrições para resolver problemas de calendarização de exames. Foi usada uma heurística

para escolher critérios para escolher de forma dinâmica um critério para melhorar a última solução encontrada no que diz respeito a critérios específicos. Foi utilizada uma medida difusa para modelar a importância de cada critério numa função agregada sobre três critérios, nomeadamente o comprimento do horário, o número de salas utilizadas e distribuição dos exames.

Em 2016 Amaral e Pais [11] propõem o uso de quatro critérios para caracterizar a distribuição dos exames ao longo da época de exames. Foi considerado um conjunto de restrições relativas à não sobreposição de exames com estudantes em comum. Para lidar com os quatro critérios foi utilizado um programa de otimização multi objetivo e uma pesquisa Tabu foi implementada para encontrar uma boa solução viável para o problema. Foram propostas duas novas funcionalidades para aumentarem a automatização do algoritmo. Primeiro, utiliza um sistema de inferência *fuzzy* baseado em regras para escolher a posse dos elementos na lista tabu. Segundo, é proposta uma versão modificada da relação de compromisso, onde os pesos habituais fixados são substituídos por funções de peso para classificarem as soluções da vizinhança em cada iteração.

2.3.6 Hiper-heurísticas

Muitas das técnicas na bibliografia dependem de como o conhecimento de domínio, por exemplo a codificação das restrições *hard* e *soft*, foi implementado na técnica, impactando o desenvolvimento da meta-heurística na calendarização de exames. O facto de a maioria destes métodos representarem abordagens personalizadas para um problema específico, neste caso calendarização de exames, apresentam desvantagens uma vez que não são facilmente adaptados, ou por vezes não o são de todo, a outras instâncias de calendarização de exames. Por vezes o esforço de adaptar os parâmetros para servir novos problemas pode ser visto como tão complicado como desenvolver novas abordagens. Complicações como estas levaram alguns investigadores a desenvolver tecnologias com o intuito de trabalharem a um nível mais geral.

Em 2003 Ahmadi *et al.*[4] desenvolveram uma pesquisa em vizinhança variável para encontrar boas combinações de heurísticas parametrizadas para diferentes problemas de calendarização de exames. Foram empregues permutações de heurísticas de baixo nível (heurísticas de seleção de sete exames, seleção de dois períodos e seleção de três salas, entre outros) e os seus parâmetros associados (pesos) para construir soluções.

Em 2004 Ross *et al.*[113] desenvolveram um algoritmo genético de estado estável geral para pesquisar dentro de um espaço de pesquisa simplificado de descrições dos estados dos problemas para construir soluções. A pesquisa do algoritmo genético foi em heurísticas e não propriamente em soluções. Três funções diferentes de *fitness* foram testadas. As descrições do estado do problema foram estudadas experimentalmente com respeito a essas funções. Os resultados promissores para os problemas de referência de calendarização de exames e aulas demonstraram direções de investigação relativo a esta abordagem para uma série de problemas.

Em 2005 Kendall e Hussin [86, 85] investigaram uma Hiper-heurística e pesquisa tabu baseada no trabalho realizado em [34] onde ambas as estratégias móveis e as heurísticas de grafos construtivas foram empregues enquanto heurísticas de baixo nível. Os algoritmos foram testados em problemas de calendarização de exames da Universidade de tecnologia MARA e mostrou-se que produzia melhores resultados comparando com soluções geradas manualmente.

Em 2005 e 2006 Burke *et al.*[27, 30] investigaram o uso de raciocínio baseado em casos, uma técnica baseada em conhecimento prévio, enquanto um seletor de heurísticas para resolver problemas de calendarização de exames e aulas.

Em 2007 Burke *et al.*[35] investigaram o uso de pesquisa tabu para encontrar sequências de heurísticas de grafos para construir soluções para problemas de calendarização. Mostrou-se que a exploração sobre o grande espaço de soluções permitiu à abordagem obter bons resultados nos problemas de calendarização de exames e aulas.

Em 2007 Bilgin *et al.*[22] analisaram sete métodos de seleção de heurísticas e cinco critérios de aceitação dentro de uma Hiper-heurísticas, conduzindo um estudo empírico tanto em funções de referência como em problemas de calendarização de exames. Concluíram que diferentes combinações de métodos de seleção e critérios de aceitação funcionavam bem em diferentes problemas, apesar de algumas combinações funcionarem ligeiramente melhor do que outras nas instâncias testadas.

Ainda em 2007 Ersoy *et al.*[68] estudaram abordagens de hiper-heurísticas onde três hill climbers foram aplicadas em diferentes ordens dentro de um algoritmo memético. As abordagens foram comparadas com hiper-heurísticas de algoritmos meméticos auto adaptativos com diferentes seleções de heurísticas e critérios de aceitação. Mostrou-se que de todas as abordagens testadas, a que teve uma melhor prestação foi a hiper-heurística do algoritmo memético com um *hill climber* de cada vez.

Em 2019 Muklason *et al.*[103] desenvolveram uma hiper-heurísticas que usa um algoritmo de coloração de grafos para encontrar uma solução inicial que entra em otimização utilizando um algoritmo do *Great Deluge*. Os exames são ordenados de acordo com o número de alunos em comum entre si, do maior para o menor. Utilizando um algoritmo de saco-mochila, os exames são atribuídos sequencialmente a salas e períodos, de acordo com a sua ordem.

2.3.7 Técnicas de decomposição/agregação

Nestas técnicas os problemas maiores são divididos em subproblemas mais pequenos, para os quais soluções de alta qualidade podem ser obtidos através de técnicas relativamente simples, uma vez que os conjuntos de soluções são significativamente menores em relação ao problema original. Apesar de ter obtido algum sucesso, técnicas de decomposição em calendarizações não têm atraído muita atenção como seria de esperar devido a duas desvantagens. Primeiro, as primeiras atribuições podem levar a soluções impossíveis, que foi também uma complicação encontrada em métodos construtivos no início da investigação em calendarização. Segundo, soluções globais de alta qualidade podem-se perder devido a certas restrições *soft* não serem consideradas quando os problemas são decompostos. Os métodos de agregação estudados no início da pesquisa em calendarização podem ser vistos enquanto abordagens de decomposição no sentido que os exames são decompostos entre grupos livres de conflito ou com baixo conflito. Outra alternativa para decompor problemas é encontrando a clique maior em grafos subjacentes.

Em 1999 Burke e Newall [28] investigaram uma abordagem de decomposição utilizando heurísticas sequenciais para atribuir o primeiro conjunto de n exames que foram considerados como os mais difíceis por heurísticas de coloração de grafos. Foram empregues técnicas de *backtracking* e de antecipação para evitar atribuições precoces que levam a inviabilidades. Os exames atribuídos em fases anteriores foram fixados e o subproblema em cada fase foi resolvido por um algoritmo memético desenvolvido em [37]. Este algoritmo reduziu dramaticamente o tempo necessário, tendo também produzido soluções de grande qualidade nos dados de Toronto e Nottingham. Na altura da publicação este artigo tinha alguns dos melhores resultados nos problemas capacitados de referência. A técnica de decomposição era na verdade independente do algoritmo memético usado em casa subconjunto decomposto.

Em 2002 Lin [91] desenvolveu um algoritmo multiagente onde os problemas foram divididos em subproblemas e resolvidos por cada agente localmente. Um *broker* foi utilizado para resolver os restantes horários incluído aqueles que foram desalocados de horários locais. As soluções globais foram obtidas

através da agregação de todos os horários gerados pelos agente e pelo *broker*. Esta abordagem funcionou bem em problemas com horários mais livres, mas não tão bem em problemas mais densos.

Em 2007 Qu e Burke [109] investigaram uma abordagem onde os exames eram decompostos em dois conjuntos (difíceis e fáceis) pela dificuldade de os agendar em iterações anteriores da construção da solução. Desta forma, a dificuldade do problema é reduzida, uma vez que se tratava de dois espaços de pesquisa menores, enquanto a qualidade global dos horários era também considerada. A pequena porção de exames difíceis obtidos pela abordagem contribuiu significativamente para os custos dos calendários gerados. Esta abordagem obteve o melhor resultado numa das instâncias do problema na altura da sua publicação.

A tabela seguinte resume as diversas abordagens apresentadas neste capítulo.

Tabela 2.3: Resumo das diferentes abordagens à calendarização de exames

Ano	Artigo	Método
1994	Corne <i>et al.</i> [51]	Algoritmo Genético
1995	Colijn e Layfield [49]	Multicritério
1996	Burke <i>et al.</i> [37]	Algoritmo Memético
1998	Ross <i>et al.</i> [112]	Algoritmo Genético
1998	Thompson e Dowsland [125]	<i>Simulated Annealing</i>
1998	David P.[58]	Programação por restrições
1999	Reis e Oliveira [111]	Programação por restrições
1999	Burke e Newall [28]	Decomposição/ Agregação
1999	Terashima-Marin <i>et al.</i> [123]	Algoritmo Genético
2001	Erben [65]	Algoritmo Genético
2001	Burke <i>et al.</i> [26]	Multicritério
2001	Di Gaspero e Schaerf [61]	Pesquisa Tabu
2001	White e Xie [130]	Pesquisa Tabu
2002	Lin [91]	Decomposição/ Agregação
2002	Sheibani [118]	Algoritmo Genético
2002	Paquete e Stützle [105]	Pesquisa Tabu
2002	Alvarez <i>et al.</i> [10]	Pesquisa Tabu
2003	Ahmadi <i>et al.</i> [4]	Hiper-heurísticas
2003	Merlot <i>et al.</i> [99]	Programação por restrições
2003	Burke e Newall [36]	Grande Dilúvio
2004	Burke <i>et al.</i> [32]	Grande Dilúvio
2004	Ross <i>et al.</i> [113]	Hiper-heurísticas
2004	Naji Azimi [17]	<i>Ant Colony</i>
2004	Daskalaki <i>et al.</i> [57]	Programação linear inteira mista
2005	Daskalaki e Birbas [56]	Programação linear inteira mista
2005	Dowsland e Thompson [62]	<i>Ant Colony</i>
2005	Côté <i>et al.</i> [55]	Algoritmo Genético

(cont...)

Ano	Artigo	Método
2005	Kendall e Hussin [86, 85]	Hiper-heurísticas
2005 e 2006	Burke <i>et al.</i> [27] e [30]	Hiper-heurísticas
2006	Le Huédé <i>et al.</i> [80]	Programação por restrições e Multicritério
2007	Burke <i>et al.</i> [35]	Hiper-heurísticas
2007	Bilgin <i>et al.</i> [22]	Hiper-heurísticas
2007	Ersoy <i>et al.</i> [68]	Hiper-heurísticas
2007	Qu e Burke [109]	Decomposição/ Agregação
2007	Eley [64]	<i>Ant Colony</i>
2007	Ulker <i>et al.</i> [127]	Algoritmo Genético
2008	Cheong <i>et al.</i> [45]	Algoritmo Genético
2008	Bakir e Aksop [20]	Programação linear inteira mista
2009	Khonggamnerd e Innet [87]	Algoritmo Genético
2012	Sabar <i>et al.</i> [115]	<i>Bee Mating</i>
2013	Kahar e Kendall [82]	Grande Dilúvio
2016	Amaral e Pais [11]	Multicritério
2016	Cheraitia e Haddadi [46]	<i>Simulated Annealing</i>
2017	Leite, Melício e Rosa [90]	<i>Simulated Annealing</i>
2017	Aslan <i>et al.</i> [14]	Programação linear inteira mista
2017	Shatnawi <i>et al.</i> [117]	Algoritmo Genético
2018	Dener e Calp [60]	Algoritmo Genético
2019	Cheiraitia <i>et al.</i> [47]	Pesquisa Local
2019	Muklason <i>et al.</i> [103]	Hiper-heurística

Capítulo 3

Coloração de grafos

3.1 O problema

Em Teoria de Grafos, o Problema da Coloração de Grafos é um caso especial de etiquetagem de grafos. Consiste na atribuição de etiquetas tradicionalmente chamadas “cores” a elementos de um grafo, sujeita a certas restrições [44].

A Coloração de Grafos é um dos subcampos mais conhecidos e estudados da teoria de grafos. Na bibliografia, a coloração de grafos é vastamente explorada devido às inúmeras aplicações em problemas do mundo real, conforme será abordado adiante neste capítulo.

3.1.1 Coloração de Nodos

Seja um grafo $G = (V, E)$, onde $V = \{1, \dots, n\}$ é o conjunto de nodos e $E = \{(i, j) : i, j \in V, i \neq j\}$ o conjunto de arestas no grafo G . O problema da coloração de nodos consiste em colorir todos os nodos de um grafo, de tal forma que a quaisquer dois nodos i e j adjacentes¹, sejam atribuídas cores distintas. Em termos de teoria de grafos, uma coloração de nodos admissível é uma função [69]:

$$f : V \rightarrow \mathbb{N} : \forall i, j \in V : \exists (i, j) \in E \Rightarrow f(i) \neq f(j)$$

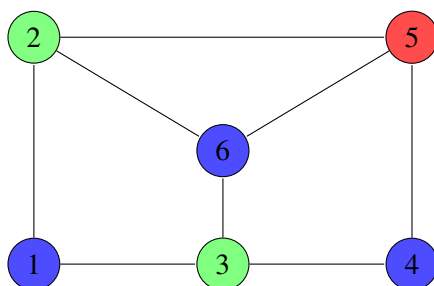


Figura 3.1: Uma coloração de nodos com três cores.

O objetivo do problema consiste em obter uma coloração de nodos admissível com o menor número de cores utilizadas. O número mínimo de cores necessárias para colorir um grafo é conhecido como o **número cromático de um grafo** denotado por $\chi(G)$. Uma coloração de G é considerada ótima se usar exatamente $\chi(G)$ cores [92, 76].

¹Dois nodos dizem-se *adjacentes* se existe uma aresta no grafo que os ligue, $(i, j) \in E$.

Em 1972, Karp [84] mostrou que o problema da coloração de grafos (nodos) é NP-completo. Garey *et al.*[72] e Stockmeyer [121] reforçaram este resultado mostrando que o problema do grafo k -colorável se mantém NP-completo para qualquer $k \geq 3$ [106]. Por essa razão investigadores têm tentado encontrar a solução para o problema adicionando novas técnicas aos algoritmos propostos, de modo a otimizar o tempo de execução [119].

A convenção da utilização de cores originou-se na coloração de regiões num mapa onde literalmente cada região é colorida. Em representações matemáticas e computacionais é habitual o uso dos primeiros números inteiros positivos ou não negativos enquanto “cores”. No geral uma pessoa pode usar qualquer conjunto finito enquanto “conjunto de cores”. A natureza do problema da coloração de grafos depende do número de cores, mas não no que estas representam. A coloração de grafos possui várias aplicações práticas assim como desafios teóricos. Diferentes limitações podem ser também definidas no grafo, ou na maneira como a cor é atribuída, ou até na cor em si. O problema até atingiu popularidade com o público em geral, disfarçado como o conhecido puzzle de números Sudoku. A coloração de grafos continua a ser uma área de pesquisa bastante ativa [44].

Principais Resultados

Os resultados apresentados de seguida foram retirados do documento Tópicos de Grafos, da cadeira de Grafos e Redes[76] da Licenciatura em Matemática Aplicada da Faculdade de Ciências da Universidade de Lisboa:

1. Seja G um grafo que contém pelo menos uma aresta. Então $\chi(G) = 2$ sse
 - (a) G é bipartido²;
 - (b) G não tem ciclos de cardinalidade ímpar.
2. Seja H um subgrafo do grafo G . Então $\chi(G) \geq \chi(H)$.
3. $\chi(G) \geq \theta(G)$. ($\theta(G)$ indica o cardinal da clique de cardinalidade máxima em G)
4. $\chi(G) \geq \left\lceil \frac{|V|}{\alpha(G)} \right\rceil$. ($\alpha(G)$ denota o cardinal do conjunto independente de cardinalidade máxima)
5. $\chi(G) \leq \max_{i \in V} \text{grau}(i) + 1$.³
6. Seja $G = (V, E)$ um grafo que não é completo⁴ nem é um ciclo de cardinalidade ímpar.
Então $\chi(G) \leq \max_{i \in V} \text{grau}(i)$.
7. Seja G um grafo planar⁵. Então $\chi(G) \leq 4$ ⁶

²Um grafo cujos nodos podem ser divididos em dois conjuntos disjuntos U e V tais que toda a aresta conecta um vértice de U a um vértice de V .

³ $\text{grau}(i)$ é o número de arestas $(i, j) \in E$.

⁴Um grafo completo é um grafo no qual cada par de nodos está conectado por uma aresta.

⁵Um grafo, que desenhado num plano de duas dimensões, admite uma representação em que as suas arestas não se intersectam.(por exemplo fig. 3.3b)

⁶Pelo teorema das quatro cores, não são necessárias mais do que quatro cores para colorir as regiões de qualquer mapa, de maneira que duas regiões adjacentes não tenham a mesma cor.

Outros resultados referentes aos números cromáticos de diferentes classes de grafos[128]:

- O número cromático de um grafo completo K_n é: $\chi(K_n) = n$;
- O número cromático de um grafo circular⁷ $C_n, n > 1$ é: $\chi(C_n) = \begin{cases} 3, & \text{se } n \text{ é ímpar;} \\ 2, & \text{se } n \text{ é par} \end{cases}$;
- O número cromático de um grafo estrela⁸ $S_n, n > 1$ é: $\chi(S_n) = 2$;
- O número cromático de um grafo roda⁹ $W_n, n > 2$ é: $\chi(W_n) = \begin{cases} 3, & \text{se } n \text{ é ímpar} \\ 4, & \text{se } n \text{ é par} \end{cases}$.

3.1.2 Coloração de Arestas

É também possível colorir as arestas (sendo esta a segunda maneira mais comum) e faces de um grafo. Equivalentemente à coloração nos nodos a coloração das arestas de um grafo G consiste em atribuir uma cor a cada aresta, satisfazendo a condição de que arestas que partilhem o mesmo vértice (arestas *incidentes* no mesmo vértice) não possam ter a mesma cor. Matematicamente tem-se uma função:

$$g : E \rightarrow \mathbb{N} : \forall a = (i_a, j_a), b = (i_b, j_b) \in E : i_a = i_b \vee i_a = j_b \Rightarrow g(a) \neq g(b)$$

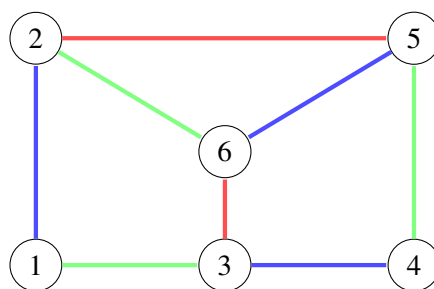


Figura 3.2: Uma coloração de arestas com três cores.

3.1.3 Coloração de faces (Mapas)

A coloração das faces de um grafo é o método utilizado para colorir áreas num mapa político. Este tipo de coloração requer que um grafo seja planar. De facto, a coloração de um mapa pode ser feita criando o *grafo dual* do grafo planar correspondente ao mapa, no qual cada nodo representa uma face do grafo planar, e cada dois nodos do grafo dual estão conectados se e só se as faces do grafo planar partilharem uma aresta. O problema da coloração do mapa passa a ser equivalente ao problema da coloração de nodos no grafo dual.

3.1.4 Coloração de Lista

Outro tipo interessante de coloração é a coloração de lista. Neste tipo de coloração, ao contrário da coloração de nodos ou arestas clássica, a cor possível atribuída a cada objeto (nodos ou arestas) de um

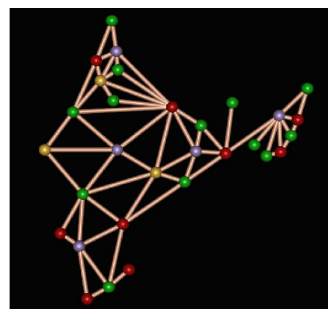
⁷Um grafo que é um único ciclo que passa por todos os seus n nodos.

⁸Grafo com n nodos, onde um dos nodos tem grau de $n - 1$ e os restantes nodos têm grau de 1.

⁹Um grafo que contém um ciclo de ordem $n - 1$, e onde cada nodo pertencente a esse ciclo está conectado a um outro nodo conhecido como o centro.

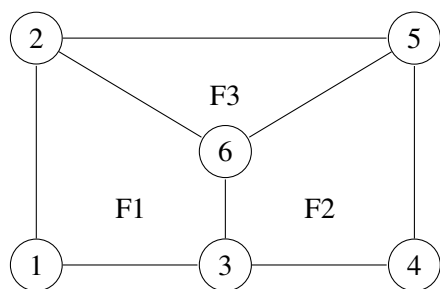


(a) Mapa da Índia

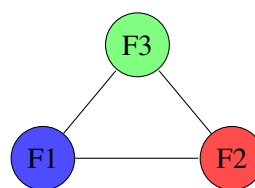


(b) Grafo dual do mapa da Índia

Figura 3.3: Um exemplo de coloração de mapas. In [5]



(a) Grafo planar com 3 faces.



(b) Grafo dual com 3 cores atribuídas.

Figura 3.4: Aplicação de coloração de faces ao grafo da Fig.3.1

dado grafo está limitada a um conjunto restrito, chamado de lista. Todos os objetos poderão ter listas diferentes. Em termos de teoria de grafos, uma coloração de nodos com lista é um método que atribui a todos os nodos uma cor das suas listas correspondentes de maneira que nenhum par de nodos adjacentes tenham a mesma cor. A coloração de arestas com lista pode ser definida analogamente, neste caso nenhum par de arestas incidentes é colorido com a mesma cor. Uma análise detalhada das propriedades e conjecturas da coloração de lista podem ser consultadas em [19] e [66].

3.1.5 Coloração de Caminhos

Um caso específico de coloração de arestas é a coloração de caminhos. Neste caso, é necessário colorir certos caminhos que ligam certos pares de nodos, e onde cada caminho contém um conjunto específico de arestas, satisfazendo a condição de não existirem dois caminhos que contenham a mesma aresta de um grafo. Erlebach e Jansen [67] mostraram uma visão geral da coloração de caminhos, assim como os seus resultados de complexidade.

3.1.6 Coloração Total

Uma última maneira de colorir um grafo é a coloração total, onde nodos e arestas são coloridos em simultâneo. Todos os pares de nodos adjacentes, quaisquer pares de arestas com um nodo em comum, assim como todos os pares incidentes (nodos, arestas) devem ser atribuídos cores diferentes. O número mínimo de cores necessário a construir uma coloração total para um dado grafo G é conhecido com o número cromático total, denotado por $\chi''(G)$

3.2 Algoritmos

Dada a dificuldade da generalidade do problema da coloração de grafos, vários investigadores têm desenvolvido algoritmos de aproximação a tempo polinomial com garantias de desempenho. A garantia de desempenho de uma heurística é o maior rácio, calculado para todos os grafos com n nodos, entre o número de cores utilizado pela heurística e o número mínimo requerido.

O primeiro algoritmo deste género a aparecer foi o de Johnson em 1974 [81] que, para qualquer grafo com n nodos dá uma garantia de desempenho de $O(n \log n)$. Foi apenas em 1983 que foi possível uma melhoria na garantia mencionada acima: Wigderson [131] desenvolveu um procedimento com uma garantia de desempenho de $O(n(\log \log n^2)/(\log n)^2)$. Berger e Rompel [21] alcançaram mais melhorias ao terem desenvolvido uma heurística com uma garantia de performance de $O(n(\log \log n)^3/(\log n)^3)$. Em 1993 Halldórsson [78] construiu uma heurística com uma garantia de desempenho de $O(n(\log \log n)^2/(\log n)^3)$. Em Karger *et al.* [83] é apresentado um algoritmo de tempo polinomial randomizado que utiliza no máximo $\min\{\tilde{O}(\Delta^{1-2/k}), \tilde{O}(n^{1-3/(k+1)})\}$ cores para colorir grafos k -coloráveis, onde Δ é o maior grau entre todos os nodos. Este algoritmo não só dá o melhor rácio de aproximação em termos de n , até à data, como também os primeiros resultados não triviais em termos de Δ [106].

Existem maioritariamente duas classes de algoritmos na bibliografia para resolver o problema da coloração de grafos: algoritmos aproximativos e algoritmos exatos. Os algoritmos exatos incluem a solução do problema da coloração de grafos baseada numa formulação em programação inteira, técnicas baseadas em decomposições lineares de um grafo, algoritmos de *branch and cut*, utilizando a formulação do conjunto de cobertura do grafo dado e baseada no algoritmo DSATUR.

A outra classe de algoritmos inclui heurísticas baseadas em sub-rotinas *greedy*, algoritmos evolucionários híbridos, reforçando a construção em otimização *ant colony* existente e colorindo nodos num grafo usando a junção de pesquisa local com programação de restrições [119].

Existem várias técnicas de heurísticas sequenciais para colorir um grafo. Uma delas é a heurística *greedy*. Esta técnica foca-se em escolher cuidadosamente o próximo nodo a ser colorido. Neste algoritmo heurístico, assim que um nodo seja colorido a sua cor nunca muda. Uma vez que não é um algoritmo exato, as diferentes ordenações dos nodos podem provocar diferentes números de cores para colorir um grafo, por vezes a mesma ordenação pode provocar a pior solução num grafo, mas a ótima noutra [89]. Algumas das diferentes técnicas de ordenação são: ordem aleatória, *first fit*, que pode ser implementado para correr em $O(n)$; ordenação baseada nos graus dos nodos: menor grau em último; maior grau primeiro, que pode ser implementado em $O(n^2)$; grau de saturação, que pode ser implementado para correr em $O(n^3)$ e por grau de incidência, que pode ser implementado para correr também em $O(n^2)$ [8].

Outro algoritmo bastante conhecido na bibliografia da coloração de grafos, e inserido nos algoritmos aproximados é o algoritmo *Recursive Largest First* (RLF). O RLF combina a estratégia do algoritmo da ordenação do nodo com maior grau com a estrutura do algoritmo AMIS. Assim como no algoritmo de maior grau, em cada passo no procedimento do RLF um nodo é selecionado para ser colorido, deixando os restantes nodos não coloridos, coloráveis no mínimo de cores possível. Assim como com o algoritmo AMIS, o procedimento do RLF completa a atribuição da cor i antes de começar a atribuir a cor $i + 1$. Se for propriamente programado, o algoritmo exhibe uma dependência temporal de $O(n^2)$ para várias aplicações [89].

