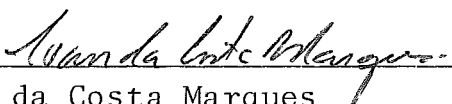


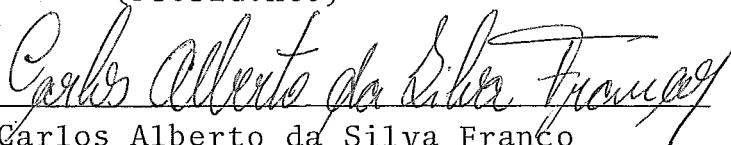
ALGORITMO DE BACKTRACK E O PROBLEMA DE HORÁRIO

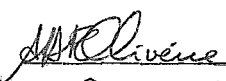
Wojciech Roman Drabik

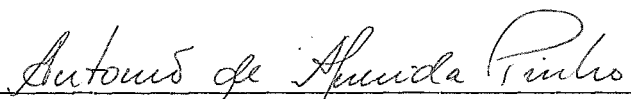
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.) EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

  
\_\_\_\_\_  
Ivan da Costa Marques  
(Presidente)

  
\_\_\_\_\_  
Carlos Alberto da Silva Franco

  
\_\_\_\_\_  
Antônio Alberto Fernandes de Oliveira

  
\_\_\_\_\_  
Antônio de Almeida Pinho

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 1984

DRABIK, WOJCIECH ROMAN

Algoritmo de Backtrack e o Problema de Horário (Rio de Janeiro) 1984.

VIII , 138 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1984).

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Algoritmo de Backtrack, Problema de Horário COPPE/UFRJ

II. Título (série).

Para a minha família e, em especial, Naly, Gustavo e Fernando.

AGRADECIMENTOS

A Deus, pela concretização deste trabalho e por termos muito a agradecer a tantos que nos distinguiram com sua amizade, orientação, apoio, compreensão, crítica, conselho e incentivo durante a realização desta pesquisa.

Ao Prof. Ivan da Costa Marques, pela sugestão do tema e pela orientação do desenvolvimento deste trabalho.

Aos Profs. Jayme Luiz Szwarcfiter e Carlos Alberto da Silva Franco, pela co-orientação e interface com a COPPE.

Ao Prof. João Lizardo R.H. de Araújo por alguns questionamentos e discussões sobre certos aspectos do presente trabalho.

Aos Profs. José Lucas M. Rangel Netto, Jano Moreira de Souza, Paulo Roberto Oliveira, Valter de Senna e Estevam Gilberto de Simone, pelo apoio e incentivo durante a realização dos cursos da COPPE.

Ao Eng. Geraldo Martins Tavares, pela objetividade das críticas e conselhos e pelo interesse sobre o andamento da pesquisa.

Aos Eng. Raimundo Carneiro Santiago e Jäder Costa Soares, pelo muito que com eles aprendemos.

Aos Profs. Denis França Leite, Guilherme Chagas Rodrigues e Ysmar Vianna e Silva Filho, pela orientação nas nossas primeiras pesquisas em Computação.

A Denise Schwartz, pelo excelente trabalho de edição e datilografia desta Tese, realizado em tempo recorde.

À Maria de Lourdes F.C. da Rocha, Selma Regina M. Martins e Solange Gonçalves de Souza, bibliotecárias do NCE/UFRJ e do CT/UFRJ, pela eficiente, inestimável e indispensável colaboração bibliográfica.

Aos amigos do Instituto de Matemática e da COPPE, pelo interesse, apoio e compreensão.



Resumo da Tese Apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ALGORITMO DE BACKTRACK E O PROBLEMA DE HORÁRIO

Wojciech Roman Drabik

Setembro de 1984

Orientador: Ivan da Costa Marques

Programa: Engenharia de Sistemas e Computação

Esta Tese examina a solução do problema de horário de aulas de Universidades por meio de um algoritmo de backtrack com verificação antecipada e reordenação da árvore de busca. Obtem-se alguns resultados teóricos referentes à redução do espaço de busca por reordenação e estabelece-se um invariante relacionando os efeitos da preclusão e da verificação antecipada. Estes resultados analíticos permitem explicar porque funcionam certas heurísticas comumente usadas no contexto do problema de horário. São descritos certos detalhes de implementação de um programa protótipo e apresentam-se alguns resultados preliminares.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

BACKTRACK ALGORITHM AND TIMETABLE PROBLEM

Wojciech Roman Drabik

September, 1984

Chairman: Ivan da Costa Marques

Department: Engenharia de Sistemas e Computação

In this Dissertation we explore the solution of the University timetabling problem by computer, using a backtrack algorithm with forward checking and search tree rearrangement. Some theoretical results are derived, concerning the reduction of the search space by rearrangement and an invariant is established relating the effects of preclusion and forward checking for the timetabling problem. These analytical results explain why some heuristics used with timetabling problems work. A computer program was developed and some implementation details and preliminary results are reported.

ÍNDICE

	Pág.
CAPÍTULO I - <u>O PROBLEMA DE HORÁRIO: APRESENTAÇÃO</u> <u>E REVISÃO DA LITERATURA</u> .....	1
I.1. INTRODUÇÃO .....	1
I.2. OS DIVERSOS PROBLEMAS DE HORÁRIO E MÉTODOS DE SOLUÇÃO .....	2
I.2.1. Uma Descrição Informal .....	2
I.2.2. Os Diversos Métodos de Solução .....	4
I.3. OS MÉTODOS HEURÍSTICOS .....	5
I.3.1. Introdução .....	5
I.3.2. O Algoritmo Heurístico de Brittan e Farley ....	6
I.3.3. Um Algoritmo que Melhora o Horário por Estágios .....	7
I.4. A ABORDAGEM DE GOTLIEB E CSIMA .....	9
I.4.1. Apresentação .....	9
I.4.2. Histórico e Evolução .....	12
I.4.3. A Utilização do Método de Gotlieb e Csimá ....	16
I.4.4. Os Aspectos Formais .....	21
I.5. UMA APLICAÇÃO DO ALGORITMO DE BACKTRACK .....	30
I.6. O PROBLEMA DE HORÁRIO E COLORAÇÃO DE GRAFOS ...	31
I.7. A PROGRAMAÇÃO LINEAR E O PROBLEMA DE HORÁRIO ..	32
CAPÍTULO II - <u>A NATUREZA DO PROBLEMA DE HORÁRIO E DO</u> <u>ALGORITMO DE BACKTRACK</u> .....	34
II.1. INTRODUÇÃO .....	34
II.2. PROBLEMAS E LINGUAGENS FORMAIS .....	36

	Pág.
II.3. A COMPLEXIDADE DE ALGORITMOS .....	39
II.4. PROBLEMAS NP-COMPLETO .....	42
II.5. A COMPLEXIDADE DO PROBLEMA DE HORÁRIO .....	45
II.6. O ALGORITMO DE BACKTRACK .....	47
II.6.1. Descrição .....	47
II.6.2. O Desempenho dos Algoritmos de Backtrack .....	51
II.7. ALGORITMOS DE APROXIMAÇÃO .....	57
II.8. COLORAÇÃO DE GRAFOS: MÍNIMA E APROXIMADA .....	60
<b>CAPÍTULO III - <u>UM ALGORITMO PARA O PROBLEMA</u></b>	
<b><u>DE HORÁRIO</u></b> .....	62
III.1. INTRODUÇÃO .....	62
III.2. DEFINIÇÕES PRELIMINARES .....	63
III.2.1. Definições .....	63
III.2.2. Alguns Comentários .....	66
III.2.3. A questão das Salas de Aula .....	68
III.3. A ESTRUTURA DO MODELO ADOTADO .....	69
III.3.1. O Modelo e as Restrições Intrínsecas .....	69
III.3.2. Os Aspectos Qualitativos .....	71
III.3.3. Uma Formulação do Problema de Horário .....	74
III.3.4. Grafos e Relações de Conflito .....	76
III.4. O MÉTODO DE SOLUÇÃO: ESCOLHA E DESCRIÇÃO .....	80
III.4.1. Os Métodos Disponíveis .....	80
III.4.2. A Escolha do Método de Solução .....	82
III.4.3. Uma Breve Descrição do Algoritmo Selecionado .....	84

III.5.	O ALGORITMO E O ESPAÇO DE ESTADOS .....	87
III.5.1.	Funções Intermediárias e Soluções .....	87
III.5.2.	O Espaço de Estados .....	92
III.5.3.	A Árvore do Espaço de Estados .....	96
III.5.4.	A Reordenação das Turmas .....	98
III.6.	A CARDINALIDADE DO ESPAÇO DE ESTADOS .....	99
III.6.1.	Introdução .....	99
III.6.2.	As Funções $q(\psi, k)$ e $r(\psi, k)$ .....	100
III.6.3.	Uma Expressão para a Cardinalidade do espaço de estados .....	105
III.7.	A REDUÇÃO DA CARDINALIDADE DO ESPAÇO DE ESTADOS .....	109
III.8.	ALGUNS ASPECTOS HEURÍSTICOS .....	119
CAPÍTULO IV - <u>O PROGRAMA PROTÓTIPO</u> .....		123
IV.1.	DESCRIÇÃO GERAL .....	123
IV.2.	ALGUNS DETALHES DE IMPLEMENTAÇÃO .....	125
IV.3.	LIMITAÇÕES .....	127
IV.4.	UMA QUESTÃO PARA ANÁLISE .....	127
IV.5.	RESULTADOS .....	128
CAPÍTULO V - <u>CONCLUSÕES E SUGESTÕES PARA PESQUISA</u> .....		130
V.1.	CONCLUSÕES SOBRE A PARTE TEÓRICA .....	130
V.2.	CONCLUSÕES SOBRE A IMPLEMENTAÇÃO .....	132
V.3.	SUGESTÕES PARA PESQUISA POSTERIOR .....	132
BIBLIOGRAFIA .....		134

CAPÍTULO I - O PROBLEMA DE HORÁRIO: APRESENTAÇÃO  
E REVISÃO DA LITERATURA

I.1. INTRODUÇÃO

A presente pesquisa foi desenvolvida visando formular um algoritmo para a construção de horários de aula de Universidades, por computador, e implementar um programa protótipo.

Neste primeiro capítulo apresentamos o problema de horário, descrevemos alguns problemas afins e relatamos o resultado da revisão da literatura sobre a construção de horários por computador.

No capítulo II introduzimos os principais conceitos necessários para uma caracterização da natureza do problema de horário e apresentamos o principal resultado teórico: o problema de horário é NP-completo, estabelecido por EVEN, ITAI e SHAMIR (16) em fins de 1976. Descrevemos o algoritmo geral de backtrack e os resultados de pesquisas recentes sobre métodos de torná-lo mais eficiente. Finalmente, introduzimos os algoritmos de aproximação e a questão da coloração aproximada de grafos.

No capítulo III formulamos o problema de horário para solução por um algoritmo de backtrack, discutimos a escolha deste método de solução e estabelecemos alguns resultados teóricos sobre a redução da cardinalidade do espaço de estados e, com base nestes resultados, examinamos alguns aspectos heurísticos.

O capítulo IV apresenta uma descrição geral da implementação do algoritmo por um programa protótipo, desenvolvido

com base nos resultados do capítulo III.

Finalmente, o capítulo V encerra este trabalho com conclusões e sugestões para pesquisa posterior.

Os grafos constituem a estrutura abstrata mais utilizada no presente trabalho, tanto no tratamento do problema de horário como do algoritmo de backtrack. Adotamos a terminologia usada pelo Prof. J.L. SZWARCFITER na sua obra recente, GRAFOS E ALGORITMOS COMPUTACIONAIS.

## I.2. OS DIVERSOS PROBLEMAS DE HORÁRIO E MÉTODOS DE SOLUÇÃO

### I.2.1. UMA DESCRIÇÃO INFORMAL

O problema de construção de horários por computador vem sendo discutido na literatura técnica há pelo menos um quarto de século. As diversas versões deste problema podem ser descritas informalmente como segue. Em primeiro lugar temos os participantes, num sentido bem amplo: professores, turmas, salas, laboratórios, equipamentos, etc, bem como um conjunto de horas, também chamadas períodos. As disponibilidades enumeram para cada participante o subconjunto de horas em que este participante pode ser alocado em cada uma das aulas, exames, conferências, ou genericamente, eventos, nos quais está envolvido. Cada evento é caracterizado pela sua duração, pelo assunto e pela coleção de participantes que devem reunir-se para sua realização. Com frequência, e por motivos diversos, são prefixadas as horas de alguns poucos eventos.

Dada uma tal situação, um horário é uma tabela que as socia a cada evento o número estipulado de horas durante as quais os respectivos participantes estão disponíveis para a sua realização e tal que seja satisfeito o requisito fundamen tal, intrínseco ao problema de horário, de que em cada hora ca da participante esteja vinculado a não mais do que um evento.

Em vista desta estrutura sofisticada, o problema de horário corresponde na prática a várias situações de interesse tais como horários de aula de escolas e universidades, horá rios de exames em univerrdidades ou de palestras em conferên cias.

O horário de aulas em geral refere-se a uma semana, sendo mantido inalterado durante todo o período letivo e assim considera-se que o seu ciclo é de uma semana. Frequentemente devem ser atendidas certas exigências adicionais que variam de instituição para instituição.

A construção de horários de exames ou de palestras em conferências, em geral visa organizâ-los de modo que possam ser realizados no menor tempo possível, considerando a quantidade e capacidade de salas e auditórios disponíveis, o espaçamento entre exames prestados por cada aluno e outros aspectos que va riam segundo as circunstâncias.

Um problema correlato, que também tem sido citado oca sionalmente no contexto do problema de horário refere-se à dis tribuição dos alunos de uma Universidade pelas múltiplas turmas correspondentes a cada uma das várias disciplinas em que se inscrevem em cada período letivo. Para cada disciplina são oferecidas diversas alternativas de horário, estabelecidas a



priori pela administração acadêmica, e os alunos devem ser distribuídos pelas turmas levando em conta fatores tais como a capacidade das salas disponíveis para a realização das aulas, as preferências eventualmente expressas pelo aluno quanto ao professor que vai lecionar a matéria, a parte do dia em que as aulas são ministradas e etc. STEWART e CLARK (42) apresentam a descrição detalhada de um tal sistema usado na University of Maryland.

### I.2.2. OS DIVERSOS MÉTODOS DE SOLUÇÃO

Os mais variados métodos foram propostos para a solução das diferentes versões do problema de horário. SCHMIDT e STROHLEIN (40) apresentam uma bibliografia com cerca de 235 referências, abrangendo a literatura disponível até 1979.

Na seção I.3 tratamos dos métodos heurísticos.

A seção I.4 é dedicada à linha de pesquisa iniciada em 1962 por GOTLIEB (20). É a mais longa deste capítulo, pela importância teórica e prática do trabalho realizado por GOTLIEB e CSIMA e outros.

Na seção I.5 descrevemos o trabalho de JOHNSTON e WOLFENDEN (27) que utilizaram um algoritmo de backtrack e em I.6 algumas abordagens por coloração de grafos. Concluindo este capítulo, na seção I.7 descrevemos uma aplicação da programação linear.

A questão das condições necessárias para a existência de um horário de aulas é tratada por SMITH (41). Um dos trabalhos mais citados sobre critérios para avaliar a qualidade dos

horários produzidos por computador foi realizado por BARRA CLOUGH (4).

### I.3. OS MÉTODOS HEURÍSTICOS

#### I.3.1. INTRODUÇÃO

A maioria dos métodos heurísticos originou-se no processo manual de construção de horários. Em cada etapa do processo manual é preciso responder a pelo menos duas questões:

- qual a próxima turma a alocar
- que períodos devem ser atribuídos a cada um dos seus eventos.

À primeira vista parece que a questão crucial é a segunda pois se sempre for respondida corretamente, não só a primeira questão será redundante mas também o horário será obtido rapidamente e nunca será preciso alterar a alocação de um período a algum evento. Entretanto, a experiência mostra que uma boa resposta à primeira questão pode contribuir para uma resposta mais adequada à segunda, na medida em que permitir antecipar, ou até evitar, o aparecimento de dificuldades em etapas subsequentes. Em virtude desta constatação, a maioria dos métodos heurísticos define e utiliza indicadores que orientam o algoritmo na escolha da próxima turma a alocar e dos períodos que serão reservados aos seus eventos. Estes indicadores são baseados num agregado de características e procuram traduzir empiricamente a situação corrente do horário que está sendo construído e a liberdade de escolha que ainda resta para alocar as turmas re

mantescientes. Exemplos destes indicadores podem ser encontrados nos trabalhos de BRITTAN e FARLEY (6) e WOOD (46), entre outros.

Como apontado pelos respectivos autores, estes algoritmos podem deixar de obter solução para um dado problema, mesmo que ele exista ou então podem entrar em "loop" por causa das trocas às quais se recorre para contornar a impossibilidade de alocação de algum evento e que podem tornar-se cíclicas. Apesar destas limitações, os métodos heurísticos são importantes pela engenhosidade no tratamento de certas exigências especiais, por permitirem a construção de horários compatíveis com as salas disponíveis e pelos excelentes resultados na construção de horários de exames em Universidades. A seguir apresentamos uma breve descrição de dois algoritmos heurísticos para a construção de horários de aulas.

### I.3.2. O ALGORITMO HEURÍSTICO DE BRITTAN E FARLEY

O sistema implementado por BRITTAN e FARLEY (6) incorpora sofisticados critérios para a escolha da próxima turma e para a resolução de conflitos em caso de impossibilidade de alocação de alguma aula. As aulas de uma mesma turma são alocadas todas de uma só vez, começando pelas mais longas e procurando assegurar uma boa distribuição destas aulas pelos dias da semana. As turmas vão sendo escaladas para atribuição de horários segundo valores crescentes de um indicador chamado folga. Este indicador exprime a maior ou menor facilidade com que as aulas de uma turma poderão encontrar um horário pois é igual ao número de períodos que continuariam disponíveis por esta turma se

ela fosse alocada.

Para cada um período inicialmente disponível por uma turma, mantem-se um registro da quantidade de turmas já alocadas neste período e conflitantes com a turma em questão. Este número exprime a quantidade de aulas que precisariam ser deslocadas para algum outro período se fosse necessário liberar o período considerado. É portanto um indicador da relativa dificuldade com que este período poderia ser tornado disponível. Assim, quando não se consegue encontrar um período para atender uma dada aula, examinam-se os períodos onde ela poderia ser alocada, em ordem crescente desta dificuldade.

### I.3.3. UM ALGORITMO QUE MELHORA O HORÁRIO POR ESTÁGIOS

O método proposto por AUST (3) para a construção de horários de escolas é inovador sob o seguinte aspecto. Praticamente todos os algoritmos para a construção de horários procuram ir alocando os eventos até que todos sejam acomodados sem violar nenhuma das restrições. A característica comum a estes algoritmos é exigir que em cada etapa da construção do horário sejam respeitadas todas as restrições. As regras usadas para escolha da próxima turma a alocar, bem como dos períodos em que suas aulas serão alocadas, visam preservar esta característica. O método de AUST (3) estende o trabalho de LAWRIE (29) e consiste em obter primeiro um "horário provisório" no sentido de que contém todos os eventos, eventualmente com violação de algumas restrições. Em seguida, este horário é corrigido, removendo as violações destas restrições. Além das aulas com horári

rios prefixados consideram-se três tipos de restrições que de vem ser respeitadas no horário final. Em primeiro lugar, alguns eventos podem exigir dois ou mais períodos consecutivos deven do ser alocados em bloco e respeitando as interrupções naturais tais como almoço e fim de dia. A segunda restrição exige que, em cada período, o número de professores alocados não ex ceda o número de professores disponíveis. Finalmente, as múlti plas aulas de uma mesma turma devem ser distribuídas uniformemente pelos dias da semana.

O algoritmo proposto resolve o problema em três está gios:

- constroi um horário que incorpora as aulas com horários pre fixados e no qual os blocos de duas ou mais aulas respeitam as interrupções naturais;
- reduz o déficit de professores em cada período onde estiver ocorrendo, mantendo o horário correto quanto aos horários pre fixados e blocos;
- corrige a distribuição das aulas, preservando as caracterís ticas já asseguradas nos estágios anteriores.

O primeiro estágio é resolvido como uma série de pro blemas de fluxo de redes. O segundo e o terceiro estágios são resolvidos por algoritmos heurísticos que realizam trocas dos eventos que compõem o horário. Para cada violação encontrada é gerado e resolvido um pequeno problema de programação inteira que determina quais as trocas que devem ser realizadas para que cada novo horário tenha menos violações do que o anterior. Se em algum estágio não se conseguir impor a observância das res trições correspondentes então o problema dado é considerado in viável, terminando o processamento.

## I.4. A ABORDAGEM DE GOTLIEB E CSIMA

### I.4.1. APRESENTAÇÃO

No Congresso do IFIP de 1962 GOTLIEB (20) apresentou um método iterativo para a construção de horários de aulas de escolas por computador, permitindo a incorporação inicial de alguns horários estabelecidos a priori. Este trabalho despertou interesse durante vários anos por duas razões.

A primeira, porque foi um dos primeiros programas para a solução do problema de horário com embasamento teórico, especificamente as condições necessárias e suficientes para a existência de um sistema de representantes distintos de subconjuntos, estabelecidas por HALL (22).

A segunda razão, provavelmente a mais ponderável, é porque enunciava, e se baseava, numa conjectura promissora:

- ou o programa conseguiria ir incorporando os horários preestabelecidos e determinaria os restantes, e isto sem nunca precisar regredir para modificar uma escolha já feita, caso existisse solução para a instância dada do problema;
- ou então, em cada estágio, seria capaz de reconhecer os horários preestabelecidos que, se aceitos, conduziriam a uma situação de inexistência de solução para a instância dada do problema.

Em linhas muito gerais, o esquema proposto por GOTLIEB recebe como entrada na matriz R de requisitos onde o elemento  $r(i,j)$  exprime quantas vezes (horas ou períodos) o professor i deve encontrar a classe j num dia de m períodos ou horas. A solução do problema de horário consiste em construir um arranjo

("array") a três dimensões: professores, classes e períodos. Cada elemento do arranjo representa o encontro potencial de um da do professor com uma dada classe num certo período do dia. Pode-se visualizar assim este arranjo: cada período corresponde a um plano horizontal e em cada plano horizontal temos uma matriz; ca da linha da matriz corresponde a um professor e cada coluna cor responde a uma classe. Os elementos do arranjo tridimensional tem valor zero ou um. Se algum elemento do arranjo tem valor ze ro isto implica a impossibilidade de um encontro entre o profes sor e a classe no período correspondente. Um elemento com valor unitário significa que o professor e a classe podem encontrar-se neste período. O arranjo é inicializado com valor um indican do que cada professor pode encontrar-se com qualquer classe em qualquer período. Os horários preestabelecidos são incluídos su cessivamente, logo no início, segundo as suas prioridades. A me dida que os cálculos se desenvolvem, certos uns vão sendo tro cados para zeros e, ao final do processamento:

- em cada período cada professor pode encontrar-se com somente uma classe e cada classe pode encontrar-se com um professor somente;
- cada professor encontra-se com cada classe o número de vezes estabelecido na matriz de requisitos R.

Em outras palavras, o processamento visa transformar este arranjo tridimensional de um estado inicial onde todos os elementos valem um em outro, final, tal que a soma de todos os elementos de uma mesma vertical, definida pelo professor  $i$  e classe  $j$ , seja exatamente igual ao valor do elemento  $r(i,j)$  dado na matriz de requisitos R.

O método de GOTLIEB exige que durante o processo computacional seja examinada cada seção plana deste arranjo. Estas seções planas são matrizes de zeros e uns. Fixando um professor, a correspondente seção plana mostra as possíveis combinações de classes e períodos para este professor. Fixando a classe, a seção plana exhibe as associações possíveis entre professores e períodos para esta classe. Finalmente, escolhendo um período, a respectiva seção plana nos dá as associações possíveis entre os professores e as classes neste período selecionado.

O exame de cada seção plana é feito em duas etapas:

- etapa 1: Teste de Viabilidade, para verificar se ainda continua sendo possível realizar cada uma das associações que faltam nesta seção plana;
- etapa 2: "Redução" da Matriz, que consiste em mudar de um para zero os valores dos elementos da matriz, definida por esta seção plana, que representam associações que nunca poderão vir a ser usadas neste plano.

Este trabalho de GOTLIEB (20) é a origem do que provavelmente pode ser considerado como um dos significativos esforços de desenvolvimento de programas para a solução do problema de construção de horários de aula por computador. As contribuições que se seguiram abordaram os seguintes aspectos:

- a conjectura de GOTLIEB
- algoritmos mais eficientes para a citada "Redução" de matrizes
- algoritmos mais eficientes para o Teste de Viabilidade
- fundamentos teóricos do método

e se estenderam pelo menos até 1971 com o artigo de LIONS (33).



#### I.4.2. HISTÓRICO E EVOLUÇÃO

Na 18ª Conferência Anual da ACM, realizada em agosto de 1963, CSIMA E GOTLIEB (11) relataram os resultados de experiências feitas com o programa em casos envolvendo nove professores e nove classes, considerando dias úteis com nove períodos de atividades escolares. Os resultados foram considerados satisfatórios pois dos 223 problemas de horário propostos todos haviam sido resolvidos com sucesso. Entretanto, em alguns poucos casos constatou-se que a conjectura original não se confirmava e em consequência a mesma foi revista e completada.

Animado pelo relato acima, DUNCAN (15) construiu um programa baseado no trabalho de GOTLIEB e CSIMA, porém mais geral no sentido de que aceitaria processar problemas com outros números de professores, classes e períodos diários que não os nove usados por GOTLIEB e CSIMA. Em outras palavras, de um problema para outro poderia ser usado um número diferente de professores, classes e períodos. Foram mantidas as restrições originais de que cada professor encontra cada classe apenas uma vez e que o número de professores e classes deve ser o mesmo dentro de cada problema. Este programa visava realizar testes para determinar se o método seria ainda válido para casos maiores e verificar como o tempo de processamento cresceria com o tamanho do problema. Os resultados dos 303 testes realizados, usando um IBM 7090 estão na tabela I.1.

TABELA I.1 - Resultados dos testes de DUNCAN (15)

Professores e Classes	Períodos	Casos	Tempo Médio (minutos)
10	8	3	.53
9	9	250	.50
10	9	1	.66
11	9	1	1.09
12	9	1	1.93
13	9	1	3.67
14	9	1	6.58
15	9	1	11.75
16	9	1	21.62
17	9	1	40.99
18	9	1	74.83
10	10	3	1.06
11	11	7	2.00
12	12	23	3.78
13	13	4	8.30
14	14	3	17.23
15	15	1	26.32

A avaliação de DUNCAN é a seguinte:

"During these tests there were no failures encountered, and the larger cases did not require any revision of the original theory.

