

UNIVERSIDADE FEDERAL DE OURO PRETO  
ESCOLA DE MINAS  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

**Programação de Horários em Escolas via GRASP e Busca Tabu**

Monografia de graduação em Engenharia de Produção

Felippe Pereira da Costa

Ouro Preto  
Fevereiro/2003.

Felippe Pereira da Costa

## Programação de Horários em Escolas via GRASP e Busca Tabu

Monografia apresentada ao Departamento de Engenharia de Produção, Administração e Economia da Escola de Minas da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Produção.

Orientador: Prof. Dr. Marcene Jamilson Freitas Souza.

Ouro Preto  
Fevereiro/2003

## **AGRADECIMENTO**

A Deus por permitir que pudéssemos participar de mais esta experiência.

Aos pais, familiares e republicanos pelo apoio.

Ao Prof. Dr. Marcone Jamilson Freitas Souza, pela oportunidade de fazer parte deste trabalho e pelos conhecimentos transmitidos com tanta dedicação e disponibilidade.

Aos amigos Irce e Júlio pelos trabalhos realizados conjuntamente e demais companheiros de turma.

Ao CNPq pela bolsa de iniciação científica concedida ao projeto que deu origem a este trabalho.

À Escola Estadual Dom Velloso de Ouro Preto pela disponibilização das informações relativas ao processo de elaboração de horários.

## SUMÁRIO

<b>LISTA DE SIGLAS.....</b>	<b>5</b>
<b>LISTA DE FIGURAS .....</b>	<b>6</b>
<b>LISTA DE TABELAS.....</b>	<b>7</b>
<b>LISTA DE GRÁFICOS .....</b>	<b>8</b>
<b>RESUMO .....</b>	<b>9</b>
<b>1 INTRODUÇÃO.....</b>	<b>10</b>
1.1 OBJETIVOS .....	11
1.1.1 Objetivo Geral.....	11
1.1.2 Objetivos Específicos.....	11
1.2 ESTRUTURA DO TRABALHO.....	11
<b>2 REVISÃO BIBLIOGRÁFICA.....</b>	<b>12</b>
2.1 INTRODUÇÃO .....	12
2.2 VIZINHANÇA DE UMA SOLUÇÃO .....	13
2.3 MÍNIMO LOCAL.....	14
2.4 BUSCA LOCAL.....	14
2.4.1 Método de Descida.....	15
2.4.2 Método Randômico de Descida .....	15
2.5 METAHEURÍSTICAS UTILIZADAS.....	15
2.5.1 GRASP (Greedy Randomized Adaptive Search Procedure) .....	16
2.5.2 Busca Tabu.....	17
<b>3 LEVANTAMENTO DAS CARACTERÍSTICAS DA ESCOLA .....</b>	<b>18</b>
3.1 DESCRIÇÃO DA ESCOLA.....	18
3.2 LEVANTAMENTO DE DADOS.....	20
<b>4 METODOLOGIA .....</b>	<b>22</b>
4.1 REPRESENTAÇÃO DO PROBLEMA .....	22
4.2 AVALIAÇÃO DE UM QUADRO DE HORÁRIOS .....	23
4.2.1 Função de Avaliação .....	23
4.2.2 Hierarquia da Função de Avaliação .....	24
4.3 ESTRUTURA DE VIZINHANÇA.....	25
4.4 GERAÇÃO DE UMA SOLUÇÃO INICIAL.....	25
4.5 REFINAMENTO DE UMA SOLUÇÃO.....	26
4.6 O ALGORITMO HÍBRIDO .....	27
<b>5 RESULTADOS COMPUTACIONAIS.....</b>	<b>28</b>
5.1 COMPARAÇÃO ENTRE OS PROCEDIMENTOS TESTADOS.....	31
5.2 COMPARAÇÃO COM O QUADRO MANUAL.....	33
<b>6 CONCLUSÕES .....</b>	<b>36</b>
<b>7 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>37</b>

**LISTA DE SIGLAS**

ABT	- O procedimento Busca Tabu partindo de soluções iniciais geradas aleatoriamente
BT	- Busca Tabu
BTMAX	- Número máximo de iterações sem melhora no procedimento Busca Tabu
GBT	- O procedimento Busca Tabu partindo de soluções iniciais geradas pelo procedimento GRASP
GRASP	- Greedy Randomized Adaptive Search Procedure
LRC	- Lista restrita de candidatos
NP	- Não polinomial
PPHE	- Problema de programação de horários em escolas
PS	- Peso associado à função que mede a quantidade de sobreposição de aulas em um quadro de horários
PL	- Peso associado à função que mede a quantidade de aulas que excedem o limite máximo de aulas por dia
PV	- Peso associado à função que mede a quantidade de dias que um professor vai à escola além do necessário.
PQ	- Peso associado à função que mede a quantidade de aulas quebradas, ou seja, aquela que em um mesmo dia acontece mais de uma vez em horários não contíguos.
PD	- Peso associado à função que mede a quantidade de aulas geminadas não atendidas.
PJ	- Peso associado à função que mede a quantidade de buracos ou janelas na programação de um professor.