Além das heurísticas previamente mencionadas, existem também heurísticas de melhoria que podem ser aplicadas, por exemplo, às heurísticas sequenciais. A heurística de coloração *greedy* iterativa de

Culberson [54] aplica a heurística de coloração sequencial *greedy* a certas ordenações de nodos, nas quais a sua ordenação para a próxima aplicação da heurística sequencial é baseada na coloração encontrada na iteração atual.

Outra melhoria a uma dada coloração é a heurística das permutações (*interchanges*). Esta heurística troca nodos previamente coloridos para classes de cor diferentes. Se uma classe de cor puder ser removida após o processo de troca, significa que foi encontrada uma coloração melhor.

3.3 Coloração equitativa

Um grafo G diz-se equitativamente k -colorável se o seu conjunto de nodos for divisível em k classes V_1, V_2, \dots, V_k tal que cada V_i seja um conjunto independente e que a condição $||V_i| - |V_j|| \leq 1$ se mantenha para todo o par (i, j) . O número inteiro k para o qual G é equitativamente k -colorável é conhecido como o **número cromático equitativo de G** e denota-se por $\chi_{=}(G)$. Uma vez que a coloração equitativa é uma coloração normal com uma restrição adicional, tem-se que $\chi(G) \leq \chi_{=}(G)$ para qualquer grafo G [70].

O conceito de coloração equitativa foi introduzido por Meyer em 1973 [100]. A sua história, no entanto, vem desde 1964 numa conjectura por Erdős, conjectura essa provada em 1970 por Hajnal e Szeemerédi [77], formando-se o seu celebrado teorema: um grafo com grau máximo Δ é equitativamente k -colorável se $k \geq \Delta + 1$. Uma das direções da pesquisa teórica inicial neste campo foi para encontrar melhores limites superiores que $\Delta + 1$ para grafos de classes especiais [93].

Não há muitos artigos acerca da complexidade algorítmica da coloração equitativa. Em primeiro lugar, uma redução direta do problema clássico da coloração de grafos para a coloração equitativa por adição de um número suficiente de nodos isolados a um grafo, prova que testar se um grafo genérico tem uma coloração equitativa com um dado número de cores (maior que dois) é NP-completo. Em segundo lugar, Bodleander e Fomin [23] mostraram que a coloração equitativa pode ser resolvida para a otimização em tempo polinomial para árvores e grafos planares.

Na coloração equitativa, conforme referida acima, é necessário que a diferença de tamanho entre qualquer par de classes de cor seja no máximo um. Para algumas necessidades práticas, esta limitação pode ser desnecessariamente rigorosa e dispendiosa para ser aplicada. Lu *et al.* no seu artigo [93] referem-se a uma variante heurística, aproximada da coloração equitativa: a coloração equilibrada. Nesta variante a restrição é relaxada, a diferença de tamanho entre classes de cor pode ser, em vez de apenas um, no máximo um número “pequeno” maior que um. Uma alternativa mais formal para declarar esta variante, é definindo que cada classe de cor é delimitada por um parâmetro l . Bodleander e Fomin [23] estudaram este problema e mostram, assim como o problema da coloração equitativa, pode ser resolvido em tempo polinomial para grafos com uma *treewidth* limitada. Lu *et al.* no seu artigo ingressaram num caminho menos formal não fixando um parâmetro l . Mais especificamente, dado um grafo $G(V,E)$, cada classe de cor deve receber aproximadamente $\frac{|V|}{C}$ nodos, onde C é o número de cores utilizadas. Nesta tese optou-se por seguir um caminho semelhante.

A coloração equitativa e equilibrada (no sentido mencionado em cima) têm várias aplicações práticas. Sempre que se tem de dividir um sistema com relações de conflito binárias em subsistemas iguais ou semelhantes, livres de conflito, a coloração equitativa e equilibrada pode ser utilizada para modelar esta situação. Outros exemplos de aplicações em casos reais incluem a partição equilibrada de carga para métodos de decomposição, cálculos de matrizes esparsas paralelas em redes irregulares e vários tipos de

problemas de escalonamento e calendarização. Num artigo de 1973, Tucker[126] discute como a teoria da coloração de grafos tem sido útil aos investigadores operacionais no Departamento de Ciência Urbana em Stony Brook, uma vez que se deparavam com um problema de roteamento desafiante que tinha como objetivo a otimização do escalonamento de camiões de recolha de lixo na cidade [93, 70].

3.4 Aplicações

Vários problemas práticos podem ser modelados enquanto problemas de coloração de grafos. A forma geral destes modelos é um grafo com nodos a representarem objetos de interesse e arestas a conectarem pares de objetos com uma relação binária. Alguns exemplos nos quais a coloração de grafos é geralmente aplicada são:

- Para a alocação de registos, um grafo de conflito G_c é construído a partir do código. Os nodos representam variáveis. Dois nodos estão conectados por uma aresta se tiverem conflito (uma variável é utilizada tanto antes como depois outra variável dentro de um breve período de tempo). A coloração de G_c é então uma atribuição livre de conflito de registos a variáveis.
- Na atribuição de frequências, dois transmissores não podem usar a mesma frequência durante o mesmo período de tempo se estiverem localizados dentro de uma dada distância um do outro. Um grafo pode ser construído, no qual os nodos representam as localizações dos transmissores e dois nodos estão conectados através de uma aresta no caso de se encontrarem mais perto do que a dada distância. O número mínimo de frequências necessárias para assegurar que não haja interferências é igual ao número cromático do grafo.
- Nos testes a placas de circuito impressos, seja G um grafo com nodos a representarem as redes e as arestas a representarem redes adjacentes. Um teste eficiente para encontrar curto-circuitos começa por colorir G com k cores. A cada rede é atribuída a uma cor em particular. Uma vez que redes adjacentes têm classes de cores diferentes, testar apenas os $\binom{k}{2}$ pares de classes de cores será necessário para descobrir qualquer curto-circuito [106].
- Para a disposição de câmaras de vigilância, por exemplo de uma galeria de arte, modelando o espaço enquanto uma figura poligonal no plano. Definindo os cantos desse polígono como nodos e possíveis locais para instalar câmaras de vigilância e ligando através de arestas as câmaras que tenham o mesmo, ou em parte, campo de visão. Através de uma coloração de grafos é possível minimizar o número de câmaras a serem instaladas.
- Para agendar torneios de todos contra todos (*round-robin*), cada jogo é um nodo. Um nodo do jogo da equipa A com a equipa B está conectado através de arestas com os outros jogos em que ambas participem, ficando com cores diferentes.
- Para agendar voos, assumindo que há k aviões que têm de ser atribuídos a n voos, onde o i -ésimo voo é durante o intervalo (a_i, b_i) . Se dois voos estão sobrepostos então não se pode afetar o mesmo avião a ambos os voos. Neste caso os nodos correspondem aos voos, dois nodos têm uma ligação se as suas durações se sobrepõem.
- Para colorir mapas, conforme descrito anteriormente, na secção 3.1.3.
- Para criar horários universitários ou escolares, considerando cada disciplina como um nodo, sendo as ligações entre nodos criadas a partir de possíveis conflitos (a disciplina é lecionada pelo mesmo

docente, os mesmos alunos participam em duas ou mais disciplinas, necessitam da mesma sala, entre outros).

- Por último, pode ser aplicado também à calendarização de exames, que será explorado na secção seguinte. [5]

3.5 Coloração de grafos em calendarização de exames

Seja $Dias = \{D_1, D_2, \dots, D_d\}$, um conjunto de dias nos quais um exame pode ser agendado e $Periodos = \{P_1, P_2, \dots, P_p\}$, um conjunto de períodos de tempo em cada dia. Seja também $Salas = \{S_1, S_2, \dots, S_s\}$, um conjunto de salas disponíveis em cada período. Cada combinação entre um dia D_i e um período P_j é considerado uma cor tal que Cores é uma lista com um comprimento de $d \times p$ que contém todas as cores. Por exemplo, a q -ésima cor da lista Cores, correspondendo ao dia D_i e ao período P_j , é denotada como $C_q^{i,j}$. Uma vaga representa a reserva do número de lugares numa dada sala num dia e período específicos para um único exame. Tem-se então que, Vagas é um conjunto de vagas às quais exames são atribuídos tal que uma única vaga seja denotada por $S_{k,w}^{i,j}$, representando a realização de um exame no dia D_i e período P_j , tendo lugar na sala S_k na qual w lugares (dentro da capacidade total da sala) estão reservados para o exame em questão. A partir da definição anterior, uma única vaga pode ser ocupada por apenas um exame. A maioria dos artigos relacionados com esta área representam uma vaga enquanto um par (dia, período). Na opinião de Hassan *et al.* [7], a sua definição é mais precisa e reflete a realidade de uma maneira mais próxima, facilita também a fase da implementação do algoritmo. Uma vez que uma cor é representada por um par de dia e período, então todas as vagas que pertençam ao mesmo dia e período têm a mesma cor.

Na coloração de grafos, a calendarização de exames pode ser descrita como se segue. Suponha-se que $Nodos = \{N_1, N_2, \dots, N_n\}$ é um conjunto de nodos num grafo não orientado tal que cada nodo representa uma disciplina. $Arestas = \{A_1, A_2, \dots, A_a\}$ é um conjunto de arestas tal que uma aresta entre dois nodos significa que pelo menos um aluno realiza exame nas duas disciplinas associadas a esses nodos. Recorde-se que quaisquer duas vagas que partilhem o mesmo dia e período partilham também a cor. Por esse motivo, o problema da calendarização de exames pode ser resolvido através da aplicação da coloração de grafos na qual cada exame é agendado numa vaga, tal que não seja atribuída uma vaga da mesma cor a dois nodos adjacentes (exames com alunos em comum).

3.5.1 Abordagens com coloração de grafos aplicadas à calendarização de exames

Em 1967, a coloração de grafos foi aplicada ao problema da calendarização de disciplinas [71]. Nesse mesmo ano, Welsh e Powell [129], num artigo que representou uma contribuição importante para a bibliografia dos problemas de calendarização, ilustraram a relação entre calendarização e coloração de grafos. Welsh e Powell desenvolveram um novo algoritmo de coloração de grafos para resolver (ou resolver aproximadamente) mais eficientemente, o problema da coloração com o menor número de cores [71], o que levou a uma quantidade significativa de investigação posterior em heurísticas de grafos aplicadas ao problema da calendarização [110].

Heurísticas de grafos que fundamentavam os métodos de construção simples desempenharam um papel muito importante nos primórdios da investigação em calendarização [40]. Embora tenham sido originalmente apresentadas enquanto técnicas (simples) por si só, estas continuam a ser empregues e adaptadas dentro de métodos híbridos na bibliografia da investigação atual. A sua grande vantagem

é a de proporcionar razoavelmente bons resultados dentro de um tempo de execução pequeno, sendo também muito simples de implementar. Estas heurísticas são frequentemente utilizadas para construir soluções iniciais, ou para construir bons conjuntos de soluções antes de serem aplicadas técnicas de melhoramentos. De seguida, encontram-se alguns exemplos de aplicações da coloração de grafos a problemas de calendarização de exames.

Em 1981, Mehta [97] aplicou o método DSM [24] para gerar um número de períodos, mas uma vez que o número cromático do seu grafo gerado era 12, utilizou também técnicas de compressão para reduzir o número de períodos para um número mais aceitável, rearranjando os períodos de modo a minimizar as penalidades.

Num artigo particularmente influente em 1996, Carter *et al.*[43] estudaram cinco estratégias de ordenação diferentes em problemas de calendarização de exames reais e gerados aleatoriamente. As maiores cliques, que são os maiores subgrafos onde cada nodo é adjacente a todos os outros, foram utilizadas para construir soluções iniciais, com base nas quais foram utilizadas heurísticas de coloração de grafos e técnicas de *backtracking* para construir as soluções. A ideia é o tamanho da clique maior determinar o número mínimo de períodos necessários para o problema. Os resultados indicaram que nenhuma das heurísticas superou as outras em todos os problemas testados. Outra contribuição importante deste trabalho foi a introdução de um conjunto de treze problemas de calendarização de exames, que se tornaram problemas padrão na área. Esses conjuntos têm sido vastamente estudados e utilizados por diferentes abordagens ao longo dos anos. A esses conjuntos de problemas chamou-se “conjuntos de dados da Universidade de Toronto”.

Em 1998, Burke *et al.*[38] estudaram o efeito da introdução de um elemento aleatório na implementação de heurísticas de grafos através do desenvolvimento de duas variantes de estratégias de seleção: (1) seleção tipo torneio que escolhe aleatoriamente um exame dentro de um subgrupo dos primeiros exames de uma lista ordenada; e (2) uma seleção “tendenciosa” que tira o primeiro exame de uma lista ordenada de um subconjunto de todos os exames. Estas técnicas, quando testadas em três dos conjuntos de dados de Toronto, melhoraram as heurísticas com *backtracking* não só em qualidade, como também em diversidade de soluções.

Em 2004, Burke e Newall [29] investigaram uma estratégia de ordenação adaptativa que ordenava os exames de uma forma dinâmica durante a resolução do problema num processo iterativo. Observaram que uma heurística fixa pré-definida (empregue enquanto uma medida de dificuldade) numa estratégia sequencial tradicional nem sempre tem um bom desempenho ao longo de todos os problemas. Um modificador heurístico foi desenvolvido para atualizar a ordenação dos exames de acordo com a experiência obtida de acordo com a dificuldade de os atribuir em iterações anteriores. Realizaram-se imensas experiências em onze dos conjuntos de dados de Toronto e também nos dados de Nottingham. Mostrou-se que esta abordagem era simples e efetiva (comparável com, e ocasionalmente melhor que, abordagens de grade sucesso). Este método não é dependente da ordem inicial dos exames.

Em 2005, a lógica *fuzzy* foi aplicado por Asmuni *et al.*[15] para ordenarem os exames a serem agendados com base em heurísticas de coloração de grafos nos conjuntos de dados de Toronto. Quando os exames são ordenados pela sua dificuldade em serem agendados, as funções *fuzzy* podem ser utilizadas para efetuarem uma avaliação apropriada. Notou-se que devem ser utilizadas funções *fuzzy* em diferentes problemas para que se obtenham melhores resultados. Ainda Asmuni *et al.*[16], em 2007 desenvolveram um sistema *fuzzy* para construir uma nova função de avaliação baseada numa série de regras para avaliarem a qualidade dos horários nos quais múltiplos critérios estavam envolvidos. A abordagem foi mais tarde melhorada através da adaptação das regras *fuzzy*.

Em 2006, Corr *et al.*[52] desenvolveram uma rede neural a partir da qual se pode obter uma medida da

dificuldade da atribuição de exames durante a construção do calendário através da introdução recorrente dos estados de construção da solução atualizada. Atribuir os exames mais difíceis numa fase inicial da construção da solução de uma maneira adaptável é o objetivo da abordagem. A rede neural foi treinada através do armazenamento dos estados de construção dos calendários (vetores de características) com o uso de três heurísticas de grafos. O trabalho demonstrou a viabilidade da aplicação de métodos baseados em redes neurais enquanto uma técnica adaptativa e geralmente aplicável a problemas de calendarização [110].

Em 2008, Malkawi *et al.*[94] apresentaram um algoritmo de coloração de grafos para agendar exames, com o objetivo de atingir igualdade, precisão e um período de tempo de exames ótimo. O algoritmo utiliza uma cor com índice duplo (R_{ij}) onde o índice (i) representa o dia do exame e (j) o período do dia. Os nodos (exames) são ordenados por ordem decrescente do grau. No caso de dois nodos com grau igual é dada prioridade aos nodos com o maior peso na sua lista de adjacência, se ambos forem o mesmo são ordenados por ordem crescente da identidade do nodo. Para alcançar a igualdade, o algoritmo define os seguintes parâmetros: distância entre duas cores com o mesmo índice (i) (R_{ij} e R_{ik}); distância entre duas cores (R_{ij} e R_{kl}); e por último a distância total entre cores (R_{ij} e R_{kg}). Em 2016 [79] baseando-se no seu algoritmo proposto previamente, Mohammad Al-Haj Hassan e Osama Al-Haj Hassan apresentam um sistema de calendarização de exames com dois focos: (1) a restrições que o sistema suporta e, (2) a interface *user-friendly*.

Em 2009, Sabar *et al.*[114] propõem uma heurística de grafos que emprega o mecanismo de seleção da roleta para a resolução de problemas da calendarização de exames. Os exames são ordenados por ordem decrescente do número de conflitos (grau) com outros exames. A dificuldade para cada exame ser agendado é estimada através do grau dos exames em conflito. O grau determina o tamanho de um segmento na roleta, sendo que um grau maior corresponde a um segmento maior. A roleta seleciona um exame se o número aleatoriamente gerado cair dentro do segmento do exame. Este método resolve o problema de escolher e agendar repetidamente a mesma sequência de exames. Os autores experimentaram a heurística nas treze instâncias do conjuntos de dados da universidade de Toronto, mostrando que produz soluções possíveis para todas as instâncias.

Em 2013, Akbulut e Yilmaz [6] propuseram um novo sistema de calendarização de exames universitárias, utilizando um algoritmo de coloração de grafos baseado em tecnologia RFID. Esta abordagem foi examinada através do uso de diferentes abordagens de inteligência artificial.

Em 2014, Abdul-Rahman *et al.*[2] apresentam uma abordagem para um caso real, o problema da calendarização de exames capacitado da universidade Utara Malaysia. Neste artigo, heurísticas de coloração de grafos são empregues para agendar exames em cada passo durante a construção do calendário. Os autores utilizam o conceito de heurísticas de empacotamento para atribuir exames a salas, tendo em conta as suas capacidades. Uma vez que se tratava de um novo conjunto de dados e não havia soluções publicadas na bibliografia até à data de publicação do artigo, apenas foi possível comparar os resultados com a solução manual existente. Mostraram que a heurística construtiva proposta conseguiu produzir soluções de boa qualidade para o problema testado, provando que as heurísticas de coloração de grafos propostas são superiores ao método empregue pela universidade até 2014.

Em 2018, Mohamed [101] apresentou uma nova abordagem para resolver a calendarização de exames através da melhoria do desempenho do algoritmo *greedy*, utilizando um algoritmo de optimização da *chicken swarm* (CSO) [98]. Este algoritmo imita o comportamento de uma galinhada. A galinhada divide-se em diversos grupos e cada grupo é composto por uma galo, galinhas e pintainhos. As galinhas seguem o galo do grupo para procurar comida. Galinhas diferentes seguem regras de movimentos diferentes. As suas identidades (galos, galinhas e pintainhos) dependem dos grupos. As galinhas com

melhores valores *fitness* são consideradas como galos, as com piores valores como pintainhos e as restantes são as galinhas. As galinhas escolhem os grupos onde vivem aleatoriamente. Na abordagem proposta por Mohamed, a CSO modificada é utilizado para otimizar a ordenação dos nodos inseridos no algoritmo *greedy*. Estatisticamente, os resultados mostraram que utilizando esta abordagem, melhorias significantes são obtidas em comparação com a criação manual de calendários de exames ou até mesmo com a utilização individual do algoritmo *greedy*. Adicionalmente, as restrições *soft* foram consideradas utilizando uma função objetivo proposta. De uma perspectiva temporal, o tempo de execução do algoritmo é de aproximadamente 1.5 minutos em média, mesmo no caso de grafos densos. Além do mais, a função custo da abordagem proposta é significativamente inferior em relação à do uso singular do algoritmo *greedy*.

Kasm *et al.*[3] apresentaram, também em 2018, uma formulação em programação inteira que abrangia todas as restrições estudadas por si, algumas das quais novas na bibliografia ao conhecimento dos autores, para facilitar a calendarização de exames no Instituto Masdar. A formulação previamente mencionada era capaz de resolver problemas pequenos utilizando um software comercial, contudo o desempenho da formulação piora à medida que a dimensão do problema aumenta. Por conseguinte, o artigo propôs heurísticas para resolverem problemas de tamanho médio e grande num tempo aceitável. Este estudo empregou algoritmos de coloração de grafos que incluem uma nova abordagem, dentro das etapas das heurísticas de calendarização de exames propostas. Quatro casos reais do Instituto foram resolvidos para ilustrar a viabilidade e competitividade das heurísticas propostas. Por fim, um estudo computacional foi apresentado para comparar as heurísticas propostas com a formulação em programação inteira. Os resultados mostraram que as heurísticas propostas foram capazes de obter soluções ótimas e quase ótimas num menor tempo de execução.

Em 2019, Hassan *et al.*[7] propuseram uma técnica de calendarização de exames que combina coloração de grafos e algoritmos genéticos. Por um lado, coloração de grafos foi utilizada para ordenar secções tal que as que fossem mais difíceis de agendar viessem em primeiro lugar e consequentemente agendadas primeiro, o que ajudou a aumentar a probabilidade de gerar horários válidos. Por outro lado, usaram algoritmos genéticos para pesquisar mais eficientemente por horários mais otimizados dentro de um largo espaço de pesquisa. Os autores propuseram um função *fitness* de duas fases com o objetivo de aumentar o bem-estar dos estudantes. Também investigaram o efeito e potencia dos operadores de cruzamento e mutação. Tendo experimentado em conjuntos de dados realísticos, os resultados mostraram que uma abordagem híbrida apenas com operadores de mutação tem um custo baixo e converge mais rapidamente para horários mais otimizados.

Além dos exemplos acima expostos, os investigadores têm explorado novos métodos alternativos para lidar com problemas de calendarização para a obtenção de melhores resultados [71]. Pode-se concluir que a coloração de grafos poderia executar uma implementação direta do problema e gerar calendários de exames sem conflitos no resultado final; porém uma vez que encontrar o número de cores necessário para colorir os nodos de um grafo é NP-difícil, como anteriormente mencionado; este método não é eficaz, exceto se for combinado com abordagens meta heurísticas para melhorar o seu desempenho [18].

A tabela seguinte resume as diversas abordagens apresentadas neste capítulo.

Tabela 3.1: Resumo das diferentes abordagens com coloração de grafos à calendarização de exames

Ano	Artigo	Método
1981	Mehta [97]	DSM + Técnicas de Compressão
1996	Carter <i>et al.</i> [43]	Algoritmo Sequencial + Técnicas de <i>Backtracking</i>
1998	Burke <i>et al.</i> [38]	Duas Novas Estratégias de Seleção
2004	Burke e Newall [29]	Algoritmo Sequencial com Ordenação Adaptativa
2005	Asmuni <i>et al.</i> [15]	Lógica <i>Fuzzy</i>
2006	Corr <i>et al.</i> [52]	Rede Neural
2007	Asmuni <i>et al.</i> [16]	Lógica <i>Fuzzy</i>
2008	Malkawi <i>et al.</i> [94]	Algoritmo de Coloração com cores de índice duplo
2009	Sabar <i>et al.</i> [114]	Coloração com Seleção da Roleta
2013	Akbulut e Yilmaz [6]	Algoritmo de Coloração baseado em Tecnologia RFID
2014	Abdul-Rahman <i>et al.</i> [2]	Heurísticas de Coloração + Heurísticas de Empacotamento
2018	Mohamed [101]	<i>Chicken Swarm</i>
2018	Kasm <i>et al.</i> [3]	Programação Inteira + Algoritmos de Coloração
2019	Hassan <i>et al.</i> [7]	Coloração de Grafos + Algoritmos Genéticos

Capítulo 4

Metodologia Utilizada

Nesta dissertação são utilizados três algoritmos diferentes: um algoritmo para colorir um grafo com um número de cores mínimo; um algoritmo para redistribuir os exames mais uniformemente pelos dias disponíveis e por último um algoritmo, que reordenando os dias, melhora a restrição *soft* que tenta garantir pelo menos um dia de intervalo entre dois exames para cada aluno.

4.1 Algoritmo de Coloração de Grafos

Este algoritmo foi inspirado no algoritmo sequencial de coloração de nodos, e o seu funcionamento pode ser consultado no Algoritmo 1.

1. Criação da Matriz de pesos e Ordenação dos nodos

No primeiro passo, é lida a instância que consiste numa lista com os códigos dos alunos e os códigos e nomes das disciplinas (para o exemplo da figura 4.1a) apenas se considera o código de aluno e nome da disciplina - sigla). A partir desta lista cria-se a matriz de pesos correspondente ao grafo, uma matriz quadrada $n \times n$, sendo n o número de exames da época especial, ou seja nodos do grafo (ver figura 4.1b), onde o peso p_{ij} corresponde ao número de alunos inscritos nos exames i e j . De seguida, os exames são ordenados por ordem decrescente do grau do respetivo nodo, como primeiro critério, e da soma dos pesos das arestas incidentes, como critério de desempate (ver figura 4.1c). No entanto, no caso de o grau de um nodo ser superior a 90% do número total de nodos, nesse caso esse nodo fica no fundo da matriz para facilitar os restantes agendamentos. A matriz obtida, após a reordenação das linhas/colunas, segundo o critério descrito (ver figura 4.1d) será então usada como *input* do algoritmo.

2. Coloração dos Nodos

Nesta fase o algoritmo, considerando a matriz criada e ordenada no passo anterior, colora todos os exames de modo a não violar as restrições *hard* a que o problema está sujeito e com o mínimo de cores que o algoritmo conseguir, e procede da seguinte maneira:

1. É criada uma lista vazia, de dimensão n na qual os índices serão os exames, e os valores as suas respetivas cores (Col);
2. O primeiro exame é colorido com a primeira cor de entre um conjunto pré-definido de k cores disponíveis;

Código de aluno	Sigla		AM	EST	IO	P1	TI	PB	B
A1	AM								
A1	EST	AM	0	3	0	0	1	0	0
A2	IO	EST	3	0	0	0	0	2	0
A3	P1	IO	0	0	0	0	0	1	0
A3	TI	P1	0	0	0	0	2	3	0
A4	EST	TI	1	0	0	2	0	0	0
A4	PB	PB	0	2	1	3	0	0	0
A5	AM	B	0	0	0	0	0	0	0
A6	PB								
A6	IO								
A7	AM								
A7	EST								
A8	IO								
A9	B	Grau	2	2	1	2	2	3	0
A10	P1	$\sum_{j=1}^n p_{ij}$	4	5	1	5	3	6	0
A10	PB								
A11	EST								
A11	PB								
A12	B								
A13	AM								
A13	TI								
A14	PB	PB	0	2	3	0	0	1	0
A14	P1	EST	2	0	0	3	0	0	0
A15	EST	P1	3	0	0	0	2	0	0
A15	AM	AM	0	3	0	0	1	0	0
A16	P1	TI	0	0	2	1	0	0	0
A16	PB	IO	1	0	0	0	0	0	0
A17	P1	B	0	0	0	0	0	0	0
A17	TI								

(b) Matriz de pesos

(c) Critérios de ordenação

(d) Matriz Ordenada de Pesos

(a) Instância

Figura 4.1: Exemplo da Criação da Matriz de Pesos e Matriz Ordenada de Pesos

3. É criada uma lista vazia ($Adj_{<}$), na qual serão registrados os exames adjuntos, já coloridos, a dado exame;
4. No caso da lista de adjuntos ser vazia o exame em questão é colorido com a primeira cor, caso contrário procura-se uma cor que seja diferente das cores dos exames adjuntos para colorir o exame em questão;
5. Após a coloração terminar verifica-se se existe algum exame que não tenha sido colorido, devido a ter conflitos com exames em todas as cores em uso;
6. Em caso afirmativo calcula-se, para cada cor em uso, a soma total de alunos que têm o exame não colorido e outro exame, colorido com a cor em questão;
7. O exame não colorido passa então a ser colorido pela cor com a menor soma dos pesos;
8. Uma lista com as cores em uso é então devolvida.