The increase in time is by a factor of about two for each teacher added, and this, although disturbing, is not surprising in view of the method of solution of the problem".

Finalmente DUNCAN conclui que a filosofia básica do método de GOTLIEB e CSIMA pode ser mantida para resolver problemas concretos de horário e informa que com base neste método seria desenvolvido um programa para a construção de horários de escolas reais.

Relata LIONS (32) que após o sucesso dos testes realizados por DUNCAN, o método de GOTLIEB e CSIMA foi considerado suficientemente promissor para ser testado em situações reais. Como isto representaria um considerável empreendimento, tanto em termos de programação como em interação com escolas, em maio de 1964 The Ontário Institute for Studies in Education confiou esta tarefa à empresa J. Kates and Associates, de Toronto, Ontário.

O próximo progresso é relatado por GRIFFITH (21) da J. Kates and Associates. A constatação de DUNCAN de que a inclusão de cada novo professor redundava em duplicação do tempo de processamento fez com que o processo de "redução" das matrizes envolvidas fossem implementado por um algoritmo mais eficiente, mantendo porém a filosofia geral do método. Para evidenciar este aperfeiçoamento, GRIFFITH (21) relata que esta modificação teve como consequência reduzir de 75 minutos para alguns segundos o tempo de processamento de um problema com 18 professores e cita que um outro problema, agora com 43 professores, exigiu cerca de 3 minutos de processamento.

A seguir LIONS (30) da J. Kates and Associates descreve a aplicação do então já conhecido "método Húngaro" na realização do Teste de Viabilidade bem como para a subsequente "Redução" das matrizes envolvidas. O "método Húngaro" é um algoritmo que permite obter uma associação de custo ótimo entre os vérti

ces de um grafo bipartite com  $2n$  vértices. Convém lembrar que cada seção plana do arranjo tridimensional que constitui o espaço onde se constroi a solução do problema de horário envolve sempre duas das três entidades: professores, classes e períodos e que esta seção plana, no método de GOTLIEB e CSIMA, é uma matriz efetivamente quadrada. Com o "método Húngaro" o Teste de Viabilidade é implementado pela aplicação sucessiva de um procedimento (que ele chama EXPAND) a determinadas filas das seções planas do arranjo tridimensional. Isto é feito até que se consiga uma associação em cada seção plana considerada ou se evidencie a impossibilidade de fazê-lo. A "redução" da matriz é obtida então mediante uma única aplicação deste procedimento. Uma descrição recente do "algoritmo Húngaro" com prova de correção e exemplo de aplicação encontra-se em PAPADIMITRIOU e STEIGLITZ (37).

A contribuição seguinte é, infelizmente, negativa. Relata LIONS (66) que após completar algumas centenas de horários com dados oriundos de escolas reais, encontrou um problema que refuta a conjectura de GOTLIEB. Neste trabalho, enviado como carta ao Editor, LIONS descreve o problema e menciona que outros casos surgiram nos quais, apesar de se verificarem as condições que foram tidas por suficientes, não foi possível construir o horário. Em cada um destes casos, pequenas variações na estratégia da conclusão do horário foram suficientes para contornar a dificuldade. Conclui afirmando que, como tais casos ainda são raros na utilização do programa, mantém a convicção de que o método de GOTLIEB e CSIMA é a base de um sistema prático e útil para a construção de horários.

O contraexemplo obtido por LIONS leva DEMPSTER a analisar o problema, explicar porque a conjectura não se confirmou e propor condições (tanto necessárias como suficientes) alternativas.

Finalmente SMITH (41) desenvolve um formalismo geral para o conjunto das condições necessárias para a existência de uma solução para um dado problema de horário, seguido de uma avaliação dos vários conjuntos. O conjunto de condições necessárias, usado por GOTLIEB e CSIMA, é incluído no grupo dos computacionalmente eficientes.

#### I.4.3. A UTILIZAÇÃO DO MÉTODO DE GOTLIEB E CSIMA

Normalmente as escolas desenvolvem as suas atividades seguindo um horário que distribui as diversas aulas ao longo da semana. O método de GOTLIEB e CSIMA resolve o problema de um dia e assim a produção de horários de cada escola é feita em duas etapas:

- a solução do problema semanal;
- a solução do problema diário.

Este processo foi usado durante algum tempo para a produção de horários para algumas escolas da Província de Ontário, Canadá e assim é interessante registrar a avaliação de LIONS (32) que participou ativamente do projeto.

#### I.4.3.1. O Método, na Prática

Embora normalmente os problemas semanal e diário sejam semelhantes, o método de GOTLIEB e CSIMA somente é usado para a solução de cada problema diário. O problema semanal consiste em dividir um conjunto de eventos que constituem as atividades semanais, em cinco ou seis subconjuntos, cada um dos quais define um problema diário. A experiência na utilização do método mostrou que o problema semanal é de solução muito mais fácil por que um professor ou uma classe pode participar de diversas atividades durante cada dia, mas só pode estar engajado em uma única atividade em cada período. Assim, o processo de solução do problema semanal pode ser consideravelmente mais simples do que para o problema diário. Esta divisão do problema, decorrente da limitação dos computadores então disponíveis, trouxe a vantagem de permitir o atendimento, em considerável proporção, de certos requisitos qualitativos, difíceis de definir, que idealmente o horário acabado deveria exibir. Por exemplo, se um professor e uma classe devem encontrar-se três vezes por semana então uma vez na segunda, uma na quarta e uma na sexta-feira é considerado excelente, enquanto uma vez na terça, uma na quinta e uma na sexta-feira é bastante aceitável mas duas vezes na quinta e uma vez na sexta-feira é considerado péssimo.

O método usado pelo programa que resolve os problemas semanais é bastante empírico. Os eventos que devem realizar-se durante a semana são examinados um por um, na ordem das prioridades estabelecidas pelas escolas. Todos os possíveis desdobramento destes eventos pelos dias da semana são considerados e avaliados por meio de uma função de mérito. Esta função leva em

conta a distribuição dos eventos ao longo da semana bem como as disponibilidades remanescentes de professores, classes e salas nos vários dias da semana. A solução do problema semanal é construída por meio de um procedimento razoavelmente simples que, a cada passo, examina a situação um pouco à frente e escolhe aquele desdobramento que permite proporcionar aos eventos do final da lista características qualitativas pelo menos aceitáveis.

Verificou-se que o algoritmo que reparte o problema semanal num conjunto de problemas diários é bastante adequado para obter uma distribuição viável, mantendo ainda razoável distribuição dos períodos correspondentes a cada matéria. Por exemplo, para as matérias com três aulas semanais foi possível virtualmente suprimir as distribuições do tipo segunda, terça e quarta, em favor de distribuições do tipo segunda, terça e quinta, nas quais não há mais do que dois dias consecutivos.

As escolas preparam os dados para os respectivos horários em formulários projetados para utilização independente da assistência de analistas. Numa primeira fase do programa os dados são lidos e criticados quanto à consistência interna, e ordenados segundo as suas prioridades. A seguir são desdobrados pelos dias da semana. Produz-se um relatório exibindo os dados e a distribuição dos eventos, numa forma adequada para verificação visual. Se os resultados não forem considerados satisfatórios, os dados podem ser modificados e submetidos a novo processamento. Dentro da filosofia de que o computador é apenas uma ferramenta, verificou-se ser extremamente útil retocar manualmente esta distribuição dos eventos pelos dias da semana e incorporar estas modificações nos processamentos subsequentes do mesmo problema semanal. Esta possibilidade tem a vantagem de

permitir às escolas a verificação dos resultados intermediários, eliminando eventuais erros e omissões na especificação dos dados.

Uma vez conseguida uma distribuição satisfatória, passa-se à solução dos problemas diários e é neste ponto que entra o algoritmo de GOTLIEB e CSIMA propriamente dito. Finalmente, produzem-se os relatórios exibindo os horários de cada professor, classe e sala.

O programa usado consiste de cerca 7000 comandos FORTRAN e 300 instruções em MAP para operações booleanas não disponíveis diretamente em FORTRAN. O computador era um IBM 7094 da Universidade de Toronto.

#### I.4.3.2. As Dificuldades e a Avaliação de LIONS (32)

Para a solução do problema diário, o algoritmo de GOTLIEB e CSIMA praticamente assegura que se obterá um horário tão logo se consiga incluir os horários pre-estabelecidos. Verificou-se que para os problemas mais complicados, não se conseguia concluir logo no 1º processamento, a fase de incorporação dos horários pre-estabelecidos. Tornava-se necessário abrir mão de certas exigências e, com base no diagnóstico fornecido pelo programa o analista ativava certos controles do programa para modificar a ordem em que os requisitos eram processados. Em alguns casos raros foram necessários até seis processamentos com tais ajustes para resolver o problema de uma escola; em média, três rodadas eram suficientes. Frequentemente a intervenção do analista para resolver um problema era frustrada pelo aparecimento



subsequente de outros problemas. Assim atingia-se um ponto além do qual a intervenção dos analistas proporcionava um retorno ca da vez menor e era preferível devolver o problema à escola para retoques finais.

Quanto aos aspectos qualitativos, observou-se que as escolas diferem significativamente na ênfase dada às diversas qualidades que um horário deve exibir. Alguns aspectos conside rados extremamente importantes por algumas escolas sequer chega ram a ser mencionados por outras.

Qualquer horário de aula é a expressão de um compromis so. Constatou-se que em geral é impossível incorporar num horá rio todas as exigências que uma escola pode formular. Embora se possa desenvolver critérios objetivos para avaliar a qualidade de um horário, como proposto por BARRACLOUGH (4), certamente um critério importante é essencialmente subjetivo: Estaria a es cola disposta a comprá-lo? Sob este prisma relata LIONS(32) que um horário produzido nas primeiras experiências com o método, em bora ainda longe de perfeito, foi considerado pela escola como de grande valor e foi convertido no horário adotado no ano leti vo 1966-1967. Por outro lado, um horário avaliado como perfeito pelos critérios propostos por BARRACLOUGH (4) foi declarado inadequado. Tais situações mostram que o sucesso de um sistema de construção de horários não pode ser totalmente avaliado ape nas com base em critérios objetivos.

#### I.4.4. OS ASPECTOS FORMAIS

Finalizamos a apresentação da abordagem de GOTLIEB e CSIMA com um enunciado da Conjectura de GOTLIEB e com a explicação de DEMPSTER para a existência de contra-exemplos que refutam esta Conjectura.

Teoricamente este método pode ser relacionado com sistemas de representantes distintos de subconjuntos, a construção de quadrados latinos, a coloração de grafos e fluxos em redes de multicomodidades.

A descrição do formalismo do método de GOTLIEB e CSIMA é baseada em DEMSTER (12) e DEMPSTER (13).

##### I.4.4.1. Algumas Definições

O algoritmo de GOTLIEB e CSIMA recebe como entrada uma matriz  $R$  e um conjunto  $S'$ .  $R$  é uma matriz de requisitos, de elementos inteiros não negativos, quadrada, de ordem  $n$  e tal que a soma dos elementos de cada fila (linha ou coluna) perfaz  $m$ , o número de períodos por dia. O valor de  $r_{ij}$  é o número de vezes que o professor  $i$  deve encontrar a classe  $j$  num dia de  $m$  períodos. A quantidade de professores e classes é dado por  $n$ . O conjunto  $S'$  é constituído pelos horários pre-estabelecidos que devem ser contidos em qualquer solução  $S$  do problema dado. Denotaremos por  $P(R, S')$  um tal problema de horário.

A solução de  $P(R, S')$  é construída com o auxílio de um arranjo  $A = (a_{ijk})$  que pode ser visualizado como um empilhamento de  $m$  matrizes  $n \times n$  cujos elementos pertencem a  $\{0, 1\}$ . Desig

naremos por  $A^0$  o estado inicial deste arranjo, quando todos os elementos tem valor 1 e por  $A^1$  o arranjo que se obtém a partir de  $A^0$  mediante a inclusão dos horários pre-estabelecidos dados em  $S'$ .

Sejam  $X = (x_{ijk})$  e  $Y = (y_{ijk})$  dois arranjos tridimensionais cujos elementos  $x_{ijk}$  e  $y_{ijk}$  pertencem a  $\{0,1\}$ . No que se segue é conveniente adotar as definições abaixo, conforme DEMPSTER (12).

A união e a interseção de  $X$  e  $Y$  são definidas por:

$$X \cup Y = (x_{ijk} + y_{ijk} - x_{ijk} \cdot y_{ijk})$$

$$X \cap Y = (x_{ijk} \cdot y_{ijk})$$

O arranjo  $X$  está contido no arranjo  $Y$ , o que denotaremos por  $X \subset Y$  se:

$$X \cap Y = X$$

A cardinalidade do arranjo  $X$ , denotada por  $|X|$ , é dada por:

$$|X| = \sum_{i,j,k} |x_{ijk}|$$

Estas definições também podem ser aplicadas a arranjos a uma e duas dimensões que contem com o mesmo número de elementos em filas correspondentes.

Qualquer arranjo  $A = (a_{ijk})$  com pelo menos um elemento não nulo e tal que  $A \subset A^0$  é chamado arranjo de disponibilidades

de  $P(R, S')$ .

Uma seção plana de um arranjo de disponibilidades  $A$  é qualquer matriz obtida fixando-se qualquer um dos três índices; uma fila do arranjo  $A$  é qualquer vetor obtido fixando-se dois dos três índices de  $A$ . O arranjo nulo, aquele onde todos os elementos tem valor zero, será denotado por  $0$ . Uma disponibilidade de  $A$  é qualquer elemento  $a_{ijk}$  de  $A$  que tem valor 1, sendo caracterizada pela tripla  $(i, j, k)$ .

Nesta representação, uma solução de  $P(R, S')$  é um arranjo de disponibilidades  $S = (s_{ijk})$  que pode ser visto como um conjunto ordenado de matrizes  $S_k$  tais que:

$$\sum_{k=1}^m S_k = R$$

e em cada matriz  $S_k$  cada linha e cada coluna contém apenas um elemento com valor 1.

O algoritmo proposto por GOTLIEB e CSIMA toma como ponto de partida o arranjo  $A^0$  e o conjunto  $S'$  e mediante a troca de certas disponibilidades para zero gera uma sequência de arranjos de disponibilidades

$$A^0 \supset A^1 \supset A^2 \supset \dots \supset S$$

que transforma  $A^0$  na solução  $S$  de  $P(R, S')$ . GOTLIEB observou que em qualquer seção plana da solução  $S$  são satisfeitas certas condições bidimensionais. Assim, procurou estabelecer um conjunto de condições aplicáveis a seções planas de arranjos de disponibilidades e que fossem suficientemente poderosas para eliminar aquelas disponibilidades que jamais poderiam estar contidas em

qualquer solução de  $P(R, S')$ . Estas são denominadas as condições (planares) fortes de Hall. Conjecturou GOTLIEB que se fosse possível por estes meios reduzir  $A^0$  a um arranjo  $A^1$  contendo os horários preestabelecidos em  $S'$  e satisfazendo estas condições (planares) fortes de Hall, que asseguram a existência de soluções em cada plano de  $A^1$  (i.e. para um professor, uma classe ou um período), estas condições seriam suficientes, bem como necessárias, para assegurar a existência de uma solução  $S \subset A^1$  para o problema (tridimensional)  $P(R, S')$ . Admitia-se também que qualquer disponibilidade de  $A^1$  poderia ser escolhida como pertencente a uma solução e que sempre seria possível reduzir  $A^1$  a um novo,  $A^2$ , que continuaria satisfazendo as condições fortes de Hall. A seguir, escolhendo uma disponibilidade qualquer de  $A^2$  e aplicando o processo de redução seria possível continuar iterando até certamente obter uma solução  $S$ . Em consequência dos testes relatados por CSIMA e GOTLIEB (11), verificou-se que as condições (planares) fortes de Hall são insuficientes para eliminar todas as disponibilidades que não poderiam pertencer a uma solução de  $P(R, S')$ . Assim a conjectura original foi modificada passando a postular a existência em  $A^1$  de alguma disponibilidade que conduziria a  $A^2$  e que por meio de alguma disponibilidade de  $A^2$  se chegaria a  $A^3$ , e assim por diante, até uma solução  $S$  para  $P(R, S')$ .

O problema  $P(R, S')$  descrito acima foi denominado por CSIMA de um problema sem folgas ("tight"). Os problemas de horário de escolas reais são representados tipicamente por matrizes de requisitos não quadradas, nas quais as somas segundo linhas e colunas podem diferir. CSIMA (10) mostra que qualquer problema de horário com tal matriz de requisitos sempre pode ser imer

so num problema sem folgas na forma descrita por  $P(R,S')$  e que uma solução de  $P(R,S')$  fornece uma solução para o problema real.

#### I.4.4.2. As Condições Fortes de Hall

Seja  $\mathfrak{D}$  o conjunto de todos os subconjuntos de  $N = \{1, 2, \dots, n\}$ . Então um arranjo de disponibilidade  $A$  satisfaz as condições (planares) de Hall se:

i) fixando  $i$ ,

$$\left| \bigcup_{j \in J} A_{ij} \right| \geq \sum_{j \in J} r_{ij}, \text{ para todo } J \in \mathfrak{D}$$

ii) fixando  $j$ ,

$$\left| \bigcup_{i \in I} A_{ij} \right| \geq \sum_{i \in I} r_{ij}, \text{ para todo } I \in \mathfrak{D}$$

iii) fixando  $k$ ,

$$\left| \bigcup_{i \in J} A_{ik} \right| \geq \sum_{j \in J} 1 = |J|, \text{ para todo } J \in \mathfrak{D}$$

Estas três condições são a expressão das condições necessárias e suficientes para a existência de um sistema de representantes distintos de subconjuntos, estabelecidas por Hall (22). No presente contexto as condições acima implicam a existência de uma solução plana contida em  $A$  para cada valor dado de  $i, j$  ou  $k$ . Um arranjo de disponibilidades que não satisfaz estas condições é dito inviável.

Uma partição  $\mathcal{P}_2$  do conjunto  $N$  é dita mais fina que outra partição  $\mathcal{P}_1$  de  $N$  se todos os conjuntos de  $\mathcal{P}_2$  são subconjuntos dos conjuntos de  $\mathcal{P}_1$  e pelo menos dois conjuntos de  $\mathcal{P}_2$  são subconjuntos de próprios de  $\mathcal{P}_1$ .

Um arranjo de disponibilidades  $A$  satisfaz as condições (planares) fortes de Hall se satisfaz as condições de Hall e se ainda:

- i) fixando  $i$ , existe uma partição  $\gamma_i$  de  $N$ , e  $\gamma_i$  é a mais fina dentre as partições de  $N$ , tal que para todo  $J \in \gamma_i$

$$\left| \bigcup_{j \in J} A_{ij} \right| = \sum_{j \in J} r_{ij}$$

e

$$A_{ij} \cap A_{ij'} = 0$$

para

$$j \in J \text{ e } j' \in N \setminus J;$$

- ii) fixando  $j$ , existe uma partição  $\gamma_j$  de  $N$ , e  $\gamma_j$  é a mais fina dentre as partições de  $N$ , tal que para todo  $I \in \gamma_j$

$$\left| \bigcup_{i \in I} A_{ij} \right| = \sum_{i \in I} r_{ij}$$

e

$$A_{ij} \cap A_{i'j} = 0$$

para

$$i \in I \text{ e } i' \in N \setminus I$$

e

- iii) fixando  $k$ , existe uma partição  $\gamma_k$  de  $N$ , e  $\gamma_k$  é a mais fina dentre as partições de  $N$ , tal que para todo  $J \in \gamma_k$

$$\left| \bigcup_{j \in J} A_{jk} \right| = \sum_{j \in J} 1 = |J|$$

e

$$A_{jk} \cap A_{j'k} = 0$$

para

$$j \in J \text{ e } j' \in N \setminus J.$$

Usamos  $\setminus$  para denotar a diferença de conjuntos.

Segundo DEMPSTER (12) e DEMPSTER (13) as condições que foram acrescentadas às condições de Hall asseguram que  $A \subset A^1$  é um arranjo de disponibilidades de cardinalidade maximal e que admite uma solução planar em cada um dos três planos que passam por cada disponibilidade de  $A$ . Uma solução de  $P(R, S')$  é um arranjo de disponibilidades particular no sentido de satisfazer as condições fortes de Hall de modo tal que todos os conjuntos de partições de índices que se referem a filas não nulas podem ser tomados como os elementos de  $N$ , i.e. somente para uma solução todas as desigualdades se verificam como igualdades, nas condições fortes de Hall.

#### I.4.4.3. A Conjectura

No trabalho de CSIMA (10) descreve-se como transformar um arranjo de disponibilidades que satisfaz as condições de Hall em um único arranjo de disponibilidades que atende as condições fortes de Hall e assim tem cardinalidade maximal. Tal transformação, aplicada a um arranjo de disponibilidades  $A$  é denominada uma redução de  $A$  e o único arranjo resultante, caso exista, é de



nominado por GOTLIEB e CSIMA de arranjo reduzido, sendo denotado por  $r(A)$ .

Finalmente, uma disponibilidade  $a_{ijk}$  de um arranjo de disponibilidade arbitrário  $A$  é dita consignada se não existir nenhuma outra disponibilidade nas três filas de  $A$  que passam por  $a_{ijk}$ . A operação de consignação consiste em mudar para zero todas as outras disponibilidades destas três filas. O novo arranjo, obtido de  $A$  mediante a consignação de  $a_{ijk}$ , será designado por  $A_a$  para uma dada tripla  $(i,j,k)$ . Naturalmente  $A_a$  pode ser inviável embora  $A$  não o seja.

Satisfeitos estes preliminares podemos enunciar a conjectura revisada de GOTLIEB e CSIMA.

#### CONJECTURA

Seja  $S$  uma solução de  $P(R,S')$  contida num arranjo de disponibilidades  $A$  que satisfaz as condições fortes de Hall e seja  $a$  uma disponibilidade de  $A$ . Então ou  $A_a$  é inviável ou existe uma solução  $S_a$  incluindo disponibilidade  $a$  contida em  $r(A_a)$ .

É a esta conjectura revisada que LIONS (31) encontrou um contra-exemplo  $12 \times 12 \times 3$ . Um outro contra-exemplo  $8 \times 8 \times 3$  é apresentado em DEMPSTER (12). Isto significa que é possível gerar um arranjo de disponibilidade que satisfaz as condições fortes de Hall mas que não contém solução de  $P(R,S')$ . Entretanto, sabe-se também que na ausência de horários pré-estabelecidos ( $S' = \emptyset$ ) sempre se pode obter um horário, CSIMA (10).

#### I.4.4.4. A Análise de DEMPSTER (12)

Os motivos da não confirmação desta Conjectura são analisados em DEMPSTER (12). Segundo ele, o problema é que embora existam soluções planares nos três planos pertencentes a cada disponibilidade de qualquer arranjo A que satisfaz as condições fortes de Hall, estas condições não são suficientemente poderosas para assegurar que um conjunto destas soluções planares sempre possa ser conjugado para formar uma solução do problema tridimensional  $P(R.S')$ .

Procurando oferecer uma contribuição construtiva, DEMPSTER (12) define "cadeias redundantes" e classifica as disponibilidades em pares e ímpares. Estabeleceu também as condições necessárias e suficientes para a existência de uma solução contendo uma disponibilidade dada de um arranjo de disponibilidades reduzido. O parágrafo seguinte é a conclusão do seu estudo.

"Ao que tudo indica são extremamente raras as instâncias em que o algoritmo de GOTLIEB e CSIMA deixa de detectar incompatibilidades durante a construção de horários reais. Não obstante, e de acordo com o espírito da abordagem de GOTLIEB e CSIMA poderia ser desejável desenvolver um algoritmo que pudesse localizar e eliminar as disponibilidades que não podem vir a pertencer a uma solução. Entretanto, como tal algoritmo deveria localizar cadeias redundantes e eliminar todas as disponibilidades ímpares que precedem o primeiro ponto de ramificação de cada cadeia, tal algoritmo provavelmente seria ineficiente".

### I.5. UMA APLICAÇÃO DO ALGORITMO DE BACKTRACK

A descrição detalhada do algoritmo de backtrack é apresentada na seção II.6 pois este foi o método por nós adotado no presente trabalho.

A utilização pioneira do algoritmo de backtrack para a construção de horários de aula de escolas foi empreendido por JOHNSTON e WOLFENDEN (27) e apresentada no Congresso da IFIP de 1968. Neste artigo os autores apresentam os documentos usados para a especificação dos dados e descrevem brevemente o algoritmo implementado. Em linhas gerais estes documentos descrevem os participantes de cada aula, sua duração e indicam quantas vezes cada uma destas aulas deve ser realizada por semana; informam a disponibilidade de cada participante (professor, classe, sala, equipamento especial) e os horários prefixados.

O programa é baseado num algoritmo de backtrack onde cada solução parcial é estendida mediante a escolha de um horário para a realização de uma aula, com duração de um ou mais períodos consecutivos. Inicialmente vão sendo selecionadas e incorporadas ao horário as aulas com horários prefixados e aquelas que, em consequência das escolhas anteriores, ficam com apenas uma alternativa, isto é, são forçadas a usar certos horários. As aulas de maior duração tem precedência. Finalmente, sempre que há alguma liberdade de escolha, avalia-se para cada aula o valor de

$$X = \frac{1}{m} \sum_{j=1}^m s_j$$

onde  $m$  é o número total de itens (professores e classes) envolvidos e  $s_j$  é o número de alternativas que o  $j$ -ésimo item possui em cada um dos períodos ainda disponíveis para esta aula. A próxima aula a ser selecionada para a atribuição de um horário tentativo é aquela que tem o menor valor de  $X$ .

#### I.6. O PROBLEMA DE HORÁRIO E COLORAÇÃO DE GRAFOS

A coloração de grafos tem sido citada frequentemente no contexto do problema de horário. Tanto a coloração de vértices, como a de arestas tem sido usada.

Vejamos a abordagem por coloração de vértices. Dado um problema de horário, contrói-se um grafo  $G(V,E)$ . Cada vértice  $v \in V$  corresponde a um evento (aula, exame) e vice-versa. O conjunto  $E$  de arestas conterá o par  $(v,v')$  se e somente se os eventos correspondentes aos vértices  $v$  e  $v'$  tiverem em comum algum participante. Uma solução do problema de horário pode ser obtida a partir de uma  $k$ -coloração de  $G$ , interpretando cada cor como um período diferente. Frequentemente a solução desejada deve incorporar horários prefixados para alguns eventos e respeitar restrições na disponibilidade dos participantes. Em termos de coloração de vértices, os horários prefixados correspondem a cores preestabelecidas para certos vértices, enquanto restrições à disponibilidade dos participantes implicam na restrição das cores que podem ser usadas para os vértices correspondentes. NEUFELD e TARTAR (35) analisam teoricamente este problema com horários prefixados e restrições à disponibilidade dos participantes e mostram como construir um grafo  $G$  sem

tais condicionamentos, equivalente ao grafo assim condicionado.