Neste algoritmo o output gerado é a lista mencionada no ponto 1, as cores utilizadas, o total de cores utilizadas e a frequência com que cada cor é utilizada.

Algoritmo 1: Coloração nos nodos

Dados: $M = [p_{ij}]$: Matriz ordenada de pesos entre exames, $Cores = \{1, \dots, k\}$: Vetor com k cores disponíveis

Resultado: Col : Lista de dimensão n na qual os índices correspondem aos exames e os valores as suas respetivas cores

$Col[i] = k + 1, i = 1, \dots, n$;

$Col[1] \leftarrow Cores[1]$;

para $i \in 2, \dots, n$ **faça**

$Adj_{<} \leftarrow \{\}$;

para $j \in 2, \dots, i$ **faça**

se $p_{i,j} > 0$ **então**

$Adj_{<} \leftarrow Adj_{<} \cup \{j\}$

fim

fim

se $Adj_{<} = \{\}$ **então**

$Col[i] \leftarrow Cores[1]$

senão

$Col[i] \leftarrow \min\{Col[i], \min_{c \in Cores} \{c : c \neq Col[j], j \in Adj_{<}\}\}$

fim

fim

para $i \in 1, \dots, n$ **faça**

se $Col[i] = k + 1$ **então**

$MC \leftarrow \arg \min_{c \in Cores} \{ \sum_{j \in Adj(i)} p_{ij} \}$;

$Col[i] \leftarrow MC$

fim

fim

De seguida, apresenta-se um exemplo de aplicação do Algoritmo 1 para $k = 5$, correspondendo à solução da Figura 4.2b (nota: a ordenação dos nodos é $[PB \ EST \ P1 \ AM \ TI \ IO \ B]$, ver Figura 4.1c).

Inicialização:		$Col = [1 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6]$
$i = EST$:	$Adj_{<} = \{PB\}$	$Col = [1 \ 2 \ 6 \ 6 \ 6 \ 6 \ 6]$
$i = P1$:	$Adj_{<} = \{PB\}$	$Col = [1 \ 2 \ 2 \ 6 \ 6 \ 6 \ 6]$
$i = AM$:	$Adj_{<} = \{EST\}$	$Col = [1 \ 2 \ 2 \ 1 \ 6 \ 6 \ 6]$
$i = TI$:	$Adj_{<} = \{P1, AM\}$	$Col = [1 \ 2 \ 2 \ 1 \ 3 \ 6 \ 6]$
$i = IO$:	$Adj_{<} = \{PB\}$	$Col = [1 \ 2 \ 2 \ 1 \ 3 \ 2 \ 6]$
$i = B$:	$Adj_{<} = \{\}$	$Col = [1 \ 2 \ 2 \ 1 \ 3 \ 2 \ 1]$

4.2 Algoritmo de Coloração Uniforme

Este algoritmo (Algoritmo 2) foi inspirado pelos algoritmos descritos em [93, 70]. O algoritmo lê a lista gerada no algoritmo anterior e, continuando a respeitar as restrições *hard*, redistribui os exames uniformemente pelo número de cores, k , para que não haja uma grande disparidade no número de exames entre os diferentes dias, procede da seguinte maneira:

1. São detetadas as cores mais usada e menos usada;
2. Enquanto a distância do número médio de nodos por cor ($\lceil n/k \rceil$) ao número de nodos coloridos com a cor mais usada e ao número de nodos coloridos com a cor menos usada for superior a um dado valor, d :
 - (a) É criado para cada nodo, colorido com a cor mais usada, uma lista com os seus nodos adjacentes;
 - (b) Se a lista mencionada no ponto acima for vazia, o nodo em questão passa a ser colorido pela cor menos usada;
 - (c) Caso contrário, se possível, o nodo em questão é colorido com uma cor que não esteja em uso por um dos nodos adjacentes;
 - (d) No caso da mudança da cor mais usada para outra cor ter sido executada com sucesso passa-se ao próximo nodo;
 - (e) As cores mais e menos utilizadas e o número de vezes que cada cor é usada são atualizadas para a próxima iteração do ciclo.

Neste algoritmo o output gerado são as cores utilizadas, o total de cores utilizadas, a frequência com que cada cor é utilizada e o calendário dos exames.

Algoritmo 2: Coloração uniforme nos nodos

Dados: Matriz de pesos entre exames (M), Vetor com k cores (Cores), Col

Resultado: Matriz com os dias como índices e os exames divididos em períodos como valores (Calendário)

Cor_MU ← cor mais utilizada;

Cor_LU ← cor menos utilizada;

Cont[c] = número de nodos coloridos com a cor c , $c = 1, \dots, k$;

enquanto ($|\text{Cont}[\text{Cor_MU}] - \lceil \frac{n}{k} \rceil|$ e $|\text{Cont}[\text{Cor_LU}] - \lceil \frac{n}{k} \rceil|$) > d **faça**

para i : Col[i] = Cor_MU **faça**

Adj ← {};

para j : Col[j] ≠ Cor_MU **faça**

se $p_{ij} > 0$ **então**

Adicionar j a Adj;

(Cont...)

```

se  $Adj = \{\}$  então
  |  $Col[i] \leftarrow Cor\_LU$ ;
  | sai do ciclo;
senão
  |  $Col[i] \leftarrow \min_{c \in Cores} \{c : Cont[c] < Cont[Cor\_MU - 1] \text{ e } c \neq Col[j], j \in Adj\}$ ;
  | sai do ciclo;
se  $Col[i] \neq Cor\_MU$  então
  | Atualizar  $Cor\_MU, Cor\_LU, Cont$ ;
  | sai do ciclo;
Atualizar  $Cor\_MU, Cor\_LU, Cont, Col$ 

```

De seguida, apresenta-se um exemplo da aplicação do algoritmo 2 à solução obtida com o algoritmo 1 (com $d = 0$ para $\lceil n/k \rceil = \lceil 7/5 \rceil = 2$), que transforma a coloração do grafo da figura 4.2b na solução final observada na figura 4.2c (relembrando que a ordenação é $[PB \ EST \ P1 \ AM \ TI \ IO \ B]$).

$$\begin{aligned}
 \text{Inicialização: } & Cor_MU = 1 \quad Cor_LU = 3 \quad Cont = [3 \ 3 \ 1] \quad Col = [1 \ 2 \ 2 \ 1 \ 3 \ 2 \ 1] \\
 i = PB: & Adj = \{EST, P1, IO\} \quad Col = [3 \ 2 \ 2 \ 1 \ 3 \ 2 \ 1] \\
 & Cor_MU = 2 \quad Cor_LU = 1 \quad Cont = [2 \ 3 \ 2]
 \end{aligned}$$

A figura seguinte apresenta os resultados obtidos com a aplicação consecutiva dos dois algoritmos anteriores. Os resultados da página 22 permitem concluir que para o grafo deste exemplo, $\chi(g) \geq 2$ (a clique máxima tem cardinalidade $\theta(G) = 2$). Por outro lado, o conjunto independente máximo tem cardinalidade $\alpha(G) = 4$, resultando em $\chi(g) \geq \lceil \frac{7}{4} \rceil = 2$. Já o resultado 6, permite concluir que $\chi(g) \leq 3$. Note-se ainda que, sendo o nodo “B” um nodo isolado, será sempre colorido com uma qualquer cor atribuída aos outros nodos e portanto, o problema da coloração apenas se aplica aos restantes nodos.

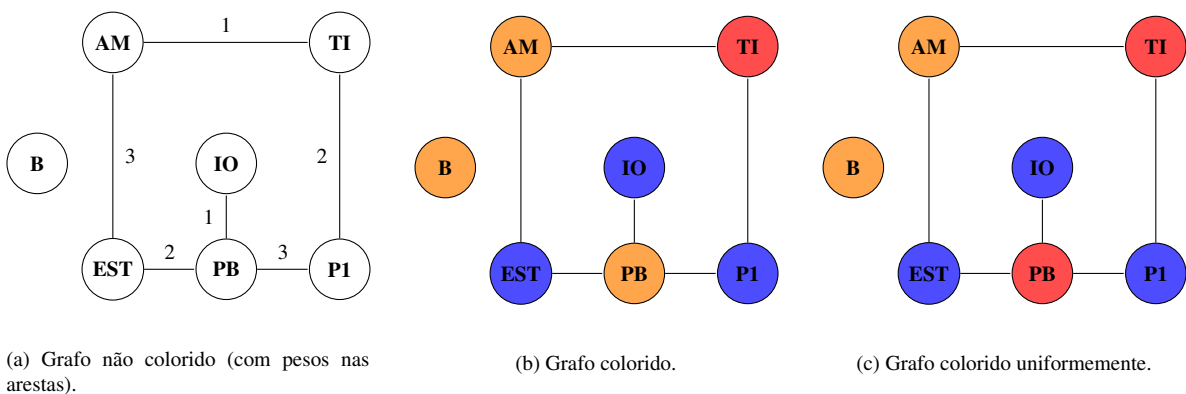


Figura 4.2: Exemplo da coloração de grafos (1=laranja, 2=azul, 3=vermelho.).

4.3 Algoritmo de Melhoria da Restrição *Soft*

Após avaliar a restrição *soft* (pelo menos 1 dia de intervalo entre dois exames para cada aluno) é criada uma matriz quadrada (de ordem k') simétrica, $Soft = [s_{ij}]$, onde s_{ij} representa o número de alunos que têm exames nos dias i e j e k' o número de dias da solução do algoritmo 2.

1. A matriz *Soft* é lida;
2. Determina-se o dia i com a maior soma total, $\sum_j s_{ij}$, ou seja o maior número de alunos com dois exames, sendo um deles um exame nesse dia i ;
3. É criada uma lista com a nova ordenação de dias, e o dia determinado no ponto anterior torna-se o primeiro elemento da lista;
4. Dos dias que ainda não pertençam à lista da nova ordenação, determina-se qual o dia com menor número de alunos com exames nesse dia e no último dia adicionado à lista, o dia escolhido é adicionado à lista;
 - i) Em caso de empate no número de alunos em dois dias ou mais, é escolhido o que tiver a maior soma total ao longo da respectiva linha da matriz.
5. Repete-se o passo 4 até todos os dias terem sido adicionados à lista da nova ordenação de dias;
6. O Calendário de exames gerado no algoritmo anterior é reordenado.

Este Algoritmo (algoritmo 3) foi inspirado em algoritmos de caminho mínimo, no entanto foi adaptado para que todos os nodos (neste caso dias) fossem incluídos entre o dia que tivesse a maior soma total de alunos, com exames nesse dia e nos restantes, e o último dia escolhido.

Algoritmo 3: Reordenação do Calendário de exames

Dados: Matriz *Soft*, Calendário

Resultado: Novo Calendário de exames

Novos_Dias \leftarrow [];

$I \leftarrow \arg \max_{i=1, \dots, k'} \sum_{j=1}^{k'} s_{ij}$;

Novos_Dias[1] \leftarrow I;

iter \leftarrow 1;

enquanto #(Novos_Dias) $<$ k' **faça**

 Menos \leftarrow $+\infty$, Mais \leftarrow 0;

para $j \in 1, \dots, k'$ **faça**

se $s_{Ij} \leq$ Menos **e** $j \notin$ Novos_Dias **E** $\sum_{i=1}^{k'} s_{ij} >$ Mais **então**

 Menos \leftarrow s_{Ij} , Mais \leftarrow $\sum_{i=1}^{k'} s_{ij}$;

 J \leftarrow j

 Novos_Dias[iter+1] \leftarrow J;

 I \leftarrow J;

(Cont...)

```

// Ordenar as colunas do Calendário atual
Novo_Calendário ← [];
para  $i \in 1, \dots, k'$  faça
  Novo_Calendário[·,  $i$ ] ← Calendário[Novos_Dias[·,  $i$ ]];

```

De seguida, apresenta-se um exemplo da aplicação do algoritmo 3 à solução obtida com o algoritmo 2 (com $k' = 3$), que transforma a coloração do grafo observada na figura 4.3a na coloração observada na figura 4.3b. Neste exemplo pode ser observado, através da matriz *Soft*, que $\sum_{i=1}^{k'-1} s_{i,i+1} = 11$, significando que onze alunos não tinham nenhum dia de intervalo entre os seus exames antes da aplicação do algoritmo. Após a sua aplicação passa a apenas $\sum_{i=1}^{k'-1} s_{i,i+1} = 4$.

$$Soft = \begin{bmatrix} 0 & 3 & 1 \\ 3 & 0 & 8 \\ 1 & 8 & 0 \end{bmatrix}$$

Inicialização: $I = 2$

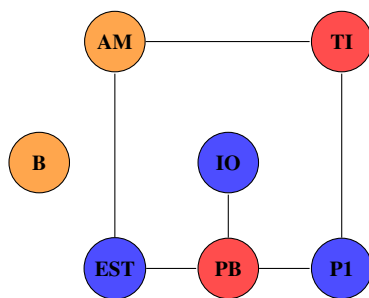
$Novos_Dias = \{2\}, I = 2$ $Menos = \infty$ $Mais = 0$ $j = 1$ $s_{21} = 3$ $\sum_{i=1}^{k'} s_{i1} = 4$
 $Menos = 3$ $Mais = 4$ $j = 3$ $s_{23} = 8$ $\sum_{i=1}^{k'} s_{i3} = 9$

$Novos_Dias = \{2, 1\}, I = 1$ $Menos = \infty$ $Mais = 0$ $j = 3$ $s_{13} = 1$ $\sum_{i=1}^{k'} s_{i3} = 9$

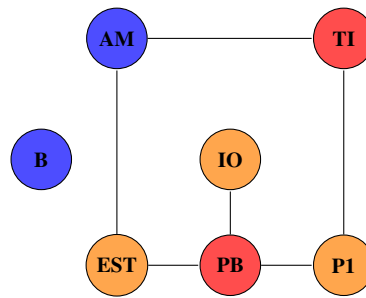
$Novos_Dias = \{2, 1, 3\}, I = 3$

Após a aplicação do algoritmo, a nova matriz *Soft* é

$$Soft = \begin{bmatrix} 0 & 3 & 8 \\ 3 & 0 & 1 \\ 8 & 1 & 0 \end{bmatrix}$$



(a) Grafo colorido uniformemente



(b) Grafo colorido uniformemente, com satisfação de restrição *soft*

Figura 4.3: Exemplo de solução após aplicação do algoritmo 3 (1=laranja, 2=azul, 3=vermelho).

Capítulo 5

Resultados computacionais

5.1 Dados e Instâncias

Nesta secção os dados, utilizados para criar as diferentes instâncias para testar o algoritmo descrito no capítulo anterior, são descritos no texto abaixo. Ainda nesta secção será também explicado como são geradas as diferentes instâncias, e em que consistem.

Dados

Para criar as instâncias, descritas e explicadas a seguir, a Faculdade de Ciências forneceu os dados, com os nomes e códigos de aluno de todos os alunos finalistas do ano letivo de 2013/14, assim como as disciplinas que cada aluno tinha por fazer no seu último ano, e os seus respetivos códigos, e por último o curso e respetivo código no qual cada aluno estava matriculado. Além do mais, a Faculdade forneceu também as taxas de aprovação, de todas as disciplinas lecionadas na Faculdade, do ano letivo de 2012/13. Apesar de alunos protegidos por algumas exceções serem também admitidos a época especial, nestas instâncias apenas se consideraram os alunos finalistas presentes nos dados.

Instâncias

Para criar as instâncias foi utilizado o Microsoft Excel. Dispondo aluno a aluno com as suas disciplinas, usando as taxas de aprovação, é calculado se o aluno é aprovado ou reprovado a cada disciplina da seguinte forma:

1. Gera-se um número pseudo aleatório (NPA) entre 0 e 1;
2. Se $NPA \leq$ taxa de aprovação, então:
 - i. considera-se que o aluno será aprovado à cadeira nas épocas normais.
 - ii. caso contrário, considera-se que o aluno reprovará à cadeira nas épocas normais.

De acordo com as normas da Faculdade, para ter acesso à época especial, cada aluno pode fazer apenas um máximo de dois exames, logo apenas são considerados para a época especial os alunos finalistas que reprovem a uma ou duas disciplinas nas épocas normais, tendo sido aprovados a todas as outras. Os alunos que forem aprovados a todas já não precisam de ir à época, e os alunos que reprovarem a pelo menos três disciplinas perdem o seu estatuto de finalista¹.

¹Não estão a ser considerados casos especiais de alunos que possam ter direito à época especial com mais do que 2 exames.

Por fim a instância gerada após o procedimento descrito anteriormente, é composta pelo código do aluno, o código da disciplina e o nome da disciplina. O nome do aluno é descartado por motivos de privacidade e também por ser irrelevante para o caso em estudo. Para testar o algoritmo criado nesta dissertação foram geradas dez instâncias, as quais estão descritas na tabela 5.1, onde “#Alunos” representa o número de alunos que vão realizar exames na época especial, “#Nodos” representa o número de exames a realizar na época especial, “#Arestas” representa a ligação entre pares de exames, isto é, um par de exames que tenha pelo menos um aluno inscrito nos dois exames. É também apresentado, na última coluna, o número de componentes conexas do grafo correspondente à instância, detalhando o número de componentes com uma dada dimensão entre parêntesis. Estas componentes foram obtidas através da determinação de árvores de suporte² das diferentes componentes conexas. Implementou-se uma versão adaptada do algoritmo Prim (visto não ser necessário levar em conta custos das arestas), aplicada iterativamente até não existirem nodos por incluir nalguma árvore.

Tabela 5.1: Descrição das Instâncias

Instância	#Alunos	#Nodos	#Arestas	#Componentes Conexas
1	394	190	168	52: 1(101) 1(13) 2(4) 3(3) 14(2) 31(1)
2	408	184	162	52: 1(38) 1(32) 1(31) 3(6) 1(4) 1(3) 14(2) 30(1)
3	391	182	167	48: 1(45) 1(39) 1(21) 1(5) 1(4) 5(3) 15(2) 23(1)
4	434	203	197	41: 1(92) 1(22) 1(20) 1(12) 2(4) 4(3) 6(2) 25(1)
5	413	199	180	45: 1(83) 1(26) 1(18) 1(6) 1(4) 5(3) 12(2) 23(1)
6	411	201	194	48: 1(119) 1(9) 5(4) 2(3) 8(2) 31(1)
7	392	199	181	48: 1(60) 1(46) 1(21) 1(5) 1(4) 5(3) 10(2) 28(1)
8	389	179	183	36: 1(87) 1(19) 1(12) 1(7) 1(5) 2(4) 3(3) 6(2) 20(1)
9	395	180	171	45: 1(38) 1(35) 1(16) 1(13) 1(9) 1(6) 1(5) 5(3) 10(2) 23(1)
10	416	193	188	45: 1(50) 1(37) 1(24) 1(17) 2(5) 1(3) 14(2) 24(1)

5.2 Resultados

Nesta secção serão apresentados os calendários de exames resultantes para cada uma das 10 instâncias. Para cada instância poderão ser consultados, o tempo total de execução (CPU) dos primeiros dois algoritmos, de coloração de grafos e coloração uniforme, o mínimo de cores (dias) necessárias para distribuir pelos nodos (exames), o número de exames por dia no calendário de exames final na Secção 5.2.1 e o número de alunos com d dias de intervalo, antes e depois de ser aplicado o algoritmo de melhoria da restrição *soft* na Secção 5.2.3. Geralmente, a época especial da Faculdade de Ciências tem uma duração de 7 dias sendo que um deles calha a um sábado no qual os exames apenas são realizados no período da manhã. Para simplificar o problema são apenas utilizados 6 dias para o calendário criado nesta dissertação, excluindo o sábado.

Os algoritmos foram programados, utilizando a linguagem de programação PythonTM 3, e implementados no Jupyter Notebook, pertencente ao *software* Project Jupyter. Os tempos de execução (CPU), em segundos, foram obtidos através de um computador com um processador AMD RyzenTM 5 3500U 2,10 GHz e 8 GB de memória RAM.

²Uma árvore de suporte é um subgrafo conexo e sem ciclos que contém todos os nodos do grafo.

Os tempos de execução, para cada uma das 10 instâncias, estão apresentados na tabela 5.2. Não existem diferenças significativas entre os diferentes tempos, sendo a variação máxima de 6,22s entre as instâncias 2 e 6. Além disso, os três maiores tempos pertencem às instâncias 4, 6 e 7, poderão ser justificados por um número superior de ligações entre os nodos no caso das instâncias 4 e 6. Os diferentes números de componentes conexas, e as suas respectivas densidades podem ter também uma influência sobre os tempos de execução.

5.2.1 Número mínimo de Cores

Para as instâncias 1, 2, 4, 7, 9 e 10, conforme pode ser consultado na tabela 5.2, o número mínimo de dias necessário, encontrado pelo algoritmo da coloração de grafos, para distribuir os exames é 4; já para as instâncias restantes são necessários apenas 3 dias. Apesar do algoritmo da coloração de grafos não ser um algoritmo exato, pelos resultados apresentados no Capítulo 3 é possível verificar que o número mínimo de cores encontrado é de facto o número cromático para todas as instâncias, dado que $\chi(G) \geq \theta(G)$. Segundo estes resultados, caso o objetivo da dissertação fosse reduzir o número de dias da época especial, seria possível diminuir a sua duração para apenas 4 dias em alguns cenários e mesmo 3 noutros. Por outro lado, não seria bom para os alunos, visto que os seus dias de intervalo seriam reduzidos consideravelmente, além disso, a logística para concretizar a época especial em apenas 3 ou 4 dias seria maior.

Tabela 5.2: Tempo de CPU e número mínimo de cores necessárias

Instância	CPU	Nº Mínimo de Cores	$\theta(G)$	$\left\lceil \frac{ V }{\alpha(G)} \right\rceil$
1	27,022 s	4	4	2
2	25,840 s	4	4	2
3	28,102 s	3	3	2
4	31,835 s	4	4	2
5	27,910 s	3	3	2
6	32,056 s	3	3	2
7	29,878 s	4	4	2
8	27,693 s	3	3	2
9	28,683 s	4	4	2
10	29,408 s	4	4	2

5.2.2 Construção do Calendário de exames

Na tabela 5.3 pode-se comparar as distribuições dos exames antes (Coluna (Min. Cores)) e depois (Coluna (Calendário)) de ser aplicado o algoritmo da coloração uniforme, para redistribuir de forma uniforme os exames pelos dias da época especial.

Como se pode notar na coluna da distribuição dos exames pelo número mínimo de cores, os exames foram atribuídos à primeira cor que estivesse disponível, não usufruindo de todos os dias disponíveis bem como não tendo uma distribuição uniforme, sobrecarregando principalmente os primeiros dois dias e, nas instâncias com 4 cores como número mínimo de cores, com apenas um, dois ou três exames no último dia.

Por outro lado, na coluna seguinte existe já uma evidente distribuição uniforme, sendo a maior diferença do número de exames entre o sexto dia e os restantes, nas instâncias 5, 7 e 10, com uma diferença de apenas cinco exames. Nas restantes instâncias a maior diferença registada é de quatro

Tabela 5.3: Distribuição de exames pelos diferentes dias

Instância	Dia	# Exames		Instância	Dia	# Exames	
		Min. Cores	Calendário			Min. Cores	Calendário
1	1	99	32	6	1	94	34
	2	79	32		2	88	34
	3	11	32		3	19	34
	4	1	32		4		34
	5		33		5		34
	6		29		6		31
2	1	94	31	7	1	96	34
	2	77	31		2	89	34
	3	11	31		3	12	34
	4	2	31		4	2	34
	5		32		5		34
	6		28		6		29
3	1	90	31	8	1	86	30
	2	77	31		2	75	30
	3	15	31		3	18	30
	4		31		4		31
	5		31		5		31
	6		27		6		27
4	1	98	34	9	1	86	30
	2	85	34		2	78	30
	3	19	34		3	14	31
	4	1	35		4	2	31
	5		35		5		31
	6		31		6		27
5	1	93	34	10	1	91	33
	2	92	34		2	79	33
	3	14	34		3	20	33
	4		34		4	3	33
	5		34		5		33
	6		29		6		28

exames, excluindo a instância 6, com uma diferença de três exames entre o sexto dia e os restantes. Como se pode analisar através da tabela 5.3, não há grandes variações entre as diferentes instâncias, diferenciando-se apenas no número de exames em cada dia do calendário de exames final, devido ao facto de cada instância ter um número diferente de exames para agendar excluindo as instâncias 5 e 7, que têm uma distribuição final igual de exames por dias.

Uma vez que, conforme mencionado anteriormente, não era o objetivo da dissertação reduzir a duração da época especial o algoritmo da coloração uniforme não foi testado com a duração de apenas 3 ou 4 dias. Quanto à distribuição por períodos, uma vez que também foge ao objetivo desta dissertação a consideração de certos fatores (preferências de professores, recursos necessários, entre outras), os exames de cada dia foram divididos de forma aleatória e uniforme por cada um dos três períodos diários, ver tabelas A.1 a A.10 dos Anexos.

5.2.3 Restrição *Soft*

Nesta secção os alunos mencionados são única e exclusivamente os alunos com dois exames a realizar, uma vez que a restrição *soft* apenas se refere a que cada aluno tenha no mínimo um dia de intervalo entre os dois exames. Esta restrição não se encontra na tabela 2.2 devido a tratar-se de uma restrição específica para a época especial, não sendo uma das restrições *soft* mais comuns encontrada na bibliografia.

Por fim, na tabela 5.4, podem ser consultadas as diferenças antes e depois da aplicação do Algoritmo de Melhoria da Restrição *Soft*. Apesar do algoritmo de Coloração de Grafos não ter em conta a restrição *soft*, considerá-la resultará numa melhor solução em termos do grau de satisfação dos alunos. Para medir o grau de satisfação dos alunos, foi tido em conta que quanto mais dias de intervalo entre dois exames maior o grau de satisfação. Assim, para cada instância foi calculado o número médio de dias de intervalo entre dois exames, \bar{d} , antes e depois da aplicação do algoritmo assim como a percentagem da sua variação $\Delta_{\bar{d}}$.

Na 5ª coluna da tabela 5.4, referente à restrição *soft*, ou seja ao número de alunos que não têm nenhum dia de intervalo entre os dois exames, é fácil perceber a redução significativa do número de alunos após a aplicação do algoritmo 3. Cada instância tinha, em média, 61 alunos sem dias de intervalo entre os seus exames, descendo para os 15,7 alunos após a aplicação do algoritmo 3 com uma redução de 45,3 alunos. Entre as dez instâncias, aquela onde se pode verificar uma redução maior é na instância 8 na qual se passa de 81 alunos sem dias de intervalo para 16, significando que 65 alunos ganharam pelo menos um dia de intervalo. Por outro lado, a menor redução foi na instância 3, com uma redução de 32 alunos. Traduzindo estes valores para percentagens, chega-se à conclusão que o número de alunos não protegido pela restrição *soft*, na solução final, representa uma percentagem reduzida de apenas 7,76% entre os alunos que têm dois exames e de 3,88% considerando todos os alunos, mais uma vez, na média das dez instâncias.

Observando ainda a tabela 5.4, note-se que existe uma melhoria, ou seja um aumento no número médio de dias de intervalo entre exames, nas dez instâncias, após a aplicação do Algoritmo de Melhoria da Restrição *Soft*. O maior aumento pode ser observado na instância 9, com um aumento de 67,66%, de 1,67 dias para 2,8 dias. Por outro lado mais negativo, apesar de não fazer parte da restrição *soft*, verifica-se um aumento no número de alunos com apenas um dia de intervalo entre exames em todas as instâncias. Antes da aplicação do algoritmo apenas 13,73% dos alunos com dois exames tinham, em média, apenas um dia de intervalo passando para 31,39% dos alunos, após a aplicação do algoritmo. No entanto as soluções finais continuam a ter um nível de satisfação superior às soluções anteriores. Por exemplo, analisando também a percentagem em média dos alunos, com dois exames, com menos de dois dias de intervalo entre exames passa de 43,90% para 39,15% após a aplicação do algoritmo. Apesar de não ser uma descida significativa, a percentagem de alunos sem dias de intervalo dentro dos 43,90% é superior à sua percentagem dentro dos 39,15%.

Além das observações já realizadas, pode-se observar que o comportamento do algoritmo nas dez instâncias foi semelhante com: uma redução no número de alunos com dois dias de intervalo; um aumento no número de alunos com 3 dias de intervalo à exceção das instâncias 6, na qual o número não varia, e 7, onde baixa de 35 alunos para 25; metade das instâncias a aumentarem o número de alunos com quatro dias de intervalo, com a outra metade a reduzir ou a não variar; uma diminuição no número de alunos com cinco dias de intervalo e, por último, um aumento significativo no número de alunos com seis dias de intervalo.

RESULTADOS COMPUTACIONAIS

 Tabela 5.4: Número de alunos com d dias de intervalo entre exames antes e depois da aplicação do Algoritmo 3

Instância	# Alunos		# Alunos	Dias de Intervalo							\bar{d}	$\Delta_{\bar{d}}$
	total	c/ 2 exames		0	1	2	3	4	5	6		
1	394	189	(antes)	64	18	42	36	16	12	1	1,80	33,33%
			(depois)	12	64	24	46	27	0	16	2,40	
2	408	186	(antes)	49	26	53	27	20	10	1	1,88	20,74%
			(depois)	13	74	26	36	17	0	20	2,27	
3	391	184	(antes)	48	24	48	39	19	5	1	1,87	25,67%
			(depois)	16	58	27	51	12	1	19	2,35	
4	434	220	(antes)	61	28	64	28	27	11	1	1,86	27,42%
			(depois)	17	76	37	38	27	1	24	2,37	
5	413	201	(antes)	60	28	51	27	23	10	2	1,82	37,36%
			(depois)	17	57	30	50	22	2	23	2,50	
6	411	228	(antes)	65	38	48	38	26	13	0	1,83	50,27%
			(depois)	17	68	28	38	36	0	41	2,75	
7	392	205	(antes)	57	28	49	35	20	16	0	1,91	53,40%
			(depois)	11	56	37	25	29	3	44	2,93	
8	389	203	(antes)	81	29	41	35	10	5	2	1,44	40,28%
			(depois)	16	80	16	48	21	2	20	2,02	
9	395	197	(antes)	68	28	45	28	17	7	4	1,67	67,66%
			(depois)	18	37	34	50	25	1	32	2,80	
10	416	210	(antes)	57	32	56	33	24	6	2	1,81	32,60%
			(depois)	20	64	29	55	16	2	24	2,40	

Capítulo 6

Conclusões e Trabalho Futuro

Esta dissertação teve como objetivo o desenvolvimento de um algoritmo para a criação de um calendário de exames para a época especial da Faculdade de Ciências da Universidade de Lisboa, de maneira a automatizar o processo. Neste capítulo serão apresentadas, de forma resumida, as conclusões obtidas ao longo da dissertação. Ainda neste capítulo são também propostas possíveis alterações e melhorias aos algoritmos desenvolvidos para um trabalho futuro.

6.1 Conclusões

Nesta dissertação foram desenvolvidos três algoritmos diferentes. O primeiro algoritmo, com o objetivo de identificar o número mínimo de dias (cores) possível para realizar os exames (nodos) da época especial, garantindo que nenhum aluno tem os dois exames no mesmo dia. Trata-se de um algoritmo de coloração de grafos. Este primeiro algoritmo está dependente da ordenação prévia dos nodos do grafo (exames) e, uma vez que se trata de um algoritmo heurístico, pode não ser possível obter a solução de valor mínimo.

Partindo da distribuição de exames obtida com o algoritmo 1, e tendo em conta que o número de dias obtido com esse algoritmo é bastante inferior ao número de dias que a época especial realmente dura, o algoritmo 2 pretende redistribuir, mas de uma forma homogênea, os exames pelos dias da época de exames, continuando a garantir que nenhum aluno tem os dois exames no mesmo dia. Mais uma vez, por ser também um algoritmo heurístico, não garante a melhor distribuição dos exames pela semana da época especial. Além dos dois algoritmos descritos anteriormente, foi ainda desenvolvido um último algoritmo para melhorar a restrição *soft*, e por sua vez melhorar a solução anterior, no que diz respeito à restrição *soft*.

Para testar os algoritmos foram criadas dez instâncias a partir de dados fornecidos pela Faculdade de Ciências referentes ao ano letivo de 2013/2014. Cada instância tinha um número diferente de alunos e exames, apesar de não variarem muito em termos de dimensão, assim como nas diferentes épocas especiais ao longo dos anos.

Quanto aos resultados computacionais, estes demonstraram que o recurso a um algoritmo de coloração de grafos, como já mencionado no capítulo 3, resulta em soluções satisfatórias, obtidas em tempos excelentes.

Conforme previamente mencionado no capítulo da Introdução1, à semelhança das épocas de exames normais, também a época especial tem um calendário de exames fixo, utilizado todos os anos. Apesar de ser uma método eficaz para a fases de exames regulares, para a época especial não é um método desejável. Comparando o calendário de exames resultante do trabalho realizado nesta dissertação com

o calendário de exames já existente da Faculdade de Ciências da Universidade de Lisboa, referente à época especial, é fácil notar a grande diferença entre os dois: o número de exames agendados num dia. Excluindo o sábado, o calendário atual da Faculdade de Ciências tem à volta de cem exames agendados por dia. Desses cem ou mais exames diários, uma vez que estes são agendados no início no ano sem qualquer noção de quantos alunos se vão inscrever, uma grande parte não se realiza e/ou existem exames no mesmo dia com alunos em comum, o que deve ser evitado para bem dos alunos. Ora, tendo em conta a falta de conhecimento de quais os exames que se realizam e quais é que não, corre-se o risco de haver dias com mais de cinquenta exames e outros dias com dez ou até menos, ocorrendo uma enorme falha numa gestão equilibrada de recursos. Por outro lado, os calendários construídos nesta dissertação podem ser gerados após a conclusão da 2ª época de exames do 2º semestre e com conhecimento real dos alunos inscritos na época especial. Em alternativa podem ser gerados no início do ano letivo, assumindo um cenário simulado, à semelhança do que foi feito na geração das instâncias, no capítulo 5. Ou ainda numa situação intermédia, após a época normal de exames do 1º semestre, Desta forma, é possível fazer uma distribuição mais homogénea, repartindo mais justamente os recursos diários sem sobrecarregar um dia muito mais do que outro.

Com a análise da solução final, foi também possível mostrar que o uso do algoritmo de melhoria da restrição *soft* resulta numa melhoria em termos de solução. O algoritmo de coloração de grafos não toma em consideração a restrição *soft*, precisando assim de outro algoritmo para conseguir melhorar a solução obtida. Através dessa análise, apesar de não haver dados para comparar com a situação real da Faculdade de Ciências, a percentagem de alunos que têm dois exames a não cumprirem a restrição *soft* é, em média, de apenas 7,76%. Contando com a totalidade dos alunos, essa percentagem baixa ainda mais para 3,88%. Trata-se de uma percentagem reduzida, traduzindo-se numa solução melhor do que antes de ser aplicado o algoritmo, onde a percentagem de alunos sem dias de intervalo entre dois exames era, em média, de 15,12% tendo decrescido mais de 10%. Além da restrição *soft*, a percentagem média de alunos com menos de dois dias de intervalo entre exames é de 39,15%, entre os alunos que têm dois exames e 19,55% na visão geral. Apesar de já ser mais elevada devido a haver mais alunos com apenas um dia de intervalo, continua a ser um resultado positivo visto que mais de 60% dos alunos que ingressam em dois exames da época especial, têm pelo menos 2 dias de intervalo entre os seus dois exames.

6.2 Trabalho Futuro

Uma vez que alguns fatores fundamentais para uma calendarização de exames mais completa não foram tidos em conta, existe uma série de propostas para desenvolver este trabalho no futuro. Algumas dessas ideias a serem exploradas num trabalho futuro são:

- Considerar as características de cada exame, de modo a saber quais os recursos necessários, ou o tipo de sala a ser utilizado;
- Ter em conta possíveis preferências dos docentes, por exemplo, se um exame não pode ser marcado num dia específico, entre outras;
- Considerar os sábados, visto que a faculdade realiza exames nos sábados no período da manhã;
- A aplicação dos algoritmos desenvolvidos às épocas normais de exames, com as devidas adaptações, uma vez que se tratam de instâncias de maior dimensão e complexidade algorítmica;
- Aplicar os algoritmos desenvolvidos aos subproblemas (componentes conexas) visto que são problemas menores, também o seu grau de complexidade será menor;

- Ganhar acesso a dados de anos anteriores, para possibilitar a comparação do número de alunos com dias de intervalo entre exames nos diferentes calendários;
- A criação de uma plataforma *user-friendly* para poder ser utilizada futuramente pelos serviços acadêmicos da faculdade.

Bibliografia

- [1] H. Abbass. Mbo: marriage in honey bees optimization-a haplometrosis polygynous swarming approach. *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 1:207 – 214 vol. 1, 02 2001.
- [2] S. Abdul-Rahman, N. S. Sobri, M. F. Omar, A. M. Benjamin, and R. Ramli. Graph coloring heuristics for solving examination timetabling problem at universiti utara malaysia. *AIP Conference Proceedings*, 1635(1):491–496, 2014.
- [3] O. Abou Kasm, B. Mohandes, A. Diabat, and S. El Khatib. Exam timetabling with allowable conflicts within a time window. *Computers & Industrial Engineering*, 127:263–273, 2019.
- [4] S. Ahmadi, R. Barone, P. C.-H. Cheng, E. K. Burke, P. I. Cowling, and B. McCollum. Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem. In *Proceedings of the 1st Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2003)*, pages 155–173, 2003.
- [5] S. Ahmed. Applications of graph coloring in modern computer science. *International Journal of Computer and Information Technology*, 3:01–07, 2013.
- [6] A. Akbulut and G. Yilmaz. University exam scheduling system using graph coloring algorithm and rfid technology. *International Journal of Innovation, Management and Technology*, 4:66–72, 2013.
- [7] O. Al-Haj Hassan, O. Qtaish, M. Abuhamdeh, and M. Al-Haj Hassan. A hybrid exam scheduling technique based on graph coloring and genetic algorithms targeted towards student comfort. *International Journal of Advanced Computer Science and Applications*, 10(3), 2019.
- [8] H. Al-Omari and K. E. Sabri. New graph coloring algorithms. *Journal of Mathematics and Statistics*, 2(4):439–441, Dec. 2007.
- [9] D. J. Albers and C. Reid. An interview with george b. dantzig: The father of linear programming. *The College Mathematics Journal*, 17(4):292–314, 1986.
- [10] R. Alvarez-Valdes, E. Crespo, and J. M. Tamarit. Design and implementation of a course scheduling system using tabu search. *European Journal of Operational Research*, 137(3):512–523, 2002.
- [11] P. Amaral and T. C. Pais. Compromise ratio with weighting functions in a tabu search multi-criteria approach to examination timetabling. *Computers & Operations Research*, 72:160–174, 2016.

- [12] J. S. Appleby, D. V. Blake, and E. A. Newman. Techniques for Producing School Timetables on a Computer and their Application to other Scheduling Problems. *The Computer Journal*, 3(4):237–245, 1961.
- [13] T. Arani and V. Lofti. A three phased approach to final exam scheduling. *IIE Transactions*, 21(1):86–96, 1989.
- [14] E. Aslan, T. Simsek, and A. Karkacier. A binary integer programming model for exam scheduling problem with several departments. *Bilgi Ekonomisi ve Yönetimi Dergisi*, 12(2):169 – 175, 2017.
- [15] H. Asmuni, E. Burke, J. Garibaldi, and B. Mccollum. Fuzzy multiple heuristic orderings for examination timetabling. In *Practice and Theory of Automated Timetabling V*, pages 334–353, 2004.
- [16] H. Asmuni, E. K. Burke, J. M. Garibaldi, and B. McCollum. A novel fuzzy approach to evaluate the quality of examination timetabling. In *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling VI, PATAT’06*, page 327–347, Berlin, Heidelberg, 2006. Springer-Verlag.
- [17] Z. N. Azimi. Comparison of metaheuristic algorithms for examination timetabling problem. *Journal of Applied Mathematics and Computing*, 16:337–354, 2004.
- [18] H. Babaei, J. Karimpour, and A. Hadidi. A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86:43–59, 2015. Applications of Computational Intelligence and Fuzzy Logic to Manufacturing and Service Systems.
- [19] C. Baber. An introduction to list colorings of graphs. Master’s thesis, Virginia Polytechnic Institute and State University, 2009.
- [20] M. A. Bakir and C. Aksop. A 0-1 integer programming approach to a university timetabling problem. *Hacettepe Journal of Mathematics and Statistics*, 37(1):41–55, 2008.
- [21] B. Berger and J. Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 5:459–466, 2005.
- [22] B. Bilgin, E. Özcan, and E. E. Korkmaz. An experimental study on hyper-heuristics and exam timetabling. In *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling VI*, volume 3867 of *PATAT’06*, page 394–412, Berlin, Heidelberg, 2006. Springer-Verlag.
- [23] H. L. Bodlaender and F. V. Fomin. Equitable colorings of bounded treewidth graphs. *Theoretical Computer Science*, 349(1):22–30, 2005.
- [24] D. Brélaz. New methods to color the vertices of a graph. *Commun. ACM*, 22(4):251–256, 1979.
- [25] S. Broder. Final examination scheduling. *Commun. ACM*, 7(8):494–498, 1964.
- [26] E. Burke, Y. Bykov, and S. Petrovic. A multicriteria approach to examination timetabling. In E. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling III*, pages 118–131, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

-
- [27] E. Burke, M. Dror, S. Petrovic, and R. Qu. Hybrid graph heuristics within a hyper-heuristic approach to exam timetabling problems. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Next Wave in Computing, Optimization, and Decision Technologies*, pages 79–91. Springer US, 2005.
- [28] E. Burke and J. Newall. A multistage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, 3(1):63–74, 1999.
- [29] E. Burke and J. Newall. Solving examination timetabling problems through adaption of heuristic orderings: Models and algorithms for planning and scheduling problems (editors: Philippe baptiste, jacques carlier, alix munier, andreas s. schulz). *Annals of Operations Research*, 129:107–134, 2004.
- [30] E. Burke, S. Petrovic, and R. Qu. Case based heuristic selection for timetabling problems. *Journal of Scheduling*, 9:115–132, 04 2006.
- [31] E. Burke, D. Werra, J. Kingston, J. Gross, and J. Yellen. Applications to timetabling. *Handbook of Graph Theory*, pages 445–474, 2004.
- [32] E. K. Burke, Y. Bykov, J. P. Newall, and S. Petrovic. A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36:509 – 528, 2004.
- [33] E. K. Burke, D. Elliman, P. H. Ford, and R. F. Weare. Examination timetabling in british universities: A survey. In E. Burke and P. Ross, editors, *Practice and Theory of Automated Timetabling*, volume 1153, pages 76–90. Springer Berlin Heidelberg, 1996.
- [34] E. K. Burke, G. Kendall, and E. Soubeiga. A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, 9:451–470, 2003.
- [35] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu. A graph-based hyper-heuristic for educational timetabling problems. *Eur. J. Oper. Res.*, 176:177–192, 2007.
- [36] E. K. Burke and J. P. Newall. Enhancing timetable solutions with local search methods. In E. Burke and P. De Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, pages 195–206, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [37] E. K. Burke, J. P. Newall, and R. F. Weare. A memetic algorithm for university exam timetabling. In E. Burke and P. Ross, editors, *Practice and Theory of Automated Timetabling*, volume 1153, pages 241–250. Springer Berlin Heidelberg, 1996.
- [38] E. K. Burke, J. P. Newall, R. F. Weare, and N. N. Rd. A simple heuristically guided search for the timetable problem. In *In: Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems, Univ. of La Laguna*, pages 574–579. Academic Press, 1998.
- [39] R. N. Bush, J. G. Caffrey, R. V. Oakford, and D. W. Allen. Using machines to make the high-school schedule. *The School Review*, 69(1):48–59, 1961.
- [40] M. W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Research*, 34(2):193–202, 1986.
-

- [41] M. W. Carter and G. Laporte. Recent developments in practical course timetabling. In E. Burke and M. Carter, editors, *Practice and Theory of Automated Timetabling II*, pages 3–19, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [42] M. W. Carter, G. Laporte, and J. W. Chinneck. A general examination scheduling system. *INFORMS Journal on Applied Analytics*, 24(3):109–120, 1994.
- [43] M. W. Carter, G. Laporte, and S. Y. Lee. Examination timetabling: Algorithmic strategies and applications. *The Journal of the Operational Research Society*, 47(3):373–383, 1996.
- [44] N. K. Cauvery. Timetable scheduling using graph coloring. *International Journal of P2P Network Trends and Technology*, 1:57–62, 2011.
- [45] C. Cheong, K. Tan, and B. Veeravalli. A multi-objective evolutionary algorithm for examination timetabling. *J. Scheduling*, 12:121–146, 2009.
- [46] M. Cheraitia and S. Haddadi. Simulated annealing for the uncapacitated exam scheduling problem. *Int. J. Metaheuristics*, 5(2):156–170, 2016.
- [47] M. Cheraitia, S. Haddadi, and A. Salhi. Hybridising plant propagation and local search for uncapacitated exam scheduling problems. *International Journal of Services and Operations Management*, 32(4):450–467, 2019.
- [48] A. J. Cole. The preparation of examination time-tables using a small-store computer. *The Computer Journal*, 7(2):117–121, 1964.
- [49] A. Colijn and C. Layfield. Conflict reduction in examination schedules. In *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling*, pages 297–306, 1995.
- [50] A. Colorni, M. Dorigo, and V. Maniezzo. A genetic algorithm to solve the timetable problem. *Computational Optimization and Application*, 1994.
- [51] D. Corne, P. Ross, and H. Ian Fang. Evolutionary timetabling: Practice, prospects and work in progress. In *In Proceedings of the UK Planning and Scheduling SIG Workshop, Strathclyde*, 1994.
- [52] P. H. Corr, B. McCollum, M. A. J. McGreevy, and P. McMullan. A new neural network based construction heuristic for the examination timetabling problem. In T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, pages 392–401, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [53] D. Costa and A. Hertz. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295–305, 1997.
- [54] J. C. Culberson. Iterated greedy graph coloring and the difficulty landscape. In *Technical Report TR 92-07*. University of Alberta Department of Computing Science, 2007.
- [55] P. Côté, T. Wong, and R. Sabourin. Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. *Lecture Notes in Computer Science*, 3616:151–168, 2005.

-
- [56] S. Daskalaki and T. Birbas. Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160:106–120, 2005.
- [57] S. Daskalaki, T. Birbas, and E. Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1):117–135, 2004.
- [58] P. David. A constraint-based approach for examination timetabling using local repair techniques. In E. Burke and M. Carter, editors, *Practice and Theory of Automated Timetabling II*, pages 169–186, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [59] D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, 1985.
- [60] M. Dener and M. H. Calp. Solving the exam scheduling problems in central exams with genetic algorithms. *Mugla Journal of Science and Technology*, 4(1):102–115, 2018.
- [61] L. Di Gaspero and A. Schaerf. Tabu search techniques for examination timetabling. In E. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling III*, pages 104–117, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
- [62] K. A. Dowsland and J. M. Thompson. Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56(4):426–438, 2005.
- [63] G. Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993.
- [64] M. Eley. Ant algorithms for the exam timetabling problem. In E. K. Burke and H. Rudová, editors, *Practice and Theory of Automated Timetabling VI*, pages 364–382, Berlin, Heidelberg, 2007. Springer.
- [65] W. Erben. A grouping genetic algorithm for graph colouring and exam timetabling. In E. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling III*, pages 132–156, Berlin, Heidelberg, 2001. Springer.
- [66] P. Erdős, A. Rubin, and H. Taylor. Choosability in graphs. *Congr. Numer.*, 26:125–157, 1979.
- [67] T. Erlebach and K. Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255(1):33–50, 2001.
- [68] E. Ersoy, E. Özcan, and S. Uyar. Memetic algorithms and hyperhill-climbers. In *Proceedings of the 3rd multidisciplinary international conference on scheduling: theory and applications*, pages 159–166, 2007.
- [69] P. Formanowicz and K. Tanaś. A survey of graph coloring - its types, methods and applications. *Foundations of Computing and Decision Sciences*, 37(3):223–238, 2012.
- [70] H. Furmanczyk, A. Jastrzebski, and M. Kubale. Equitable coloring of graphs. recent theoretical results and new practical algorithms. *Archives of Control Sciences*, 26:281–295, 2016.
- [71] R. Ganguli and S. D. Roy. A study on course timetable scheduling using graph coloring approach. *International Journal of Computational and Applied Mathematics*, 12:469–485, 2017.
-

- [72] M. Garey, D. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.
- [73] R. Gashgari, L. Alhashimi, R. Obaid, T. Palaniswamy, L. Aljawi, and A. Alamoudi. A survey on exam scheduling techniques. In *2018 1st International Conference on Computer Applications Information Security (ICCAIS)*, pages 1–5, 2018.
- [74] F. Glover. Tabu search—part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [75] C. C. Gotlieb. The construction of class-teacher time-tables. In *IFIP Congress*, pages 73–77, 1962.
- [76] L. Gouveia. Tópicos de grafos. *Disciplina de Grafos e Redes, Licenciatura em Matemática Aplicada, Faculdade de Ciências da Universidade de Lisboa*, 2018.
- [77] A. Hajnal and E. Szemerédi. Proof of a conjecture of P. Erdős. In *Combinatorial theory and its applications, II*, pages 601–623, 1970.
- [78] M. Halldorsson. A still better guarantee for approximate graph coloring.”. In *Report No. 90-44*. Department of Computer Science, Rutgers University New Brunswick, NJ, 1990.
- [79] M. Hassan and O. Hassan. Constraints aware and user friendly exam scheduling system. *The International Arab Journal of Information Technology*, 13:156–162, 2016.
- [80] F. Huédé, M. Grabisch, C. Labreuche, and P. Savéant. Mcs - a new algorithm for multicriteria optimisation in constraint programming. *Annals OR*, 147:143–174, 10 2006.
- [81] D. W. JOHNSON. Worst case behavior of graph coloring algorithms. *Proceedings of the 5th Southeast Conference on Combinatorics, Graph Theory, and Computing, 1974*, pages 513–527, 1974.
- [82] M. Kahar and G. Kendall. A great deluge algorithm for a real-world examination timetabling problem. *Journal of the Operational Research Society*, 66(1):116–133, 2015.
- [83] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 2–13, 1994.
- [84] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, Boston, M, 1972. Springer US.
- [85] G. Kendall and N. Mohd Hussin. An investigation of a tabu-search-based hyper-heuristic for examination timetabling. In G. Kendall, E. K. Burke, S. Petrovic, and M. Gendreau, editors, *Multidisciplinary Scheduling: Theory and Applications*, pages 309–328. Springer US, 2005.
- [86] G. Kendall and N. Mohd Hussin. A tabu search hyper-heuristic approach to the examination timetabling problem at the mara university of technology. In E. Burke and M. Trick, editors, *Practice and Theory of Automated Timetabling V*, pages 270–293. Springer Berlin Heidelberg, 2005.
- [87] P. Khonggamnerd and S. Innet. On improvement of effectiveness in automatic university timetabling arrangement with applied genetic algorithm. In *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, pages 1266–1270, 2009.