Em 1968, WELSH e POWELL (45) examinaram a construção de horários de exames por coloração de vértices. Estabeleceram uma cota superior para o número cromático de um grafo e apresentaram um algoritmo heurístico que respeita esta cota superior. Entretanto, alertam para o fato de que a diferença entre esta cota superior e o número cromático pode tornar-se arbitrariamente grande para certos grafos. É interessante comparar este resultado como o obtido por GAREY e JOHNSON (17) em 1976, apresentado na seção II.8.

A coloração de arestas também tem sido usada. Um trabalho nesta linha é o de MCDIARMID (34).

### I.7. A PROGRAMAÇÃO LINEAR E O PROBLEMA DE HORÁRIO

O artigo de AKKOYUNLU (2) descreve um modelo matemático para a alocação de turmas de um departamento de uma universidade e apresenta um algoritmo para obtenção de uma solução globalmente ótima, mediante a associação de um custo a cada característica exibida pelo horário. Os custos podem ser infinitos (quando se tornam restrições), bem como positivos ou negativos (para características indesejáveis e desejáveis, respectivamente). O modelo é suficientemente poderoso para atender todas as exigências usuais. Entretanto, admite-se que não há limitação para o número total de salas de aulas disponíveis; em compensação, o problema mais simples e mais concreto, de competição de várias turmas por recursos especiais como um laboratório, pode ser incorporado facilmente no modelo.

O algoritmo consiste em reduzir o problema dado de horário ao que em essência é um problema de programação linear que é então resolvido para produzir um horário de custo global mínimo mediante um procedimento Simplex adaptado.

A utilização deste método evidenciou algumas dificuldades da abordagem por programação matemática. Em primeiro lugar, são escassos os princípios que orientam a atribuição de custos a cada uma das características que podem ser levadas em conta pelo modelo. Mais ainda, é extremamente difícil aquilatar a importância relativa das características de tipos diversos, para efeito de atribuição de custos. Esta dificuldade de fixar os custos é bastante inquietante pois uma ligeira perturbação dos seus valores pode resultar num horário drasticamente diferente. Uma segunda dificuldade reside no fato de que a solução ótima pode penalizar injustamente algum professor ou alguma turma e às suas custas obter um ótimo global. Como podem existir soluções próximas da ótima, mais equitativas, o programa implementado imprime também algumas poucas destas soluções.

CAPÍTULO II - A NATUREZA DO PROBLEMA DE HORÁRIO  
E DO ALGORITMO DE BACKTRACK

II.1. INTRODUÇÃO

No capítulo I apresentamos algumas das abordagens usadas na prática para solução do problema de horário. Agora introduzimos os principais conceitos que permitem caracterizar a natureza do problema de horário do ponto de vista teórico e que proporcionam um contexto para descrição do algoritmo de backtrack, escolhido para a implementação de um programa protótipo para construção de horários de aula.

Este capítulo pode ser dividido em três partes. A primeira, seções II.2 a II.4, trata dos problemas em geral, complexidade de algoritmo e problemas NP-completo. A segunda parte, seção II.5, apresenta o principal resultado teórico conhecido sobre o problema de horário: é um problema NP-completo. A terceira parte, seções II.6 a II.8, resume os dois temas: problemas NP-completo e algoritmos, descrevendo duas das várias técnicas de abordagem dos problemas NP-completo: algoritmo de backtrack, seção II.6, e algoritmos de aproximação, seção II.7. Finalmente, em II.8 apresentamos um resultado sobre existência de algoritmos de aproximação para o problema de coloração de grafos, cuja estrutura é bastante semelhante à do problema de horário.

Completamos esta introdução com um breve comentário de ordem geral sobre computabilidade, complexidade e dificuldade intrínseca dos problemas.

A teoria da computabilidade exprime o conceito intuitivo de procedimento computacional por meio de modelos matemáticos da computação e, com base nestes modelos procura estabelecer o que pode e o que não pode ser feito por meio de algoritmos. Visa descobrir as limitações últimas, absolutas, intrínsecas à natureza da computação.

A teoria da complexidade trata dos aspectos quantitativos, em termos de consumo dos recursos computacionais. Estuda, por meio de uma teoria geral, as propriedades que devem ser satisfeitas por qualquer possível medida da dificuldade, ou complexidade de problemas e algoritmos. A questão importante, e onde está o desafio maior, reside em determinar para cada problema e quantidade mínima de recursos computacionais que são necessários para sua solução, por qualquer algoritmo existente ou que venha a ser proposto. Esta quantidade mínima de recursos constitui um limite inferior da dificuldade intrínseca, inerente ao problema.

Por sua vez, a análise dos algoritmos, ao exprimir o consumo de recursos computacionais por meio destas medidas, permite classificar os algoritmos segundo a sua complexidade, compará-los e verificar as possibilidades do seu aperfeiçoamento.

Ainda não foram estabelecidos os limites inferiores da complexidade de muitos problemas de grande importância prática. Assim, adota-se provisoriamente como indicador desta dificuldade a complexidade dos melhores algoritmos conhecidos para resolvê-los. Desta forma, a teoria da complexidade computacional proporciona um melhor conhecimento da natureza dos problemas, ao permitir classificá-los segundo a sua dificuldade intrínseca, demonstrada ou presumida.



Quanto à possibilidade de solução algorítmica os problemas são decidíveis ou indecidíveis. Os decidíveis, em ordem decrescente de dificuldade intrínseca são classificados em: problemas que provavelmente exigem tempo pelo menos exponencial, NP-difícil, NP-completo e polinomiais.

O problema de horário de aulas na sua forma mais interessante do ponto de vista prático é NP-completo, embora em certos casos particulares favoráveis, seja polinomial.

## II.2. PROBLEMAS E LINGUAGENS FORMAIS

Visando tornar mais clara a apresentação deste trabalho, descrevemos informalmente alguns termos relacionados com problemas algorítmicos.

Um problema algorítmico é uma questão geral a ser respondida, denominada objetivo do problema, possuindo diversos parâmetros cujos valores são deixados em aberto. Geralmente estes parâmetros são estruturas abstratas tais como matrizes, grafos, conjuntos, funções, linguagens e etc. Define-se um problema fornecendo (i) uma descrição geral dos seus parâmetros e (ii) a especificação das propriedades que a resposta ou solução deve exibir. Obtem-se uma instância do problema atribuindo valores a todos os parâmetros do problema; estes valores constituem dos dados do problema. Assim, um problema é o conjunto de todas as suas instâncias.

Para cada problema algorítmico P define-se uma função que associa a cada instância I de P o seu tamanho, geralmente expresso por um inteiro. Esta função não é única e sua escolha

é ditada por considerações teóricas e práticas relacionadas com a discussão do problema considerado.

É conveniente destacar três classes de problemas algorítmicos: problemas de otimização, de localização e de decisão.

Um problema de otimização  $P$  é um problema de minimização ou de maximização. Consiste de um conjunto  $D$  de instâncias. Cada instância  $I \in D$  é constituída por um par  $(S, m)$  onde  $S$  é um conjunto de soluções viáveis para a instância  $I$  enquanto  $m$  é uma função que associa a cada  $I \in D$  e a cada solução viável  $s \in S$  um número positivo chamado valor desta solução. Se  $P$  é um problema de minimização (maximização) então uma solução ótima da instância  $I \in D$  é uma solução viável  $s^* \in S$  tal que, para todo  $s \in S$

$$m(I, s^*) \leq m(I, s) \quad (\text{ou} \quad m(I, s^*) \geq m(I, s)).$$

O objetivo de um problema de otimização é obter uma solução ótima, caso exista.

Um problema de localização tem por objetivo encontrar uma estrutura de certo tipo e que satisfaça um dado conjunto de propriedades. O objetivo de um problema de decisão é decidir se a resposta é SIM ou NÃO a uma questão referente a alguma estrutura abstrata dada.

Os problemas de decisão são, em geral, mais simples do que os problemas associados de localização e de otimização. Assim, alguma prova da sua eventual intratabilidade pode ser estendida ao correspondente problema de localização ou de otimização.

A teoria formal da complexidade dos problemas foi desenvolvida em termos de linguagens formais, com as quais teve

origem. Embora um tanto artificial, isto foi feito assim por duas razões. A primeira, porque simplifica a notação. A segunda razão, é porque permite tratar, por meio de um modelo comum, problemas oriundos de disciplinas diversas tais como teoria dos números, banco de dados, grafos, otimização combinatória, etc. Assim, o tratamento intuitivo da complexidade dos problemas é desenvolvido em termos de problemas de decisão, enquanto o tratamento rigoroso realiza-se por meio de linguagens formais. Finalmente, a transição entre estas duas formulações, quando necessária, é feita adotando um esquema de codificação adequado que permite apresentar o problema de decisão como um problema de reconhecimento de uma linguagem.

Concluindo esta breve revisão de terminologia mencionaremos apenas que um tratamento detalhado destes aspectos e da complexidade de problemas algorítmicos é apresentado por GARREY e JOHNSON (18) e HOPCROFT e ULLMAN (25), entre outros. Uma abordagem rigorosa e pioneira, pois baseada apenas em problemas de decisão, foi desenvolvida por SZWARCFITER (43).

### II.3. A COMPLEXIDADE DE ALGORITMOS

A teoria da complexidade computacional trata dos aspectos quantitativos da solução de problemas algorítmicos. Em geral existem vários algoritmos para a solução de um dado problema. A cada um destes algoritmos podem ser associadas certas medidas da sua eficiência de utilização dos recursos computacionais. Avalia-se esta eficiência exprimindo a medida de cada uma das grandezas consideradas como significativas, como uma função do tamanho da entrada. Algumas das grandezas mais importantes são o número de passos executados pelo algoritmo, o espaço de memória usado e, para algoritmos implementados em hardware, o tamanho e a profundidade dos circuitos. Dentre os vários critérios de eficiência citados acima, o mais importante é o tempo de processamento, em função do tamanho da entrada.

A obtenção das expressões analíticas para a eficiência de tempo e espaço dos algoritmos envolve três aspectos: um modelo matemático da computação, uma medida do tamanho da entrada e saída do algoritmo, e características dos dados em relação ao algoritmo: pior caso, caso médio e melhor caso. Os modelos de computação mais usados são a máquina de Turing, introduzida por A. Turing em 1936, e o modelo RAM - "Random Access Machine" - introduzido por COOK e RECKHOW (9) em 1973. Para a análise de algoritmos prefere-se o modelo RAM por modelar mais naturalmente os computadores digitais e por induzir uma hierarquia de classes de complexidade mais fina do que aquela proporcionada pela máquina de Turing.

A conceituação mais clara e didática da complexidade de tempo dos algoritmos, em relação ao modelo RAM, encontra-se

em SZWARCFITER (43) e será apresentada, em resumo, a seguir.

Seja  $P$  um programa que implementa um algoritmo  $\alpha$  e seja  $E$  a sua entrada.

DEFINIÇÃO II.1 - Passo de um algoritmo  $\alpha$  é a computação de uma instrução do programa  $P$  que o implementa.

DEFINIÇÃO II.2 - Complexidade local do algoritmo  $\alpha$  é o número total de passos necessários para a computação completa de  $P$ , para uma certa entrada  $E$ .

DEFINIÇÃO II.3 - Complexidade (local) assintótica de um algoritmo é um limite superior da sua complexidade local, para uma certa entrada suficientemente grande.

DEFINIÇÃO II.4 - Complexidade de pior caso (ou simplesmente complexidade) de um algoritmo, é o valor máximo dentre todas as suas complexidades assintóticas, para entradas suficientemente grandes.

DEFINIÇÃO II.5 - Sejam  $f(n)$  e  $g(n)$  funções de inteiros em reais; escrevemos  $f(n) = O(g(n))$ , o que se lê " $f(n)$  é da ordem de  $g(n)$ ", se existirem constantes positivos  $C$  e  $N_0$  tais que  $f(n) \leq C \cdot g(n)$  para todo  $n > N_0$ .

A notação apresentada na definição (II.5) é usada para exprimir a complexidade de algoritmos.

A medida que os algoritmos para muitos problemas importantes foram sendo analisados e aperfeiçoados, começaram a

emergir certos padrões surpreendentes. Em primeiro lugar verificou-se que a complexidade de muitos algoritmos é polinomial, isto é, limitada superiormente por um polinômio em  $n$ , onde  $n$  é o tamanho da instância do problema. Curiosamente, estes polinômios são de baixo grau v.g.  $n$ ,  $n^{\frac{3}{2}}$ ,  $n^2$ ,  $n^3$ . Em segundo lugar constatou-se que existe uma classe constituída por muitos problemas de grande importância prática para os quais não se conhecem algoritmos de complexidade polinomial, mas que sempre podem ser resolvidos por algoritmos exponenciais i.e. algoritmos cuja complexidade é uma função exponencial de  $n$ . Assim, o estudo da complexidade dos algoritmos acabou por revelar algo sobre a própria natureza dos problemas, permitindo classificá-los em tratáveis quando possuem algoritmos de complexidade polinomial, e intratáveis em caso contrário.

Neste sentido, o caso geral do problema de horário de aulas é intratável, como exposto na seção II.5.

#### II.4. PROBLEMAS NP-COMPLETO

A preocupação com a eficiência dos algoritmos praticamente acompanha o desenvolvimento da computação. Como cita COOK (8) pp. 402, já em 1953 Von Neumann destacava a distinção entre algoritmos polinomiais e exponenciais. Segundo GAREY e JOHNSON (18) pp. 7, a natureza fundamental desta distinção em face da natureza intrínseca dos problemas, foi discutida primeiramente por A. Cobham em 1964 e J. Edmonds em 1965. Edmonds foi o primeiro a qualificar como "bons" os algoritmos polinomiais. Esta é a caracterização aceita atualmente, sendo considerados eficientes os algoritmos polinomiais.

Ao longo dos anos foi possível ir construindo algoritmos cada vez mais eficientes para muitos problemas. Uma excelente ilustração deste fato é apresentada por TARJAN (44) pp. 475-483 onde mostra a evolução dos algoritmos e das suas complexidades para dez problemas importantes na prática. Apenas como exemplo, citamos o algoritmo de planaridade. Se baseado diretamente no teorema de Kuratowski, teria complexidade  $O(n^6)$ ; em 1963 foi proposto um algoritmo que, implementado em 1969 apresentava um desempenho de  $O(n^3)$ . Usando estruturas de dados mais adequadas, em 1972 foi obtido um algoritmo  $O(n \log n)$  o qual deu lugar a um algoritmo  $O(n)$  em 1974.

O exemplo acima é uma amostra do sucesso obtido na construção de algoritmos eficientes. Entretanto, há uma grande quantidade de problemas, também de extrema importância prática, para os quais ainda não se conseguiu construir algoritmos polinomiais, nem tampouco provar que tais algoritmos não possam existir. Os melhores algoritmos conhecidos para estes proble

mas são exponenciais. A consequência positiva deste aparente fracasso foi a identificação de uma nova classe de problemas, denominada NP e o reconhecimento da sua maior dificuldade intrínseca. A característica fundamental desta classe é a possibilidade de verificar, por um algoritmo polinomial, se uma solução proposta realmente satisfaz as condições que são exigidas para qualificar-se como uma solução da instância dada do problema - isto constitui o "passo de reconhecimento" referido na definição (II.7).

As definições apresentadas a seguir, e que conduzem à definição da classe NP-completo, também são baseadas em SZWARCFITER (43).

DEFINIÇÃO II.6 - A classe P é constituída pelos problemas de decisão que admitem algoritmo polinomial para a sua solução.

DEFINIÇÃO II.7 - A classe NP é constituída por todos os problemas de decisão tais que existe uma justificativa à resposta SIM cujo passo de reconhecimento pode ser realizado por um algoritmo polinomial no tamanho da entrada do problema considerado.

A relação entre as classes P e NP é um tópico extremamente importante em Ciência da Computação. Sabe-se que  $P \subseteq NP$ . Entretanto, é um problema em aberto se  $P = NP$ . Todas as evidências sugerem que  $P \neq NP$ . Um argumento a favor desta conjectura é que a classe NP incorpora um grande número de problemas para os quais inúmeros pesquisadores já desenvolveram esforços para construir algoritmos eficientes. Entretanto,



não se conseguiu construir algoritmos polinomiais para qual quer um destes problemas.

Seja  $P(D,Q)$  um problema de decisão onde  $D$  representa o conjunto de todas as instâncias de  $P$ , e  $Q$  é a questão correspondente.

DEFINIÇÃO II.8 - Uma transformação polinomial de um problema de decisão  $P_1(D_1,Q_1)$  no problema  $P_2(D_2,Q_2)$  é uma função  $f:D_1 \rightarrow D_2$  tal que:

- i)  $f$  pode ser computada por um algoritmo polinomial
- ii) para toda instância  $I \in D_1$  do problema  $P_1$  tem-se que  $P_1$  possui resposta SIM, se e somente se  $P_2$  também possui resposta SIM para a instância  $f(I)$ .

DEFINIÇÃO II.9 - Um problema de decisão  $P$  é denominado NP-completo quando as seguintes condições forem ambas satisfeitas

- i)  $P \in NP$
- ii) todo problema de decisão  $P' \in NP$  pode ser transformado polinomialmente no problema  $P$ .

A condição (ii) da definição (II.9) implica que, se um problema NP-completo  $P$  pode ser resolvido em tempo polinomial, então todo problema de NP admite também um algoritmo polinomial. A classe NP-completo corresponde aos problemas de maior dificuldade dentre os pertencentes a NP.

Caso somente a condição (ii) da definição (II.9) seja considerada, não importando se (i) é satisfeito ou não, o problema é denominado NP-difícil. Conseqüentemente, a dificuldade intrínseca de um problema NP-difícil é não menor do que a de um problema NP-completo.

## II.5. A COMPLEXIDADE DO PROBLEMA DE HORÁRIO

Em fins de 1976 foram publicados os resultados obtidos por EVEN, ITAI e SHAMIR (16) sobre a complexidade do problema de horário de aulas. Os autores demonstram que o problema de decisão constituído pela versão geral do problema de horário é NP-completo. Outrossim mostram que um caso particular do problema de horário pode ser resolvido em tempo polinomial. Apresentamos a seguir sua formulação do problema de horário e os principais resultados.

DEFINIÇÃO II.10 - Sendo dados

- i) um conjunto finito  $H$  de horas da semana;
- ii) uma coleção  $\{P_1, P_2, \dots, P_n\}$ , onde  $P_i \subseteq H$ ; (isto é, existem  $n$  professores e  $P_i$  é o conjunto das horas em que o  $i$ -ésimo professor pode lecionar);
- iii) uma coleção  $\{C_1, C_2, \dots, C_m\}$ , onde  $C_j \subseteq H$ ; (isto é, existem  $m$  classes e  $C_j$  é o conjunto das horas em que a  $j$ -ésima classe pode ter aula);
- iv) uma matriz  $R$  de inteiros não negativos, de  $m$  linhas e  $n$  colunas; (o elemento  $r_{ij}$  exprime o número de horas de aula que o  $i$ -ésimo professor deve ministrar para a  $j$ -ésima classe).

O problema de horário consiste em decidir se existe uma função horário

$$f(i, j, h) : \{1, \dots, n\} \times \{1, \dots, m\} \times H \rightarrow \{0, 1\}$$

(onde  $f(i, j, h) = 1$  se e somente se o professor  $i$  leciona para a classe  $j$  durante a hora  $h$ ) satisfazendo:

- a)  $f(i,j,h) = 1$  implica  $h \in P_i \cap C_j$ ;
- b)  $\sum_{h \in H} f(i,j,h) = r_{i,j}$ , para todo  $1 \leq i \leq n$  e  $1 \leq j \leq m$ ;
- c)  $\sum_{i=1}^n f(i,j,h) \leq 1$ , para todo  $1 \leq j \leq m$  e  $h \in H$ ;
- d)  $\sum_{j=1}^m f(i,j,h) \leq 1$ , para todo  $1 \leq i \leq n$  e  $h \in H$ .

Na definição acima, a condição (a) assegura que a aula só se realiza quando o professor e a classe estão disponíveis; (b) assegura que o número de aulas ministradas pelo professor  $i$  para a classe  $j$  tem o valor estipulado por  $r_{i,j}$ ; (c) assegura que cada classe encontra um só professor em cada aula e (d) impõe que em cada hora cada professor leciona para não mais do que uma classe.

TEOREMA II.1 - É NP-completo o problema de horário da definição (II.10).

A demonstração foi realizada por EVEN et.al. (16) mediante a transformação de uma versão mais simples do problema acima num problema de 3-SATISFABILIDADE, que foi demonstrado ser NP-completo por S. Cook em 1971.

DEFINIÇÃO II.11 - Um problema de horário é sem restrições de horário se para todo  $1 \leq i \leq n$  e  $1 \leq j \leq m$ ,  $P_i = C_j = H$ ; é dito aparentemente viável se nem os professores nem as classes estão sobrecarregadas, isto é,

- i) para todo  $1 \leq i \leq n$  ,  $\sum_{h=1}^m r_{ij} \leq |H|$
- ii) para todo  $1 \leq j \leq m$  ,  $\sum_{i=1}^n r_{ij} \leq |H|$

Neste mesmo artigo, EVEN et.al.(16) mostram que a versão particular do problema de horário, especificada na definição (II.11), sempre tem solução e esta solução sempre pode ser obtida por um algoritmo de associação em grafos bipartites, que é polinomial.

## II.6. O ALGORITMO DE BACKTRACK

Na falta de um processo melhor, instâncias de tamanho razoável de certos problemas combinatórios podem ser resolvidas por um processo equivalente a uma busca exaustiva de todas as possibilidades, porém organizada de modo a ser bem mais eficiente do que o exame exaustivo. Esta busca pode ser estruturada sob a forma de uma árvore que é percorrida em profundidade e foi chamada de backtrack por D.H. LEHMER na década de 50. Inicialmente descrevemos o método e em seguida apresentamos alguns fatos sobre o desempenho destes algoritmos.

### II.6.1. DESCRIÇÃO

Para aplicar o método de backtrack, é preciso formular o problema de modo que as soluções procuradas possam ser

expressas por n-tuplas  $(x_1, \dots, x_n)$  onde cada  $x_i$  é selecionado de um conjunto finito  $X_i$  e definir uma função

$$f_n: X_1 \times X_2 \times \dots \times X_n \rightarrow \{\text{FALSO}, \text{VERDADEIRO}\}$$

tal que  $f_n(x_1, \dots, x_n) = \text{VERDADEIRO}$  se e somente se  $(x_1, \dots, x_n)$  é uma solução da instância dada do problema.

Seja  $m_i$  a cardinalidade de  $X_i$ . Existem  $m = m_1 \cdot m_2 \cdot \dots \cdot m_n$  n-tuplas, das quais em geral apenas umas poucas satisfazem  $f_n$ . A abordagem por força bruta consistiria em formar todas estas n-tuplas e ir testando e separando como soluções aquelas que satisfazem  $f_n$ . O algoritmo de backtrack propõe-se chegar ao mesmo resultado porém gastando substancialmente menos esforço.

A idéia básica é ir construindo a n-tupla componente a componente e definir funções  $f_k(x_1, \dots, x_k)$ ,  $0 \leq k \leq n$ , para descobrir se a k-tupla construída até este ponto, chamada solução parcial, é tal que não pode ser estendida para formar uma n-tupla que satisfaz  $f_n$ . Isto é, procura-se identificar e rejeitar, o quanto antes, as não soluções encontradas enquanto se busca as soluções. A grande virtude deste processo reside no seguinte: se for possível assegurar que a solução parcial  $(x_1, \dots, x_k)$  é tal que  $(x_1, \dots, x_k, x_{k+1}, \dots, x_n)$  nunca poderá vir a ser uma solução, para quaisquer valores de  $x_{k+1}, \dots, x_n$  então podem ser excluídas as  $m_{k+1} \cdot m_{k+2} \cdot \dots \cdot m_n$  n-tuplas que tem  $x_1, \dots, x_k$  como os k primeiros componentes, sem sequer se dar ao trabalho de construí-las a partir de  $x_1, \dots, x_k$ .

Assim, além de construir  $f_n$  é preciso inventar funções intermediárias  $f_k(x_1, \dots, x_k)$ ,  $0 \leq k < n$ , tais que, se  $f_k(x_1, \dots, x_k) = \text{FALSO}$  então  $(x_1, \dots, x_k)$  certamente não po-

de ser estendida para formas  $n$ -tuplas que sejam soluções da instância dada do problema. Segundo KNUTH (28), estas funções intermediárias devem ser tais que

$$f_{k+1}(x_1, \dots, x_k, x_{k+1}) \text{ implica } f_k(x_1, \dots, x_k)$$

para  $0 \leq k < n$ . Isto é, se  $f_k(x_1, \dots, x_k) = \text{FALSO}$  então nenhuma solução parcial  $(x_1, \dots, x_k, x_{k+1})$  poderá satisfazer  $f_{k+1}$  e, por indução, nenhuma  $n$ -tupla  $(x_1, \dots, x_k, x_{k+1}, \dots, x_n)$  poderá satisfazer  $f_n$ .

O método de backtrack funciona assim. Começamos ( $k=0$ ) com a tupla vazia  $()$ , escolhemos um primeiro elemento de  $X_1$  como valor de  $x_1$  e avaliamos  $f_1(x_1)$ . Se  $f_1(x_1) = \text{FALSO}$  então  $\{x_1\} \times X_2 \times \dots \times X_n$  não contém  $n$ -tupla que satisfaça  $f_n$ . Assim escolhemos um segundo elemento de  $X_1$  e avaliamos  $f_1(x_1)$ , até que seja encontrado um elemento tal que  $f_1(x_1) = \text{VERDADEIRO}$ . Se um tal elemento não existe em  $X_1$  então a instância dada do problema não tem solução. Encontrado um valor para  $x_1$ , os elementos de  $X_2$  são examinados até obter-se um valor para  $x_2$  tal que  $f_2(x_1, x_2) = \text{VERDADEIRO}$ . Se tal elemento foi encontrado então passamos a  $X_3$  em busca de um elemento que será o valor de  $x_3$  tal que  $f_3(x_1, x_2, x_3) = \text{VERDADEIRO}$  e etc. Entretanto, se  $f_k(x_1, \dots, x_k) = \text{VERDADEIRO}$  mas é impossível escolher um elemento de  $X_{k+1}$  como valor de  $x_{k+1}$  que permite estender  $(x_1, \dots, x_k)$  para  $(x_1, \dots, x_k, x_{k+1})$  com  $f_{k+1}(x_1, \dots, x_k, x_{k+1}) = \text{VERDADEIRO}$  então torna-se necessário regredir até o conjunto  $X_k$  e escolher um novo valor para o  $k$ -ésimo componente. Este ato de "back track" deu o nome ao método. Em seguida, recomeça-se o exame dos elementos de  $X_{k+1}$ , desde o primeiro, até encontrar um valor para  $x_{k+1}$  que permita estender a solução parcial com  $k$  com

ponentes para uma solução parcial com  $(k+1)$  componentes e tal que  $f_{k+1}(x_1, \dots, x_k, x_{k+1}) = \text{VERDADEIRO}$ . Cabe observar que os  $k$  primeiros componentes já selecionados em geral restringem a um subconjunto de  $X_{k+1}$  a escolha do valor de  $x_{k+1}$ . Muitas vezes pode ser necessário regredir várias etapas até que se obtenha uma solução parcial que novamente possa ser estendida. Por fim, há de ocorrer uma dentre as duas situações seguintes. Ou não se encontra nenhuma  $n$ -tupla que satisfaz  $f_n$ , após ter regredido e experimentado todos os elementos de  $X_1$ , caso em que a instância dada do problema não tem solução; ou então obtém-se uma ou várias soluções, que são  $n$ -tuplas com as características desejadas, isto é, tais que  $f_n(x_1, \dots, x_n) = \text{VERDADEIRO}$ .

Cada  $k$ -tupla ( $0 \leq k \leq n$ ) gerada no processo de busca é denominada estado do problema, satisfaça ou não a função  $f_k$ . O conjunto destas  $k$ -tuplas constitui o chamado espaço de estados do problema. Este espaço de estados pode ser visto como um grafo estruturado em árvore enraizada e rotulada, que é percorrida em profundidade ("depth-first") à medida que os estados vão sendo gerados e testados em busca de soluções. Os vértices correspondem aos estados e as arestas, aos valores escolhidos para os componentes das  $k$ -tuplas. A árvore assim obtida é a chamada árvore de backtrack ou árvore do espaço de estados. Como as soluções são  $n$ -tuplas que satisfazem  $f_n$  segue-se que, se existirem, estas soluções estarão no nível  $n$  da árvore.

Na descrição acima admitimos que as soluções procuradas são tuplas com uma quantidade fixa de componentes e que satisfazem uma determinada função. Entretanto, o método também pode ser aplicado a problemas onde as soluções podem ser formuladas como tuplas com quantidade variável de componentes, como

ilustrado por HOROWITZ e SAHNI (26) às pp. 327-333. Neste caso, as soluções podem ser encontradas em vários níveis da árvore.

### II.6.2. O DESEMPENHO DOS ALGORITMOS DE BACKTRACK

Inicialmente é preciso reconhecer que é exponencial a complexidade destes algoritmos. Entretanto, para certas instâncias o algoritmo pode gerar  $O(n)$  estados enquanto para outros pode vir a gerar quase todos os vértices da árvore do espaço de estados. Por outro lado, segundo GAREY e JOHNSON (18), às vezes é possível reduzir esta complexidade por uma escolha mais engenhosa dos objetos sobre os quais se fará a busca.

A eficiência dos algoritmos de backtrack depende muito de quatro fatores:

- o tempo necessário para gerar o próximo componente da tupla;
- a quantidade de elementos em  $X_k$ ;
- o tempo gasto para avaliar as funções  $f_k$  para cada um dos estados;
- a quantidade de valores  $x_k$  que são compatíveis com os valores previamente selecionados para  $x_1, \dots, x_{k-1}$ .

As funções  $f_k$  são consideradas boas quando reduzem substancialmente a quantidade de estados; o seu projeto porém é um compromisso pois as mais eficazes demandam, em geral, mais tempo para ser avaliadas. Muitas abordagens procuram reduzir o tempo total de processamento mediante a redução do número de estados gerados na busca. As técnicas clássicas de obter esta redução são a preclusão, a fusão de subárvores e a reordenação. Uma descrição pioneira é apresentada por GOLOMB e BAUMERT (19).



A preclusão consiste em identificar no problema considerado alguma relação entre os componentes das soluções parciais que possa ser usada para impedir que sequer venham a ser gerados valores para  $x_k$  que são incompatíveis com os valores já selecionados para  $x_1, \dots, x_{k-1}$ . A fusão de subárvores procura identificar simetrias no problema para reduzir a uma fração o número de casos a examinar, obtendo-se as demais soluções por simetria. A reordenação é apresentada na literatura como um princípio geral de busca eficiente. A idéia é a seguinte: em muitos problemas os conjuntos  $X_i$  podem ser tomados em qualquer ordem e assim é mais eficiente selecionar o próximo componente da tupla daquele conjunto que tem menos elementos. A seguir resumidos os resultados de duas pesquisas recentes sobre estes temas, publicados em 1980 e 1983.

Em 1980 HARALICK e ELLIOTT (23) publicaram um excelente estudo sobre algumas técnicas de aumentar a eficiência dos algoritmos de backtrack. Os autores resumem a filosofia deste trabalho em dois princípios:

- "Remember what you have done to avoid repeating the same mistake"

- "To succeed, try first where you are most likely to fail".

Antes de descrever os seus resultados convém apresentar sua terminologia peculiar. Assim, existem  $N$  unidades e a cada unidade corresponde um conjunto de  $M$  possíveis valores ou rótulos. O problema examinado consiste em atribuir um rótulo a cada unidade de modo a satisfazer um conjunto de restrições. Um conjunto  $R$  de pares (unidade, rótulo) satisfazendo um dado conjunto de restrições é dito consistente; dois pares que podem pertencer a  $R$  são ditos consistentes. As unidades às quais já foram

atribuídos rótulos são as unidades passadas, a unidade corrente é aquela cujo rótulo está sendo atribuído e são futuras as unidades às quais ainda falta atribuir um rótulo. É utilizada uma tabela para cada nível da árvore de busca, que registra os rótulos ainda possíveis para cada unidade. Para ilustrar o desenvolvimento do seu trabalho usam o problema das  $N$  rainhas num tabuleiro de xadrez.

Na primeira parte do trabalho discutem cinco técnicas de utilização da tabela para memorizar e recordar os resultados dos testes de consistência que são realizados ao longo do processo de busca, para evitar repeti-los no futuro. Estas técnicas são por eles denominadas "look ahead", "partial look ahead", "forward checking", "backchecking" e "backmarking".

"Look ahead" consiste em assegurar que cada unidade futura tem pelo menos um rótulo compatível com os rótulos das unidades passadas e corrente, e pelo menos um rótulo compatível com algum dos rótulos possíveis das unidades futuras. Como tal procedimento não pode memorizar os resultados da maioria dos testes de consistência entre unidades futuras para usar em etapas subsequentes de "look ahead", dá-se preferência ao "partial look ahead", que testa seletivamente as unidades futuras entre si. "Forward checking" tenta evidenciar uma inconsistência o quanto antes, testando se existe alguma unidade futura sem rótulo consistente com o par corrente. Se cada unidade futura tem rótulos consistentes, memoriza-se isto copiando os respectivos pares (unidade, rótulo) na tabela do próximo nível da árvore. "Backchecking" é semelhante ao "forward checking": os testes de consistência são realizados entre o par (unidade, rótulo) corrente e os pares passados, mas não com pares futuros. Fi

nalmente, "backmarking" é "backchecking" com uma característica adicional: após regredir do nível  $u$  ao nível  $v$  e avançar novamente, testa-se a consistência dos rótulos da unidade do nível  $u$  apenas com os pares do nível  $v$  até  $u-1$ , isto é, somente com aqueles pares cujos rótulos podem ter mudado.

A seguir comparam experimentalmente o desempenho dos algoritmos de backtrack simples e de backtrack enriquecido com cada uma das cinco técnicas citadas, quando aplicados ao problema de satisfação de restrições aleatórias. A partir destas experiências concluem que o backtrack simples é o menos eficiente na maioria dos casos e que "backmarking" e "forward checking" são os mais eficientes para os problemas considerados. As medidas de complexidade usadas baseiam-se no número: de testes de consistência efetuados, de consultas às tabelas e de vértices gerados na busca de todas as soluções.

Na segunda parte de seu trabalho, HARALICK e ELLIOTT (23) examinam mais duas aplicações do princípio de que quanto mais cedo for detectada alguma inconsistência, tanto melhor: é o chamado princípio de "fail first". Em certo sentido este princípio já foi aplicado em "look ahead" e "forward checking". A primeira das duas novas aplicações refere-se à ordem em que são feitos os testes de consistências entre a unidade corrente e as unidades passadas. Pelo citado princípio, devemos fazer primeiro os testes que tem maior probabilidade de apontar inconsistências pois com isto não precisaremos fazer os demais testes. Para o problema das  $N$  rainhas isto significa que os testes de consistência do rótulo da unidade corrente com os rótulos das unidades passadas devem ser feitas por níveis decrescentes.

A segunda aplicação adicional deste princípio propõe-se determinar a ordem em que as unidades devem ser tomadas para atribuição dos rótulos. Pelo princípio citado a melhor ordem é aquela que toma como a próxima unidade a que tem menos rótulos ainda disponíveis. Experimentalmente constatam que a observância deste princípio tem o efeito de reduzir o número de testes de consistência a realizar por problema, em relação a uma ordenação aleatória das unidades, além de reduzir a variância em relação às médias obtidas. Finalmente, provam analiticamente que esta ordenação minimiza os valores esperados dos comprimentos dos caminhos com origem na raiz da árvore de busca. Esta demonstração é baseada em duas hipóteses: a probabilidade de sucesso num teste de consistência de um dado rótulo de uma unidade depende apenas do nível em que é usado e independente de quais são as unidades passadas. Seja  $P_i(n)$  a probabilidade de que algum rótulo da unidade  $u$  satisfaça os testes de consistência quando no nível  $i$ , em relação a uma unidade arbitrária, e seja  $P(u)$  a mesma probabilidade referida ao nível 1 da árvore; então a segunda hipótese é que

$$P_i(u) = \alpha^{i-1} \cdot P(u) \quad \text{para } 0 < \alpha \leq 1.$$

Concluem que "forward checking" combinado com ordenação ótima das unidades conduz a algoritmos de busca mais eficientes do que "look ahead" ou "backmarking".

Por último, em fins de 1983 PURDOM e BROWN (38) publicaram uma análise do desempenho do algoritmo de backtrack aplicado à obtenção de todas as soluções que satisfazem fórmulas booleanas na forma normal conjuntiva com  $t$  termos, e  $s$  literais por termo, sobre um conjunto de  $v$  variáveis. Supõe-se que

$t = v^\alpha$  para algum  $\alpha$ . Os objetivos sobre os quais se faz a busca são as  $v$  variáveis, donde árvores de backtrack com níveis 0 a  $v$ . As funções intermediárias  $f_k$  são tomadas como conjunções daqueles termos que só envolvem as  $(k-1)$  variáveis definidas em níveis anteriores e a variável a ser selecionada para o nível  $k$ .

Sendo as variáveis booleanas, cada vértice tem 0, 1 ou 2 filhos. Os com zero filhos correspondem a vértices onde a função intermediária  $f_k$  tem valor FALSO para qualquer valor escolhido para a variável corrente. Os vértices com um filho indicam que só existe um valor que satisfaz a função intermediária e os vértices com dois filhos, chamados vértices binários, indicam que ambos os valores da variável são aceitáveis, pelo menos por ora. PURDOM e BROWN (38) propõem-se a determinar a quantidade média de vértices binários nas árvores de backtrack obtidas para este problema quando a próxima variável selecionada para inclusão na árvore é aquela que tem a menor quantidade (dentre 1 ou 2) de valores que satisfazem a função intermediária correspondente ao nível considerado; o desempate é feito pelo subscrito da variável.

Após uma análise muito extensa concluem que as árvores obtidas, quando as variáveis são selecionadas na ordem descrita, possuem em média  $N$  vértices binários onde:

i) para  $s/2 < \alpha < s-1$  e  $s \geq 3$

$$N = \exp\left\{\theta\left(v^{(s-\alpha-1)/(s-2)}\right)\right\}$$

ii) enquanto, para  $1 < \alpha \leq s/2$  e  $s \geq 3$

$$\exp\left\{\theta\left(v^{(s-\alpha-1)/(s-2)}\right)\right\} \leq N \leq \exp\left\{\theta\left\{\left(v^{(s-\alpha-1)/(s-2)}\right)_{(1nv)}^{(s-1)/(s-2)}\right\}\right\}$$

Por outro lado, BROWN e PURDOM (7) estabeleceram que o valor esperado para o número de vértices numa árvore de backtrack comum (isto é, sem nenhuma ordenação especial das variáveis), para o mesmo problema é

$$N = \exp\left\{\theta\left(v^{(s-\alpha)/(s-1)}\right)\right\}$$

Assim, concluem que a mencionada ordenação das variáveis conduz a uma redução do valor do expoente e portanto resulta numa redução drástica do tempo de processamento do algoritmo de back track.

## II.7. ALGORITMOS DE APROXIMAÇÃO

A classe NP-difícil contém muitos problemas de otimização combinatoria importantes na prática, donde a necessidade de resolver instâncias grandes destes problemas num tempo razoável. Isto estimulou a pesquisa de algoritmos heurísticos de baixa complexidade polinomial e que produzem soluções aproximadas. Uma abordagem, que deu lugar aos algoritmos de aproximação consiste em relaxar o sentido de "resolver" um problema de otimização: ao invés de exigir que o algoritmo produza sempre uma solução ótima, exige-se apenas que produza uma solução viável, próxima da ótima.

Em geral os algoritmos heurísticos são considerados imunes a uma análise precisa da qualidade dos seus resultados. Recentemente verificou-se que em alguns casos é possível demonstrar rigorosamente que os resultados gerados por um algoritmo heurístico para um dado problema nunca diferem da solu-

ção ótima por mais do que uma constante ou uma percentagem, conhecida a priori. Isto é, oferece-se uma garantia para qualidade dos resultados aproximados.

Seja  $I$  uma instância de um problema de otimização combinatória  $P$  e  $A$  um algoritmo. Designaremos por  $OPT(I)$  o valor da solução ótima e por  $A(I)$  o valor da solução viável gerada por  $A$  para a mesma instância  $I$  de  $P$ . Admitimos que  $OPT(I) > 0$  e  $A(I) > 0$  para toda instância  $I$  de  $P$ . As próximas definições são de GAREY e JOHNSON (18).

DEFINIÇÃO II.12 -  $A$  é um algoritmo de aproximação para  $P$ , se dada qualquer instância  $I$  de  $P$ ,  $A$  gera uma solução viável para a instância considerada. Se  $A(I) = OPT(I)$  então  $A$  é um algoritmo de otimização para o problema  $P$ .

Os problemas diferem na forma como podem ser aproximados, o que dá lugar a vários tipos de garantias de desempenho para os algoritmos de aproximação.

DEFINIÇÃO II.13 -  $A$  é um algoritmo de aproximação absoluta para  $P$  se, e somente se, para toda instância  $I$  de  $P$

$$|OPT(I) - A(I)| \leq k$$

para alguma constante  $k$ .

DEFINIÇÃO II.14 - Um esquema de aproximação para o problema de otimização  $P$  é um algoritmo  $A$  que recebe como entrada uma instância  $I$  de  $P$  e uma constante  $\epsilon > 0$ , que prescreve a precisão desejada, e gera uma solução viável  $A_\epsilon(I)$  tal que

$$|\text{OPT}(I) - A_{\epsilon}(I)| / \text{OPT}(I) \leq \epsilon$$

Usamos o termo "esquema" na definição (II.14) pois  $A$ , na realidade, oferece um conjunto de algoritmos de aproximação para  $P$ , um para cada valor fixo de  $\epsilon$ .

DEFINIÇÃO II.15 - Um esquema de aproximação polinomial é um esquema de aproximação tal que, fixado um  $\epsilon > 0$ , a sua complexidade é polinomial no tamanho da instância  $I$  de entrada.

DEFINIÇÃO II.16 - Um esquema de aproximação cuja complexidade é polinomial no tamanho da instância  $I$  e do valor de  $1/\epsilon$ , é um esquema de aproximação plenamente polinomial.

O que pode, e o que não pode ser conseguido pelo algoritmos de aproximação é examinado entre outros por HOROWITZ e SAHNI (26), GAREY e JOHNSON (18), PAPADIMITRIOU e STEIGLITZ (37). Estes autores mostram que os melhores resultados que podem ser garantidos por tais algoritmos não dependem tanto da nossa engenhosidade em construí-los mas, primordialmente, decorrem da própria natureza de cada problema. Assim, por exemplo, se  $P \neq NP$  então simplesmente não pode existir esquema de aproximação polinomial para o problema de coloração mínima dos vértices de um grafo, comentado a seguir.



## II.8. COLORAÇÃO DE GRAFOS: MÍNIMA E APROXIMADA

É útil examinar agora a natureza do problema de coloração de grafos, quer mínima, quer aproximada. Em primeiro lugar, porque o problema de horário pode ser formulado com naturalidade, pelo menos nos seus aspectos essenciais como um problema de coloração dos vértices de um grafo (conforme seção I.6). Em segundo lugar, porque ilustra o tipo de limitação que a natureza dos problemas impõe aos algoritmos de aproximação.

Seja  $A(G)$  o número de cores usadas por um algoritmo  $A$  para colorir os vértices de um grafo  $G$  e  $x(G)$  o número cromático de  $G$ , isto é, o número mínimo de cores que são necessárias para colorí-los.

Vejamos a coloração mínima. Em 1972 R.M. KARP provou que é NP-completo o problema de decisão associado ao problema geral de coloração dos vértices de um grafo, como documentado por GAREY e JOHNSON (18). Isto significa que, se  $P \neq NP$ , então não pode existir algoritmo polinomial  $A$  tal que se possa garantir que  $A(G) = x(G)$ , para qualquer grafo  $G$ . Diante destes fatos poderíamos tentar construir um algoritmo polinomial que, embora não garantindo a obtenção da coloração mínima, seja capaz de produzir coloração tal que se possa garantir que  $A(G)/x(G)$  nunca exceda um valor bem próximo de 1. Entretanto, nem isto pode ser realizado em face do seguinte resultado de GAREY e JOHNSON (17).

TEOREMA II.2 - Se, para alguma constante  $r < 2$  e constante  $d$ , existir um algoritmo polinomial  $A$  para colorir os vértices de um grafo  $G$  com  $A(G)$  cores e tal que

$$A(G) \leq r \cdot x(G) + d$$

então existe um algoritmo polinomial  $\bar{A}$  tal que  $\bar{A}(G) = x(G)$ .

Este teorema mostra que, se  $P \neq NP$ , então também não pode existir esquema de aproximação polinomial para o qual se possa garantir que sempre produzirá boas colorações aproximadas.

## CAPÍTULO III - UM ALGORITMO PARA O PROBLEMA DE HORÁRIO

### III.1. INTRODUÇÃO

No presente capítulo apresentamos um desenvolvimento teórico para a construção e implementação de um algoritmo para a construção de horários de aula de Universidades. Constitui a parte principal desta pesquisa, do ponto de vista teórico.

Em face da natureza do problema, optamos por um algoritmo de backtrack, embora os algoritmos heurísticos sempre sejam importantes para a solução de problemas associados a problemas de decisão NP-completo. A análise da árvore do espaço de estados gerada pelo algoritmo de backtrack permitiu obter alguns resultados teóricos sobre a redução do número de estados gerados na busca. Estes resultados teóricos também podem ser úteis no desenvolvimento de algoritmos heurísticos ou então de heurísticas para melhorar o desempenho de algoritmos de backtrack.

Em linhas gerais, este capítulo compõe-se de mais sete seções que podem ser agrupadas em três partes. A primeira, constituída pelas seções III.2 e III.3, põe uma terminologia e um modelo para a formulação do problema de horário visando um tratamento preciso do problema e do algoritmo proposto. A segunda parte, seções III.4 e III.5, trata da escolha do método de solução e da descrição do algoritmo de backtrack aplicado ao problema de horário. Finalmente as seções III.6, III.7 e III.8 apresentam os principais resultados teóricos referentes à cardinalidade do espaço de estados e à sua redu

ção, quer por reordenação quer por considerações heurísticas.

### III.2. DEFINIÇÕES PRELIMINARES

A seguir reunimos os termos geralmente usados no âmbito da administração acadêmica no contexto do problema de horário de aulas, com a aceção que terão no restante do presente trabalho. Estes termos serão usados na formulação do problema de horário e do algoritmo proposto para resolvê-lo, bem como na definição de outros termos, relativos ao algoritmo e à organização dos seus dados de entrada. Admitiremos que o ciclo do horário é constituído por uma semana. Apresentamos também alguns comentários referentes às definições e uma breve discussão da importante questão das salas de aula.

#### III.2.1. DEFINIÇÕES

DEFINIÇÃO III.1 - Corpo discente de uma disciplina é o conjunto de todos os alunos inscritos nesta disciplina num dado período letivo.

DEFINIÇÃO III.2 - Carga horária semanal de uma disciplina é o número nominal de horas de aula que a Universidade deve proporcionar em cada semana a cada aluno inscrito nesta disciplina.

DEFINIÇÃO III.3 - Configuração da carga horária semanal de uma disciplina é cada um dos possíveis desdobramentos de sua

carga horária semanal em uma ou mais aulas, tendo cada aula uma duração nominal especificada.

DEFINIÇÃO III.4 - Curso de uma disciplina é a sequência das aulas desta disciplina ao longo do período letivo, ministrada pelo mesmo professor, para os mesmos alunos, visando cumprir um programa estabelecido de atividades didáticas. Em cada semana a soma das durações destas aulas perfaz a carga horária semanal da disciplina.

DEFINIÇÃO III.5 - Classe de uma disciplina é um subconjunto do corpo discente desta disciplina, constituído por todos os alunos que pretendem participar de um mesmo curso desta disciplina. A classe constitui o menor grupo indivisível de alunos para efeito de construção do horário de aulas.

DEFINIÇÃO III.6 - Período é cada uma das partes de igual duração em que os dias da semana são divididos para efeito de construção do horário de aulas. A duração do período é igual ao máximo divisor comum das durações das aulas, quando expressas em minutos.

DEFINIÇÃO III.7 - Recurso é um professor, uma classe ou algum equipamento especial (v.g. Laboratório, retroprojektor, etc.) necessário à realização das aulas de um curso.

DEFINIÇÃO III.8 - Disponibilidade de um recurso é o conjunto dos períodos em que este recurso está livre para participar das aulas.

DEFINIÇÃO III.9 - Turma é a associação de uma disciplina e dos recursos necessários à realização das aulas de um curso desta disciplina.

DEFINIÇÃO III.10 - Disponibilidade de uma turma é o conjunto dos períodos em que esta turma está livre para a realização de aulas. É portanto a interseção das disponibilidades dos seus recursos.

DEFINIÇÃO III.11 - Horário de uma turma é a especificação, para cada aula, do dia da semana em que será realizada e dos seus períodos de início e término.

Para cada turma devem ser indicadas as características qualitativas que o seu horário deve apresentar, tais como:

- a configuração da carga horária semanal escolhida entre várias eventualmente possíveis;
- se as aulas podem ser todas elas realizadas em dias consecutivos, ou não, além de outras, examinadas na subseção III.3.2.

DEFINIÇÃO III.12 - Duas ou mais turmas são ditas conflitantes quando exigem um mesmo recurso para a realização das suas aulas. Os horários de turmas conflitantes devem ser disjuntos.

DEFINIÇÃO III.13 - A matriz de conflito de um problema de horário  $P$  envolvendo as  $n$  turmas  $t_1, t_2, \dots, t_n$  é a matriz quadrada  $X = |x_{ij}|$  de ordem  $n$  tal que  $x_{ij} = 1$  se, e somente se, as turmas  $t_i$  e  $t_j$  são conflitantes e  $x_{ij} = 0$  caso contrário.

### III.2.2. ALGUNS COMENTÁRIOS

Para a imensa maioria dos problemas de Engenharia, Física, Matemática e outras ciências tradicionais dispõe-se de uma terminologia já consagrada e que permite formulá-los precisamente para um tratamento analítico. Uma das primeiras dificuldades com que nos deparamos no desenvolvimento do presente trabalho foi a necessidade de tornar um pouco mais precisa a terminologia usada correntemente com relação ao problema de horário. As definições apresentadas em geral refletem o significado com que os termos correspondentes são usados na Universidade Federal do Rio de Janeiro e, provavelmente, em outras instituições de ensino superior. Entretanto, convém ressaltar que vários dos termos aqui definidos também são usados com outros significados no ambiente acadêmico. Por exemplo, "curso" pode designar cada uma das diversas modalidades de formação profissional oferecidas pela Universidade; "turma" indica também o conjunto de todos os alunos inscritos nas mesmas disciplinas, que assistem às mesmas aulas e pretendem a mesma formação profissional. Finalmente, "classe" parece ser pouco usada na Universidade, mas foi adotada para usar "turma" com o significado da definição (III.9).

O número de alunos que compõem o corpo discente de uma disciplina determina a quantidade de turmas desta disciplina num dado período letivo. Muitas disciplinas, principalmente do Ciclo Básico, possuem um corpo discente tão numeroso que precisa ser desdobrado em diversas classes para a realização das aulas. Assim, é necessário poder especificar várias turmas para a mesma disciplina. São variados os critérios que

presidem a distribuição dos alunos pelas diversas turmas de uma mesma disciplina. Um critério natural consiste em particionar os alunos segundo a formação profissional que pretendem obter. Se algum dos conjuntos assim obtidos ainda for considerado por demais numeroso, ele pode ser dividido em classes menores até obter-se turmas com quantidade adequada de alunos. Por outro lado, para disciplinas eletivas por exemplo, este critério pode conduzir a várias classes com número reduzido de alunos e neste caso a administração acadêmica pode preferir tratar cada uma destas classes como "subclasses" e reuní-las numa classe que dará lugar a uma só turma. Neste último caso, devem ser tratadas como conflitantes as turmas cujas classes contêm alguma "subclasse" em comum. Isto sugere que, ao invés de considerar os alunos estruturados em classes, deveriam sê-lo em "subclasses". Entretanto, preferimos manter a classe como o menor grupo indivisível de alunos porque a grande maioria dos problemas de horário pode ser organizada convenientemente por meio de classes. Por outro lado, uma utilização liberal de "subclasses" poderia conduzir a relações de conflito tão numerosas que seria impossível construir um horário de aulas com ciclo de uma semana. Para especificar as relações de conflito decorrentes de "subclasses" comuns, podem ser usadas as "cliques de turmas", definidas na subseção III.3.4.

Como as relações de conflito dificultam a solução de um problema de horário, é sempre conveniente reduzi-las a um mínimo. Assim, as turmas devem ser organizados de tal modo que os recursos que delas participam acarretem a menor quantidade possível de conflitos. Quanto mais esparsa a matriz de conflito, tanto melhor.



### III.2.3. A QUESTÃO DAS SALAS DE AULA

Um aspecto importante e frequentemente associado ao problema de horário de aulas, refere-se à determinação das salas onde serão realizadas estas aulas. Entretanto, após examinar esta questão decidimos omitir seu tratamento no presente trabalho, e isto por vários motivos.