-
- [88] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [89] F. T. Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the National Bureau of Standards*, 84 6:489–506, 1979.
- [90] N. Leite, F. Melício, and A. C. Rosa. A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications*, 122:137–151, 2019.
- [91] S. Lin. A broker approach for multi-agent scheduling. In D. Scott, editor, *Artificial Intelligence: Methodology, Systems, and Applications*, pages 193–202. Springer Berlin Heidelberg, 2002.
- [92] V. Lotfi and S. Sarin. A graph coloring algorithm for large scale scheduling problems. *Computers & Operations Research*, 13(1):27–32, 1986.
- [93] H. Lu, M. Halappanavar, D. Chavarría-Miranda, A. H. Gebremedhin, A. Panyala, and A. Kalyanaraman. Algorithms for balanced graph colorings with applications in parallel computing. *IEEE Transactions on Parallel and Distributed Systems*, 28(5):1240–1256, 2017.
- [94] M. Malkawi, M. Hassan, and O. Al-Haj Hassan. A new exam scheduling algorithm using graph coloring. *Int. Arab J. Inf. Technol.*, 5:80–86, 2008.
- [95] B. McCollum. A perspective on bridging the gap between theory and practice in university timetabling. In E. K. Burke and H. Rudová, editors, *Practice and Theory of Automated Timetabling VI*, pages 3–23, Berlin, Heidelberg, 2007. Springer.
- [96] B. Mccollum, P. McMullan, B. Paechter, R. Lewis, A. Schaerf, L. Di Gaspero, A. Parkes, R. Qu, and E. Burke. Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing*, 22:120–130, 2010.
- [97] N. K. Mehta. The application of a graph coloring method to an examination scheduling problem. *Interfaces*, 11(5):57–65, 1981.
- [98] X.-B. Meng, Y. Liu, X. Gao, and H. Zhang. A new bio-inspired algorithm: Chicken swarm optimization. In Y. Tan, Y. Shi, and C. A. C. Coello, editors, *Advances in Swarm Intelligence*, pages 86–94. Springer International Publishing, 2014.
- [99] L. T. G. Merlot, N. Boland, B. D. Hughes, and P. J. Stuckey. A hybrid algorithm for the examination timetabling problem. In E. Burke and P. De Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, pages 207–231, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [100] W. Meyer. Equitable coloring. *The American Mathematical Monthly*, 80(8):920–922, 1973.
- [101] T. Mohamed. Enhancing the performance of the greedy algorithm using chicken swarm optimization: An application to exam scheduling problem. *Egyptian Computer Science Journal*, 42(1), 2018.
- [102] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts - towards memetic algorithms. In *Technical Report Caltech Concurrent Computation Program Report 826*, Pasadena, California, 1989. Caltech.
-

- [103] A. Muklason, G. B. Syahrani, and A. Marom. Great deluge based hyper-heuristics for solving real-world university examination timetabling problem: New data set and approach. *Procedia Computer Science*, 161:647–655, 2019. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.
- [104] Z. Naji Azimi. Hybrid heuristics for examination timetabling problem. *Appl. Math. Comput.*, 163(2):705–733, 2005.
- [105] L. Paquete and T. St. Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In *4th International Conference on the Practice and Theory of Automated Timetabling*, pages 21–23. Springer-Verlag, 2002.
- [106] P. M. Pardalos, T. Mavridou, and J. Xue. The graph coloring problem: A bibliographic survey. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization: Volume 1–3*, pages 1077–1141, Boston, MA, 1998. Springer US.
- [107] S. Petrovic and Y. Bykov. A multiobjective optimisation technique for exam timetabling based on trajectories. In E. Burke and P. De Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, pages 181–194, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [108] PORDATA, Base de dados Portugal contemporâneo. Alunos matriculados no ensino superior: Total e por tipo de ensino. <https://www.pordata.pt/Portugal/Alunos+matriculados+no+ensino+superior+total+e+por+tipo+de+ensino-1018-8153>, 2021. Visitado: 2022-02-02.
- [109] R. Qu and E. Burke. Adaptive decomposition and construction for examination timetabling problems. *Proceedings of the Multidisciplinary international scheduling: theory and applications*, pages 418–425, 2007.
- [110] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee. A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12:55–89, 2009.
- [111] L. Reis and E. Oliveira. A constraint logic programming approach to examination scheduling. *Livro de Atas de Conferência Internacional*, 1999.
- [112] P. Ross, E. Hart, and D. Corne. Some observations about ga-based exam timetabling. In E. Burke and M. Carter, editors, *Practice and Theory of Automated Timetabling II*, pages 115–129, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [113] P. Ross, J. Marin-Blazquez, and E. Hart. Hyper-heuristics applied to class and exam timetabling problems. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, volume 2, pages 1691–1698 Vol.2, 2004.
- [114] N. Sabar, M. Ayob, G. Kendall, and R. Qu. Roulette wheel graph colouring for solving examination timetabling problems. In *Combinatorial Optimization and Applications*, pages 463–470. Springer Berlin Heidelberg, 2009.
- [115] N. Sabar, M. Ayob, G. Kendall, and R. Qu. A honey-bee mating optimization algorithm for educational timetabling problems. *European Journal of Operational Research*, 216(3):533–543, 2012.

-
- [116] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13:87–127, 2004.
- [117] A. Shatnawi, M. Fraiwan, and H. S. Al-Qahtani. Exam scheduling: A case study. In *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, pages 137–142, 2017.
- [118] K. Sheibani. An evolutionary approach for the examination timetabling problems. In E. Burke and P. De Causmaecker, editors, *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling*, pages 387–396, 2002.
- [119] A. N. Shukla, V. Bharti, and M. L. Garag. A linked list-based exact algorithm for graph coloring problem. *Revue d’Intelligence Artificielle*, 33(3):189–195, 2019.
- [120] K. Socha, M. Sampels, and M. Manfrin. Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In S. Cagnoni, C. G. Johnson, J. J. R. Cardalda, E. Marchiori, D. W. Corne, J.-A. Meyer, J. Gottlieb, M. Middendorf, A. Guillot, G. R. Raidl, and E. Hart, editors, *Applications of Evolutionary Computing*, pages 334–345. Springer Berlin Heidelberg, 2003.
- [121] L. J. Stockmeyer. Planar 3-colorability is polynomial complete. *SIGACT News*, 5:19–25, 1973.
- [122] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. *Expert Systems with Applications*, 165:113943, 2021.
- [123] H. Terashima-Marín, P. Ross, and M. Valenzuela-Rendón. Evolution of constraint satisfaction strategies in examination timetabling. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 1, GECCO’99*, page 635–642, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
- [124] The World Bank. School enrollment, tertiary (% gross). <https://data.worldbank.org/indicator/SE.TER.ENRR>, 2020. Visitado: 2022-02-02.
- [125] J. M. Thompson and K. A. Dowsland. A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25(7):637–648, 1998.
- [126] A. Tucker. Perfect graphs and an application to optimizing municipal services. *SIAM Review*, 15(3):585–590, 1973.
- [127] Ö. Ülker, E. Özcan, and E. E. Korkmaz. Linear linkage encoding in grouping problems: Applications on graph coloring and timetabling. In *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling VI*, volume 3867, pages 347–363. Springer-Verlag, 2007.
- [128] E. W. Weisstein. Chromatic number. <https://mathworld.wolfram.com/ChromaticNumber.html>. Visitado: 2022-01-21.
- [129] D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, 1967.
- [130] G. M. White and B. S. Xie. Examination timetables and tabu search with longer-term memory. In *Practice and Theory of Automated Timetabling III*, pages 85–103, 2001.
-

BIBLIOGRAFIA

- [131] A. Wigderson. Improving the performance guarantee for approximate graph coloring. *J. ACM*, 30(4):729–735, 1983.
- [132] M. Zelany. A concept of compromise solutions and the method of the displaced ideal. *Computers & Operations Research*, 1(3):479–496, 1974.

Anexos

Tabela A.1. Calendário de exames da instância 1

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
44311	53226	22701	62806	44341	34726	13536	62828	13540
71731	13504	13528	66507	22725	53237	44420	66524	26730
71723	53238	26733	34708	22751	66534	44332	34713	85104
66503	62825	44317	44365	66513	13546	26736	34706	62807
44425	34737	34741	44304	66518	26725	53243	62808	66505
13505	71708	34731	62805	44314	22737	62823	22741	53241
13557	71740	62819	53233	71706	13527	44430	26735	62820
44331	62814	22749	44409	13530	13524	71725	22745	71737
66522	44340	13548	62818	34722	71727	66523	26703	425135
22727	62812		13560	13513	66511	53216	53227	71712
			22736	34753		62813	22738	
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
71703	53221	66521	62824	13501	34710	66509	44376	13550
53214	66536	26714	34750	13552	71714	44386	13502	71739
34739	62803	34736	13538	22728	53220	13539	44359	66530
85109	22746	34751	53239	44320	53228	53224	26715	71732
66510	62831	13525	26729	26717	34738	26731	71721	13535
53208	34705	44426	26719	71730	34759	71718	26732	53229
44353	22754	71735	71736	26737	34725	34748	66512	44383
71729	71713	53219	53215	44301	66517	53210	13547	34728
44321	26718	53211	44319	66533	66520	13512	26723	53242
34744	13516	22723	13503	44348	34730	13509	44313	62822
441114	53240		66515	26734	44355	22747	26727	

Tabela A.2. Calendário de exames da instância 2

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
26735	62803	62822	53231	62831	26707	22722	34750	62812
66524	71730	22727	13525	66515	66510	62824	13530	66517
22735	44318	34725	66521	53206	53208	71725	71737	71708
26704	34740	26729	62805	44313	44331	22738	34711	34734
44332	34763	22706	71713	26732	13560	13528	62808	26717
62825	13538	71718	34708	13548	44337	66519	13502	13539
44376	44319	62807	44344	44334	66531	13557	71736	26737
34730	66511	34748	66530	66520	26723	66533	34731	13540
13504	13536	13501	44409	13552	71731	85104	71712	26730
34706			13547	66523	13535	22701	66509	44320
			26716			44321	66513	
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
62828	44359	22720	44340	22733	53242	71714	53241	26719
71738	34739	22749	62813	22723	71723	66518	22728	13550
85109	53243	53239	53216	26725	34741	44430	44355	34707
62827	44314	34755	26727	62819	26718	26733	53227	71729
53224	34762	26744	22741	71735	62820	66512	44317	66522
34736	44350	93102	44311	62821	13503	34726	44348	62823
34716	13524	44310	34728	34705	44420	62814	26724	13527
66536	71707	66505	22714	53215	26736	44304	62806	66534
71720	34710	71719	44383	71740	71706	26734	22754	13505
44433	53228	44386	26701	71721	13546	13518	26732	44425
44316			53238			34714		

Tabela A.3. Calendário de exames da instância 3

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
44355	34763	13560	13501	34740	34705	71736	66517	13531
13504	26723	71740	66534	22727	66519	66512	22751	13539
71735	26729	34741	66505	44384	34708	13546	71714	44311
44313	22719	53243	53247	13530	53228	44304	71725	71718
22741	62825	34710	93102	44383	85104	62806	13552	34765
44320	66522	22749	53215	62818	44420	13558	71708	13502
62803	62813	26730	34736	465102	34750	44340	13528	34707
66524	66518	26727	62816	13550	34762	71723	66509	22724
13536	62824	13547	71707	62823	62812	13540	22738	62822
			66511	62821	44334	26734	13518	53237
			62820			13527		
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
44301	13557	44365	13535	26724	22722	71721	44425	71737
13510	13513	425135	34748	34731	26718	26701	26731	22701
71703	44359	53246	13542	71702	13538	26732	66521	44318
22720	34755	53226	13559	62819	34759	34701	13505	13533
26719	71719	44426	71726	71730	44319	44348	34728	13512
53208	66523	62808	44331	13503	53241	26733	26703	53238
26736	71738	22711	44376	44316	53214	44310	53216	71710
44321	44432	13524	22728	66533	26716	71712	34730	22723
66504	44388	53206	71731	44317	66513	44332	66530	66510
26717	44308	53227	71706	26735	26737	22754	62807	34706
26725			34725			26707		

Tabela A.4. Calendário de exames da instância 4

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
34721	62822	26729	71728	62823	44310	71730	26734	71726
62820	34735	26737	53240	66520	13527	22701	44311	13558
22725	13502	71738	22747	44316	13534	22723	34741	34762
34727	13536	13501	22724	34708	22720	44332	71703	66522
34728	53246	13561	13552	62821	13512	66507	53227	34714
34731	13504	13547	62814	66537	13559	53215	71708	71710
44359	34713	22741	13549	26721	53222	22738	44331	44340
66516	71702	66521	66531	13530	71712	44425	44376	66530
13538	44344	34750	62811	62812	425135	62819	71731	66524
26702	62824	66505	53208	34722	22737	34730	13557	26735
53242	66534	26730	44318	62805	53239	22743		
44365	71737		71719					
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
13513	44373	44348	13505	22728	22736	71725	71736	66518
62809	53203	26709	22714	34763	44313	53243	53238	13531
22733	13516	53220	34734	53237	44355	13518	44304	71723
13524	66509	71701	62807	26723	26718	66513	22706	34711
13503	22719	34738	62831	26715	66533	44317	26719	13528
26744	44423	66517	62813	13550	26733	66510	71706	22727
85104	71740	71739	62806	34705	53228	53226	44383	44420
26724	66519	22749	13533	44320	44431	44341	13510	13540
26714	71707	66536	34701	62808	44321	62825	34739	71735
44426	62803	26704	26701	71729	13546	44319	441140	71721
82601	34706	26716	44334	66512	53216	26732	13535	26727
34725			34748	66511		53231		

Tabela A.5. Calendário de exames da instância 5

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
71710	26730	22741	62808	34750	53213	13552	53215	13542
13548	44425	13546	34728	34759	26725	66518	26719	22722
26736	66533	22724	66522	26706	26714	71731	44433	53245
53219	13501	71708	62820	53236	62807	62823	66512	66524
62822	34738	71725	53206	53226	34708	34763	13561	44332
26701	66517	44319	62821	44308	34762	34730	22738	44386
44340	22749	71721	66519	71736	44355	44311	13540	66509
13528	13539	62806	13530	26744	62812	13527	53242	71707
44376	53238	26727	26716	66507	53221	44320	26707	13535
34741	34727		26731	44318	44359	13547	22727	66511
			44365	22754	66513	22725	71723	44304
			44409			62824		
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
53224	44420	34751	71740	22720	34765	34726	62813	44334
66523	71701	62803	26704	34748	62819	66515	13557	26734
66505	62811	53246	13538	425135	44341	26737	22747	71705
44424	34710	62834	13502	13549	34735	53227	22723	13550
53208	53207	62831	13518	71730	71739	71732	44317	26732
62818	13524	66521	44321	53239	53216	71724	93102	44430
13516	26715	34739	13504	34706	71720	44313	26703	22701
53228	13562	26717	26723	22733	26729	44337	34725	44316
71711	53243	13512	85104	26735	13513	13519	44331	26732
71728	53220	71737	66510	71712	13503	26718	71729	71718
44383	62825	66530	26733	22728	13536	13505	71735	13559
44432			34731			62814		

Tabela A.6. Calendário de exames da instância 6

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
66511	66522	62827	13535	62812	66523	26701	53224	26732
13503	71712	13552	44359	34714	26716	34739	62807	66519
53242	44320	26735	22719	22751	53221	66521	44420	22712
53227	66513	26729	62834	53206	66531	34725	13557	26717
26724	13538	34731	22723	13527	34740	71740	13505	66512
34741	13504	425135	13530	22724	66536	66524	71710	44304
62813	71707	66505	53208	13558	66517	71730	13528	44319
26737	22722	71725	44311	66515	22720	62819	53238	34763
26705	66533	71736	13559	44430	34706	62805	66530	22749
44317	62822	26707	13510	44337	13533	34750	13501	26733
71720	44383	13502	44409	13512	53219	26727		
71726			13548					
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
44318	53228	44321	26725	26731	71729	13546	71738	22738
53211	44348	66504	62820	26734	44376	44313	44332	71706
85109	26706	26732	34753	34707	62808	26723	62824	71708
34751	62816	13524	62831	26719	13536	44310	34765	22741
53233	22748	22728	26704	71735	71723	44365	34759	71739
44314	53226	26744	13540	34762	13547	34734	34730	71719
26714	66509	26715	71731	71721	53215	26718	34728	71737
13525	53236	26721	22736	22701	26741	22747	34748	62803
53229	66537	53246	62825	62814	34705	44340	34726	13539
66516	44355	26709	85104	44334	26703	44331	71718	26730
53222	44425	53243	93102	44341	53241	66510	13550	62806
22727			66518			13518		

Tabela A.7. Calendário de exames da instância 7

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
13536	34739	34708	44340	34748	22724	26734	13550	13502
44355	66524	71723	66515	22714	26716	22747	66505	22720
44304	44425	44376	53214	66537	44316	26730	53243	13559
66533	26737	26727	62816	66517	26725	44430	62807	22723
13538	53213	71721	26718	44341	13558	71731	22737	26735
53220	22741	44321	44334	44308	71712	22754	26732	13527
71729	62824	44317	34726	71720	34740	13528	62822	22722
44311	13518	13547	53228	71719	22708	13533	71706	22701
26731	13513	62820	71738	66521	71724	62823	66512	34763
62813	71736		34738	53216	53242	66536	13504	44318
			13530	62831	22727	62806	62825	66503
			53238			26703		
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
62821	53244	34759	34728	22728	53215	62818	26733	13505
22730	44337	53206	22749	71708	13501	44314	26729	71735
22735	71707	22736	71737	66534	26741	71718	22738	53226
66523	71702	71732	44331	34706	66522	44420	62811	66513
62814	34736	34762	34753	44409	26724	66510	13524	62819
53241	66516	53219	34725	26723	66509	34741	44332	26701
34750	28002	22733	26707	71725	62803	44313	34731	66511
13503	66531	71710	13562	34764	44319	34707	13552	26736
13512	441114	26719	44365	53227	85104	26732	13516	53221
66519	26744	34755	34730	34705	13548	44383	71730	53222
44359	34710	66530	13546	44431	13549	425135	44320	66518
62808			13535			34713		

Tabela A.8. Calendário de exames da instância 8

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
13550	13513	34707	13558	62827	44355	34730	71737	71730
44334	26701	34706	22720	13546	44337	34705	66522	13528
13538	26725	66521	13524	66536	93102	26727	22741	26707
71731	13527	53215	26703	22754	71724	66516	13552	22725
26733	62803	26735	66509	13503	53233	66520	26723	62824
13504	26708	34763	66507	425135	66505	44321	22749	26729
62813	62825	62822	85104	53221	34708	53216	44311	44340
71723	53242	71721	22747	13530	71727	62819	26731	71739
13501	34750	62807	82611	62814	44310	66533	44304	26715
53228	71710	71714	34740	62812	71740	71720	26705	53226
53238						13535		
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
66524	22722	26743	13547	53237	13557	66513	66512	53241
71707	22724	71708	44376	26737	26734	22728	34739	53224
44431	44316	44365	53239	34725	44320	53240	26732	34738
62828	53245	13559	71725	13502	66510	22727	13505	13518
34736	22711	44386	71736	34741	71718	34726	13548	62820
34744	62818	22701	22738	34731	13536	66518	71729	44430
26704	34711	66511	44319	62831	71735	34748	62811	44425
53208	26718	44331	26730	71712	34734	62823	44420	22723
66506	13516	22735	34728	13540	53236	26719	62806	26724
53227	26716	22751				44318	71726	44332

Tabela A.9. Calendário de exames da instância 9

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
22728	22738	44420	53222	22727	425148	26723	22754	44383
53245	53241	13510	66513	13512	66523	66530	44386	66518
34741	71730	26733	53228	44409	26714	22747	26727	66517
13501	26729	13550	62811	13524	66519	53240	34759	13535
44311	26735	44376	34727	66515	13558	34705	71712	53227
26731	71723	34731	66506	62816	66509	53216	62823	66522
53239	62814	26724	26741	71726	34728	62821	13552	71729
71736	62803	13538	44319	71737	66516	66521	62812	13562
44430	62807	71708	62818	34710	71710	53237	44317	26701
13561	71706	13546	44321	34726	13530	62819	71718	66505
44320						44331		
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
44425	53226	26726	13528	34763	22749	22723	66510	66512
66511	13534	71739	44365	13548	44340	26737	71721	44433
26743	66534	71705	53243	44304	62827	62822	26707	34706
34762	34738	71740	34725	26732	53220	13557	13536	26725
66536	44318	71713	71725	26730	13540	53215	66524	44310
34754	26718	62834	66533	34750	44355	13533	13502	13513
62820	22733	22730	62813	13504	13505	13547	34748	53224
13516	44301	22735	13527	22701	53238	34701	22741	34730
62808	44313	22708	26719	62825	71731	71738	71735	26734
26704	53212	44359				62806	22737	62824
						44332		

Tabela A.10. Calendário de exames da instância 10

Dia 1			Dia 2			Dia 3		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
26736	62811	66534	66520	26741	53222	13518	53239	71725
44311	26716	13534	66522	66523	44339	62803	71721	13548
62818	71719	44314	34714	34736	13512	66509	66516	13502
71729	66505	34740	44383	22724	71707	26737	44310	62806
62831	34753	53233	13513	71713	53214	44425	26732	71731
53247	71727	22751	44318	62825	22735	22728	13527	22747
22725	66513	34744	34755	66519	22714	44386	44321	26704
13552	44409	44355	26702	62827	66517	34725	26703	34730
22719	71702	34751	53203	22736	85109	34763	53227	66524
13542	22749	53237	22737	66507	53241	53228	13550	62812
44365	62805	34734	44301	44341	93102	71735	53216	26719
Dia 4			Dia 5			Dia 6		
Período 1	Período 2	Período 3	Período 1	Período 2	Período 3	Período 1	Período 2	Período 3
34731	34707	66530	66512	34748	44334	53226	34741	85104
44376	62819	13535	425135	44331	34705	34750	71730	26727
26734	26743	22745	66518	34728	66511	62822	22743	62824
13538	22701	22727	53204	26705	22738	13504	44420	13501
26725	34762	71738	44316	53215	71712	13540	71710	62807
66531	62808	26732	13530	13546	44319	13547	26723	53245
44340	44317	62823	53208	62813	26735	26733	66510	26731
66536	26730	13524	71723	13557	34726	13533	44304	26701
71708	13536	66533	44430	26718	22741	53243	53242	62820
34706	71740	44359	71718	441114	66521	26729		
34710	53238	71736	13528	44373	13505			

Tabela A.11. Alunos e respectivos exames (instância 1).

CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2
A1	34731	-	A100	71731	13536	A199	53224	53211	A298	62807	66522
A2	34708	-	A101	44383	44331	A200	44321	-	A299	62822	-
A3	13557	22741	A102	62814	53243	A201	62823	-	A300	66524	-
A4	26733	26734	A103	62807	13504	A202	13504	62828	A301	66523	-
A5	34731	34730	A104	26729	22701	A203	66533	-	A302	66511	-
A6	44332	13547	A105	26727	26735	A204	62825	66505	A303	66534	-
A7	44311	44320	A106	13538	-	A205	62805	-	A304	66513	-
A8	13557	13509	A107	26730	13538	A206	26715	66505	A305	66520	62820
A9	22747	22745	A108	26730	-	A207	66530	-	A306	66518	-
A10	34706	13528	A109	13538	-	A208	66517	-	A307	62818	-
A11	13547	44331	A110	13538	-	A209	22738	-	A308	62813	62822
A12	62823	-	A111	13528	-	A210	62807	-	A309	66522	-
A13	34731	-	A112	13528	26735	A211	62819	-	A310	66510	-
A14	34741	-	A113	44331	-	A212	66533	66522	A311	66515	66524
A15	13505	53243	A114	26729	-	A213	22738	62822	A312	66507	-
A16	13547	-	A115	34741	-	A214	22738	62825	A313	22738	66517
A17	44386	44430	A116	22727	22728	A215	13504	22738	A314	66536	-
A18	26725	-	A117	13557	-	A216	62819	-	A315	66524	-
A19	26703	13528	A118	13512	22749	A217	66510	-	A316	62812	66517
A20	13547	-	A119	22751	-	A218	22738	44376	A317	62806	-
A21	13528	-	A120	71732	71713	A219	44376	-	A318	62807	66533
A22	44340	44319	A121	71725	-	A220	22738	66512	A319	44376	22738
A23	22741	-	A122	13536	-	A221	66536	66530	A320	66517	-
A24	22701	26715	A123	71721	-	A222	62820	44313	A321	62825	62808
A25	26732	-	A124	71725	71730	A223	66523	66521	A322	66509	-
A26	26727	-	A125	44304	-	A224	13504	-	A323	62823	44376
A27	26737	34706	A126	53226	-	A225	66513	-	A324	66513	-
A28	26734	-	A127	53216	53215	A226	13504	-	A325	66510	-
A29	26735	-	A128	62814	62820	A227	44420	-	A326	22738	-
A30	34759	34741	A129	22737	-	A228	71714	71712	A327	66522	-
A31	13538	13540	A130	66533	62819	A229	26736	26732	A328	66512	66533
A32	13528	-	A131	22738	-	A230	13528	26736	A329	62807	62822
A33	34726	-	A132	13504	62822	A231	26735	44348	A330	66515	-
A34	53216	53210	A133	13504	-	A232	26735	-	A331	62807	-
A35	13539	34706	A134	44376	-	A233	26732	-	A332	22738	-
A36	26734	-	A135	13528	26729	A234	34728	34730	A333	62824	62823
A37	44301	13505	A136	53227	13505	A235	34731	-	A334	62813	-
A38	71712	-	A137	13547	-	A236	34713	13525	A335	66533	13504
A39	71736	71723	A138	34750	34741	A237	53233	-	A336	62807	62819
A40	71706	-	A139	71730	71718	A238	22738	44376	A337	53243	-
A41	44355	-	A140	71708	71735	A239	53241	53214	A338	13504	22738
A42	71725	-	A141	71740	22741	A240	53226	53228	A339	62819	66533
A43	13504	-	A142	44409	-	A241	53229	-	A340	62822	66533
A44	22725	-	A143	53215	53227	A242	13503	-	A341	66522	66509
A45	22738	-	A144	66530	-	A243	53226	-	A342	66530	-
A46	22738	62819	A145	62819	-	A244	53220	53226	A343	44321	85104
A47	53242	13503	A146	66510	-	A245	53219	53215	A344	44317	-
A48	13528	425135	A147	66533	66522	A246	53224	53227	A345	44320	-
A49	22728	-	A148	22738	-	A247	53237	-	A346	44420	-
A50	71725	-	A149	62808	-	A248	53239	53240	A347	44313	44311
A51	13535	-	A150	13504	62823	A249	71731	-	A348	44317	-
A52	13504	62807	A151	22738	-	A250	71731	71730	A349	62831	44317
A53	71708	71712	A152	66533	-	A251	26719	71703	A350	44313	-
A54	34731	-	A153	66510	66533	A252	26717	26714	A351	44321	-
A55	44365	-	A154	44314	-	A253	13502	26718	A352	85109	44340
A56	13546	-	A155	44320	-	A254	13502	13501	A353	26733	26737
A57	13552	71708	A156	22741	71736	A255	13550	-	A354	26737	-
A58	66524	71723	A157	26731	26735	A256	34731	-	A355	22741	-
A59	66509	66503	A158	13528	-	A257	13512	-	A356	71727	-
A60	15021	-	A159	13540	13538	A258	13524	-	A357	71737	-
A61	22754	44311	A160	13528	-	A259	22727	13516	A358	13501	-
A62	22701	13538	A161	26727	26735	A260	44425	34710	A359	26718	-
A63	13538	-	A162	13528	425135	A261	44332	44359	A360	22746	22747
A64	13540	-	A163	34728	-	A262	44425	44355	A361	22741	71731
A65	13502	-	A164	34731	34728	A263	71730	53238	A362	71731	71725
A66	34731	-	A165	34731	34739	A264	71730	13536	A363	66515	44376
A67	71731	-	A166	13505	-	A265	71739	-	A364	71730	71739
A68	13540	-	A167	53215	53226	A266	22741	-	A365	71731	71725
A69	15021	-	A168	53227	-	A267	71730	-	A366	44341	71740
A70	22741	-	A169	53229	53208	A268	71723	-	A367	71723	34705
A71	26737	-	A170	53215	-	A269	71739	22741	A368	71723	71730
A72	34730	-	A171	13524	-	A270	71730	-	A369	34741	34725
A73	34748	13548	A172	13547	44331	A271	71730	71740	A370	34738	22741
A74	22741	-	A173	44359	-	A272	53238	66524	A371	34753	-
A75	62819	44376	A174	71740	-	A273	71725	-	A372	22741	34751
A76	13504	-	A175	71725	-	A274	71731	71736	A373	34738	-
A77	44376	-	A176	71721	71730	A275	71730	-	A374	13550	-
A78	62813	62825	A177	22741	-	A276	71723	71736	A375	22741	34750
A79	44320	85104	A178	71721	-	A277	71725	71730	A376	34738	22741
A80	71730	22741	A179	71725	71740	A278	34748	-	A377	34725	22741
A81	26723	26735	A180	34741	441114	A279	34741	-	A378	34750	22741
A82	26730	-	A181	34741	-	A280	34725	-	A379	34748	34744
A83	13560	-	A182	13550	34725	A281	34748	-	A380	26734	-
A84	13552	13536	A183	34725	-	A282	22741	-	A381	22736	-
A85	13501	-	A184	22741	53238	A283	26729	-	A382	44340	-
A86	22701	13501	A185	13528	-	A284	62814	-	A383	44425	44426
A87	13555	22741	A186	13530	-	A285	62824	62814	A384	66512	66522
A88	22741	-	A187	71737	71729	A286	22701	26737	A385	26703	26732
A89	44331	-	A188	62819	-	A287	22701	26729	A386	26729	-
A90	13536	13535	A189	13504	-	A288	71731	71725	A387	34741	-
A91	71725	71730	A190	62813	62819	A289	13535	-	A388	22738	-
A92	22741	-	A191	62813	62822	A290	44425	-	A389	13504	66512
A93	71725	71730	A192	44304	-	A291	34722	-	A390	62803	44340
A94	26727	-	A193	71712	-	A292	71731	-	A391	44320	44420
A95	22728	-	A194	22741	-	A293	66533	-	A392	44353	44386
A96	13502	-	A195	13538	-	A294	34710	44340	A393	62822	62824
A97	22727	22723	A196	13546	-	A295	13501	13502	A394	34737	34736
A98	44425	44383	A197	53224	53221	A296	13513	-			
A99	71730	-	A198	71730	-	A297	22738	-			

Tabela A.12. Alunos e respectivos exames (instância 2).