Em primeiro lugar, porque consideramos esta questão suficientemente complexa para merecer um estudo à parte, que pode ser extenso. Acreditamos que uma solução adequada deve minimizar o deslocamento dos alunos e levar em conta fatores tais como a capacidade, localização e jurisdição administrativa das salas disponíveis, além de prover salas para realização de conferências, seminários e etc.

Em segundo lugar, como relatado na seção II.5 é NP-completo o problema de decisão correspondente à versão mais interessante do problema de horário e assim julgamos desaconselhável tentar resolver, concomitantemente, os problemas de construção do horário de aulas e da determinação das suas salas.

Por outro lado, na prática constata-se que os campi Universitários dispõem de salas em quantidade tal que geralmente é possível atender um horário de aulas pronto. Assim, a solução do problema de salas de aula poderia ser feita por um algoritmo independente que determinaria, para cada aula de uma turma, a sala onde ela pode ser realizada.

Embora as salas de aula não constem do modelo adotado, admitimos que cada aula exige exatamente uma sala para sua realização.

### III.3. A ESTRUTURA DO MODELO ADOTADO

A escolha e a organização das definições apresenta das na seção III.2 foi norteada pela identificação dos principais fatos que devem ser levados em conta na formulação do modelo em relação ao qual será desenvolvido um algoritmo para a construção dos horários de aula. Este modelo refere-se ao funcionamento da Universidade sob o aspecto limitado à construção e utilização dos horários de aula.

#### III.3.1. O MODELO E AS RESTRIÇÕES INTRÍNSECAS

Em essência, consideramos as disciplinas e os recursos - professores, classes e equipamentos especiais - como entidades independentes, que são associadas para a formação de turmas. A administração acadêmica, de comum acordo com os interessados, define as turmas e as características qualitativas que espera dos seus horários, informa a identificação e a disponibilidade de cada recurso e estabelece os horários prefixados, se houver. Os recursos usados por cada turma determinam tanto a disponibilidade da turma, como as relações de conflito entre as turmas; estas relações de conflito podem ser expressas de forma unificada por uma matriz de conflito. A única relação prevista entre as turmas é a relação de conflito. Para cada turma deverá ser determinado um horário e o conjunto dos horários das turmas constitui o horário de aulas da Universidade.

A própria natureza das entidades envolvidas, o signific

ficado da sua associação e a finalidade de um horário de aulas exigem que seja satisfeito um conjunto de restrições que denominamos restrições intrínsecas do problema de horário. Adjetivamos estas restrições como intrínsecas porque elas sempre devem ser satisfeitas por qualquer solução de um problema de horário de aulas e, no que couber, também devem ser atendidas pelos dados.

Em relação ao modelo adotado, consideramos como restrições intrínsecas as seguintes:

- a) as turmas que envolvem uma mesma classe devem ter disciplinas distintas;
- b) as turmas não podem diferir apenas pelo equipamento especial;
- c) os horários de turmas conflitantes devem ser disjuntos;
- d) o horário de uma turma somente pode usar os períodos contidos na sua disponibilidade;
- e) o horário de uma turma deve ser tal que a soma das durações das aulas iguale a carga horária semanal da disciplina correspondente;
- f) o horário de uma turma deve ser tal que as suas aulas sejam realizadas em diferentes dias da semana;
- g) toda aula tem a duração de um ou mais períodos consecutivos;
- h) toda aula exige um professor, uma classe e, eventualmente, um equipamento especial.

Estas restrições intrínsecas também poderiam ser enunciadas como proposições que devem ter o valor lógico VERDADEIRO para todas as soluções de uma instância do problema de horário

e pelos seus dados. Por conveniência, designamos por  $A_0$  tanto o conjunto destas oito restrições como das proposições correspondentes.

Em linhas gerais, as restrições (a) e (b) referem-se à definição das turmas, (c) a (f) dizem respeito aos horários das turmas enquanto (g) e (h) referem-se às aulas. Cabe destacar que muitos autores consideram a restrição (f) apenas como um requisito desejável e não como uma restrição intrínseca. Naturalmente, o modelo permite que qualquer professor participe de duas ou mais turmas com a mesma disciplina ou com disciplinas diferentes; permite também que qualquer professor participe de duas ou mais turmas com a mesma classe, embora isto seja bem mais raro nas Universidades.

Além de satisfazer estas oito restrições intrínsecas geralmente espera-se que um horário de aulas atenda certos requisitos qualitativos, analisados a seguir.

### III.3.2. OS ASPECTOS QUALITATIVOS

Dado um problema de horário e duas soluções diferentes, eventualmente uma delas poderá vir a ser considerada melhor do que a outra. Embora uma tal avaliação tenha sempre um componente subjetivo, é possível enumerar algumas características que contribuem para uma melhor avaliação de um dado horário de aulas.

Assim, em geral é desejável que:

a) a carga horária semanal da disciplina seja desdobrada em

- aulas com aproximadamente a mesma duração;
- b) todas as aulas de uma mesma turma comecem à mesma hora;
  - c) as aulas de uma mesma turma não sejam todas em dias consecutivos;
  - d) seja facultado aos professores, principalmente aos de tempo parcial, escolher dia da semana em que obrigatoriamente deve haver uma aula de cada uma das suas turmas;
  - e) a carga horária semanal de aulas de cada aluno distribua-se equitativamente pelos dias da semana;
  - f) a carga horária semanal de aulas de cada professor utilize o menor número possível de dias;
  - g) as aulas ministradas pelo mesmo professor num mesmo dia guardem entre si um intervalo não inferior a um número da do de períodos.

Em relação ao modelo adotado, as características (a) a (d) constituem as características qualitativas do horário de uma turma e devem ser especificadas pelo usuário. O atendimento de (e) é proporcionado, ainda que indiretamente, pela indicação da disponibilidade de cada recurso, e em particular de cada classe, pelo usuário. Deixamos de considerar as características (f) e (g) pois cada uma delas introduz uma nova relação entre turmas que já são conflitantes, parece bastante difícil atendê-las em toda generalidade, e não está claro se seria adequado fazê-lo em face da complexidade do problema de horário. Por outro lado, a possibilidade de especificar a disponibilidade de cada recurso, aliada à liberdade de especificar qualquer combinação destas quatro características qualitativas para cada turma, proporcionam ao usuário meios

bastante significativos para moldar o horário de aulas segundo as suas necessidades. Como em certo sentido estas quatro características (a) e (d) constituem restrições à disponibilidade da turma, devem ser usadas criteriosamente para que a instância dada do problema de horário tenha pelo menos uma solução.

As características qualitativas são especificadas indicando, para cada turma:

- a configuração da carga horária semanal;
- se todas as suas aulas devem começar à mesma hora, ou podem começar em horas diferentes;
- se é permitido, ou não, que todas as suas aulas sejam realizadas em dias consecutivos;

e, para cada professor, quando isto for considerado importante

- o dia da semana em que obrigatoriamente deve haver aula de cada uma das suas turmas.

Analogamente às restrições intrínsecas, estas características qualitativas também poderiam ser formuladas como proposições, que devem ter o valor lógico VERDADEIRO para toda solução da instância dada do problema de horário.

### III.3.3. UMA FORMULAÇÃO DO PROBLEMA DE HORÁRIO

Sendo dados

- um conjunto  $T = \{t_1, t_2, \dots, t_n\}$  de  $n$  turmas, onde cada turma é a associação de uma disciplina e dos recursos necessários à realização de um curso desta disciplina;
- a identificação e a disponibilidade de cada recurso;
- as características qualitativas que devem ser exibidas pelo horário de cada turma;
- os horários prefixados para algumas das turmas de  $T$ ;

determinar caso exista,

- um conjunto de horários para as turmas de  $T$  satisfazendo o conjunto  $A_0$  de restrições intrínsecas e apresentando as características qualitativas especificadas.

É conveniente designar por  $A_n$  o conjunto de todas as proposições que devem ser satisfeitas por qualquer solução de uma instância dada do problema descrito acima. A proposição (III.1) apresenta algumas relações entre  $A_0$ ,  $A_n$  e outros conjuntos de proposições que serão usados na seção III.5 e seguintes.

PROPOSIÇÃO III.1 - Sejam  $P_i$  uma instância do problema de horário constituído pelo conjunto de turmas  $\{t_1, t_2, \dots, t_i\}$  e demais dados, e  $A_i$  o conjunto de todas as proposições que devem ser satisfeitas por qualquer solução de  $P_i$ . Os conjuntos  $A_i$  correspondentes às instâncias  $P_i$ , para  $i = 1, 2, \dots, n$  são

tais que:

$$A_0 \subset A_1 \subset A_2 \subset \dots \subset A_n$$

onde  $A_0$  é o conjunto de proposições correspondentes às restrições intrínsecas do problema de horário.

Demonstração: as instâncias consideradas  $P_i$  e os respectivos conjuntos de proposições  $A_i$ ,  $i = 1, 2, \dots, n$  são

Instância	Conjunto de Turmas	Conjunto de Proposições
$P_1$	$\{t_1\}$	$A_1$
$P_2$	$\{t_1, t_2\}$	$A_2$
...	...	...
$P_n$	$\{t_1, t_2, \dots, t_n\}$	$A_n$

Inicialmente, para  $i = 1, 2, \dots, n$

$$A_0 \subset A_i \tag{1}$$

pela própria definição de  $A_0$  e  $A_i$ .

Por outro lado, qualquer solução de  $P_i$  contém uma solução de  $P_{i-1}$  e assim o conjunto diferença  $A_i \setminus A_{i-1}$  é não vazio e contém as proposições correspondentes às características qualitativas impostas ao horário de turma  $t_i$ , donde

$$A_{i-1} \subset A_i, \text{ para } i = 1, 2, \dots, n \tag{2}$$

combinando (1) e (2) teremos

$$A_0 \subset A_1 \subset A_2 \subset \dots \subset A_n$$



### III.3.4. GRAFOS E RELAÇÕES DE CONFLITO

As relações de conflito entre as turmas de um problema de horário exprimem a interação entre estas turmas. Estas relações restringem os períodos que podem ser usados pelas aulas de uma dada turma, em decorrência da sua utilização por turmas conflitantes com a turma considerada. É conveniente representar esta relações de conflito por meio de grafos, como descrito a seguir.

DEFINIÇÃO III.14 - Seja  $T = \{t_1, t_2, \dots, t_n\}$  um conjunto de  $n$  turmas de um problema de horário  $P$ . O grafo de conflito de  $P$  é o grafo  $G(V, E)$  rotulado em vértices, tal que

- existe uma correspondência biunívoca entre os conjuntos  $V$  de vértices e  $T$  de turmas e a turma  $t_i \in T$  é o rótulo do vértice  $v_i \in V$ , para  $i = 1, 2, \dots, n$ .
- o conjunto  $E$  de arestas de  $G$  contém a aresta  $(v_i, v_j)$  se, e somente se as turmas  $t_i$  e  $t_j$  são conflitantes, para  $i = 1, 2, \dots, n$  e  $j = 1, 2, \dots, n$ .

Das definições (III.13) e (III.14) segue-se que a matriz de adjacência do grafo de conflito de um problema de horário  $P$  é igual à matriz de conflito de  $P$ . O grafo de conflito permite examinar com naturalidade a decomposição do problema de horário e oferece uma alternativa para a especificação das relações de conflito.

### III.3.4.1. A Decomposição do Problema de Horário

Uma vez obtido o grafo de conflito de uma instância do problema de horário  $P$  com  $n$  turmas, podemos verificar se ele é conexo ou não. Se este grafo for não conexo então existem  $k \geq 2$  componentes conexos. Estes componentes conexos particionam o conjunto dos vértices do grafo de conflito (e dos seus rótulos) em subgrafos com  $n_1, n_2, \dots, n_k$  vértices e  $n_1 + n_2 + \dots + n_k = n$ . Ora, a cada componentes conexo do grafo de conflito corresponde um conjunto de turmas, constituído pelos rótulos dos seus vértices. Assim, esta partição do conjunto dos vértices do grafo de conflito induz a mesma partição do conjunto das  $n$  turmas de  $P$ , isto é, em  $k$  subconjuntos de turmas tais que, para  $i = 1, 2, \dots, k$  o  $i$ -ésimo subconjunto é constituído pelos rótulos dos vértices do  $i$ -ésimo componente conexo do grafo de conflito. Sendo a relação de conflito a única prevista entre as turmas e por não existir interação entre turmas pertencentes a componentes distintos, segue-se que esta partição do grafo de conflito implica uma partição análoga do problema dado, em instâncias independentes e menores. Estas instâncias são "menores" no sentido de que cada uma delas envolve uma quantidade menor de turmas do que a instância original do problema. Esta decomposição também é aplicável quando, além da relação de conflito, define-se outras relações entre as turmas, desde que estas relações sejam definidas apenas entre turmas conflitantes. Por exemplo, esta decomposição conduziria a problemas independentes mesmo que tivessem sido definidas as relações correspondentes às características (f) e/ou (g) da seção III.3.2. A proposição III.2 resume esta análise.

PROPOSIÇÃO III.2 - Se o grafo de conflito de um problema de horário  $P$  tem  $k$  componentes conexos então  $P$  pode ser decomposto em  $k$  problemas independentes: uma para cada componente conexo.

Na prática, esta decomposição é extremamente útil porque:

- a) em geral, a soma dos tempos de processamento necessários para obter a solução de cada uma destas  $k$  instâncias será menor do que o tempo de processamento exigido para a solução do problema não decomposto;
- b) se não existe solução para um dado problema de horário  $P$ , então não existe solução para uma, ou várias, das  $k$  instâncias em que  $P$  pode ser decomposto; entretanto, é mais fácil determinar os motivos que levam algumas destas  $k$  instâncias a não ter solução isoladamente, e providenciar as alterações dos dados, do que fazê-lo para o problema não decomposto.

Em face de (a) e (b) acima, e de ser NP-completo o problema de decisão associado ao problema de horário considerado neste trabalho, conclui-se que é importante organizar as turmas de modo que a instância dada do problema possa ser decomposta em instâncias independentes e menores, tais que seja o menor possível o número de turmas na maioria destas instâncias.

### III.3.4.2. Alternativas de Especificação de Turmas Conflitantes

Em princípio, as relações de conflito entre as turmas de um dado problema podem ser obtidas a partir do exame dos recursos que participam de cada turma. Entretanto, há situações em que é conveniente especificar explicitamente que certas turmas de um dado problema de horário  $P$  são conflitantes. Isto poderia ser feito indicando que os elementos correspondentes da matriz de conflito devem ter valor 1. Uma forma mais concisa de fazê-lo baseia-se num conceito de grafos. Seja  $\{t_i, t_j, \dots, t_k\}$  um subconjunto de turmas de  $P$  tal que cada turma é conflitante com todas as outras deste subconjunto. Se construirmos o grafo de conflito de  $P$  veremos que os vértices correspondentes a estas turmas  $t_i, t_j, \dots, t_k$  constituem um subgrafo completo que, em teoria de grafos, é denominado clique. Convém esclarecer que alguns autores como HARRY (24) e REINGOLD, NIEVERGELT e DEO (39) definem clique como um subgrafo completo maximal. Já BERGE (5), PAPADIMITRIOU e STEIGLITZ (37), SZWARCFITER (43) denominam qualquer subgrafo completo como clique. Preferimos a aceção adotada por SZWARCFITER (43). Incidentalmente citamos que muitos problemas envolvendo determinação de cliques são NP-completo, como documentado por GAREY e JOHNSON (18). A próxima definição adapta este conceito para turmas.

DEFINIÇÃO III.15 - Clique de turmas é um subconjunto de turmas de um problema de horário tal que, neste subconjunto, cada turma conflita com todas as demais.

Assim, para indicar que as turmas  $t_i, t_j, \dots, t_k$  são todas conflitantes entre si, basta especificar que constituem uma clique e serão tratadas como conflitantes, independentemente dos seus recursos. Isto resolve, por exemplo, a dificuldade no tratamento de turmas que são conflitantes por causa de "subclasses" comuns, discutida ao final de III.2.2.

#### III.4. MÉTODO DE SOLUÇÃO: ESCOLHA E DESCRIÇÃO

Neste ponto passamos a examinar a seguinte questão:

Propor um algoritmo para resolver o problema de horário descrito na seção III.3.3 sabendo que o respectivo problema de decisão é NP-completo, e considerando a experiência relatada na literatura disponível, resumida no Capítulo I.

##### III.4.1. OS MÉTODOS DISPONÍVEIS

Dentre as técnicas gerais de solução de problemas combinatórios consideramos como candidatos os seguintes:

- programação matemática
- busca exaustiva estruturada em árvore (enumeração implícita)
- métodos heurísticos (enumeração incompleta)

ou eventualmente uma combinação destas técnicas; vejamos brevemente em que consiste cada uma delas.

Em programação matemática, o problema é formulado abstratamente sob a forma

$$\text{otimizar } f(\underline{x}) \text{ , } \underline{x} \in S \subseteq \mathbb{R}^n$$

onde  $\mathbb{R}^n$  é o conjunto de todos os vetores n-dimensionais de números reais e  $f$  é uma função em reais definida sobre  $S$ .  $S$  é o conjunto de restrições, sendo definido por igualdades e desigualdades envolvendo os componentes de  $\underline{x} = (x_1, x_2, \dots, x_n)$  e  $f$  é a função objetiva. Todo  $\underline{x} \in S$  é uma solução viável, também chamada programa. Um problema de programação matemática visa estabelecer se existe uma solução viável que otimiza (maximiza ou minimiza) o valor de  $f(\underline{x})$  e, em caso afirmativo, obter uma ou talvez as soluções ótimas. Existem diversas modalidades de programação matemática tais como programação linear, não linear, inteira e etc; frequentemente estas técnicas são adaptadas para problemas de fluxos em redes.

Nos métodos de busca exaustiva estruturada em árvore, a busca é organizada sob a forma de uma árvore e a enumeração é dita implícita porque, tão logo se constate que um dado vértice não pode pertencer à solução procurada, exclui-se de qualque consideração posterior toda a subárvore com raiz neste vértice. De acordo com a forma de exploração dos vértices da árvore e a existência ou não de uma função objetiva a otimizar, distinguem-se basicamente três variantes: "backtrack" (sequencial), programação dinâmica (paralela) e "branch and bound" (em parte sequencial e em parte paralela).

Os métodos heurísticos usam estratégias intuitivas, ora bem simples ora sofisticadas, baseadas em observações empíricas, para produzir com rapidez alguma solução viável do pro

blema. Para problemas de otimização propõem-se gerar soluções viáveis próximas da solução ótima. Em geral estas estratégias são sob medida para cada problema e pouco sujeitas a tratamento matemático, razão pela qual praticamente não há restrições quanto à estrutura das abordagens heurísticas. Estas estratégias procuram orientar a construção de uma solução usando apenas parte das alternativas disponíveis e por isto são de enumeração incompleta. Não sendo exaustivos, podem deixar de encontrar soluções com as características desejadas, mesmo que tais soluções existam. Para alguns problemas de otimização combinatória é possível construir algoritmos heurísticos com garantia de desempenho, no sentido de que a solução por eles gerada não difere da solução ótima por mais do que uma constante ou percentagem. Tais são os algoritmos de aproximação descritos sumariamente na seção II.7 onde também apresentamos algumas referências bibliográficas.

#### III.4.2. A ESCOLHA DO MÉTODO DE SOLUÇÃO

Inicialmente devemos decidir se a solução buscada deve otimizar alguma função objetiva e, em caso afirmativo, construir esta função ou se basta obter uma solução viável, satisfazendo o conjunto  $A_n$  de restrições descrito em III.3.3.

Quanto à função objetiva, o problema de horário de aulas não possui uma função objetiva tão natural quanto a disponível por muitos outros problemas. Ou melhor, o valor ótimo da função objetiva, obtido na prática por horários construídos manualmente, exprime o resultado de uma negociação envolvendo de

departamentos, professores e classes, iterada por várias tentativas de conciliar interesses legítimos mas muitas vezes mutuamente exclusivos. Em suma, este ótimo exprime o melhor compromisso entre critérios objetivos e juízos subjetivos dos interessados. A dificuldade não está no compromisso, pois este é o papel da função objetiva, mas na construção de uma função que exprima fidedignamente, e para uso computacional, a interação entre os inúmeros aspectos a considerar. Por outro lado, as dificuldades constatadas na prática com tais funções, relatadas na seção I.7 para modelos de programação linear do problema de horário de aulas, mostram que a técnica de construção destas funções para o problema considerado requer pesquisas adicionais para tornar-se o instrumento valioso que é para outros problemas.

Assim, abrimos mão da otimização e consideramos que resolver o problema de horário de aulas consiste em obter uma solução viável. Dentre os métodos citados, restam-nos dois: a busca exaustiva por "backtrack" e os métodos heurísticos, excepto os algoritmos de aproximação descritos na seção II.7. Entre estes dois, optamos por um algoritmo de "backtrack", pelos motivos abaixo:

- a) a não confirmação da conjectura de GOTLIEB e CSIMA, subseção I.4.4.3, e a análise de DEMPSTER, subseção I.4.4.4, mostram que não existe nenhum processo eficiente para escolher o horário de uma turma de tal modo que ele nunca precise ser alterado; assim, qualquer algoritmo de construção de horários deve prever esta possibilidade.



- b) os métodos heurísticos, quando baseados numa boa idéia, são bastante satisfatórios, como atestado pelo trabalho de BRITAN e FARLEY (6) apresentados em I.3.2; entretanto, parece que resta pouco a fazer para melhorá-lo.
- c) um algoritmo de backtrack examina sistematicamente uma árvore finita e portanto não há risco de entrar em "loop", mas para certos algoritmos heurísticos de construção de horários é pelo menos complicado evitar que isto suceda, conforme relato de experiências prática citadas em I.3.
- d) finalmente, quanto ao desempenho, ambos tem seus pontos fracos. Ambos, pela sua própria natureza, podem deixar de obter uma solução: os heurísticos, porque fazem uma enumeração incompleta e os de backtrack porque a enumeração implícita que fazem, para certas instâncias, pode demandar um tempo de processamento que exceda os limites práticos.

#### III.4.3. UMA BREVE DESCRIÇÃO DO ALGORITMO SELECIONADO

Na seção II.6 descrevemos o algoritmo de backtrack e as principais técnicas de melhorar o seu desempenho. Assim, para completar a escolha do algoritmo restam ainda as seguintes tarefas:

- formular o problema de modo que as suas soluções sejam tu-  
plas;
- definir a função que identificará uma tupla como solução;
- construir as funções intermediárias que identificarão as tu-  
plas que não podem ser estendidas para formas soluções;

- propor alguma técnica para melhorar o desempenho do algoritmo.

Um aspecto importante do problema de horário é a existência de relações de conflito entre as turmas. Uma primeira forma construtiva de aproveitar este fato seria a seguinte: dada uma instância do problema cujo grafo de conflito é não conexo, ela poderia ser decomposta segundo os componentes conexos do seu grafo de conflito e o algoritmo de backtrack seria usado para obter o horário de aulas de todas as turmas de cada componente. O horário da instância dada do problema seria a união dos horários das turmas que constituem cada um dos componentes conexos. Em face da natureza do problema de horário e do algoritmo de backtrack, tal decomposição só pode contribuir para a redução do tempo de processamento. Entretanto, sem perda de generalidade e visando simplificar um pouco a apresentação, preferimos admitir que as instâncias consideradas do problema de horário tem grafo de conflito conexo.

Segue-se breve exame das tarefas citadas acima, com base na seção II.6, dedicada ao algoritmo de backtrack e no desenvolvimento apresentado nas seções III.2 e III.3 sobre o problema de horário de aulas.

Seguindo HARALICK e ELLIOTT (23), num dado instante da execução do algoritmo de backtrack as turmas passadas são aquelas para as quais já foi definido um horário, a turma corrente é aquela cujo horário está sendo selecionado e são futuras as turmas para as quais ainda falta escolher um horário.

Inicialmente, dada uma instância do problema de horário com  $n$  turmas, as soluções são  $n$ -tuplas nas quais cada com

ponente é um par (turma, horário desta turma). Uma  $n$ -tupla é uma solução se e somente se satisfaz o conjunto  $A_n$  de proposições apresentado em III.3.3. Assim, qualquer solução, caso exista, está no nível  $n$  da árvore do espaço de estados.

As funções intermediárias serão os conjuntos  $A_k$  de proposições ( $0 \leq k < n$ ) também citados em III.3.3, o que implica que o horário escolhido para a turma corrente deve ser compatível com os horários já escolhidos para as turmas passadas. Além dos conjuntos  $A_k$  deverão ser usadas proposições que assegurem que o horário selecionado para a turma corrente é tal que reste pelo menos um horário disponível para cada turma futura. Neste ponto temos uma segunda oportunidade de aproveitar bem as relações de conflito, restringindo estes testes de consistência às turmas conflitantes com a turma corrente.

Finalmente, para melhorar a eficiência, além dos testes de consistência citados, procuraremos reduzir o tamanho do espaço de estados mediante reordenação das turmas: em cada nível da árvore será selecionada como a turma corrente aquela que tem a menor quantidade de horários disponíveis. Por outro lado, o problema de horário não tem simetrias nem translações que possam ser usadas para reduzir ainda mais a cardinalidade do espaço de estados.

### III.5. O ALGORITMO E O ESPAÇO DE ESTADOS

Abordamos agora alguns detalhes do algoritmo de back-track adotado para a solução do problema de horário e examinamos o espaço de estados gerado pela sua execução. Em III.5.1 formulamos como  $n$ -tuplas as soluções de uma instância dada do problema e discutimos as funções intermediárias que serão usadas. Em III.5.2 descreveremos o espaço de estados e examinamos sua estrutura, concluindo que é uma árvore. Em III.5.3 verificamos o efeito destas funções sobre a árvore do espaço de estados e em III.5.4 introduzimos a questão da ordem em que as turmas devem ser tomadas para reduzir o número de estados gerados na busca.

#### III.5.1. FUNÇÕES INTERMEDIÁRIAS E SOLUÇÕES

Pela definição (III.11), o horário de uma turma é a especificação, para cada aula, do dia da semana em que será realizada e dos seus períodos de início e término.

Consideremos uma instância  $I$  do problema de horário de aulas descrito em III.3.3 e seja  $T = \{t_1, \dots, t_n\}$  o conjunto das  $n$  turmas que constituem esta instância. Designaremos por  $H_i$  o conjunto de todos os horários  $h_{i,1}, \dots, h_{i,d_i}$  da turma  $t_i$  considerada isoladamente, isto é, ignorando a interação de  $t_i$  com cada uma das demais turmas de  $T$ . Assim, os horários de  $H_i$  devem satisfazer o conjunto  $A_0$  de restrições intrínsecas (subseção III.3.1) e ainda apresentar as características qualitativas (final de III.3.2) especificadas pelo usuário pa

ra o horário da turma  $t_i$  em algum estágio da execução do algoritmo de backtrack. Com esta notação, as soluções da instância dada do problema serão  $n$ -tuplas de pares (turma, horário desta turma), que satisfazem o conjunto  $A_n$  de proposições, isto é,

$$A_n((t_1, h_{1,j_1}), \dots, (t_n, h_{n,j_n})) = \text{VERDADEIRO} .$$

Tratemos agora das funções intermediárias que serão usadas para orientar a escolha do horário da turma corrente. Consideremos uma solução da instância dada do problema, representada pela  $n$ -tupla

$$((t_1, h_{1,j_1}), \dots, (t_k, h_{k,j_k}), (t_{k+1}, h_{k+1,j_{k+1}}), \dots, (t_n, h_{n,j_n}))$$

Ora, nesta solução o horário da turma genérica  $t_{k+1}$  deve ser compatível com os horários das turmas à sua esquerda - turmas passadas - e com os horários das turmas à sua direita - turmas futuras. No momento da escolha do horário de  $t_{k+1}$  os horários das turmas passadas são conhecidos, bastando escolher  $h_{k+1,j_{k+1}}$  de modo que

$$((t_1, h_{1,j_1}), \dots, (t_k, h_{k,j_k}), (t_{k+1}, h_{k+1,j_{k+1}}))$$

seja uma solução da instância do problema de horário constituída pelas turmas  $t_1, \dots, t_k, t_{k+1}$ , isto é, devemos ter

$$A_{k+1}((t_1, h_{1,j_1}), \dots, (t_k, h_{k,j_k}), (t_{k+1}, h_{k+1,j_{k+1}})) = \text{VERDADEIRO}$$

Como o conjunto de proposições  $A_{k+1}$  é a conjunção de  $A_k$  com as exigências qualitativas impostas ao horário da turma  $t_{k+1}$  segue-se que

$A_{k+1} ((t_1, h_1, j_1), \dots, (t_k, h_k, j_k), (t_{k+1}, h_{k+1}, j_{k+1}))$  implica

$A_k ((t_1, h_1, j_1), \dots, (t_k, h_k, j_k))$

e assim podemos tomar os conjuntos de proposições  $A_k$ ,  $0 \leq k < n$ , como as funções intermediárias. Reunimos estas considerações na definição (III.16).

DEFINIÇÃO III.16 - Horário parcial de ordem (k+1),  $0 \leq k < n$ , é a (k+1)-tupla que se obtém estendendo um horário parcial de ordem k, mediante a inclusão do par  $(t_{k+1}, h_{k+1}, j_{k+1})$  como o (k+1)-ésimo componente e tal que esta (k+1)-tupla satisfaça o conjunto  $A_{k+1}$  de proposições referentes às turmas  $t_1, \dots, t_k, t_{k+1}$ , isto é,

$A_{k+1} ((t_1, h_1, j_1), \dots, (t_k, h_k, j_k), (t_{k+1}, h_{k+1}, j_{k+1})) = \text{VERDADEIRO}$  ;

o horário parcial de ordem zero é a tupla vazia ( ).

Considerando a definição (III.16), as soluções de uma instância do problema de horário de aulas com n turmas são horários parciais de ordem n.

Vejamos agora a compatibilidade do horário da turma corrente com os horários das turmas futuras. Infelizmente não se conhece nenhum critério que permita escolher um horário para a turma  $t_{k+1}$  de modo que o horário parcial obtido sempre possa ser estendido para formar uma solução da instância dada do problema. Entretanto, é possível reconhecer um horário parcial que certamente não pode ser estendido para formar uma solução. A idéia é a seguinte. Suponhamos que estamos estendendo o horário

rio parcial de ordem zero para formar um horário parcial de ordem um, mediante a escolha de um horário para a turma  $t_1$ . Como ainda não temos turmas passadas, podemos tomar qualquer horário de  $H_1$ . Uma vez escolhido este horário de  $t_1$ , para cada turma  $t_c$  futura e conflitante com  $t_1$  marcaremos como não mais disponíveis por  $t_c$  aqueles horários de  $H_c$  que tem algum período em comum com o horário escolhido para  $t_1$ . Suponhamos que este mesmo procedimento seja adotado imediatamente depois da escolha do horário de  $t_2, \dots, t_k$ . Assim, no momento de escolher um horário para  $t_{k+1}$  bastará tomar um horário não marcado de  $H_{k+1}$  e seguramente ele será compatível com os horários de todas as turmas passadas. Repetiremos este mesmo processo de marcação, agora para todas as turmas  $t_c$  futuras e conflitantes com  $t_{k+1}$ . O ponto importante é o seguinte: como resultado desta marcação (e das marcações anteriores) pode vir a suceder que para alguma turma futura  $t_c$ , o respectivo conjunto  $H_c$  acaba contendo apenas horários marcados. Isto significa que não mais existe em  $H_c$  horário compatível com os horários selecionados para as turmas  $t_1, \dots, t_{k+1}$  isto é, com o horário parcial de ordem  $(k+1)$  construído. Segue-se que não adianta estender este horário parcial, devendo-se desmarcar os horários que foram marcados devido ao horário escolhido para  $t_{k+1}$  e tentar um outro horário para  $t_{k+1}$ . Adotaremos portanto o seguinte critério:

Dado um horário parcial de ordem  $(k+1)$ , o horário selecionado para a turma  $t_{k+1}$  deve ser tal que cada turma futura e conflitante com  $t_{k+1}$ , tenha pelo menos um horário disponível e compatível com o horário parcial considerado.

Designaremos por  $B_k$ ,  $0 \leq k < n$ , o conjunto de proposições correspondentes ao critério descrito acima; o domínio de  $B_k$  é o conjunto dos horários parciais de ordem  $k$ . Se

$$B_{k+1}((t_1, h_{1,j_1}), \dots, (t_k, h_{k,j_k}), (t_{k+1}, h_{k+1,j_{k+1}})) = \text{FALSO}$$

então este horário parcial de ordem  $(k+1)$  certamente não pode ser estendido até uma solução porque existe pelo menos uma turma futura, conflitante com  $t_{k+1}$ , para a qual não existe nenhum horário compatível com o horário parcial considerado. Neste caso deveremos escolher um outro horário para a turma  $t_{k+1}$ . Por outro lado, se

$$B_{k+1}((t_1, h_{1,j_1}), \dots, (t_k, h_{k,j_k}), (t_{k+1}, h_{k+1,j_{k+1}})) = \text{VERDADEIRO}$$

então talvez este horário parcial de ordem  $(k+1)$  possa ir sendo estendido até obter-se uma solução da instância dada do problema.

DEFINIÇÃO III.17 - Um horário parcial de ordem  $k$ , é viável se  $B_k((t_1, h_{1,j_1}), \dots, (t_k, h_{k,j_k})) = \text{VERDADEIRO}$  é inviável em caso contrário,  $0 \leq k < n$ .

É importante frisar que a marca usada para assinalar os horários não mais disponíveis por cada turma  $t_c$  futura e conflitante com  $t_{k+1}$  deve permitir identificar a turma  $t_{k+1}$  como a responsável pela marcação destes horários. Como todas as turmas de um horário parcial são distintas, segue-se que esta marca pode ter o valor do próprio índice  $(k+1)$  da turma  $t_{k+1}$ . Mais ainda, em cada estágio esta marca deve ser aplicada somente aos



horários ainda não marcados dos conjuntos  $H_c$  e que usam algum período também usado pelo horário corrente de  $t_{k+1}$ . Com isto, se um horário parcial de ordem  $(k+1)$  revelar-se inviável, obrigando a escolha de um novo horário para  $t_{k+1}$ , será possível desmarcar em cada conjunto  $H_c$  apenas os horários que foram marcados devido ao horário abandonado de  $t_{k+1}$ . As eventuais marcas devidas aos horários escolhidos para as turmas passadas  $t_1, \dots, t_k$  serão mantidas inalteradas. Escolhido um novo horário para  $t_{k+1}$ , novamente serão marcados com  $(k+1)$  alguns horários de cada conjunto  $H_c$  correspondente a cada turma  $t_c$  futura e conflitante com  $t_{k+1}$ .

Em resumo: com auxílio deste processo de marcação, o algoritmo proposto constroi  $(k+1)$ -tuplas que satisfazem o conjunto  $A_{k+1}$  de proposições isto é, são horários parciais de ordem  $(k+1)$ . Obtido um tal horário parcial, procedemos à marcação citada e testamos se este horário parcial é viável ou não isto é, se satisfaz ou não o conjunto  $B_{k+1}$  de proposições. Finalmente, a função intermediária aplicável a uma  $(k+1)$ -tupla em cada estágio da execução do algoritmo de backtrack é a conjunção dos conjuntos  $A_{k+1}$  e  $B_{k+1}$  de proposições.

### III.5.2. O ESPAÇO DE ESTADOS

De acordo com subseção II.6.1, o espaço de estados de um problema é o conjunto das  $k$ -tuplas geradas durante o processo de busca de todas as soluções de uma instância dada do problema. Segundo DIJKSTRA (14) pp. 11

"The state space describes the amount of freedom of the system; it just has nowhere else to go".

O grafo do espaço de estados é um grafo rotulado; cada vértice corresponde a um estado, sendo rotulado com a descrição do estado correspondente e cada aresta é associada ao operador que transforma o estado correspondente a uma das suas extremidades, no estado correspondente à outra extremidade. Segundo NILSSON (36) pp. 23 este grafo pode ser especificado explícita ou implicitamente. Na especificação explícita, todos os vértices e arestas são enumerados, por exemplo, por meio de uma tabela. Na especificação implícita, são dados um conjunto finito de vértices que são os vértices iniciais, os operadores que aplicados a um certo vértice permitem obter cada um dos vértices adjacentes a este vértice, e os critérios que caracterizam um dado vértice como correspondendo a um estado solução.

Usando a terminologia de NILSSON (36) pp. 11, adotaremos a seguinte definição:

DEFINIÇÃO III.18 - O espaço de estados do problema de horário de aulas com  $n \geq 1$  turmas, onde cada turma  $t_i$  tem um conjunto  $H_i$  com  $d_i > 0$  horários, é assim constituído:

- i) estados: são horários parciais de ordem  $k$ ,  $0 \leq k \leq n$ ; o estado inicial é o horário parcial de ordem zero (i.e., vazio).
- ii) operadores: são os procedimentos computacionais que transformam um horário parcial de ordem  $k$  num horário parcial de ordem  $(k+1)$ , quando o algoritmo de backtrack avança,

ou num horário parcial de ordem  $(k-1)$ , quando regride.

- iii) critério de sucesso: os estados que correspondem às soluções são horários parciais de ordem  $n$ , contendo portanto os horários das  $n$  turmas dadas.

DEFINIÇÃO III.19 - O grafo do espaço de estados do problema de horário é o grafo especificado implicitamente pela definição (III.18), onde os vértices correspondem aos estados e as arestas, aos operadores.

TEOREMA III.1 - O grafo do espaço de estados do problema de horário com  $n \geq 1$  turmas é uma árvore enraizada.

#### Demonstração

- i) Por definição deste grafo, existe exatamente um vértice inicial, que corresponde ao estado inicial e cujo rótulo é o horário parcial de ordem zero, isto é,  $( )$ . Designaremos por  $v_0$  este vértice.
- ii) Este grafo é conexo devido à sua especificação implícita: qualquer novo vértice  $v_k$  é gerado a partir de um vértice já existente mediante a aplicação de um operador. Assim, qualquer vértice é alcançável a partir do vértice inicial. Seja  $v_k$  um vértice cujo rótulo é um horário de ordem  $k$ . Um caminho desde  $v_0$  até  $v_k$  contém os vértices  $v_0, v_1, \dots, v_{k-1}, v_k$  e os seus rótulos são:

$$\begin{aligned}
v_0 &: ( ) \\
v_1 &: ((t_1, h_1, j_1)) \\
&\dots \\
v_{k-1} &: ((t_1, h_1, j_1), \dots, (t_{k-1}, h_{k-1}, j_{k-1})) \\
v_k &: ((t_1, h_1, j_1), \dots, (t_{k-1}, h_{k-1}, j_{k-1}), (t_k, h_k, j_k))
\end{aligned}$$

Como todos estes vértices são distintos, o caminho é simples (segundo terminologia de SZWARCFITER (43)) e tem comprimento  $k$ .

iii) Se  $v_k$  tem como rótulo um horário parcial de ordem  $k$  então todo caminho simples entre  $v_0$  e  $v_k$  tem exatamente  $k$  arestas. Com efeito, partindo do estado inicial cujo rótulo é o horário parcial vazio, só chegaremos a um vértice  $v_k$  cujo rótulo é um horário parcial de ordem  $k$  mediante pelo menos  $k$  aplicações do operador. Por outro lado, se usarmos  $k' > k$  aplicações do operador então não é mais "simples" o caminho construído por que há de ter regredido pelo menos uma vez, passando assim por um vértice já gerado anteriormente, pois do contrário o horário parcial obtido seria de ordem  $k' > k$ .

iv) O caminho simples entre  $v_0$  e  $v_k$  é único. Suponhamos que existe um outro caminho simples de  $v_0$  até  $v_k$ ; então percorrendo este outro caminho, obteríamos o seguinte horário parcial de ordem  $k$  como rótulo de  $v_k$ :

$$((t_\alpha, h_\alpha, j_\alpha), \dots, (t_\beta, h_\beta, j_\beta), (t_\gamma, h_\gamma, j_\gamma))$$

Como estes dois horários parciais de ordem  $k$  rotulam o mesmo vértice, estes horários parciais são iguais, donde

$$(t_\alpha, h_{\alpha, j_\alpha}) = (t_1, h_{1, j_1})$$

...

$$(t_\beta, h_{\beta, j_\beta}) = (t_{k-1}, h_{h-1, j_{k-1}})$$

$$(t_\gamma, h_{\gamma, j_\gamma}) = (t_k, h_{k, j_k})$$

e portanto os dois caminhos simples reduzem-se a um só.

- v) Entre dois vértices quaisquer  $v_i$  e  $v_j$  existe um único caminho simples pois do contrário entre um deles e  $v_0$  haveria mais de um caminho simples, contradizendo o resultado do item anterior.

Assim, concluímos que o grafo do espaço de estados considerado é uma árvore pois é conexo e entre dois vértices quaisquer existe um único caminho simples. Tomaremos  $v_0$  como a raiz desta árvore pois todos os demais vértices são gerados a partir de  $v_0$ .

### III.5.3. A ÁRVORE DO ESPAÇO DE ESTADOS

Na subseção anterior demonstraremos que o grafo do espaço de estados especificado para o problema de horário de aulas é uma árvore. A seguir definimos esta árvore de forma descritiva e apontamos o efeito das funções intermediárias  $A_k$  e  $B_k$  sobre sua constituição.

DEFINIÇÃO III.20 - A árvore do espaço de estados do problema de horário é uma árvore enraizada, rotulada, cuja raiz representa o estado inicial, tem por rótulo o horário parcial de or

dem zero e está no nível zero. Nas subárvores com raiz no nível  $k$ ,  $0 \leq k \leq n$ , os horários parciais viáveis de ordem  $k$  são os rótulos da raiz e seus filhos tem como rótulo todos os horários parciais (viáveis e inviáveis) de ordem  $(k+1)$ , onde somente o horário  $h_{k+1,j}$  do  $(k+1)$ -ésimo componente varia satisfazendo

$$A_{k+1} ((t_1, h_{1,j_1}), \dots, (t_k, h_{k,j_k}), (t_{k+1}, h_{k+1,j})) = \text{VERDADEIRO};$$

cada aresta entre o horário parcial de ordem  $k$  e os horários parciais de ordem  $(k+1)$  é rotulada com o par  $(t_{k+1}, h_{k+1,j})$  correspondente.

Observe-se que somente os horários parciais viáveis tem sucessores; como cada turma  $t_i$  tem um conjunto não vazio de horários, a raiz do nível zero sempre tem filhos. Consideremos um horário parcial viável de ordem  $k$  e seja  $t_{k+1}$  a turma corrente. Os horários escolhidos para estas  $k$  turmas passadas e a marcação descrita em III.5.1 restringem a um subconjunto de  $H_{k+1}$  os horários de  $t_{k+1}$  que podem ser usados para estender o horário parcial considerado para formar horários parciais de ordem  $(k+1)$ , isto é,  $k$ -tuplas que satisfazem  $A_{k+1}$ . Assim, o efeito de  $A_{k+1}$  é fazer com que não sejam sequer construídas, não aparecendo na árvore, as  $(k+1)$ -tuplas que não satisfariam  $A_{k+1}$  porque utilizariam horários de  $t_{k+1}$  incompatíveis com os horários já selecionados para as  $k$  turmas passadas. Na terminologia de GOLOMB e BAUMERT (19), isto constitui a preclusão.

Por outro lado, as  $(k+1)$ -tuplas que satisfazem  $A_{k+1}$  são os horários parciais e constituem a árvore do espaço de es

tados; cada horário parcial pode ser viável ou inviável. Os horários parciais inviáveis de ordem  $(k+1)$  são aqueles que não satisfazem  $B_{k+1}$ ; são gerados, aparecem na árvore, mas não tem sucessores. Assim, as funções  $B_{k+1}$  realizam a poda da árvore. Precisam ser gerados porque somente é possível constatar que são inviáveis após construí-los, realizar a marcação descrita em III.5.1 e só então testar se cada turma conflitante com  $t_{k+1}$  continua tendo pelo menos um horário disponível. Por último, os horários parciais de ordem  $(k+1)$  que satisfazem  $B_{k+1}$  são ditos viáveis e são os únicos que tem sucessores.

#### III.5.4. A REORDENAÇÃO DAS TURMAS

A reordenação é apresentada como um princípio geral de busca eficiente para algoritmos de backtrack. Consiste em selecionar o próximo componente da tupla que está sendo construída daquele conjunto que tem menos elementos disponíveis; para o problema de horário estes podem ser os conjuntos  $H_i$ , ou seus subconjuntos, de horários das turmas dadas  $t_i$ ,  $1 \leq i \leq n$ . Classificaremos a reordenação em estática e dinâmica.

Na reordenação estática, as turmas seriam tomadas numa ordem fixa estabelecida a priori. Na reordenação dinâmica, a próxima turma seria escolhida segundo algum critério baseado no estado corrente da computação. A ordenação dinâmica pode ser restrita ou irrestrita. Na ordenação dinâmica restrita todos os horários parciais de ordem  $k$ ,  $0 \leq k \leq n$ , envolvem as mesmas turmas e portanto na mesma ordem. Um problema de horário com  $n$  turmas tem  $n!$  espaços de estado distintos, sob orde

nação dinâmica restrita. A árvore de um dado espaço de estados é caracterizada pela ordem em que as turmas vão sendo associadas aos seus níveis. Na reordenação dinâmica irrestrita, os horários parciais de ordem  $k$ ,  $0 \leq k \leq n$ , podem diferir pelas turmas que envolvem, ou pela ordem em que foram usadas.

No restante do presente trabalho usaremos ordenação dinâmica restrita.

### III.6. A CARDINALIDADE DO ESPAÇO DE ESTADOS

#### III.6.1. INTRODUÇÃO

Consideremos uma instância do problema de horário com  $n$  turmas  $t_1, \dots, t_n$ . Seja  $H_i$  o conjunto de horários da turma  $t_i$  considerada isoladamente, como descrito no início da subseção III.5.1 e seja  $d_i$  a cardinalidade de  $H_i$ , para  $1 \leq i \leq n$ . Sem perda de generalidade suporemos que é conexo o grafo de conflito da instância dada do problema. Propomo-nos estabelecer uma expressão para a cardinalidade do espaço de estados gerado por um algoritmo de backtrack que no nível  $k$  aplica as funções intermediárias  $A_k$  e  $B_k$ , supondo uma ordenação dinâmica restrita das  $n$  turmas dadas. Como já mencionado, isto significa que o  $k$ -ésimo componente de todos os horários parciais de ordem  $k$  gerados é sempre um horário da mesma turma  $t_k$ , isto é um par  $(t_k, h_{k,j})$  onde  $h_{k,j} \in H_k$ ,  $0 < k \leq n$ . Assim, ao estender o horário parcial de ordem zero para horários parciais de ordem 1, tomamos sempre horários da turma  $t_1$ ; ao estender os horários



parciais de ordem 1 para os de ordem 2, tomamos sempre horários da turma  $t_2$  e etc. e ao estender horários parciais de ordem  $(n-1)$  para formar os de ordem  $n$  tomaremos horários de turma  $t_n$ . Nestas condições teremos  $n!$  espaços de estados distintos, um para cada uma das  $n!$  permutações dos inteiros de 1 e  $n$ . Dado um espaço de estados  $\psi$ , fica perfeitamente definida a turma cujos horários constituem o  $k$ -ésimo componente dos horários parciais de ordem  $k$  e de ordens maiores do que  $k$ .

Por concisão chamaremos de vértices viáveis (inviáveis) os que representam horários parciais viáveis (inviáveis). Finalmente, suporemos que a instância dada do problema tem pelo menos uma solução, o que implica que, para  $1 \leq i \leq n$ ,  $d_i > 0$  e também são estritamente maiores do que zero a quantidade total de vértices e a quantidade de vértices viáveis, em cada nível da árvore.

Uma vez obtida a expressão da cardinalidade do espaço de estados, examinaremos em que ordem as turmas devem ser tomadas para reduzir a cardinalidade do espaço de estados gerado.

### III.6.2. AS FUNÇÕES $q(\psi, k)$ e $r(\psi, k)$

Suponhamos então construída a árvore do espaço de estados  $\psi$ , gerada na busca de todas as soluções de uma dada instancia do problema. O espaço  $\psi$  é obtido tomando as turmas numa certa ordem que representamos por  $t_1, t_2, \dots, t_n$ . No nível zero desta árvore temos exatamente um vértice, correspondente ao horário parcial de ordem zero, que é sempre viável. No nível  $k$  desta árvore temos vértices viáveis e inviáveis.

DEFINIÇÃO III.21 - Representamos por  $s_{in}(\psi, k)$  a quantidade total de vértices (isto é, viáveis e inviáveis) no nível  $k$  da árvore do espaço de estados  $\psi$ . Analogamente,  $s_{out}(\psi, k)$  representa a quantidade de vértices viáveis no nível  $k$  da mesma árvore.

Infelizmente não se sabe determinar antecipadamente os valores de  $s_{in}(\psi, k)$  e de  $s_{out}(\psi, k)$  a partir dos dados de uma instância do problema e da ordem em que as turmas são tomadas pelo algoritmo de backtrack para a construção dos horários parciais. Sabemos apenas que

$$s_{in}(\psi, 0) = s_{out}(\psi, 0) = 1 \quad \text{e} \quad s_{in}(\psi, 1) = d_1 .$$

Assim, adotamos a atitude mais simples de supor construída a árvore e contar todos os vértices, e os vértices viáveis, em cada nível  $k$ . Obteremos assim os valores exatos de  $s_{in}(\psi, k)$  e  $s_{out}(\psi, k)$  que, neste sentido, ficam bem definidos. Usamos os subscritos "in" e "out" porque ao percorrer a árvore da raiz para as folhas,  $s_{in}(\psi, k)$  representa a quantidade de vértices que "chegam" ao nível  $k$  enquanto  $s_{out}(\psi, k)$  é o número de vértices do nível  $k$  que tem sucessores, e é por meio destes vértices que passamos do nível  $k$  ao nível  $(k+1)$  da árvore. Como por hipótese a instância dada do problema tem pelo menos uma solução,

$$0 < s_{out}(\psi, k) \leq s_{in}(\psi, k)$$

para todo os espaços de estado  $\psi$  e todo nível  $k$ ,  $0 \leq k \leq n$ .

Vejamos agora o que sucede quando estendemos os horários parciais viáveis de ordem  $k$  para obter os horários parciais (viáveis e inviáveis) de ordem  $(k+1)$ , tomando horários da turma  $t_{k+1}$ . Ora, fixado um horário parcial de ordem  $k$ , em

geral apenas uma parte dos  $d_{k+1}$  horários de  $H_{k+1}$  é compatível com os horários já selecionados para cada uma das  $k$  turmas passadas. Mais ainda, em geral são diferentes os subconjuntos de  $H_{k+1}$  que são formados pelos horários de  $t_{k+1}$  ainda compatíveis com os diferentes horários parciais de ordem  $k$ . Este aliás é o efeito da preclusão proporcionada pelas funções intermediárias  $A_{k+1}$ . Assim, o número total de vértices no nível  $(k+1)$ , representado por  $s_{in}(\psi, k+1)$ , satisfaz a desigualdade

$$0 < s_{in}(\psi, k+1) \leq s_{out}(\psi, k) \cdot d_{k+1}$$

que pode ser reescrito na forma

$$0 < s_{in}(\psi, k+1) / (s_{out}(\psi, k) \cdot d_{k+1}) \leq 1 .$$

Os termos que figuram no numerador e no denominador da expressão acima são bem definidos e não nulos. Representaremos por  $q(\psi, k+1)$  o valor desta razão, isto é, por definição

$$q(\psi, k+1) = s_{in}(\psi, k+1) / (s_{out}(\psi, k) \cdot d_{k+1})$$

e assim

$$0 < q(\psi, k+1) \leq 1 .$$

Multiplicando por  $d_{k+1}$  os membros da última igualdade teremos

$$q(\psi, k+1) \cdot d_{k+1} = s_{in}(\psi, k+1) / s_{out}(\psi, k)$$

o que nos permite interpretar  $q(\psi, k+1) \cdot d_{k+1}$  como a quantidade média de sucessores no nível  $(k+1)$ , por cada vértice viável do nível  $k$ . Em outras palavras, é a quantidade média de horários da turma  $t_{k+1}$  que podem ser usados para estender horários parciais viáveis de ordem  $k$ , gerando horários parciais (viáveis e inviáveis) de ordem  $(k+1)$ . Também podemos interpretar  $q(\psi, k+1)$  como a fração dos horários de  $t_{k+1}$  que são compatíveis

com os horários já selecionados para as  $k$  turmas passadas, em relação ao número de horários inicialmente disponíveis pela turma corrente  $t_{k+1}$ .

Uma vez obtido cada vértice do nível  $(k+1)$ , devemos aplicar a função intermediária  $B_{k+1}$  para determinar quais destes vértices são viáveis. Seja  $s_{out}(\psi, k+1)$  a quantidade de vértices viáveis e  $s_{in}(\psi, k+1)$  a quantidade total de vértices neste nível  $(k+1)$ . Aqui completamos a construção de um novo nível da árvore e assim teremos novamente

$$0 < s_{out}(\psi, k+1) \leq s_{in}(\psi, k+1) ,$$

que pode ser reescrito na forma

$$0 < s_{out}(\psi, k+1) / s_{in}(\psi, k+1) \leq 1$$

Ambos os termos da expressão acima são bem definidos e convém representá-la por  $r(\psi, k+1)$  fazendo

$$r(\psi, k+1) = s_{out}(\psi, k+1) / s_{in}(\psi, k+1) .$$

Naturalmente,  $0 < r(\psi, k+1) \leq 1$ .