CD_ALUNO	CD_DISCIP.1	CD_DISCIP.2	CD_ALUNO	CD_DISCIP.1	CD_DISCIP.2	CD_ALUNO	CD_DISCIP.1	CD_DISCIP.2	CD_ALUNO	CD_DISCIP.1	CD_DISCIP.2
A1	44334	-	A103	13538	-	A205	66521	-	A307	44318	44311
A2	34708	-	A104	13538	-	A206	62805	-	A308	62824	44376
A3	13557	22741	A105	13528	-	A207	34714	26744	A309	62812	-
A4	26732	-	A106	44331	-	A208	66513	-	A310	66518	66513
A5	26729	-	A107	13547	-	A209	22738	66518	A311	62806	62807
A6	26729	-	A108	26735	26737	A210	62813	-	A312	66517	62822
A7	34734	34730	A109	22714	-	A211	62807	66512	A313	22738	62822
A8	44332	44383	A110	34748	-	A212	62819	-	A314	66524	-
A9	13538	-	A111	22727	22728	A213	66518	62822	A315	66513	-
A10	34711	34728	A112	13536	-	A214	62824	-	A316	66524	-
A11	34706	13528	A113	34731	34763	A215	34714	13504	A317	66518	-
A12	13547	-	A114	13518	13557	A216	13504	-	A318	66517	-
A13	13504	22738	A115	22706	26719	A217	62819	-	A319	44376	22738
A14	34763	34741	A116	13557	-	A218	62807	62824	A320	66513	-
A15	44332	-	A117	13536	-	A219	62819	-	A321	62813	62824
A16	34706	-	A118	71718	71725	A220	62820	62822	A322	66511	-
A17	26730	-	A119	71725	-	A221	62831	-	A323	66523	-
A18	13547	-	A120	13536	-	A222	13504	22738	A324	62807	66512
A19	71735	71737	A121	53216	-	A223	44316	62803	A325	22738	-
A20	44430	13546	A122	13505	53242	A224	44311	-	A326	66524	-
A21	26704	26725	A123	53215	-	A225	44320	-	A327	62806	22738
A22	26727	34707	A124	53243	53238	A226	22754	44310	A328	66513	-
A23	13546	-	A125	62819	66533	A227	71714	71712	A329	62813	-
A24	26727	26735	A126	22738	44376	A228	26737	-	A330	62825	22738
A25	44317	-	A127	62819	22738	A229	26736	-	A331	22738	-
A26	22741	-	A128	13504	62812	A230	44348	-	A332	66533	-
A27	26730	26724	A129	13528	-	A231	13538	-	A333	62824	62819
A28	13540	26734	A130	13547	-	A232	26730	26735	A334	62807	66533
A29	26737	34706	A131	34741	-	A233	34728	34739	A335	62823	62828
A30	26734	-	A132	71735	-	A234	34730	34731	A336	62822	62823
A31	26735	-	A133	44409	-	A235	13525	-	A337	66534	13504
A32	13540	-	A134	13505	-	A236	53241	53239	A338	66522	-
A33	13528	-	A135	62822	-	A237	62825	22738	A339	62812	66511
A34	13557	-	A136	66510	-	A238	53231	-	A340	53206	-
A35	34726	26701	A137	66522	-	A239	53228	-	A341	22738	13504
A36	34741	-	A138	66511	62812	A240	53216	-	A342	66515	-
A37	13540	34706	A139	66511	-	A241	53216	53224	A343	66519	66511
A38	26723	-	A140	66530	-	A242	53238	93102	A344	66533	-
A39	13538	-	A141	62822	-	A243	53227	-	A345	62824	66505
A40	13505	-	A142	66510	-	A244	53241	53242	A346	66531	-
A41	71712	-	A143	13504	66518	A245	71729	-	A347	66512	66533
A42	71723	-	A144	44304	-	A246	71731	-	A348	66513	-
A43	71706	71729	A145	62825	62820	A247	71731	-	A349	66513	-
A44	44355	-	A146	44320	-	A248	71729	22741	A350	62821	22738
A45	22741	13536	A147	44331	-	A249	13502	-	A351	62806	62822
A46	13504	-	A148	13528	-	A250	13502	-	A352	44320	-
A47	13518	22722	A149	71730	-	A251	13502	-	A353	44304	44420
A48	22738	-	A150	26732	26736	A252	13504	22738	A354	85104	-
A49	22738	62814	A151	26730	26704	A253	26717	26718	A355	85104	62803
A50	53227	13503	A152	26733	13538	A254	34755	-	A356	44317	-
A51	13528	-	A153	13538	13540	A255	34731	34728	A357	44420	44321
A52	13528	-	A154	26735	-	A256	13550	-	A358	44340	62803
A53	13538	-	A155	13538	13528	A257	22749	-	A359	44316	-
A54	22728	-	A156	34763	34728	A258	44425	44383	A360	44304	44319
A55	13535	-	A157	85109	34730	A259	44383	-	A361	44316	-
A56	22741	-	A158	13505	-	A260	44425	34710	A362	44319	44320
A57	13539	26735	A159	53215	53228	A261	71725	71721	A363	44321	44314
A58	71712	22741	A160	53208	-	A262	71730	53238	A364	44317	44311
A59	34731	-	A161	53215	-	A263	13536	71725	A365	44313	-
A60	44425	-	A162	62814	44376	A264	71725	-	A366	44319	-
A61	66510	-	A163	22720	22714	A265	71740	-	A367	44311	-
A62	15021	-	A164	71725	-	A266	22741	71730	A368	44318	44340
A63	26737	-	A165	26716	-	A267	71723	71725	A369	26732	26729
A64	13540	34706	A166	22741	-	A268	22741	71736	A370	26730	44348
A65	13502	-	A167	22733	22749	A269	71736	-	A371	22741	-
A66	34763	-	A168	44350	44332	A270	71725	71723	A372	26718	26719
A67	22741	-	A169	44344	-	A271	22741	-	A373	13501	-
A68	44348	13538	A170	44332	44359	A272	71725	-	A374	26707	-
A69	13538	13540	A171	71740	-	A273	22741	-	A375	22728	-
A70	15021	-	A172	71725	-	A274	71730	-	A376	71730	-
A71	22741	-	A173	71725	-	A275	34748	34741	A377	71730	71723
A72	13530	13527	A174	22741	-	A276	34725	-	A378	44376	-
A73	34730	34731	A175	71721	71730	A277	34725	22741	A379	71725	-
A74	13550	-	A176	71725	71730	A278	26733	26729	A380	71736	22741
A75	44376	-	A177	22741	-	A279	26733	13528	A381	34762	22741
A76	13504	62813	A178	71721	71730	A280	13504	-	A382	34755	-
A77	62824	44376	A179	71725	71740	A281	26729	85104	A383	34750	-
A78	34705	71708	A180	34741	-	A282	22701	26729	A384	22741	34750
A79	26735	34716	A181	34741	-	A283	13528	-	A385	22741	-
A80	26730	-	A182	34741	34725	A284	44334	-	A386	13550	34755
A81	13503	13505	A183	62822	62814	A285	71730	71725	A387	22741	34750
A82	13560	-	A184	13528	26735	A286	66522	66509	A388	34725	22741
A83	13552	-	A185	13530	-	A287	62806	62813	A389	26730	-
A84	13501	71720	A186	71735	-	A288	44337	-	A390	44433	44355
A85	22701	13501	A187	71723	-	A289	71731	-	A391	44383	-
A86	13535	-	A188	71713	-	A290	13501	13502	A392	62803	44340
A87	71718	-	A189	22738	-	A291	22723	13524	A393	66513	-
A88	26701	71718	A190	62822	62823	A292	22741	-	A394	66510	-
A89	71725	22741	A191	44304	-	A293	66534	-	A395	22738	-
A90	26735	26727	A192	22741	-	A294	66512	-	A396	22738	66518
A91	13502	-	A193	13548	-	A295	66519	-	A397	66524	66536
A92	22723	-	A194	13538	-	A296	66534	66533	A398	66531	-
A93	71730	-	A195	13547	-	A297	66510	-	A399	66533	62822
A94	71730	-	A196	71714	71738	A298	66518	62819	A400	62820	-
A95	71735	13536	A197	53224	-	A299	66513	-	A401	62803	-
A96	34741	34750	A198	71730	-	A300	66513	-	A402	44318	44340
A97	13538	26734	A199	44321	44319	A301	66533	66522	A403	22735	71707
A98	26729	-	A200	62827	22738	A302	66515	-	A404	62821	22738
A99	71718	71719	A201	62823	-	A303	13504	62808	A405	62825	-
A100	26730	-	A202	22738	13504	A304	62819	62822	A406	66520	62825
A101	26730	13538	A203	13504	-	A305	62822	-	A407	22754	44386
A102	13538	26737	A204	66519	-	A306	62822	-	A408	34740	34736

Tabela A.13. Alunos e respectivos exames (instância 3).

CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2	CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2	CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2	CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2
A1	34731	-	A99	71718	71719	A197	66517	-	A294	62818	-
A2	13547	13542	A100	26730	-	A198	62813	-	A295	66513	62822
A3	26734	26731	A101	13528	13538	A199	62819	-	A296	22738	-
A4	34707	26727	A102	13538	-	A200	62819	-	A297	22738	-
A5	34708	-	A103	13538	-	A201	22738	-	A298	62818	-
A6	53247	-	A104	13538	13528	A202	13504	-	A299	66510	66522
A7	22723	22724	A105	44331	-	A203	13504	-	A300	66533	66534
A8	62813	44376	A106	26729	-	A204	66518	-	A301	44320	44311
A9	26737	26734	A107	22741	-	A205	62824	-	A302	66530	62825
A10	44332	13547	A108	34741	34725	A206	22738	62824	A303	44376	66512
A11	44311	-	A109	34748	34741	A207	66530	66513	A304	66512	66504
A12	34701	13527	A110	465102	-	A208	44319	66509	A305	66517	-
A13	13538	-	A111	22727	-	A209	62819	44313	A306	66524	-
A14	26707	-	A112	71731	-	A210	13504	-	A307	22738	66533
A15	44331	-	A113	13536	71725	A211	66513	66524	A308	22738	66521
A16	62822	13504	A114	26707	13557	A212	62807	-	A309	62824	62822
A17	71730	-	A115	22723	13510	A213	13504	22738	A310	66510	-
A18	34763	-	A116	13536	-	A214	13546	62803	A311	62821	-
A19	13505	-	A117	71730	-	A215	44304	44310	A312	62822	-
A20	22754	44355	A118	71730	-	A216	44318	-	A313	66519	-
A21	34706	-	A119	71721	-	A217	44310	13547	A314	66511	-
A22	71707	-	A120	71736	71735	A218	44311	-	A315	66517	66524
A23	44331	13547	A121	62803	44331	A219	13535	-	A316	62807	-
A24	71723	71726	A122	53226	-	A220	71710	-	A317	62822	44319
A25	26703	-	A123	53216	-	A221	71710	13535	A318	66519	-
A26	26734	26727	A124	66522	-	A222	44348	13540	A319	62813	62822
A27	13546	-	A125	13503	53227	A223	44348	26736	A320	62819	62807
A28	13528	-	A126	22738	13504	A224	26735	-	A321	62807	-
A29	26727	26735	A127	62808	22738	A225	26729	-	A322	66512	66522
A30	44340	-	A128	66512	66518	A226	26737	26729	A323	22738	-
A31	22741	-	A129	44376	13504	A227	34730	-	A324	66518	22738
A32	26730	26724	A130	13547	-	A228	34730	34731	A325	62822	-
A33	13540	-	A131	71708	71735	A229	53238	53237	A326	62819	66518
A34	13538	26734	A132	22741	-	A230	53228	-	A327	62806	-
A35	26735	-	A133	53215	-	A231	53216	53226	A328	53243	53206
A36	34763	-	A134	13505	-	A232	53227	-	A329	66519	66513
A37	26703	13539	A135	62822	-	A233	53237	-	A330	66513	-
A38	13538	13540	A136	13504	66533	A234	53238	-	A331	62823	-
A39	13528	-	A137	66510	-	A235	53237	53243	A332	44376	62806
A40	26701	-	A138	62808	-	A236	13536	71725	A333	62819	62822
A41	13538	13528	A139	62822	-	A237	71737	-	A334	66511	-
A42	34741	34765	A140	62822	-	A238	71707	-	A335	62816	-
A43	26723	26737	A141	66533	-	A239	71731	-	A336	22738	44376
A44	53214	13505	A142	66510	66522	A240	26716	71703	A337	66509	62807
A45	71714	71712	A143	66518	-	A241	26717	13502	A338	44308	44320
A46	13536	71723	A144	44320	-	A242	66521	44376	A339	85104	-
A47	71706	71725	A145	13538	-	A243	34750	-	A340	44420	-
A48	62819	-	A146	26730	26735	A244	34763	34731	A341	44320	44304
A49	13558	22722	A147	26730	-	A245	34731	22741	A342	44340	-
A50	22738	-	A148	13538	-	A246	13524	13512	A343	93102	-
A51	13528	-	A149	13538	425135	A247	22749	22751	A344	44318	44420
A52	13533	13535	A150	34763	34728	A248	44383	-	A345	44319	44313
A53	22741	-	A151	44301	13505	A249	44331	34710	A346	44340	-
A54	34763	34759	A152	53226	-	A250	44304	44313	A347	44340	-
A55	34731	-	A153	53241	53246	A251	44359	13547	A348	44316	-
A56	22741	71731	A154	53208	53216	A252	44425	-	A349	44319	-
A57	13518	13559	A155	53215	-	A253	71730	22741	A350	44313	-
A58	15021	-	A156	26718	26719	A254	53238	-	A351	26729	-
A59	44318	13547	A157	22701	-	A255	71730	13536	A352	26730	26733
A60	26734	34706	A158	22741	-	A256	71731	-	A353	22741	-
A61	13502	-	A159	22720	22728	A257	71725	-	A354	22741	-
A62	13536	71708	A160	44432	44425	A258	71725	-	A355	26719	-
A63	22741	-	A161	44334	-	A259	71710	22741	A356	34759	34728
A64	13538	13540	A162	26701	71730	A260	71731	71740	A357	13513	13512
A65	13538	13540	A163	71725	-	A261	71736	71740	A358	66524	-
A66	26724	26727	A164	71725	-	A262	53238	71726	A359	26701	22741
A67	15021	-	A165	71721	71730	A263	53243	71730	A360	34741	34725
A68	22741	-	A166	71730	71736	A264	71731	71721	A361	22741	-
A69	34730	34731	A167	71702	71740	A265	71726	-	A362	22741	34748
A70	34748	34741	A168	71730	-	A266	71725	71731	A363	22741	-
A71	62813	-	A169	71725	-	A267	71725	-	A364	22741	-
A72	62824	44376	A170	34741	-	A268	71730	-	A365	22741	26701
A73	44316	-	A171	34725	-	A269	71725	-	A366	22741	-
A74	71735	71708	A172	62819	-	A270	34741	-	A367	22741	-
A75	26735	-	A173	13528	-	A271	34725	-	A368	34725	13550
A76	13531	13560	A174	13530	-	A272	34762	34725	A369	34731	34740
A77	13552	-	A175	71740	71725	A273	22741	-	A370	26733	26735
A78	13501	-	A176	71725	-	A274	26732	26729	A371	44425	44426
A79	13501	-	A177	71737	71735	A275	13504	-	A372	44383	44355
A80	13535	22741	A178	66533	-	A276	26729	13538	A373	44321	44317
A81	22741	-	A179	22738	-	A277	26729	-	A374	44365	44340
A82	44332	62825	A180	13504	-	A278	26737	-	A375	44331	44384
A83	44331	-	A181	62822	-	A279	34763	34731	A376	66533	66512
A84	13535	-	A182	13546	-	A280	71730	13533	A377	26732	26729
A85	71723	-	A183	22741	71712	A281	66518	62806	A378	26729	22701
A86	22741	-	A184	71723	-	A282	44311	-	A379	34731	34741
A87	71723	71730	A185	71738	71708	A283	26735	-	A380	22738	-
A88	71725	-	A186	71718	-	A284	66524	66523	A381	62803	44316
A89	13540	26727	A187	13552	22711	A285	13501	-	A382	44340	-
A90	22728	22719	A188	34705	-	A286	13513	-	A383	22728	22751
A91	13502	-	A189	13538	-	A287	66518	66533	A384	34755	22741
A92	71730	-	A190	71730	-	A288	66524	66517	A385	13505	-
A93	71730	-	A191	44317	-	A289	66510	66533	A386	71725	71737
A94	22741	-	A192	66518	-	A290	66512	-	A387	22738	-
A95	22738	-	A193	62812	-	A291	66518	66512	A388	62820	-
A96	71725	-	A194	62824	-	A292	66517	-	A389	22754	44388
A97	26729	22701	A195	66505	-	A293	66524	-	A390	62819	62813
A98	26737	26725	A196	62808	-						

Tabela A.14. Alunos e respectivos exames (instância 4).

CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2
A1	13510	13561	A110	22728	22725	A218	62824	62819	A326	62812	-
A2	34731	44334	A111	13518	13557	A219	62805	-	A327	66533	66518
A3	13547	13546	A112	82601	22706	A220	66524	-	A328	62807	66522
A4	34708	-	A113	13516	22725	A221	44319	62824	A329	62822	-
A5	22723	22728	A114	13536	-	A222	62824	62813	A330	62824	62813
A6	13531	34701	A115	22741	-	A223	62824	-	A331	62822	-
A7	34731	34734	A116	22741	71725	A224	62824	-	A332	66533	-
A8	44332	-	A117	71730	44341	A225	66533	-	A333	66520	-
A9	44311	-	A118	71725	-	A226	66516	66530	A334	62831	-
A10	13538	-	A119	34705	13535	A227	34714	13504	A335	66513	66531
A11	22747	-	A120	71725	-	A228	13504	22738	A336	62812	-
A12	26729	22701	A121	71735	71730	A229	62819	-	A337	66531	-
A13	13528	-	A122	44331	-	A230	66510	-	A338	44319	66534
A14	34730	34735	A123	53216	53226	A231	62824	-	A339	66518	-
A15	13547	44331	A124	53237	53243	A232	62819	-	A340	22738	66519
A16	34731	34730	A125	53216	53215	A233	44376	-	A341	66517	-
A17	26701	13535	A126	62809	53215	A234	62820	-	A342	62819	-
A18	71730	-	A127	22737	-	A235	62824	66522	A343	62819	62808
A19	34763	34741	A128	66533	-	A236	22738	-	A344	66517	-
A20	53246	53215	A129	62811	-	A237	66524	66521	A345	22738	66521
A21	44355	44332	A130	62808	66505	A238	62824	-	A346	62824	-
A22	71728	-	A131	66512	62822	A239	13504	22738	A347	22738	-
A23	44331	13547	A132	44317	44313	A240	44310	-	A348	66522	62821
A24	13546	-	A133	26729	26723	A241	71710	71738	A349	66512	66510
A25	26734	-	A134	66511	13504	A242	26730	26704	A350	62819	62822
A26	26723	13528	A135	13547	-	A243	13528	-	A351	62807	-
A27	13547	-	A136	34741	-	A244	425135	-	A352	22738	-
A28	13528	-	A137	71723	-	A245	26735	44348	A353	66512	66534
A29	26727	-	A138	71708	71735	A246	26730	-	A354	62823	-
A30	22741	71726	A139	22741	-	A247	34730	34739	A355	62822	62823
A31	34727	34730	A140	13505	-	A248	34730	34731	A356	22738	13504
A32	13502	26715	A141	62822	-	A249	34713	34730	A357	62813	-
A33	13540	26734	A142	62819	-	A250	53239	-	A358	62824	62813
A34	13538	26734	A143	66512	66522	A251	53227	53231	A359	53215	53203
A35	26729	-	A144	66533	-	A252	13503	53227	A360	13504	22738
A36	26735	-	A145	66513	22738	A253	53226	53227	A361	62823	-
A37	13538	-	A146	66530	62808	A254	53216	-	A362	22738	-
A38	26737	13528	A147	66530	66517	A255	53242	53227	A363	66533	66522
A39	53228	53231	A148	62822	-	A256	53237	-	A364	66524	-
A40	13538	13540	A149	66521	62825	A257	53238	53242	A365	62824	66512
A41	22741	34763	A150	13504	-	A258	71725	71729	A366	62822	62807
A42	71712	-	A151	44313	-	A259	71731	-	A367	66530	66536
A43	71706	-	A152	44420	44331	A260	26702	26744	A368	62820	66530
A44	44355	-	A153	13528	-	A261	71729	71707	A369	62814	62822
A45	22738	-	A154	26737	26735	A262	71730	-	A370	66530	-
A46	13528	-	A155	22741	71725	A263	71725	71729	A371	44376	-
A47	13536	71731	A156	26724	26735	A264	71730	13536	A372	62813	-
A48	26701	13535	A157	26732	26737	A265	26718	71703	A373	44420	-
A49	22741	-	A158	13538	13528	A266	13502	26714	A374	44320	44420
A50	13504	-	A159	13528	-	A267	13502	26718	A375	44304	44423
A51	44341	44365	A160	26721	-	A268	26715	13501	A376	62803	44304
A52	13547	-	A161	13538	-	A269	22738	-	A377	44317	-
A53	13552	-	A162	34763	34728	A270	26709	26719	A378	44321	44317
A54	22741	71723	A163	34713	34739	A271	22743	34763	A379	44321	44340
A55	13559	-	A164	13505	-	A272	34728	-	A380	44340	85104
A56	15021	-	A165	53215	53228	A273	13550	-	A381	44316	44313
A57	44311	13547	A166	53227	-	A274	13524	-	A382	44317	-
A58	34711	34721	A167	53240	-	A275	13524	22727	A383	44310	-
A59	26734	34706	A168	53208	-	A276	13524	-	A384	44320	44340
A60	13502	-	A169	53215	-	A277	44383	44373	A385	44313	-
A61	441140	34763	A170	53242	53215	A278	13547	44426	A386	44340	62831
A62	13535	71708	A171	53222	-	A279	44355	-	A387	44321	44318
A63	13538	13540	A172	13510	-	A280	71730	-	A388	44316	-
A64	15021	-	A173	26719	26716	A281	71725	-	A389	44340	44320
A65	22741	-	A174	22701	-	A282	22741	71730	A390	62831	44420
A66	13528	26734	A175	71706	22701	A283	22741	-	A391	26734	26732
A67	34731	-	A176	22714	13536	A284	34705	22741	A392	26729	26733
A68	13550	34741	A177	22741	-	A285	22741	-	A393	34763	34731
A69	34748	22741	A178	22733	-	A286	71730	22741	A394	34762	22741
A70	13550	22741	A179	44331	44344	A287	71730	-	A395	13557	22725
A71	44376	62824	A180	44431	44332	A288	13536	71730	A396	71701	13501
A72	13504	62813	A181	44359	44431	A289	22741	71736	A397	13501	-
A73	62824	-	A182	71725	-	A290	71726	-	A398	13557	22741
A74	62819	-	A183	71730	-	A291	22741	-	A399	22724	22749
A75	71730	71736	A184	71721	71730	A292	71730	71736	A400	22733	22727
A76	26735	-	A185	71730	-	A293	34741	-	A401	71730	13535
A77	26735	-	A186	22741	71740	A294	34725	-	A402	66530	-
A78	66533	-	A187	34741	-	A295	34725	22741	A403	53243	71731
A79	13531	13558	A188	34741	13550	A296	13534	-	A404	22741	71735
A80	13552	-	A189	26727	-	A297	22741	-	A405	71739	22741
A81	22701	13501	A190	13538	-	A298	26729	-	A406	71731	71723
A82	13535	22741	A191	44311	-	A299	26729	-	A407	13550	13549
A83	22720	-	A192	44376	66512	A300	34763	-	A408	22741	-
A84	44425	-	A193	44376	62806	A301	71725	13533	A409	34725	-
A85	71721	-	A194	13528	-	A302	66510	66534	A410	22741	-
A86	13533	13535	A195	13530	-	A303	66524	-	A411	22741	-
A87	71725	71730	A196	71725	71730	A304	44311	-	A412	34750	-
A88	26701	71730	A197	71737	71735	A305	26735	-	A413	34762	22741
A89	22741	-	A198	62822	66518	A306	22741	-	A414	22741	34738
A90	26737	-	A199	13504	66507	A307	13501	-	A415	22741	26701
A91	13558	-	A200	26701	62819	A308	22723	13513	A416	22741	-
A92	71730	-	A201	13546	-	A309	22741	-	A417	22741	34741
A93	13550	-	A202	34705	-	A310	66512	66509	A418	34750	-
A94	34741	34748	A203	53227	53238	A311	62825	22738	A419	26733	26735
A95	66510	-	A204	71702	71735	A312	62822	-	A420	26727	22736
A96	26734	-	A205	34750	-	A313	66522	-	A421	44340	44355
A97	71725	-	A206	53242	-	A314	66509	-	A422	44425	44383
A98	26729	22701	A207	13535	13533	A315	66509	-	A423	22736	22706
A99	71719	-	A208	34750	34741	A316	62812	-	A424	71730	-
A100	26727	-	A209	22741	-	A317	62819	62824	A425	71725	71731
A101	26737	-	A210	13538	-	A318	66519	-	A426	22738	66516
A102	13538	13540	A211	13547	13546	A319	66517	-	A427	22738	-
A103	13538	26735	A212	71738	71708	A320	66513	-	A428	62820	22738
A104	13528	-	A213	53226	-	A321	66520	-	A429	34701	13512
A105	13552	-	A214	71730	-	A322	13504	-	A430	34762	-
A106	34741	-	A215	44317	-	A323	62824	-	A431	13505	53220
A107	22719	22741	A216	62813	-	A324	62822	66511	A432	66537	-
A108	71731	-	A217	22738	-	A325	66512	-	A433	66520	-
A109	71725	-									

Tabela A.15. Alunos e respectivos exames (instância 5).

CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2
A1	13561	13519	A105	22741	-	A208	22722	22723	A311	22738	66533
A2	44433	44334	A106	13540	26729	A209	34725	34741	A312	66524	-
A3	13542	13546	A107	71710	71731	A210	13538	-	A313	66518	66533
A4	34706	26727	A108	22728	-	A211	13547	-	A314	66517	22738
A5	34708	-	A109	13502	-	A212	71725	71735	A315	13504	-
A6	53213	-	A110	71730	-	A213	71730	-	A316	62812	-
A7	22723	22725	A111	71730	-	A214	44308	-	A317	66512	66533
A8	26727	26732	A112	22741	-	A215	13504	-	A318	62822	-
A9	13504	44337	A113	13504	22738	A216	66517	-	A319	62822	66511
A10	26729	-	A114	26729	22701	A217	62813	-	A320	66533	-
A11	26729	26730	A115	71718	71720	A218	66509	62819	A321	66521	-
A12	44332	-	A116	26730	26735	A219	66511	-	A322	66511	66530
A13	44320	-	A117	13538	-	A220	62819	-	A323	93102	-
A14	13557	22741	A118	26737	13538	A221	62822	-	A324	62822	-
A15	13538	-	A119	13538	13540	A222	44319	-	A325	66519	-
A16	22747	26707	A120	13528	-	A223	62822	22738	A326	66511	-
A17	34706	13528	A121	44331	-	A224	13504	-	A327	66533	-
A18	34730	34735	A122	26735	-	A225	62819	-	A328	22738	-
A19	13547	44331	A123	22733	13552	A226	22738	-	A329	66523	66524
A20	13504	22738	A124	34748	34741	A227	66530	-	A330	66524	-
A21	34731	34730	A125	71725	-	A228	13504	-	A331	62806	13504
A22	71730	-	A126	13557	-	A229	62819	-	A332	66519	-
A23	34763	-	A127	13513	-	A230	62819	-	A333	62813	-
A24	44332	-	A128	22724	26719	A231	66510	-	A334	62808	-
A25	34706	-	A129	71712	71705	A232	62824	-	A335	66512	-
A26	26725	-	A130	13536	71725	A233	66518	62811	A336	62819	-
A27	71707	-	A131	22741	71730	A234	13504	62825	A337	44319	66510
A28	44331	-	A132	44304	44331	A235	66530	-	A338	66517	-
A29	71735	-	A133	13505	-	A236	44319	44321	A339	66518	-
A30	44386	44430	A134	13505	53219	A237	44420	-	A340	62819	62824
A31	26727	26704	A135	53216	-	A238	13535	71711	A341	62822	66509
A32	13528	-	A136	62825	-	A239	13536	13535	A342	62824	-
A33	13546	13547	A137	53220	53215	A240	26735	26731	A343	62823	62819
A34	13528	-	A138	53215	71732	A241	26735	-	A344	62821	66515
A35	26735	-	A139	53243	53242	A242	26734	26735	A345	53227	-
A36	44318	-	A140	66533	62819	A243	26730	13538	A346	66524	-
A37	13528	26737	A141	44376	66518	A244	26736	-	A347	62806	62807
A38	26730	26734	A142	13504	62822	A245	34730	34731	A348	13504	-
A39	26729	-	A143	66521	66517	A246	53227	-	A349	66524	-
A40	26735	-	A144	44311	44313	A247	53227	53224	A350	22738	-
A41	13538	-	A145	44376	22738	A248	53238	53236	A351	62824	-
A42	13528	-	A146	13528	13538	A249	53216	-	A352	44320	85104
A43	34726	26701	A147	13547	-	A250	53227	-	A353	44304	44317
A44	13539	13538	A148	34741	-	A251	53216	-	A354	44317	-
A45	26723	26737	A149	71723	71730	A252	53227	53228	A355	44340	85104
A46	53245	13505	A150	71708	71735	A253	53238	53239	A356	44321	-
A47	71712	-	A151	71723	-	A254	53238	53242	A357	44321	-
A48	71730	-	A152	44409	-	A255	71724	71725	A358	44317	-
A49	22722	-	A153	13505	-	A256	26744	-	A359	85104	-
A50	22738	-	A154	22727	-	A257	71707	71724	A360	44316	-
A51	22738	-	A155	62819	-	A258	71731	-	A361	44340	44420
A52	53215	13503	A156	62822	-	A259	22741	13536	A362	44316	44321
A53	13528	425135	A157	66510	-	A260	26714	-	A363	44316	44311
A54	22701	13528	A158	66522	-	A261	13502	26718	A364	44313	44320
A55	13538	-	A159	66517	-	A262	13502	13501	A365	44340	62831
A56	22728	-	A160	62813	62819	A263	26715	13501	A366	44311	44317
A57	71739	-	A161	13504	66505	A264	71705	71701	A367	44320	-
A58	26701	13536	A162	66533	-	A265	13550	-	A368	44319	44317
A59	71712	22741	A163	66533	-	A266	34731	-	A369	26733	26730
A60	34738	34765	A164	44313	-	A267	13513	13512	A370	26730	26732
A61	34731	-	A165	44311	-	A268	22728	22749	A371	22741	-
A62	44341	44425	A166	71705	71712	A269	13524	22749	A372	13527	13557
A63	13547	13546	A167	26735	-	A270	22727	22749	A373	13516	13527
A64	44332	-	A168	26735	26703	A271	44355	-	A374	71739	71728
A65	22741	-	A169	26736	26735	A272	44425	34710	A375	62834	71729
A66	66510	-	A170	13528	-	A273	71730	53238	A376	26718	-
A67	13518	13559	A171	26704	-	A274	71731	-	A377	26717	13501
A68	15021	-	A172	13538	13528	A275	71725	71730	A378	26718	13501
A69	22724	-	A173	34728	-	A276	71723	71740	A379	26707	22724
A70	22754	-	A174	34731	-	A277	71730	-	A380	22727	22724
A71	26737	-	A175	34731	34739	A278	71725	71740	A381	71721	13536
A72	13540	34706	A176	53226	-	A279	71721	-	A382	71725	-
A73	13502	-	A177	53221	-	A280	71731	-	A383	71731	71740
A74	34731	34763	A178	53242	53246	A281	71736	-	A384	66524	71740
A75	13535	13536	A179	53208	53216	A282	34748	34741	A385	22741	34725
A76	22741	-	A180	53215	-	A283	34741	-	A386	13559	13562
A77	13540	-	A181	62825	-	A284	34725	-	A387	34750	-
A78	26703	26727	A182	26706	-	A285	22741	-	A388	22741	-
A79	15021	-	A183	26719	26716	A286	13528	-	A389	22741	-
A80	22741	-	A184	13536	13552	A287	22738	62814	A390	34750	-
A81	34730	34731	A185	22733	22749	A288	13504	-	A391	34762	-
A82	13550	34741	A186	13547	44331	A289	26729	13528	A392	34751	22741
A83	34748	13550	A187	71740	-	A290	26729	26734	A393	34738	-
A84	13549	13548	A188	71725	-	A291	71725	-	A394	26733	26734
A85	62819	-	A189	71730	71736	A292	44425	-	A395	26727	26733
A86	13504	-	A190	71730	-	A293	34763	34731	A396	44365	44383
A87	62824	44376	A191	22741	-	A294	71731	-	A397	44383	-
A88	62814	-	A192	34741	-	A295	62813	62806	A398	62803	44319
A89	44320	-	A193	34741	-	A296	44376	-	A399	44424	44355
A90	71735	-	A194	34741	34725	A297	44340	-	A400	66511	93102
A91	26735	26730	A195	26725	-	A298	53242	53239	A401	26734	-
A92	13528	26732	A196	13528	13538	A299	66524	44337	A402	34726	34759
A93	34727	34731	A197	62814	62806	A300	13501	13502	A403	66517	-
A94	13552	13536	A198	62823	93102	A301	66522	-	A404	22738	13504
A95	13502	13501	A199	13528	-	A302	66505	-	A405	66507	-
A96	22701	13501	A200	13530	-	A303	66513	-	A406	62819	22738
A97	13535	-	A201	71730	-	A304	66518	66533	A407	71712	71732
A98	22741	22720	A202	71737	71729	A305	66512	66533	A408	22749	22727
A99	44332	44425	A203	22738	-	A306	66512	66533	A409	26737	-
A100	44331	-	A204	13504	66515	A307	66517	-	A410	53206	53207
A101	71721	-	A205	13547	-	A308	62822	62824	A411	22738	-
A102	13535	-	A206	71712	-	A309	66517	62818	A412	62820	-
A103	22741	-	A207	53238	-	A310	62822	-	A413	44359	44432
A104	26701	71718									

Tabela A.16. Alunos e respectivos exames (instância 6).

CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2
A1	34731	44334	A104	71730	-	A207	22738	-	A309	62822	-
A2	26733	26741	A105	13536	71725	A208	62819	13504	A310	22738	62822
A3	34707	26727	A106	71725	-	A209	66505	-	A311	22738	-
A4	22723	-	A107	71721	-	A210	22738	62807	A312	66517	-
A5	26729	26732	A108	44341	22741	A211	22738	62825	A313	66517	-
A6	44332	44383	A109	44304	44331	A212	66513	-	A314	66518	66510
A7	13538	-	A110	13505	53222	A213	22738	-	A315	44376	66524
A8	22747	26707	A111	13505	53236	A214	66509	66510	A316	66524	66516
A9	34730	34731	A112	62805	-	A215	22738	44376	A317	62812	-
A10	44320	13547	A113	53215	53224	A216	62822	-	A318	62813	62806
A11	34739	34730	A114	53238	-	A217	66530	22738	A319	66522	66510
A12	34741	-	A115	66522	93102	A218	62820	62822	A320	22738	-
A13	13505	-	A116	66533	-	A219	66513	62831	A321	66531	-
A14	34706	-	A117	66505	66512	A220	62824	-	A322	62822	62808
A15	26725	26732	A118	62808	-	A221	13504	22738	A323	62822	-
A16	44331	-	A119	62827	62806	A222	13546	-	A324	66518	-
A17	44430	-	A120	44317	-	A223	44310	44319	A325	66521	-
A18	13528	34707	A121	71712	71738	A224	71710	71738	A326	66504	22738
A19	13546	-	A122	13538	-	A225	26730	26704	A327	62819	62822
A20	13528	-	A123	13547	-	A226	26732	26734	A328	66509	-
A21	26735	26734	A124	34741	-	A227	26732	-	A329	62813	-
A22	62803	-	A125	71723	-	A228	13538	-	A330	66533	-
A23	71719	71726	A126	71708	-	A229	34728	34739	A331	66510	66530
A24	26727	34707	A127	71710	22741	A230	34730	34731	A332	62813	62824
A25	13538	13528	A128	44409	-	A231	34730	-	A333	62822	-
A26	34759	34741	A129	53215	53246	A232	13504	22738	A334	62822	62819
A27	26703	13539	A130	62827	62819	A233	53232	53229	A335	44319	62813
A28	13540	-	A131	62822	-	A234	13503	53226	A336	66510	-
A29	26737	13528	A132	66533	-	A235	53227	53233	A337	44319	62824
A30	34726	26701	A133	62813	62819	A236	53219	-	A338	62812	-
A31	13540	13528	A134	66518	-	A237	53232	53228	A339	53206	-
A32	34741	34765	A135	13504	62820	A238	53238	53242	A340	66519	62816
A33	22741	34763	A136	66533	-	A239	71737	71729	A341	13504	66536
A34	71712	-	A137	13504	62820	A240	71737	71707	A342	66533	62807
A35	71721	71725	A138	44313	44314	A241	71730	-	A343	22738	66533
A36	71706	71729	A139	44320	44313	A242	22741	-	A344	66516	-
A37	26705	62806	A140	62820	-	A243	26717	26714	A345	22738	-
A38	13518	22722	A141	44314	-	A244	13502	26719	A346	66511	-
A39	22738	-	A142	26701	26744	A245	13501	-	A347	66511	-
A40	13528	425135	A143	13528	13538	A246	13504	22738	A348	66530	22738
A41	13535	-	A144	13538	13528	A247	26709	26719	A349	44304	44320
A42	13504	62806	A145	26730	-	A248	34731	34728	A350	62803	44318
A43	44359	-	A146	71730	71725	A249	34731	22741	A351	44320	44304
A44	34731	-	A147	26723	26737	A250	13512	-	A352	62803	44420
A45	44341	44365	A148	26731	26737	A251	22727	22749	A353	85104	-
A46	13547	-	A149	13528	26704	A252	44425	44383	A354	44420	-
A47	13559	-	A150	13528	13538	A253	44383	-	A355	44320	-
A48	15021	-	A151	13538	13540	A254	44425	-	A356	44420	44313
A49	13546	-	A152	26730	26737	A255	44313	44317	A357	44319	44313
A50	13547	-	A153	26727	26721	A256	71730	-	A358	44320	44420
A51	26737	-	A154	34763	34728	A257	71736	-	A359	44310	93102
A52	26734	-	A155	34763	34728	A258	53243	71739	A360	44317	44340
A53	34731	34763	A156	13505	-	A259	22741	-	A361	62803	62831
A54	22741	-	A157	53221	-	A260	34705	22741	A362	44311	-
A55	425135	-	A158	53208	-	A261	66524	44341	A363	44340	44420
A56	26724	26703	A159	13510	-	A262	71731	-	A364	44311	-
A57	15021	-	A160	71725	-	A263	71736	22741	A365	13528	26729
A58	26737	85104	A161	26716	-	A264	53243	-	A366	26730	-
A59	34730	34731	A162	22701	-	A265	22741	-	A367	34753	34750
A60	13548	-	A163	26706	22741	A266	71726	-	A368	71723	22741
A61	34741	22741	A164	22741	-	A267	71725	71731	A369	22741	71725
A62	62824	-	A165	13525	22749	A268	22741	-	A370	13501	-
A63	44340	85104	A166	22749	13524	A269	71723	71730	A371	34753	22741
A64	71731	66524	A167	71718	34705	A270	71731	-	A372	26707	22728
A65	71735	71708	A168	71739	71736	A271	34741	34762	A373	22724	22748
A66	34739	34734	A169	71725	-	A272	34725	-	A374	22741	71725
A67	13558	-	A170	71725	-	A273	34725	22741	A375	71723	-
A68	13552	13536	A171	71730	-	A274	34741	-	A376	66505	-
A69	13501	71720	A172	22741	-	A275	26733	26732	A377	71730	-
A70	22701	13501	A173	71721	71730	A276	62807	13504	A378	71736	-
A71	22741	-	A174	71725	71740	A277	13504	-	A379	71736	71730
A72	22720	-	A175	34741	-	A278	71731	71725	A380	34725	-
A73	13535	-	A176	34741	-	A279	26729	-	A381	34762	22741
A74	13536	-	A177	13550	34725	A280	71725	-	A382	22741	34750
A75	22741	-	A178	34741	34725	A281	13550	44348	A383	34725	-
A76	22741	-	A179	26727	26725	A282	53238	53241	A384	34751	26701
A77	71730	71723	A180	62814	62806	A283	13533	-	A385	22741	-
A78	13540	26727	A181	26735	-	A284	62813	62819	A386	34725	22741
A79	13558	-	A182	13530	-	A285	66518	-	A387	34750	22741
A80	22719	-	A183	71725	-	A286	53238	53241	A388	22741	-
A81	71735	-	A184	71723	71730	A287	44337	-	A389	26735	-
A82	13550	34725	A185	34705	-	A288	13501	13502	A390	22736	26733
A83	22741	-	A186	66533	-	A289	13512	-	A391	44383	44355
A84	34748	34725	A187	62819	22738	A290	66512	66533	A392	85109	62803
A85	66510	62807	A188	66530	66511	A291	62825	66521	A393	66515	-
A86	26729	-	A189	44304	-	A292	66518	66533	A394	66533	66510
A87	26725	-	A190	71712	-	A293	66522	66512	A395	26718	26715
A88	22701	13538	A191	71710	71708	A294	66509	-	A396	26732	13538
A89	26727	26735	A192	53232	53238	A295	66524	66513	A397	26729	26727
A90	13538	-	A193	71730	-	A296	62822	-	A398	71725	62834
A91	13540	-	A194	13548	-	A297	66518	-	A399	66511	-
A92	13538	26731	A195	71736	22741	A298	66513	-	A400	66531	-
A93	13528	-	A196	13547	13546	A299	62808	-	A401	34714	-
A94	13528	26729	A197	71735	71730	A300	62807	62806	A402	62803	44321
A95	22741	22712	A198	53221	-	A301	62822	-	A403	44304	44317
A96	22741	34725	A199	71730	-	A302	66510	66533	A404	44317	-
A97	34748	-	A200	13505	53211	A303	66510	-	A405	34750	22741
A98	71731	-	A201	22738	13504	A304	62807	62806	A406	22736	26737
A99	26707	13557	A202	66513	-	A305	62814	62827	A407	13504	22738
A100	26718	-	A203	22738	66505	A306	62827	-	A408	13505	-
A101	22751	-	A204	62805	62808	A307	62825	66513	A409	22741	71723
A102	71710	22741	A205	66523	-	A308	66510	66533	A410	66537	22738
A103	13536	71725	A206	66524	62822						

Tabela A.17. Alunos e respectivos exames (instância 7).

CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2	CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2	CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2	CD_ALUNO	CD_DISCIP_1	CD_DISCIP_2
A1	34731	-	A99	71725	-	A197	62824	62819	A295	66531	-
A2	26732	26741	A100	26731	26735	A198	66505	-	A296	22738	-
A3	26729	26727	A101	26729	22701	A199	62820	62819	A297	66509	62819
A4	22723	22728	A102	26730	-	A200	66524	-	A298	62808	-
A5	26727	-	A103	26735	13538	A201	62819	-	A299	62824	62823
A6	44332	13547	A104	26737	-	A202	62807	66509	A300	62822	-
A7	44311	44320	A105	26732	-	A203	62824	-	A301	66522	-
A8	13538	-	A106	44331	-	A204	62819	13504	A302	62824	62822
A9	22747	26707	A107	26735	26729	A205	66510	66505	A303	62822	62820
A10	34706	13528	A108	13552	-	A206	62807	-	A304	62813	22738
A11	13547	44331	A109	13549	22741	A207	13504	-	A305	66515	-
A12	22738	-	A110	34741	-	A208	13504	22738	A306	62822	-
A13	34739	34731	A111	22727	-	A209	66510	-	A307	62814	62822
A14	71730	-	A112	13536	71725	A210	62819	62813	A308	62807	62824
A15	34763	34741	A113	34731	34763	A211	22738	44376	A309	66524	-
A16	26725	-	A114	13513	13516	A212	66511	66536	A310	66536	-
A17	71729	-	A115	71731	-	A213	62814	-	A311	22738	-
A18	44331	13547	A116	13536	-	A214	66523	62822	A312	66531	-
A19	71735	71723	A117	22741	71730	A215	13504	-	A313	62806	-
A20	44430	13546	A118	13536	71730	A216	13504	22738	A314	66519	62822
A21	26727	-	A119	71718	13535	A217	44321	-	A315	66522	66518
A22	26727	-	A120	71725	-	A218	44311	44314	A316	66515	-
A23	13528	-	A121	13505	53220	A219	71732	22741	A317	62831	-
A24	26727	26735	A122	53216	-	A220	71738	-	A318	66533	-
A25	71719	-	A123	53216	-	A221	26703	26737	A319	66524	-
A26	13528	26737	A124	34708	53222	A222	26732	-	A320	66524	-
A27	26729	-	A125	62818	13504	A223	13528	26732	A321	22738	62821
A28	34763	26701	A126	13504	-	A224	26735	425135	A322	44337	62822
A29	13538	-	A127	13504	22738	A225	26735	26736	A323	66524	66511
A30	26737	13528	A128	44311	-	A226	26730	13538	A324	62824	62819
A31	34726	-	A129	13538	26729	A227	34730	34739	A325	22738	-
A32	34764	34741	A130	13505	53213	A228	34730	34731	A326	62824	-
A33	26723	26734	A131	13547	-	A229	34713	34739	A327	66513	-
A34	53214	-	A132	34753	34741	A230	13504	44376	A328	66524	66531
A35	34731	34763	A133	71708	-	A231	53228	-	A329	22738	-
A36	71712	-	A134	22741	-	A232	53222	-	A330	62822	-
A37	71706	71725	A135	44409	-	A233	53216	-	A331	62819	-
A38	34730	34739	A136	13505	-	A234	53241	-	A332	62806	-
A39	44355	-	A137	13504	66533	A235	53243	13503	A333	22738	13504
A40	62806	-	A138	66510	-	A236	53227	53241	A334	62806	66533
A41	13518	22722	A139	66513	-	A237	22735	71723	A335	66517	-
A42	62813	62822	A140	62822	-	A238	71729	26744	A336	53215	53206
A43	53242	-	A141	62822	-	A239	71729	71737	A337	66524	66513
A44	13528	-	A142	62820	-	A240	71731	-	A338	62825	62813
A45	13538	-	A143	44313	44304	A241	71720	-	A339	44319	-
A46	71725	13536	A144	62825	62820	A242	13502	-	A340	66510	66522
A47	26701	13535	A145	44304	-	A243	13502	13501	A341	66524	66516
A48	22741	-	A146	13528	-	A244	26718	-	A342	66530	66536
A49	66533	66512	A147	26729	-	A245	13550	-	A343	44317	44320
A50	34763	22741	A148	71736	71725	A246	34731	34728	A344	62803	44318
A51	22754	44365	A149	26737	26736	A247	22730	13513	A345	22754	44331
A52	34731	-	A150	34707	13528	A248	34710	44355	A346	62803	-
A53	44383	13547	A151	13528	-	A249	44331	-	A347	44304	-
A54	44409	71723	A152	26727	26735	A250	44332	44431	A348	44318	44321
A55	66503	66534	A153	34763	34728	A251	13547	53244	A349	44321	-
A56	13518	13559	A154	13505	-	A252	44431	13547	A350	62803	44320
A57	15021	-	A155	53215	53226	A253	34710	-	A351	44420	44317
A58	22724	-	A156	53227	53221	A254	71730	53238	A352	44317	44420
A59	13547	-	A157	71725	-	A255	71730	71725	A353	62803	-
A60	44321	22754	A158	26718	-	A256	71725	71731	A354	44317	44319
A61	26737	-	A159	22708	-	A257	71723	-	A355	44319	-
A62	13538	-	A160	22741	-	A258	71731	-	A356	44321	44318
A63	13502	-	A161	22749	13524	A259	71730	71736	A357	44316	-
A64	34763	-	A162	13547	44359	A260	71730	-	A358	44340	-
A65	22741	-	A163	44431	44332	A261	71725	53243	A359	44321	44320
A66	13538	-	A164	71723	34705	A262	13536	-	A360	26731	26733
A67	26724	26703	A165	71725	-	A263	22741	-	A361	26730	26731
A68	13527	13562	A166	71725	-	A264	71731	71736	A362	26703	26733
A69	15021	-	A167	71721	-	A265	71730	71723	A363	26731	-
A70	22741	-	A168	71730	-	A266	71725	-	A364	34759	22741
A71	34730	34731	A169	71702	22741	A267	34741	-	A365	34750	-
A72	34748	-	A170	71725	-	A268	34725	-	A366	71707	71729
A73	13548	34741	A171	34741	441114	A269	13504	-	A367	26716	26719
A74	13549	22741	A172	34741	-	A270	26734	26737	A368	22733	22749
A75	44376	-	A173	13550	34725	A271	26729	26727	A369	44376	66505
A76	62806	62813	A174	34741	34725	A272	22736	26729	A370	22735	-
A77	44376	-	A175	26727	-	A273	71731	-	A371	44341	71730
A78	66512	-	A176	62819	62806	A274	44334	44383	A372	22741	34750
A79	13533	71708	A177	71730	-	A275	34763	34731	A373	22741	-
A80	26730	-	A178	44311	44320	A276	71730	71731	A374	22741	34753
A81	66533	-	A179	13528	-	A277	62823	-	A375	22741	-
A82	22737	13505	A180	13530	-	A278	66533	66518	A376	22741	13550
A83	13558	-	A181	71725	-	A279	13504	22738	A377	34755	22741
A84	13552	13536	A182	71729	71735	A280	44319	62807	A378	34738	-
A85	13501	-	A183	71736	71730	A281	44311	62803	A379	22741	34762
A86	13535	-	A184	62822	66518	A282	26735	-	A380	22741	-
A87	22714	-	A185	62819	-	A283	71737	71729	A381	26733	-
A88	22741	22720	A186	66524	-	A284	22741	-	A382	44383	44355
A89	44332	44425	A187	13547	13546	A285	13501	13502	A383	26729	-
A90	13533	-	A188	34705	22741	A286	13513	13512	A384	22738	-
A91	62811	62822	A189	71710	71708	A287	22741	-	A385	44340	-
A92	71718	-	A190	71721	71735	A288	66512	66518	A386	44308	85104
A93	71725	22741	A191	34750	-	A289	66522	-	A387	85104	44420
A94	26737	-	A192	13538	-	A290	62823	62808	A388	28002	26729
A95	13502	-	A193	13505	53219	A291	66510	-	A389	71724	-
A96	71730	-	A194	62819	62813	A292	66519	-	A390	66537	-
A97	71735	13536	A195	22738	66533	A293	66530	-	A391	66521	-
A98	13504	22738	A196	62816	-	A294	66511	62822	A392	34740	34736

Tabela A.18. Alunos e respectivos exames (instância 8).

CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2
A1	34731	44334	A99	22738	-	A196	62813	62819	A293	22738	62822
A2	26732	26733	A100	22701	-	A197	62818	22738	A294	66521	-
A3	34706	26727	A101	71718	71720	A198	62822	66516	A295	22738	66511
A4	22728	-	A102	26730	-	A199	22738	66518	A296	44319	93102
A5	26727	-	A103	13538	-	A200	62813	-	A297	66522	66513
A6	44332	13547	A104	13538	-	A201	22738	62806	A298	66507	-
A7	44320	-	A105	13528	-	A202	13504	62824	A299	22738	-
A8	13557	22741	A106	13528	-	A203	62819	-	A300	66524	-
A9	13538	-	A107	44331	-	A204	66505	-	A301	62819	62806
A10	26707	-	A108	26729	-	A205	22738	-	A302	66513	-
A11	34711	34728	A109	13527	22723	A206	22738	62825	A303	66524	-
A12	34706	13528	A110	22724	22727	A207	66524	-	A304	13504	-
A13	13547	-	A111	71725	-	A208	13504	22738	A305	22738	-
A14	22738	-	A112	13518	26707	A209	62819	-	A306	13504	-
A15	34739	34730	A113	22723	22749	A210	44376	-	A307	62824	62822
A16	34763	34741	A114	22751	-	A211	44319	62824	A308	22738	-
A17	53215	13505	A115	22741	-	A212	13504	-	A309	66518	66533
A18	26725	26734	A116	13536	71730	A213	66513	-	A310	62824	62807
A19	71729	71707	A117	71723	-	A214	44320	62803	A311	62824	-
A20	13547	-	A118	71718	34705	A215	22754	-	A312	62822	-
A21	71737	71726	A119	71735	-	A216	44311	-	A313	62813	-
A22	44386	44430	A120	53216	-	A217	71714	71712	A314	62812	-
A23	26727	26725	A121	53216	-	A218	26733	-	A315	66513	-
A24	26734	34707	A122	44337	-	A219	26735	-	A316	62822	-
A25	13546	-	A123	13503	-	A220	26735	-	A317	62819	62822
A26	26727	26735	A124	62819	-	A221	26737	26735	A318	66512	62807
A27	44340	44318	A125	62811	62819	A222	34730	34739	A319	62813	62823
A28	26735	26724	A126	62824	62822	A223	34730	34731	A320	66507	-
A29	26719	26715	A127	44420	-	A224	34730	34739	A321	62813	66509
A30	13540	-	A128	26731	26733	A225	53242	53236	A322	66536	-
A31	26727	34707	A129	13538	26729	A226	13504	44376	A323	66511	-
A32	13528	34706	A130	13547	-	A227	53226	53228	A324	66511	-
A33	13528	26730	A131	44430	44431	A228	53227	53224	A325	62806	62819
A34	26729	-	A132	34750	34741	A229	53237	53240	A326	66524	-
A35	26735	-	A133	71708	-	A230	53241	53237	A327	44304	44310
A36	34763	-	A134	71710	22741	A231	53238	53241	A328	44420	44321
A37	13538	-	A135	53215	-	A232	53238	53239	A329	44321	62803
A38	26737	13528	A136	13505	-	A233	71731	-	A330	44304	44318
A39	34726	26701	A137	53242	53237	A234	71727	-	A331	62803	-
A40	71721	71739	A138	13513	22749	A235	26708	26743	A332	62803	62831
A41	34708	-	A139	66512	-	A236	71707	-	A333	44420	-
A42	26723	26737	A140	66512	66533	A237	71731	-	A334	62803	-
A43	53245	13505	A141	62822	-	A238	13502	-	A335	85104	-
A44	13536	-	A142	66533	-	A239	13550	-	A336	62831	44311
A45	44355	-	A143	13504	62820	A240	13524	-	A337	44420	44316
A46	13504	26705	A144	44304	-	A241	22749	-	A338	44316	-
A47	22725	22722	A145	44319	-	A242	22728	13516	A339	44320	-
A48	22738	-	A146	66521	66520	A243	44425	-	A340	62803	44340
A49	66516	-	A147	44311	-	A244	71723	71725	A341	44311	-
A50	13535	71736	A148	44331	62803	A245	71730	53238	A342	26735	-
A51	26701	13535	A149	22741	-	A246	22741	71725	A343	34705	71723
A52	22741	-	A150	13528	13538	A247	71731	-	A344	13527	22723
A53	13504	-	A151	26730	26735	A248	71730	-	A345	71724	71726
A54	71735	71708	A152	26730	26729	A249	22741	-	A346	71737	-
A55	34731	-	A153	26723	26735	A250	22741	71731	A347	26716	26719
A56	44365	44425	A154	13528	26704	A251	71725	-	A348	26718	13501
A57	13547	-	A155	13528	13540	A252	71731	-	A349	22747	13536
A58	22741	71723	A156	26734	13528	A253	71723	71736	A350	22727	22751
A59	62807	-	A157	425135	-	A254	71730	-	A351	22728	26707
A60	13518	13559	A158	34763	34728	A255	13550	22741	A352	22741	66524
A61	15021	-	A159	34739	34728	A256	34725	-	A353	71723	22741
A62	22724	-	A160	53221	-	A257	26729	-	A354	22735	71730
A63	26737	-	A161	53208	53216	A258	62824	62823	A355	71730	71723
A64	13540	13538	A162	13524	-	A259	13504	-	A356	71723	71735
A65	13540	34706	A163	44331	-	A260	13538	13528	A357	26701	71735
A66	13502	-	A164	13536	-	A261	26729	-	A358	71730	71731
A67	34731	34763	A165	71730	71725	A262	13528	26733	A359	34750	22741
A68	71731	-	A166	71725	-	A263	26729	-	A360	34748	34741
A69	13535	71712	A167	71740	-	A264	44334	44425	A361	34750	-
A70	22741	-	A168	71730	-	A265	34734	13550	A362	22741	-
A71	15021	-	A169	34741	-	A266	71730	-	A363	34738	22741
A72	34730	34731	A170	34741	-	A267	62822	66512	A364	26701	-
A73	13548	34741	A171	34741	-	A268	13504	66513	A365	34738	34750
A74	34741	-	A172	26727	-	A269	53233	53241	A366	34725	34738
A75	62819	44376	A173	62819	-	A270	53240	53238	A367	44376	22741
A76	66521	22738	A174	13528	26735	A271	62820	-	A368	22741	34744
A77	22741	71736	A175	13530	-	A272	22741	-	A369	26733	26734
A78	34705	-	A176	71730	71710	A273	13501	13502	A370	26727	26733
A79	26723	26735	A177	71736	71730	A274	26715	13501	A371	44425	-
A80	13540	-	A178	71737	71718	A275	22723	-	A372	44355	-
A81	26735	-	A179	71718	-	A276	66533	66510	A373	26703	13538
A82	13552	13536	A180	34705	-	A277	66505	-	A374	22741	-
A83	13502	13501	A181	44304	-	A278	66510	62822	A375	26733	26737
A84	22701	13501	A182	22741	-	A279	66518	-	A376	71725	-
A85	82611	-	A183	53238	-	A280	66511	-	A377	13504	-
A86	22720	-	A184	71730	71721	A281	66518	-	A378	22738	-
A87	44331	-	A185	13548	34750	A282	22738	-	A379	62813	62819
A88	71718	-	A186	13535	22711	A283	13504	-	A380	62820	22738
A89	71730	-	A187	71736	-	A284	66511	-	A381	44340	62803
A90	22741	-	A188	13538	-	A285	66510	66522	A382	34750	22741
A91	71725	22741	A189	13547	-	A286	66522	-	A383	13504	44376
A92	26737	26727	A190	22741	71712	A287	44319	62824	A384	71725	71737
A93	13558	-	A191	53224	53226	A288	66522	66510	A385	62825	22738
A94	13502	-	A192	13505	53215	A289	62814	66506	A386	62820	66520
A95	71730	-	A193	44318	-	A290	44319	62813	A387	13504	62827
A96	71730	-	A194	22738	62828	A291	66524	-	A388	62822	62811
A97	44334	44332	A195	66524	-	A292	22738	62807	A389	34740	34736
A98	13550	34748									

Tabela A.19. Alunos e respectivos exames (instância 9).

CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2
A1	34731	44433	A100	34731	34763	A199	22738	66505	A297	66522	66510
A2	26734	-	A101	22723	22749	A200	62808	-	A298	66505	62825
A3	34735	26729	A102	13536	71725	A201	62819	-	A299	66515	-
A4	13562	34701	A103	22741	-	A202	66523	-	A300	66533	66512
A5	26734	-	A104	13536	71730	A203	22738	-	A301	66518	66512
A6	26729	26737	A105	71721	71723	A204	62813	22738	A302	66512	-
A7	26729	26732	A106	71739	13536	A205	62819	13504	A303	66513	-
A8	44320	-	A107	71725	-	A206	62819	-	A304	66518	62824
A9	13538	-	A108	71725	71735	A207	66510	66533	A305	66519	-
A10	22747	26707	A109	44331	-	A208	66518	62822	A306	62818	-
A11	34706	13528	A110	13505	-	A209	22738	-	A307	66524	-
A12	34727	-	A111	53224	-	A210	13504	-	A308	22738	62821
A13	44331	-	A112	53240	13505	A211	13504	22738	A309	22738	13504
A14	13504	62823	A113	53216	53215	A212	66518	-	A310	13504	62818
A15	26701	13533	A114	13505	-	A213	66533	-	A311	66513	62825
A16	34741	-	A115	53228	-	A214	62819	62813	A312	62827	66512
A17	44332	-	A116	22737	44301	A215	44376	62812	A313	66523	-
A18	34706	-	A117	66533	-	A216	66530	62820	A314	62822	-
A19	26727	26725	A118	62819	22738	A217	62807	66522	A315	66506	62807
A20	71729	-	A119	66521	-	A218	66511	-	A316	66516	-
A21	44331	13547	A120	44311	44313	A219	62822	-	A317	66513	-
A22	44386	44430	A121	66517	-	A220	13504	-	A318	66511	-
A23	26734	-	A122	13528	26729	A221	66513	-	A319	62824	62813
A24	13546	-	A123	13505	-	A222	13504	66517	A320	66518	66512
A25	26727	-	A124	13547	-	A223	44319	-	A321	44376	66536
A26	26730	26724	A125	13513	22749	A224	44310	44304	A322	66518	22738
A27	13540	26734	A126	34750	34741	A225	22741	71712	A323	62808	-
A28	26729	26734	A127	71708	71735	A226	71712	71738	A324	62813	-
A29	13538	26730	A128	71710	-	A227	71713	34705	A325	66524	62821
A30	13540	-	A129	44409	-	A228	13538	-	A326	62813	22738
A31	26737	-	A130	53215	-	A229	26733	13540	A327	62822	66533
A32	13527	13510	A131	53237	53241	A230	425148	-	A328	66513	-
A33	34726	-	A132	53238	53241	A231	26734	-	A329	62824	62807
A34	34741	-	A133	62814	62827	A232	34730	34731	A330	62808	62824
A35	26723	-	A134	62819	-	A233	13504	-	A331	66512	62807
A36	53245	13505	A135	13504	-	A234	53224	53226	A332	22738	-
A37	34731	-	A136	66510	66533	A235	53226	-	A333	66510	66533
A38	71706	71729	A137	66533	-	A236	53222	-	A334	66518	66533
A39	44355	-	A138	66530	-	A237	53241	53240	A335	62813	-
A40	44376	66533	A139	62808	-	A238	53224	-	A336	62824	-
A41	22738	-	A140	66530	-	A239	53216	-	A337	66534	66512
A42	13528	-	A141	66518	-	A240	53215	53216	A338	62806	62819
A43	13538	-	A142	66510	-	A241	53238	53237	A339	66517	-
A44	22741	-	A143	13504	-	A242	53238	53239	A340	66519	-
A45	66510	-	A144	62820	-	A243	53243	53227	A341	22738	62822
A46	34763	34759	A145	62803	-	A244	53238	53239	A342	66513	-
A47	22754	44365	A146	44420	-	A245	71737	22735	A343	66530	-
A48	13547	-	A147	13528	13538	A246	71731	-	A344	62811	62827
A49	15021	-	A148	26730	-	A247	71729	26743	A345	62816	-
A50	26737	-	A149	26732	26731	A248	71731	71730	A346	44376	-
A51	13540	34706	A150	13528	26704	A249	26718	-	A347	44304	44320
A52	13502	-	A151	13538	13528	A250	26714	-	A348	44340	44420
A53	71708	-	A152	26735	-	A251	13502	-	A349	44321	-
A54	13528	-	A153	13538	-	A252	13502	13501	A350	44320	-
A55	13538	13540	A154	34763	-	A253	44376	-	A351	44319	-
A56	15021	-	A155	53216	-	A254	34763	34728	A352	44340	44317
A57	26741	-	A156	53226	-	A255	13512	-	A353	44420	44317
A58	13561	13557	A157	53220	53241	A256	22749	13557	A354	44317	44420
A59	34730	34731	A158	22701	-	A257	22728	13513	A355	44340	-
A60	62819	44376	A159	26719	13502	A258	22727	-	A356	44340	44318
A61	62824	-	A160	22708	13552	A259	34710	-	A357	44319	-
A62	44320	-	A161	22749	22730	A260	44355	-	A358	44311	-
A63	34705	-	A162	44331	44359	A261	71725	22741	A359	62803	-
A64	71708	-	A163	71723	13536	A262	71730	-	A360	44318	-
A65	26723	13538	A164	26701	-	A263	71730	71725	A361	26734	26733
A66	26735	-	A165	71725	-	A264	71739	-	A362	26735	26737
A67	53212	13505	A166	22741	-	A265	22741	71730	A363	26734	26735
A68	13558	-	A167	71721	-	A266	71725	22741	A364	26723	26726
A69	13536	-	A168	71725	71730	A267	22741	71736	A365	22741	-
A70	13501	-	A169	22741	71740	A268	71736	71735	A366	13516	13527
A71	22701	13501	A170	71721	-	A269	71731	-	A367	71723	22741
A72	13535	22741	A171	34741	-	A270	71726	-	A368	26718	13501
A73	22741	-	A172	34741	34725	A271	13536	-	A369	26707	22749
A74	44332	66521	A173	26727	26725	A272	22741	71736	A370	71735	71736
A75	71731	71718	A174	13538	26723	A273	22741	-	A371	66511	66517
A76	13536	13535	A175	53238	71736	A274	71731	-	A372	71736	22741
A77	71723	-	A176	62819	62807	A275	71725	-	A373	34725	22741
A78	22741	-	A177	71730	71735	A276	71736	-	A374	34754	-
A79	71718	-	A178	44311	-	A277	34748	34741	A375	34748	-
A80	71725	22741	A179	62813	62819	A278	26701	-	A376	22741	-
A81	13536	53227	A180	13528	26735	A279	34741	-	A377	34750	-
A82	13558	-	A181	13530	-	A280	34725	-	A378	22741	13550
A83	22728	-	A182	71723	71718	A281	34725	22741	A379	34762	26701
A84	13502	-	A183	71713	-	A282	13548	13534	A380	22741	-
A85	22741	-	A184	66524	66530	A283	62806	62813	A381	34725	13550
A86	22738	-	A185	44304	13546	A284	13538	-	A382	22741	34738
A87	71725	-	A186	22741	71712	A285	26729	13528	A383	26735	-
A88	26729	-	A187	34705	71708	A286	44425	44383	A384	44383	44355
A89	71718	-	A188	53232	-	A287	53239	-	A385	44320	44340
A90	26730	26735	A189	13548	-	A288	34731	-	A386	22738	66533
A91	26730	-	A190	53239	53237	A289	62812	-	A387	71731	62834
A92	26737	26732	A191	34750	-	A290	26735	-	A388	13504	-
A93	13538	-	A192	13547	-	A291	71731	-	A389	44376	13504
A94	13528	13538	A193	44321	-	A292	13501	13502	A390	62824	62819
A95	26737	-	A194	62827	-	A293	22723	13524	A391	62803	44310
A96	22733	13552	A195	62812	13504	A294	71725	13536	A392	34754	22741
A97	34741	22741	A196	66521	13504	A295	22741	-	A393	13504	-
A98	34748	-	A197	66509	-	A296	71738	71705	A394	62825	22738
A99	13527	22723	A198	62822	66518						

Tabela A.20. Alunos e respectivos exames (instância 10).

CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2	CD.ALUNO	CD.DISCIP.1	CD.DISCIP.2
A1	13542	-	A105	26733	26737	A209	66513	-	A313	66530	62812
A2	26732	-	A106	13538	13528	A210	62813	-	A314	66509	-
A3	34707	-	A107	13538	26735	A211	62813	-	A315	66510	-
A4	22725	-	A108	13528	-	A212	62819	-	A316	22738	-
A5	22741	-	A109	26731	26732	A213	66533	66522	A317	66519	-
A6	26734	26727	A110	22714	-	A214	66509	66518	A318	66524	-
A7	44376	13504	A111	22741	-	A215	13504	-	A319	44376	62822
A8	26729	-	A112	22724	22728	A216	62820	66536	A320	62824	-
A9	34734	-	A113	13536	71725	A217	66511	66523	A321	22738	62808
A10	44311	-	A114	13518	-	A218	22738	-	A322	44376	-
A11	13538	-	A115	13524	-	A219	62820	66536	A323	62807	66533
A12	22747	22745	A116	22751	-	A220	13504	-	A324	66524	-
A13	44420	34728	A117	22741	-	A221	62813	-	A325	22738	-
A14	26737	26727	A118	13536	-	A222	13504	-	A326	66505	-
A15	34706	13528	A119	22741	-	A223	44316	44310	A327	44319	62822
A16	34731	-	A120	13536	-	A224	44319	85104	A328	62819	66518
A17	34763	34741	A121	71725	-	A225	44420	62803	A329	62813	-
A18	13505	-	A122	71721	-	A226	71738	22741	A330	62824	62823
A19	44355	-	A123	13535	-	A227	71710	71738	A331	66518	13504
A20	26727	26725	A124	71725	-	A228	26730	26741	A332	66518	66533
A21	71729	-	A125	53215	53222	A229	13528	13540	A333	62813	62807
A22	44331	13547	A126	53226	-	A230	26734	26735	A334	53227	53203
A23	44386	44430	A127	53215	-	A231	26735	-	A335	66516	66519
A24	26727	26704	A128	44301	-	A232	26729	-	A336	13504	93102
A25	13528	34707	A129	66533	62807	A233	26734	26703	A337	66509	-
A26	13546	13547	A130	62819	-	A234	34728	34730	A338	62824	62823
A27	26737	13528	A131	62813	-	A235	34731	-	A339	66517	66524
A28	26727	-	A132	62819	62822	A236	22738	44376	A340	62822	66512
A29	13540	-	A133	13504	66521	A237	53247	-	A341	66516	-
A30	26727	-	A134	66524	-	A238	53227	-	A342	62819	66509
A31	34706	-	A135	13505	-	A239	53216	53226	A343	66511	-
A32	13538	13528	A136	71723	71730	A240	53215	53226	A344	66524	-
A33	34741	34763	A137	71710	71723	A241	53226	53228	A345	62813	62822
A34	13538	13540	A138	53215	53226	A242	53242	53239	A346	44304	-
A35	53204	53228	A139	13505	22737	A243	53238	53239	A347	44420	-
A36	34726	26701	A140	53238	-	A244	53239	53241	A348	44304	44319
A37	26723	26734	A141	22738	-	A245	53214	53227	A349	44321	44304
A38	53245	13505	A142	62808	-	A246	26702	26743	A350	44340	44321
A39	34763	-	A143	62819	-	A247	71729	-	A351	44340	-
A40	71721	71723	A144	66531	-	A248	71730	-	A352	62803	44316
A41	44355	-	A145	66510	-	A249	13502	-	A353	44316	62831
A42	26705	62806	A146	44304	-	A250	13504	22738	A354	44316	-
A43	22738	-	A147	62820	66521	A251	22749	-	A355	44317	-
A44	13528	-	A148	26743	26701	A252	44373	44383	A356	44317	85104
A45	13538	-	A149	44331	-	A253	44304	44310	A357	62803	44316
A46	22728	-	A150	22741	-	A254	44359	13547	A358	44317	-
A47	71730	13536	A151	13528	-	A255	44425	-	A359	62831	-
A48	13536	-	A152	26729	-	A256	71730	-	A360	85104	44340
A49	66518	66510	A153	26723	26737	A257	71731	71740	A361	44420	44316
A50	26733	26735	A154	26732	-	A258	71725	-	A362	44420	44318
A51	34731	-	A155	26725	13528	A259	71731	-	A363	44316	-
A52	44425	34710	A156	13538	-	A260	71740	71730	A364	44314	-
A53	13547	13546	A157	13528	-	A261	71731	-	A365	26732	-
A54	44409	-	A158	26735	-	A262	71736	22741	A366	26729	26736
A55	13518	-	A159	13538	425135	A263	71725	-	A367	22741	-
A56	15021	-	A160	34763	-	A264	22741	-	A368	66534	-
A57	22701	-	A161	44301	13505	A265	71723	71730	A369	13527	-
A58	13540	-	A162	53226	-	A266	71730	71723	A370	22741	71707
A59	13502	-	A163	53208	53216	A267	71730	-	A371	71729	-
A60	13528	26729	A164	53215	-	A268	34748	34741	A372	26718	26716
A61	22741	34763	A165	53216	13505	A269	13550	-	A373	13501	26718
A62	71731	-	A166	22741	-	A270	34741	-	A374	26719	13501
A63	13536	71712	A167	22714	-	A271	34725	-	A375	34725	34744
A64	22741	-	A168	26718	26719	A272	13534	-	A376	13518	13512
A65	13538	-	A169	22701	-	A273	22741	-	A377	66505	-
A66	15021	-	A170	22714	13536	A274	26727	13528	A378	71730	22741
A67	22741	-	A171	13524	-	A275	26732	26729	A379	71723	71736
A68	62813	13504	A172	44359	44334	A276	13504	-	A380	71736	71730
A69	13530	13527	A173	71735	71718	A277	26734	-	A381	71740	44341
A70	34730	34731	A174	71730	71725	A278	26729	13538	A382	34741	-
A71	13548	34741	A175	71730	22741	A279	22701	26729	A383	34762	22741
A72	34750	13550	A176	71730	-	A280	26730	26729	A384	22741	-
A73	44376	-	A177	71702	-	A281	71735	13535	A385	34750	-
A74	62806	62813	A178	71730	-	A282	22743	44339	A386	22741	-
A75	44340	-	A179	34741	441114	A283	53242	53237	A387	22741	-
A76	71712	71708	A180	34762	-	A284	71731	13533	A388	13550	-
A77	13538	26732	A181	26725	-	A285	26735	-	A389	22741	34741
A78	13538	13528	A182	71730	22741	A286	53233	53238	A390	34762	22741
A79	26735	-	A183	62819	-	A287	71731	22741	A391	34751	-
A80	26735	26730	A184	44311	-	A288	13501	13502	A392	34750	22741
A81	13552	-	A185	13528	-	A289	13513	13524	A393	34753	34755
A82	13502	13501	A186	13530	-	A290	66510	66518	A394	26733	-
A83	13501	-	A187	71725	71736	A291	66510	66533	A395	22736	-
A84	13535	22741	A188	71736	71718	A292	66533	-	A396	44365	-
A85	22741	-	A189	34705	71713	A293	66533	66512	A397	62803	44317
A86	22727	13557	A190	66518	-	A294	62813	62807	A398	26703	13538
A87	71718	53243	A191	13504	-	A295	66531	-	A399	22701	13540
A88	13536	-	A192	44304	13546	A296	62818	-	A400	13528	-
A89	22741	-	A193	71712	-	A297	66511	-	A401	34741	34726
A90	26701	71723	A194	53238	-	A298	62812	-	A402	66536	-
A91	71725	22741	A195	53215	53242	A299	34714	62820	A403	62811	-
A92	22719	-	A196	22741	22735	A300	66521	62819	A404	62819	22738
A93	13502	-	A197	22741	34725	A301	13504	-	A405	22738	-
A94	71730	71731	A198	71730	22741	A302	62819	62813	A406	85109	44420
A95	13550	34750	A199	13538	-	A303	66516	66530	A407	44319	-
A96	22741	-	A200	13547	34710	A304	66533	62807	A408	71712	71710
A97	34741	13550	A201	71730	71721	A305	62824	62819	A409	71725	-
A98	66510	-	A202	71730	-	A306	62825	62812	A410	22736	26730
A99	13528	13538	A203	62803	-	A307	62822	-	A411	13504	-
A100	71725	-	A204	62819	22738	A308	66518	62807	A412	71725	71727
A101	71719	-	A205	62813	-	A309	66531	-	A413	22738	-
A102	26727	-	A206	13504	62827	A310	22738	-	A414	62813	-
A103	26729	26725	A207	66505	-	A311	66507	66531	A415	66533	66520
A104	13538	-	A208	62805	-	A312	22738	66524	A416	34740	34736