A função  $r(\psi, k+1)$  traduz o fato de que dentre os horários de  $t_{k+1}$  compatíveis com os horários já selecionados para as  $k$  turmas passadas, e que dão lugar aos  $s_{in}(\psi, k+1)$  vértices do nível  $(k+1)$ , apenas a fração  $r(\psi, k+1)$  também é compatível com os horários das  $(n-(k+1))$  turmas futuras, no sentido de que cada uma delas continue tendo pelo menos um horário disponível.

Podemos resumir esta análise como segue. Dado um espaço de estados  $\psi$ , a cada turma corrente selecionada para o nível  $(k+1)$  são associadas as funções

$$q(\psi, k+1) \quad \text{e} \quad r(\psi, k+1) .$$

A primeira exprime quantitativamente o efeito da preclusão so

bre os horários da turma  $t_{k+1}$ , efetuada pela função intermediária  $A_{k+1}$  e a segunda traduz quantitativamente o efeito da poda da árvore realizada pela função intermediária  $B_{k+1}$ . O valor de  $q(\psi, k+1)$  é determinado pelas turmas passadas enquanto o valor de  $r(\psi, k+1)$  é determinado pelas turmas futuras. Segue-se a definição formal.

DEFINIÇÃO III.22 - As funções  $q(\psi, k)$  e  $r(\psi, k)$  são definidas pelas igualdades abaixo:

$$q(\psi, k) = s_{in}(\psi, k) / (s_{out}(\psi, k-1) \cdot d_k) , \quad 0 < k \leq n$$

$$r(\psi, k) = s_{out}(\psi, k) / s_{in}(\psi, k) , \quad 0 \leq k \leq n$$

onde  $k$  é um nível genérico da árvore do espaço de estado  $\psi$ .

As funções  $q$  e  $r$ , apesar da simplicidade das respectivas definições são bastante úteis no exame da árvore do espaço de estados. Apresentam algumas propriedades interessantes que serão estabelecidas e utilizadas no restante deste capítulo. Algumas propriedades triviais são reunidas aqui para efeito de referência.

Para toda árvore do espaço de estados  $\psi$  e nível  $k$ :

i)  $0 < q(\psi, k) \leq 1$  ,  $0 < k \leq n$

ii)  $0 < r(\psi, k) \leq 1$  ,  $0 \leq k \leq n$

iii)  $q(\psi, 1) = 1$

iv)  $r(\psi, 0) = 1$

v)  $r(\psi, n) = 1$

III.6.3. UMA EXPRESSÃO PARA A CARDINALIDADE DO  
ESPAÇO DE ESTADOS

Seja  $\psi$  o espaço de estados onde os horários da turma  $t_i$  constituem o  $i$ -ésimo componente dos horários parciais de ordem  $i$ , para  $1 \leq i \leq n$ . Isto é, os vértices do nível  $i$  da árvore são obtidos pela escolha de horários para a turma  $t_i$ .

TEOREMA III.2 - A quantidade de vértices no nível  $k$  da árvore do espaço de estados  $\psi$ , para  $0 < k \leq n$ , é dada por

$$s_{in}(\psi, k) = \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

Demonstração

Será feita por indução sobre o nível  $k$  da árvore.

Para  $k=1$ , a turma corrente é  $t_1$  e como ainda não temos turmas passadas, todos os  $d_1$  horários da turma  $t_1$  podem ser usados para formar os horários parciais de ordem 1. Assim, teremos  $d_1$  vértices no nível 1. Por outro lado, para  $k=1$  a expressão proposta fornece

$$s_{in}(\psi, 1) = r(\psi, 0) \cdot q(\psi, 1) \cdot d_1 = d_1$$

pois  $r(\psi, 0) = q(\psi, 1) = 1$

Assim, a expressão proposta fornece o número correto de vértices para o nível 1. Suponhamos então que seja válida até e inclusive o nível  $(k-1)$ . Portanto

$$s_{in}(\psi, k-1) = \prod_{i=1}^{k-1} \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

Multiplicando  $s_{in}(\psi, k-1)$ , o número de vértices no nível  $(k-1)$ , por  $r(\psi, k-1)$  obteremos o número de vértices viáveis no nível  $(k-1)$ , o qual multiplicado por  $\{q(\psi, k) \cdot d_k\}$  nos dará  $s_{in}(\psi, k)$ , o número total de vértices no nível  $k$ . Teremos então

$$s_{in}(\psi, k) = r(\psi, k-1) \cdot q(\psi, k) \cdot d_k \cdot s_{in}(\psi, k-1) \quad \text{donde}$$

$$s_{in}(\psi, k) = r(\psi, k-1) \cdot q(\psi, k) \cdot d_k \cdot \prod_{i=1}^{k-1} \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

Ora, os três fatores que prêmultiplicam o produtório são o seu argumento para  $i = k$  e portanto

$$s_{in}(\psi, k) = \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

o que demonstra o teorema.

TEOREMA III.3 - Na árvore do espaço de estados  $\psi$ , a quantidade total de vértices até e inclusive o nível  $m$ , para  $1 \leq m \leq n$ , é dada por

$$N(\psi, m) = 1 + \sum_{k=1}^m \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

### Demonstração

Será feita por indução sobre o valor de  $m$ .

Com efeito, para  $m=1$  a árvore é constituída pela raiz e pelos  $d_1$  vértices do nível 1. Assim,  $N(\psi, 1) = 1 + d_1$ , que é o valor da expressão proposta para  $m=1$ . Supondo que a expressão seja verdadeira para  $2, 3, \dots, (m-1)$ , teremos

$$N(\psi, m-1) = 1 + \sum_{k=1}^{m-1} \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

Por outro lado, o valor de  $N(\psi, m)$  é dado por

$$N(\psi, m) = N(\psi, m-1) + \prod_{i=1}^m \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

onde o produtório é a quantidade de vértices no nível  $m$  da árvore, de acordo com o teorema (III.2). Substituindo o valor de  $N(\psi, m-1)$  na última igualdade pela sua expressão teremos

$$N(\psi, m) = 1 + \sum_{k=1}^{m-1} \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\} + \prod_{i=1}^m \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

donde

$$N(\psi, m) = 1 + \sum_{k=1}^m \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

o que demonstra o teorema.

Com base no teorema (III.3) obtemos a seguinte expressão para a cardinalidade do espaço de estados  $\psi$

$$N(\psi, n) = 1 + \sum_{k=1}^n \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}.$$

TEOREMA III.4 - Dada uma instância do problema de horário com  $n$  turmas e pelo menos uma solução, o valor de

$$\prod_{i=1}^n \{r(\lambda, i-1) \cdot q(\lambda, i) \cdot d_{(a_i)}\}$$

é invariante sobre todos os  $n!$  espaços de estados  $\lambda$ , obtidos a partir do espaço  $\psi$  por reordenação dinâmica restrita das  $n$  turmas;  $(a_1, a_2, \dots, a_n)$  é uma permutação dos inteiros de 1 a  $n$ .



Demonstração

Consideremos a árvore do espaço de estados  $\lambda$  onde a turma  $t_{a_i}$  é associada ao nível  $i$ , sendo  $(a_1, a_2, \dots, a_n)$  uma permutação dos inteiros de 1 a  $n$ .

Pelo teorema (III.2)

$$s_{in}(\lambda, n) = \prod_{i=1}^n \{r(\lambda, i-1) \cdot q(\lambda, i) \cdot d_{a_i}\} \quad e$$

$$s_{in}(\psi, n) = \prod_{i=1}^n \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

Como todos os horários parciais parciais de ordem  $n$  são viáveis porque não mais existem turmas futuras, segue-se que  $s_{in}(\lambda, n) = s_{in}(\psi, n)$  pois ambos exprimem a quantidade de soluções de uma mesma instância do problema de horário. Assim

$$\prod_{i=1}^n \{r(\lambda, i-1) \cdot q(\lambda, i) \cdot d_{a_i}\} = \prod_{i=1}^n \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

o que prova o teorema.

TEOREMA III.5 - Nas mesmas hipóteses do teorema (III.4), o valor de

$$\prod_{i=1}^n \{r(\lambda, i-1) \cdot q(\lambda, i)\}$$

também é invariante.

Demonstração

Pelo teorema III.4

$$\prod_{i=1}^n \{r(\lambda, i-1) \cdot q(\lambda, i)\} \cdot \prod_{i=1}^n d_{a_i} = \prod_{i=1}^n \{r(\psi, i-1) \cdot q(\psi, i)\} \cdot \prod_{i=1}^n d_i$$

Como  $\prod_{i=1}^n d_{a_i} = \prod_{i=1}^n d_i$  e são não nulos por hipótese, fica provada a invariância de  $\prod_{i=1}^n \{r(\lambda, i-1) \cdot q(\lambda, i)\}$  nas condições propostas.

III.7. A REDUÇÃO DA CARDINALIDADE DO ESPAÇO DE ESTADOS

Nesta seção estabelecemos um critério para orientar a escolha da turma que deve ser associada ao nível  $(k+1)$  da árvore do espaço de estados, de modo a reduzir o número total de vértices neste nível, em face da escolha já realizada para os  $k$  primeiros níveis.

TEOREMA III.6 - Consideremos uma instância do problema de horário com  $n$  turmas. Seja  $\lambda$  o espaço de estados onde os horários da turma  $t_i$  constituem o  $i$ -ésimo componente dos horários parciais de ordem  $i$ ,  $1 \leq i \leq k$ . Seja  $\theta$  o espaço de estados onde os horários da turma  $t_{(a_i)}$  constituem o  $i$ -ésimo componente dos horários parciais de ordem  $i$ , onde  $\pi = (a_1, \dots, a_k)$  é uma permutação dos inteiros  $1$  a  $k$ . Então, a cada horário parcial viável de ordem  $k$  de  $\theta$  corresponde um horário parcial viável de ordem  $k$  de  $\lambda$  e vice-versa, e ambos usam os mesmos horários de turmas.

Demonstração

Seja  $\pi = (a_1, a_2, \dots, a_k)$  a permutação considerada dos inteiros de 1 até k. Para simplificar um pouco a notação, faremos  $a_1 = \alpha$ ,  $\dots$ ,  $a_k = \gamma$ . Seja

$$v_k^\theta = ((t_\alpha, h_{\alpha, j_\alpha}), \dots, (t_\gamma, h_{\gamma, j_\gamma}))$$

um horário parcial viável de ordem k de  $\theta$ .

Para provar o teorema mostraremos que existe em  $\lambda$  uma k-tupla  $v_k^\lambda$  com os mesmos componentes que em  $v_k^\theta$ , porém na ordem que caracteriza o espaço  $\lambda$ . Mostraremos que esta k-tupla é um horário parcial e que este horário parcial é viável. De pois trataremos da recíproca.

Por definição (III.16) e (III.17) os horários parciais de ordem k são k-tuplas de horários de turmas, isto é, de pares (turma, horário desta turma) que satisfazem o conjunto  $A_k$  de proposições; são viáveis aqueles horários parciais que satisfazem o conjunto  $B_k$  de proposições.

Consideremos a k-tupla

$$v_k^\lambda = ((t_1, h_{1, j_1}), \dots, (t_k, h_{k, j_k}))$$

obtida de  $v_k^\theta$  por uma permutação dos seus horários de turmas que consiste em colocar o par  $(t_\beta, h_{\beta, j_\beta})$  de  $v_k^\theta$ , onde  $\beta = a_i$ , como o  $a_i$ -ésimo componente de  $v_k^\lambda$ . Assim por exemplo, se

$$v_3^\theta = ((t_2, h_{2, j_2}), (t_3, h_{3, j_3}), (t_1, h_{1, j_1}))$$

então  $a_1 = 2$ ,  $a_2 = 3$  e  $a_3 = 1$  donde

$$v_3^\lambda = ((t_1, h_{1, j_1}), (t_2, h_{2, j_2}), (t_3, h_{3, j_3}))$$

É claro que a  $k$ -tupla  $v_k^\lambda$  sempre pode ser construída, é única e os seus horários de turmas estão na ordem que caracteriza o espaço  $\lambda$ . Resta mostrar que satisfaz  $A_k$  e  $B_k$ .

Ora, por construção  $v_k^\lambda$  e  $v_k^\theta$  contêm os mesmos horários de turmas e se  $A_k(v_k^\theta) = \text{VERDADEIRO}$  então também  $A_k(v_k^\lambda) = \text{VERDADEIRO}$  pois  $v_k^\lambda$  é uma enumeração dos mesmos horários de turma que em  $v_k^\theta$ , só que com outra ordenação. Assim,  $v_k^\lambda$  satisfaz  $A_k$  e portanto é um horário parcial de ordem  $k$ , restando mostrar que é viável. Como as turmas futuras em relação ao nível  $k$  são as mesmas nos espaços  $\lambda$  e  $\theta$ , segue-se que  $v_k^\lambda$  satisfaz  $B_k$  se  $B_k(v_k^\theta) = \text{VERDADEIRO}$ . Mais ainda, como em  $v_k^\lambda$  e  $v_k^\theta$  os horários de turmas são os mesmos, para cada turma futura seriam marcados como não disponíveis os mesmos horários, tanto em  $\lambda$  como em  $\theta$  e assim cada turma futura continua dispondo dos mesmos horários tanto em  $\lambda$  como em  $\theta$ . Assim, se  $B_k(v_k^\theta) = \text{VERDADEIRO}$  então também devemos ter  $B_k(v_k^\lambda) = \text{VERDADEIRO}$ . Resta a possibilidade de que, para algum  $i < k$ ,  $v_i^\lambda$  se tenha tornado inviável. Mas isto também não pode ser sustentado porque, sendo  $v_k^\lambda$  compatível com os horários das turmas futuras em relação ao nível  $k$ ,  $v_i^\lambda$  só poderia ser incompatível com o horário de alguma turma  $t_{i+1}, \dots, t_k$ , o que contradiria o fato de que  $v_k^\lambda$  satisfaz  $A_k$ . Segue-se portanto que  $v_k^\lambda$  também é viável.

Para provar a recíproca, bastaria tomar  $v_k^\lambda$ , um horário parcial viável de ordem  $k$  do espaço de estados  $\lambda$ , reordenar os horários de turmas segundo a permutação  $\pi$ , obtendo uma  $k$ -tupla  $v_k^\theta$ . Aplicando a  $v_k^\theta$  os argumentos já usados acima, mostramos que  $v_k^\theta$  é um horário parcial viável de ordem  $k$  no espaço de estados  $\theta$ , que usa os mesmos horários de turmas que  $v_k^\lambda$ . Isto conclui a demonstração deste teorema.

COROLÁRIO III.2 - As árvores dos espaços de estados  $\lambda$  e  $\theta$  especificados no teorema (III.6) possuem a mesma quantidade de vértices viáveis no nível  $k$ . Isto é,  $s_{out}(\lambda, k) = s_{out}(\theta, k)$ .

### Demonstração

Como para cada horário parcial viável de ordem  $k$  de  $\lambda$  existe um correspondente horário parcial viável de ordem  $k$  em  $\theta$ , e reciprocamente, segue-se que ambos os espaços de estado possuem a mesma quantidade de horários parciais viáveis de ordem  $k$ .

TEOREMA III.7 - Sejam  $\lambda$  e  $\theta$  dois espaços de estado como no teorema (III.6). Seja  $t_{k+1}$  a turma cujos horários constituem o  $(k+1)$ -ésimo componente das tuplas em  $\lambda$  e  $\theta$ . Então, a cada horário parcial de ordem  $(k+1)$  de  $\lambda$  corresponde um horário parcial de ordem  $(k+1)$  em  $\theta$  e, vice-versa, e ambos são formados pelos mesmos horários de turmas.

### Demonstração

Inicialmente lembramos que estes horários parciais de ordem  $(k+1)$  podem ser viáveis ou inviáveis.

Sejam  $v_k^\lambda$  e  $v_k^\theta$  os horários parciais viáveis de ordem  $k$  que usam os mesmos horários de turmas, como no teorema (III.6). Então, tanto no espaço  $\lambda$  como em  $\theta$  a turma  $t_{k+1}$  terá disponíveis os mesmos horários. Assim,  $v_k^\lambda$  e  $v_k^\theta$  podem ser estendidos com os mesmos horários da turma  $t_{k+1}$ . Se a  $(k+1)$ -tupla obtida estendendo  $v_k^\lambda$  satisfaz  $A_{k+1}$  então a  $(k+1)$ -tupla obtida esten-

dendo  $v_k^\theta$  com o mesmo horário de  $t_{k+1}$  também satisfará  $A_{k+1}$  e reciprocamente.

COROLÁRIO III.3 - Os espaços de estados  $\lambda$  e  $\theta$  do teorema (III.7), incluindo agora a turma  $t_{k+1}$ , terão a mesma quantidade de horários parciais de ordem  $(k+1)$ , isto é,  $s_{in}(\lambda, k+1) = s_{in}(\theta, k+1)$ .

#### Demonstração

Se a cada horário parcial de ordem  $(k+1)$  do espaço  $\lambda$  corresponde um horário parcial de ordem  $(k+1)$  de  $\theta$ , e vice-versa, então ambos os espaços terão a mesma quantidade de horários parciais de ordem  $(k+1)$ . Assim, as respectivas árvores terão a mesma quantidade de vértices no nível  $(k+1)$  donde

$$s_{in}(\lambda, k+1) = s_{in}(\theta, k+1)$$

TEOREMA III.8 - Como no teorema (III.7), consideremos as árvores dos espaços de estados  $\lambda$  e  $\theta$  e seja  $t_{k+1}$  a turma selecionada para estender os horários parciais viáveis de ordem  $k$  para horários parciais de ordem  $(k+1)$ . Então

$$q(\lambda, k+1) \cdot d_{k+1} = q(\theta, k+1) \cdot d_{k+1}$$

#### Demonstração

Por definição (III.22) da função  $q$  teremos

$$q(\lambda, k+1) \cdot d_{k+1} = s_{in}(\lambda, k+1) / s_{out}(\lambda, k)$$

$$q(\theta, k+1) \cdot d_{k+1} = s_{in}(\theta, k+1) / s_{out}(\theta, k)$$

Pelos corolários (III.2) e (III.3) os dois denominadores são iguais entre si nas duas expressões acima, bem como os dois numeradores donde, de fato

$$q(\lambda, k+1) \cdot d_{k+1} = q(\theta, k+1) \cdot d_{k+1}$$

o que demonstra o teorema

O teorema (III.8) mostra que é a escolha das  $k$  primeiras turmas, e não a ordem em que estas turmas são tomadas, quem determina o valor da função  $q$ , referido ao nível  $(k+1)$  da árvore, para cada uma das  $(n-k)$  turmas restantes. Neste sentido podemos dizer que o valor da função  $q$  para a turma corrente é determinado pelo conjunto das turmas passadas.

Por outro lado, o valor da função  $r$  do nível  $(k+1)$  somente pode ser determinado após a escolha da turma corrente, construção dos horários desta turma e exame da sua interação com os horários das turmas futuras conflitantes com a turma considerada. Neste sentido, dizemos que o valor da função  $r$  é determinado pela turma corrente e pelas turmas futuras.

Vejamos agora a redução da cardinalidade do espaço de estados. Sob reordenação dinâmica restrita, a passagem do nível  $k$  ao nível  $(k+1)$  da árvore é feita tomando-se uma determinada turma que designaremos por  $t_{k+1}$  e rotulando todas as arestas entre os níveis  $k$  e  $(k+1)$  com algum horário da turma  $t_{k+1}$ . Convém resumir num só teorema (III.9) os resultados já obtidos nos teoremas (III.6), (III.7) e (III.8), e seus corolários.

TEOREMA III.9 - Dada uma instância do problema de horário com  $n$  turmas, são invariantes, sobre todos os espaços de estado  $\psi$  obtidos por reordenação dinâmica restrita das turmas  $t_1, t_2, \dots, t_k$  associadas aos níveis 1 até  $k$ ,

- i) o número de horários parciais viáveis de ordem  $k$
- ii) o número total de horários parciais (viáveis e inviáveis) de ordem  $(k+1)$  que tem como  $(k+1)$ -ésimo componente horários da turma  $t_{k+1}$
- iii) o valor de  $q(\psi, k+1) \cdot d_{k+1}$  correspondente à esta turma  $t_{k+1}$ .

Este teorema (III.9) mostra que uma vez escolhidas as turmas cujos horários constituem os  $k$  primeiros componentes dos horários parciais, ficam determinados o número de vértices no nível  $k$  e, para cada turma futura  $t_{k+1}$  o valor de  $q(\psi, k+1) \cdot d_{k+1}$  que ela terá se for associada ao nível  $(k+1)$ . Como estes valores não dependem da ordem em que as  $k$  primeiras turmas foram selecionadas, conclui-se que o que realmente importa é quem são estas  $k$  primeiras turmas.

A redução da cardinalidade do espaço de estados deve visar a redução do número total de vértices, quer viáveis quer inviáveis. Assim, o problema geral é:

$$\text{Minimizar } N(\psi, n) = 1 + \sum_{k=1}^n \pi_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

sujeito ao resultado do teorema (III.4), segundo o qual

$$\sum_{i=1}^n \pi \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\} = \text{constante.}$$



Dentre as dificuldades de obter esta minimização está o fato de que cada valor  $r(\psi, k+1)$  somente é conhecido depois que se decide qual a turma  $t_{k+1}$  que será associada ao nível  $(k+1)$ , para ir construindo o espaço de estados  $\psi$ .

O algoritmo de backtrack começa com um horário parcial vazio e, em cada estágio procura estender um horário parcial viável de ordem  $k$ ,  $1 \leq k < n$ , para um horário parcial (viável ou inviável) de ordem  $(k+1)$ . Destarte abordaremos o problema da redução da cardinalidade do espaço de estados da seguinte forma: dadas as turmas já associadas aos níveis  $1$  até  $k$ , qual a melhor escolha da turma  $t_{k+1}$ , dentre todas as turmas futuras  $t_{k+1}$ ,  $t_{k+2}$ , ...,  $t_n$ ? Adotaremos então o seguinte critério: escolheremos para  $t_{k+1}$  aquela turma que conduz ao menor número de vértices (viáveis e inviáveis) no nível  $(k+1)$ . Esta turma é caracterizada pelo teorema (III.10).

TEOREMA III.10 - Seja  $\psi$  o espaço de estados que está sendo construído e sejam  $t_1, \dots, t_k$  as turmas já associadas aos níveis  $1, 2, \dots, k$  da árvore do espaço de estados  $\psi$ , para  $1 \leq k < n$ . Obtem-se o menor número total de vértices no nível  $(k+1)$  selecionado para este nível aquela turma  $t_{k+1}$  que dentre todas as turmas futuras possui o menor valor de  $q(\psi, k+1) \cdot d_{k+1}$ .

### Demonstração

Pela definição (III.22) o número total de vértices no nível  $(k+1)$  é dado por

$$s_{in}(\psi, k+1) = s_{out}(\psi, k) \cdot q(\psi, k+1) \cdot d_{k+1}$$

Pelo corolário (III.2), o valor de  $s_{out}(\psi, k)$  depende apenas das turmas já selecionadas para os níveis  $1, 2, \dots, k$  e não depende da ordem em que estas turmas foram tomadas. Pelo teorema (III.8), o valor de  $q(\psi, k+1) \cdot d_{k+1}$  é determinado pelas  $k$  turmas já selecionadas e independe da ordem em que elas foram associadas aos níveis  $1, 2, \dots, k$ . Assim, dadas estas  $k$  turmas, o valor de

$$s_{out}(\psi, k) \cdot q(\psi, k+1) \cdot d_{k+1}$$

depende apenas de  $q(\psi, k+1) \cdot d_{k+1}$ . Nestas condições  $s_{in}(\psi, k)$  será mínimo se e somente se for selecionada para o nível  $(k+1)$  aquela turma que tem o menor valor de  $q(\psi, k+1) \cdot d_{k+1}$ .

Este teorema mostra que podemos realizar uma minimização local, selecionado para cada nível aquela turma  $t_{k+1}$  que tem o menor valor de  $q(\psi, k+1) \cdot d_{k+1}$ . Não podemos concluir que se trata de uma minimização global porque a escolha da turma corrente para o nível  $(k+1)$  determina o valor da função  $r$  para este nível, e da função  $q$  para cada uma das turmas conflitantes com a turma selecionada para o nível  $(k+1)$ . Este aspecto será examinado mais detidamente na seção III.8.

A análise desenvolvida até aqui foi baseada numa árvore de backtrack totalmente construída. É importante ressaltar que o algoritmo de backtrack realiza uma busca em profundidade. Assim, ao passar pela primeira vez do nível  $k$  ao nível  $(k+1)$ , não se conhece o valor exato de  $q(\psi, k+1) \cdot d_{k+1}$  para cada uma das  $(n-k)$  turmas futuras. Como exposto na subseção III.6.2,  $q(\psi, k+1) \cdot d_{k+1}$  pode ser interpretado como o valor médio do número de horários disponíveis por uma turma  $t_{k+1}$ , su

pondo já construído todo o nível imediatamente anterior ( $k$ ). A cada instante o algoritmo de backtrack conhece a quantidade de horários disponíveis por cada turma futura, em relação a um dado horário parcial de ordem  $k$ . Embora possam ser construídas aproximações melhores, acumulando conhecimento adquirido em outros vértices do mesmo nível  $k$ , convém ter presente que a redução da cardinalidade é um meio para reduzir o esforço computacional. Assim, na implementação será selecionada para o nível  $(k+1)$  aquela turma que tiver a menor quantidade de horários disponíveis no momento em que esta decisão for tomada.

Por outro lado, o que importa é saber qual a turma que tem o menor valor de  $q(\psi, k+1) \cdot d_{k+1}$ , e não este valor exato. Como as quantidades de horários disponíveis por cada turma futura são comparadas entre si "mantendo fixo" um mesmo horário parcial de ordem  $k$ , é altamente possível que a ordenação baseada nos valores correntes seja muito próxima da ordenação que seria obtida com o conhecimento dos valores exatos. Mais ainda, se houver alguma diferença, ela certamente será entre turmas cujos valores de  $q(\psi, k+1) \cdot d_{k+1}$  são muito próximos.

### III.8. ALGUNS ASPECTOS HEURÍSTICOS

A redução da cardinalidade do espaço de estados apresentada na seção anterior foi baseada na relação entre a quantidade de vértices viáveis num dado nível  $k$  e a quantidade média de sucessores. A informação usada pelo algoritmo de back-track restringia-se ao valor de  $q(\psi, k+1) \cdot d_{k+1}$  para a turma a ser selecionada. Os resultados obtidos permitiram uma melhor compreensão do processo de busca e sugerem uma outra abordagem. Esta abordagem é baseada no teorema (III.4) segundo o qual o valor de

$$\prod_{i=1}^n \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

é invariante sobre todos os espaços de estado, para uma dada instância do problema de horário com  $n$  turmas; seja  $C$  este valor constante.

Consideremos então este invariante, reescrito de modo a evidenciar a parte correspondente às turmas passadas, que designaremos por  $x$ , a parte  $y$  referente ao valor de  $q(\psi, k+1) \cdot d_{k+1}$  da turma corrente e a parte  $Z$  que inclui  $r(\psi, k+1)$  e os fatores referentes às turmas futuras.

Assim

$$x = \prod_{i=1}^k \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\} \cdot r(\psi, k)$$

$$y = q(\psi, k+1) \cdot d_{k+1}$$

$$Z = \prod_{i=k+2}^n \{r(\psi, i-1) \cdot q(\psi, i) \cdot d_i\}$$

com  $x \cdot y \cdot z = C$ .

No momento em que estão escolhidas as  $k$  primeiras turmas, o valor de  $x$  fica irrevogavelmente determinado.

Inicialmente suponhamos que ao selecionar a turma corrente para o nível  $(k+1)$  de acordo com o teorema (III.10) encontramos duas turmas,  $t_{k+1}$  e  $t'_{k+1}$  que apresentam

$$q(\psi, k+1) \cdot d_{k+1} = q(\psi, k+1) \cdot d'_{k+1}$$

Isto implica que a escolha de  $t_{k+1}$  ou  $t'_{k+1}$  poderá afetar apenas os fatores de  $z$ , mantendo  $z$  com valor constante e já determinado por  $x \cdot y$ . Um critério de desempate pode basear-se em outras informações conhecidas sobre estas duas turmas, tais como

- a carga horária semanal
- o número de turmas futuras conflitantes com cada uma destas turmas.

O efeito da escolha da turma corrente manifesta-se sobre os fatores de  $z$  tanto pela valor de  $r(\psi, k+1)$  para o nível corrente  $(k+1)$ , como pela redução do número de horários disponíveis por cada turma futura conflitante com a turma corrente, isto é, pela redução dos respectivos valores de  $q(\psi, i) \cdot d_i$  em  $z$ .

O ponto importante é o seguinte: em geral

- o valor de  $r(\psi, k+1)$  decrece com o crescimento tanto da carga horária semanal como do número de turmas conflitantes com a turma corrente  $t_{k+1}$ .
- com o crescimento da carga horária de  $t_{k+1}$  decrece o valor de  $q(\psi, i) \cdot d_i$  de cada turma futura conflitante com  $t_{k+1}$ .
- quanto maior for o número destas conflitantes, maior será o número de fatores  $q(\psi, i) \cdot d_i$  de  $z$  que serão reduzidos.

Como o valor de  $z$  deve manter-se constante, segue-se o crescimento do valor do produto  $r(\psi, k+2) \cdot r(\psi, k+3) \cdot \dots \cdot r(\psi, n-1)$  na expressão  $z$ . Isto por sua vez indica que cresce a proporção de vértices viáveis em relação ao número total de vértices no restante da árvore. Por outro lado, uma redução mais severa de  $r(\psi, k+1)$  do nível corrente e dos valores de  $q(\psi, i) \cdot d_i$  de algumas turmas futuras podem conduzir a uma redução do número total de vértices (viáveis e inviáveis) no restante da árvore. O efeito combinado poderia ser uma redução efetiva do número de vértices inviáveis, e não apenas da fração que constituem em relação ao número total de vértices. Somente para ressaltar este efeito, observe-se que se fosse possível ir escolhendo em cada nível uma turma de modo que

$$\prod_{i=1}^n r(\psi, i-1) = 1$$

teríamos uma árvore do espaço de estados sem vértices inviáveis! Embora isto não possa ser conseguido para o problema de horário, em vista da sua complexidade, é certamente uma meta da qual devemos procurar aproximar-se.

Estas considerações, introduzidas sob o pretexto de uma regra de desempate, sugerem o uso de outros indicadores, mais informados para selecionar a turma corrente. Estes indicadores poderiam ser construídos usando o valor de  $q(\psi, k+1) \cdot d_{k+1}$ , a carga horária semanal, o número de aulas, e a quantidade de turmas futuras conflitantes com a turma a ser escolhida. Entretanto, nas instâncias do problema onde todas as turmas tem aproximadamente a mesma carga horária semanal, o mesmo número de aulas e cada turma conflita com todas as outras, este indi

cador teria pouca discriminação, além daquela proporcionada por  $q(\psi, k+1) \cdot d_{k+1}$ .

Finalmente, como pretendemos usar o algoritmo de back-track para obter uma solução da instância dada do problema, passa a ser mais importante a ordem em que os horários de uma dada turma são selecionados para a formação de horários parciais. Por ser realizada sobre um mesmo nível da árvore, esta reordenação poderia ser denominada de "horizontal". Este aspecto não foi examinado no presente trabalho.

## CAPÍTULO IV - O PROGRAMA PROTÓTIPO

Neste capítulo apresentamos uma breve descrição do programa que foi construído. O desenvolvimento deste programa teve mais as características de pesquisa do que implementação pura e simples de um programa totalmente especificado segundo o estado da arte. Além de confirmar que é possível construir um programa de acordo com o desenvolvimento teórico apresentado no capítulo III, esta implementação permitiu identificar algumas dificuldades de ordem prática, solucionadas satisfatoriamente, e levantar um aspecto teórico para pesquisa posterior.

### IV.1. DESCRIÇÃO GERAL

O programa foi escrito em FORTRAN IV por questões de portabilidade. Compõe-se de um programa principal e 32 subrotinas, perfazendo 4500 comandos. No computador B6700 da BURROUGHS demanda 60s de compilação e o código resultante ocupa 5500 palavras.

Os dados de entrada são organizadas essencialmente como discutido de forma abstrata nas seções III.2 e III.3 e resumidos em III.3.3. Para facilitar a criação de relações de conflito arbitrárias entre classes, o que é útil em experiências, foram usadas as cliques de turmas (definição III.15).

A medida que os dados vão sendo lidos, procede-se a uma crítica detalhada, cuja eficiência e utilidade foram comprovadas nas vezes em que o programa foi usado. Obtidos os dados,



e se permitido pela ausência de erros, passa-se à fase de montagem das estruturas usadas em fases subsequentes. A instância dada do problema é decomposta em seus componentes conexos, de acordo com a seção III.3.4. Passa-se então à construção do horário de cada componente por vez, usando o algoritmo de back-track descrito em III.5. Inicialmente determina-se se quantidade de horários disponíveis por cada turma  $t_i$  do componente, isto é os valores de  $d_i$  descritos no início da seção III.6.1; a construção destes horários é comentada mais adiante, em IV.2. A seguir seleciona-se a turma com o menor valor de  $q(\psi, l) \cdot d_i$  que fica então definitivamente associada ao nível 1 da árvore de busca. O algoritmo prossegue a busca em profundidade e para cada nível seleciona-se aquela turma que, naquele instante, possui a menor quantidade de horários disponíveis, de acordo com a análise apresentada ao final da seção III.7.

Em cada componente pode ser atingida qualquer uma das duas situações: ou foi obtido um horário para a instância constituída pelas turmas do componente considerado, ou então foi atingido um nível, digamos  $j$ , além do qual é impossível passar por não existir horário parcial de ordem  $j$  que satisfaça  $B_j$ . Neste último caso, arbitrariamente é excluída do componente a turma que havia sido associada ao nível  $j$  e é impressa uma mensagem. Como esta exclusão pode fazer com que o componente considerado deixe de ser conexo, determinam-se os seus componentes conexos e o programa retoma o(s) componente(s) resultante(s) desta exclusão. Ao final do processamento de cada componente é emitido um relatório indicando quais as turmas excluídas, se alguma, e os horários obtidos para as demais. Os componentes vão sendo processados um a um até que todos tenham sido

processados. Ao término, é emitido um relatório geral com as informações sobre cada componente do problema dado.

A saída é bastante detalhada indicando o horário de cada turma dada, respectivo professor, classe e disciplina. Mais ainda, para cada clique de turmas o horário é apresentado sob a forma de matriz horas x dias da semana. Embora outros relatórios possam ser produzidos, como por exemplo os horários de todas as turmas de uma mesma disciplina, não o foram pois não há nenhuma dificuldade em fazê-lo.

#### IV.2. ALGUNS DETALHES DE IMPLEMENTAÇÃO

Segue-se a descrição de uns poucos detalhes de implementação, alguns dos quais - itens b e c - ofereceram certa dificuldade até serem equacionados satisfatoriamente.

a) A disponibilidade de cada turma é obtida pela interseção das disponibilidades dos seus recursos, sendo determinada de uma vez só antes de ser iniciado o backtrack.

b) Os horários disponíveis por cada turma (que constituem os conjuntos  $H_i$  do início da seção III.5.1) podem ser bastante numerosos. Ao invés de construí-los todos de uma só vez para uso posterior, desdobra-se este processo em duas etapas. Primeiro, constroem-se as "distribuições" e durante o processo de backtrack vão sendo gerados os horários possíveis em cada distribuição, em face dos períodos ainda disponíveis. Cada "distribuição" é, simplesmente, uma tupla de dias da semana em que a

turma considerada pode ter aula, em face das características qualitativas (final da seção III.3.2) e da disponibilidade obtida para a turma. Assim por exemplo uma turma com três aulas por semana de cinco dias teria até  $C_5^3 = 10$  combinações de dias da semana do tipo  $(2^a, 4^a, 5^a)$ ,  $(2^a, 4^a, 6^a)$ , etc. Cada combinação destas é uma "distribuição". Algumas destas dez distribuições seriam eventualmente rejeitadas por usarem todos os dias consecutivos como por exemplo  $(2^a, 3^a, 4^a)$ .

c) O status de um período da disponibilidade de uma turma refere-se à possibilidade de sua utilização para as aulas da turma considerada. Por exemplo, este status poderia ser não disponível, disponível, não mais disponível porque usado pelo horário de alguma turma passada e conflitante com a turma considerada. O problema está no fato de que, quando o algoritmo de backtrack regride, deve ser mudado o status de certos períodos contidos na disponibilidade de turmas futuras, conflitantes com a turma corrente. Os períodos que devem ter mudado o status são apenas aqueles que são comuns ao horário abandonado da turma corrente e que foram marcados como não mais disponíveis exatamente devido a este horário. Isto é implementado marcando cada período que inicialmente estava disponível, com o número do nível onde estava o algoritmo de backtrack no momento em que este deixou de ser disponível por cada turma futura conflitante com a turma considerada. Mais ainda, se algum período já está assim marcado ele não é marcado de novo quando o algoritmo avança. Assim, quando o algoritmo regride do nível  $k$ , somente voltam a ser disponíveis os períodos marcados com  $k$  na disponibilidade de cada turma futura conflitante com a turma associada ao nível  $k$ .

d) Vejamos agora o teste de compatibilidade do horário da turma corrente em relação às turmas futuras. A cada vez que é selecionado um horário para a turma corrente, é necessário fazer a marcação descrita e verificar se cada turma futura conflitante com a turma corrente continua tendo pelo menos um horário disponível, como apresentado na seção III.5.1. Isto é feito mantendo as turmas conflitantes com a turma corrente numa lista autoorganizada, de tal maneira que ao ser encontrada uma turma futura que deixa de ter horários disponíveis, ela é passada ao início da lista. Assim, ela será a primeira a ser testada ao ser selecionado um novo horário para a turma corrente.

#### IV.3. LIMITAÇÕES

Foram feitas três simplificações na construção deste programa protótipo. São elas: as aulas de cada disciplina são de igual duração e até três por semana, embora possam diferir de turma para turma. Omitiu-se o tratamento de equipamentos especiais na especificação das turmas.

#### IV.4. UMA QUESTÃO PARA ANÁLISE

Durante a construção do programa surgiu a seguinte questão: se o algoritmo de backtrack precisar regredir, será sempre necessário regredir ao nível imediatamente anterior? Designemos por  $t_{k+1}$  a turma associada ao nível  $(k+1)$ . Se for impossível obter um horário para a turma corrente  $t_{k+1}$  então se

r  necess rio regredir. Mas observe-se que, se a turma  $t_k$  n o conflita com a turma  $t_{k+1}$  hem com nenhuma turma futura conflitante com  $t_{k+1}$  ent o podemos regredir diretamente pelo menos ao n vel  $(k-1)$ . A quest o  : dado um n vel  $(k+1)$  ao qual est  associada a turma  $t_{k+1}$ , como caracterizar o n vel mais pr ximo da raiz ao qual   poss vel regredir diretamente, e sem eliminar nenhuma solu o? Embora isto tenha sido resolvido na implementa o, demandaria uma an lise mais formal.

#### IV.5. RESULTADOS

Em face do grande volume de trabalho investido no desenvolvimento deste programa prot tipo, foram bastante limitadas as experi ncias j  realizadas. Ainda assim foi poss vel obter uma boa ilustra o do efeito da reordena o din mica restrita das turmas proposta pelo teorema (III.10): ao avan ar pela primeira vez do n vel  $k$  ao n vel  $(k+1)$  devemos associar ao n vel  $(k+1)$  aquela turma que tem menos hor rios dispon veis.

A tabela (IV.1) foi produzida pelo programa junto com a solu o de um problema exemplo constituido por cinco turmas, todas conflitantes entre si, e designadas pelos n meros 101, 102, 103, 104, 105 apresentadas ao programa nesta mesma ordem.

TABELA IV.1

NÍVEL	TURMA	NUMERO DE HORÁRIOS DISPONIVEIS	
		INICIAL	FINAL
1	104	12	12
2	102	22	15
3	103	19	14
4	105	104	11
5	101	90	3

A coluna 1 indica o nível da árvore ao qual a turma da coluna 2 foi associada. A coluna 3 indica a quantidade de horários disponíveis por cada turma da coluna 2, independentemente de qualquer outra turma. Finalmente a coluna 4 mostra o número de horários que cada uma tinha no momento em que foi selecionada para ser associada ao nível indicado na coluna 1.

Em primeiro lugar, se por exemplo a turma nº 101 tivesse sido associada ao nível 1, a árvore teria 90 vértices neste nível, ao invés dos 12 indicados na tabela. Por outro lado, a turma nº 105 inicialmente tinha 104 alternativas de horário. Foi associado ao nível 4 e neste instante tinha (aproximadamente) 11 horários disponíveis, pois os demais 93 horários precisariam dos períodos já usados pelos horários das três turmas já associadas aos níveis anteriores da árvore.

## CAPÍTULO V - CONCLUSÕES E SUGESTÕES PARA PESQUISA

Dividimos este capítulo em três partes: conclusões referentes à parte teórica, à implementação e sugestões para pesquisa.

### V.1. CONCLUSÕES SOBRE A PARTE TEÓRICA

- a) Foi desenvolvida uma formulação do problema de horário de aulas incluindo as restrições intrínsecas do problema e algumas características qualitativas - seções III.2 e III.3.
- b) Quanto à escolha do espaço de estados adotamos como unidade a turma e alocamos de uma só vez todas as aulas de cada turma antes de passar à próxima. Esta escolha revelou-se adequada pela facilidade com que certos aspectos qualitativos referentes ao horário puderam ser incorporados e pela relativa simplicidade deste tratamento no programa implementado.
- c) Com base nesta formulação foi proposto um algoritmo de backtrack e definido rigorosamente o seu espaço de estados - seção III.5 - de modo a permitir uma análise quantitativa.
- d) A análise do espaço de estados - seção III.6 - permitiu definir as funções  $q(\psi, k)$  e  $r(\psi, k)$ , que exprimem quantitativamente o efeito da aplicação das funções intermediárias  $A_k$  e  $B_k$  do algoritmo de backtrack.

- e) Foi estabelecido rigorosamente um critério para minimizar o número total de vértices no nível  $(k+1)$  da árvore de back-track, dadas as turmas já selecionadas para os  $k$  primeiros níveis - teorema III.10. Segundo a discussão do final da seção III.7, este critério pode ser implementado satisfatoriamente com base no número de horários disponíveis por cada turma futura.
- f) Foi estabelecido um invariante envolvendo as funções  $q(\psi, k)$  e  $r(\psi, k)$  - teorema III.4. Este invariante permite examinar o efeito de heurísticas para a redução da cardinalidade do espaço de estados, como descrito em III.8.
- g) O critério geral de tomar como a próxima turma aquela que tem menos horários disponíveis, decorrente do teorema III.10, tem o mérito de incluir com naturalidade as turmas com horários prefixados, pois estas tem exatamente um horário disponível. Lembramos que muitas abordagens exigem um tratamento particular para as turmas com horários prefixados.
- h) Analogamente, o invariante do teorema III.4 e a discussão sobre aspectos heurísticos, justificam por meios quantitativos porque são adequadas certas heurísticas no tratamento preferencial de turmas com maior carga horária semanal ou maior número de turmas conflitantes.



## V.2. CONCLUSÕES SOBRE A IMPLEMENTAÇÃO

- a) Em face do tamanho do programa obtido, é preferível admitir que qualquer programa de construção de horários de aula de verá ser implementado como um sistema constituído por vários programas correlatos, mas independentes.
- b) Seria bem mais desejável construir um sistema interativo que permita ao usuário ir retirando ou acrescentando restrições, para obter horários qualitativamente cada vez mais satisfatórios.
- c) Os poucos resultados obtidos mostram que é viável implementar um programa baseado no desenvolvimento teórico apresentado no capítulo III e ilustram o efeito favorável da reordenação adotada, em termos de redução da cardinalidade do espaço de estados.

## V.3. SUGESTÕES PARA PESQUISA POSTERIOR

- a) Estabelecer critérios para decidir em que ordem os horários de cada turma devem ser usados, isto é, como realizar a ordenação "horizontal" mencionada na seção III.8.
- b) Estabelecer critérios envolvendo por exemplo a função  $q(\psi, k)$ ,  $d_k$ , a carga horária semanal, o número de turmas futuras conflitantes com a turma considerada, conforme seção III.8, para reduzir o tempo de processamento e não apenas a cardinalidade do espaço de estados.

- c) É condição necessária para que o espaço de estados seja mínimo que a turma associada ao nível  $k$  tenha menos alternativas de horário disponíveis que a turma associada ao nível  $(k+1)$ , ambas referidas ao nível  $(k-1)$ ?
- d) A questão do nível mais próximo da raiz, ao qual se poderia regredir, como mencionado ao final da seção IV.4.
- e) No algoritmo de backtrack proposto, no momento em que todas as turmas conflitantes com uma turma dada já estão alocadas, o horário da(s) turma(s) que resta alocar pode ser escolhido livremente entre os horários por ela(s) disponíveis. Seja  $U$  o conjunto de todas as turmas dadas e  $Y$  o subconjunto contendo as turmas tais que todas as suas conflitantes já foram associadas aos níveis da árvore e já tem horários de finidos. Com isto o algoritmo de backtrack poderia limitar-se as turmas do subconjunto  $X = U \setminus Y$  e teria apenas  $|X|$  níveis, com  $|X| < |U|$ . Como particionar  $U$  de modo que o espaço de estados referente apenas às turmas de  $X$  seja de cardinalidade, ou custo, mínimo?

BIBLIOGRAFIA

- (1) AHO, A.V., HOPCROFT, J.E., ULLMAN, J.A., The Design and Analysis of Computer Algorithms, Reading, Addison-Wesley Publ. Co., 1974.
- (2) AKKOYUNLU, E.A., "A Linear Algorithm for Computing the Optimum University Timetable", The Computer Journal, Vol. 16, n° 4, pp. 347-350, (1973).
- (3) AUST, R.J., "An Improvement Algorithm for School Timetabling", The Computer Journal, Vol. 19, n° 4, pp. 339-343, (1976).
- (4) BARRACLOUGH, E.D., "The Application of a Digital Computer to the Construction of Timetables", The Computer Journal, Vol. 8, pp. 136-146, (1965).
- (5) BERGE, C., Graphs and Hypergraphs, Amsterdam, North-Holland Publ. Co., 1973.
- (6) BRITTAN, J.N.G. e FARLEY, F.J.M., "College Timetable Construction by Computer", The Computer Journal, Vol. 14, n° 4, pp.361-365, (1971).
- (7) BROWN, C.A. e PURDOM, P.W., "An Average Time Analysis of Backtracking", SIAM J. Comp., Vol. 10, n° 3, pp. 583-593, (1981).
- (8) COOK, S.A., "An Overview of Computational Complexity", Comm. of the ACM, Vol. 26, n° 6, pp. 401-408, (1983).
- (9) COOK, S.A., e RECKHOW, R.A., "Time Bounded Random Access Machines", Journal of Computer and System Sciences, Vol. 7, pp. 354-375, (1973).

- (10) CSIMA, J., "Investigations on a Timetable Problem", Tese de Doutorado, Universidade de Toronto, Toronto, 1965.
- (11) CSIMA, J. e GOTLIEB, C.C., "Tests on a Computer Method for Constructing School Timetables", Comm. of the ACM, Vol. 7, n° 3, pp. 160-163, (1964).
- (12) DEMPSTER, M.A.H., "On the Gotlieb-Csima Timetabling Algorithm", Canadian J. Math. Vol. 20, pp. 103-119, (1968).
- (13) DEMPSTER, M.A.H., "Two Algorithms for the Timetable Problem", in WELSH, D.J.A. (ed.), Combinatorial Mathematics and its Applications, pp. 63-65, London, Academic Press, 1971.
- (14) DIJKSTRA, E.W., A Discipline of Programming, Englewood Cliffs, Prentice Hall, 1976.
- (15) DUNCAN, A.K., "Further Results on Computer Construction of School Timetables", Comm. of the ACM, Vol. 8, n° 1, pp. 72, (1965).
- (16) EVEN, S., ITAI, A., SHAMIR, A., "On the Complexity of Timetable and Multicommodity Flow Problems", SIAM J. Comput., Vol. 5, n° 4, (1976).
- (17) GAREY, M.R. e JOHNSON, D.S., "The Complexity of Near-Optimal Graph Coloring", J. of the ACM, Vol. 23, n° 1, pp. 43-49, (1976).
- (18) GAREY, M.R. e JOHNSON, D.S., Computers and Intractability - A Guide to the Theory of NP-Completeness, San Francisco, W.H. Freeman and C°, 1979.
- (19) GOLOMB, S.W. e BAUMERT, L.D., "Backtrack Programming", J. of the ACM, Vol.12, n° 4, pp.516-524, (1965).

- (20) GOTLIEB, C.C., "The Construction of Class-Teacher Timetables", Proc. IFIP Congress 1962 - Munich, pp. 73-77, Amsterdam, North-Holland Publ. C°, 1963.
- (21) GRIFFITH, B.A., "Remarks on a Computer Program for the Construction of School Timetables", Comm. of the ACM, Vol. 9, n° 1, pp. 35, (1966).
- (22) HALL, P., "On Representatives of Subsets", J. London Math. Soc., Vol. 10, pp. 26-30, (1935).
- (23) HARALICK, R.M. e ELLIOTT, G.L., "Increasing Tree Search Efficiency for Constraint Satisfaction Problems", Artificial Intelligence, Vol. 14, n° 3, pp. 263-313, (1980).
- (24) HARARY, F., Graph Theory, Reading, Addison-Wesley Publ.C°, 1969.
- (25) HOPCROFT, J.E. e ULLMAN, J.A., Introduction to Automata Theory, Languages and Computation, Reading, Addison-Wesley Publ. C°, 1979.
- (26) HOROWITZ, E. e SAHNI, S., Fundamentals of Computer Algorithms, Rockville, Computer Science Press, 1978.
- (27) JOHNSON, H.C. e WOLFENDEN, K., "Computer Aided Construction of School Timetables", in: MORREL, A.J.H. (ed.): Information Processing 68, Proc. IFIP Congress 1968, pp. 1368-1376, Amsterdam, North-Holland Publ. C°, 1969.
- (28) KNUTH, D.E., "Estimating the Efficiency of Backtrack Programs", Mathematics of Computation, Vol. 29, n° 129, pp. 121-136, (1975).

- (29) LAWRIE, N.L., "An Integer Linear Programming Model of a School Timetabling Problem", The Computer Journal, Vol. 12, pp. 307-316, (1969).
- (30) LIONS, J., "Matrix Reduction Using the Hungarian Method for the Generation of School Timetables", Comm. of the ACM, Vol. 9, n° 5, (1966).
- (31) LIONS, J., "A Counter-Example for Gotlieb's Method for the Construction of School Timetables", Comm. of the ACM, Vol. 9, n° 9, (1966).
- (32) LIONS, J., "The Ontário School Scheduling Program", The Computer Journal, Vol. 10, pp. 14-21, (1967).
- (33) LIONS, J., "Some Results Concerning the Reduction of Binary Matrices", J. of the ACM, Vol. 18, n° 3, pp. 424-430, (1971).
- (34) McDIARMID, C.J.H., "The Solution of a Timetabling Problem", J. Inst. Maths, Applics., Vol. 9, pp. 23-34, (1972).
- (35) NEUFELD, G.A. e TARTAR, J., "Generalized Graph Colorations", SIAM J. Appl. Math., Vol. 29, pp. 91-98, (1975).
- (36) NILSSON, N.J., Problem-Solving Methods in Artificial Intelligence, New York, McGraw-Hill, 1971.
- (37) PAPANIMITRIOU, C.H. e STEIGLITZ, K., Combinatorial Optimization: Algorithms and Complexity, Englewood Cliffs, Prentice-Hall, 1982.
- (38) PURDOM, Jr., P.W. e BROWN, C.A., "An Analysis of Backtracking with Search Rearrangement", SIAM J. Comput., Vol. 12, n° 4, pp. 717-733, (1983).

- (39) REINGOLD, E.M., NIEVERGELT, J., DEO, N., Combinatorial Algorithms: Theory and Practice, Englewood Cliffs, Prentice-Hall, 1977.
- (40) SCHMIDT, G. e STROHLEIN, T., "Timetable Construction - an Annotated Bibliography", The Computer Journal, Vol. 23, n° 4, pp. 307-315, (1980).
- (41) SMITH, G., "On Maintenance of the Opportunity List for Class-Teacher Timetable Problems", Comm. of the ACM, Vol. 18, n° 4, pp. 203-208, (1975).
- (42) STEWART, J. e CLARK, R.L., "University of Maryland Student Scheduling Program", Technical Report 68-83, Maryland, University of Maryland Computer Science Center, (1968).
- (43) SZWARCFITER, J.L., Grafos e Algoritmos Computacionais, Rio de Janeiro, Editora Campus, 1984.
- (44) TARJAN, R.E., "Complexity of Combinatorial Algorithms", SIAM Review, Vol. 20, n° 3, pp. 457-491, (1978).
- (45) WELSH, D.J.A. e POWELL, M.B., "An upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems", The Computer Journal, Vol. 10, pp. 85-86, (1967).
- (46) WOOD, D.C., "A System for Computing University Examination Timetables", The Computer Journal, Vol. 11, pp. 41-47, (1968).