**LISTA DE FIGURAS**

Figura 1	Representação do espaço de soluções.....	14
Figura 2	Algoritmo <i>GRASP</i> .....	16
Figura 3	Algoritmo Busca Tabu.....	17
Figura 4	Questionário de levantamento das características da escola.....	20
Figura 5	Fragmento da representação de um quadro de horários .....	22
Figura 6	Representação de um movimento.....	25
Figura 7	Algoritmo híbrido <i>GRASP</i> + Busca Tabu.....	27
Figura 8	Quadro gerado manualmente.....	34
Figura 9	Quadro gerado pelo procedimento proposto.....	35

**LISTA DE TABELAS**

Tabela 1 Carga horária das turmas.....	21
Tabela 2 Carga horária dos professores .....	21
Tabela 3 Parâmetros do sistema.....	28
Tabela 4 Pesos associado às restrições.....	28
Tabela 5 Resultados da geração de soluções iniciais .....	31
Tabela 6 Resultados do refinamento .....	32
Tabela 7 Comparação entre os resultados .....	33

**LISTA DE GRÁFICOS**

Gráfico 1 Comportamento típico do procedimento ABT .....	30
Gráfico 2 Comportamento típico do procedimento GBT .....	30
Gráfico 3 Comportamento das melhores soluções geradas.....	32



## RESUMO

Um dos grandes problemas que está inserido no contexto das instituições educacionais brasileiras ao iniciar o seu período letivo é a programação dos horários das aulas. Este problema se torna de difícil resolução devido ao grande número de possibilidades a serem analisadas e a necessidade de verificação de uma série de requisitos de várias naturezas, muitos dos quais conflitantes entre si, tornando o espaço de busca vasto e altamente restrito.

Devido à sua complexidade combinatorial, normalmente esse problema é abordado via técnicas heurísticas. Dentre as técnicas utilizadas com relativo sucesso para tratar esse problema destacam-se as metaheurísticas, entre as quais: *Simulated Annealing*, Busca Tabu e Programação Genética. Essas técnicas, regra geral, são dependentes da qualidade das soluções iniciais.

Por outro lado, GRASP é um procedimento iterativo de duas fases, cuja primeira fase tem por objetivo construir uma boa solução inicial para a segunda fase, que é de busca local na vizinhança da solução construída.

Neste trabalho é apresentada uma técnica híbrida, baseada nas metaheurísticas GRASP e Busca Tabu, como proposta para a resolução deste problema. Resultados computacionais mostram que, com esse procedimento, soluções viáveis de melhor qualidade são encontradas mais rapidamente.

**PALAVRAS CHAVES:** Programação de horários de escolas, Metaheurísticas, Busca Tabu, GRASP.

## 1 INTRODUÇÃO

O problema de programação de horários em escolas (PPHE) diz respeito à alocação das aulas de uma escola a um conjunto restrito de horários, satisfazendo diversas restrições (Schaefer 1999).

A solução manual deste problema é uma tarefa árdua e normalmente requer vários dias de trabalho (Bardadym 1996). Além do mais, a solução obtida pode ser insatisfatória com relação a vários aspectos. Por exemplo, em função da programação feita, pode haver aulas de uma mesma matéria sendo ministradas em dias consecutivos, com prejuízo na sedimentação da aprendizagem.

Em função de situações como essa, uma atenção especial vem sendo dada à sua automação. O PPHE é considerado um problema NP-difícil (Even. et al, 1976), ou seja, de elevada complexidade, sendo assim, é comumente abordado através de técnicas heurísticas. Dentre essas técnicas, destacam-se as chamadas metaheurísticas, as quais, ao contrário das heurísticas convencionais, têm caráter geral e são providas de mecanismos para escapar de soluções ótimas locais.

Como exemplos de aplicações bem sucedidas de metaheurísticas a problemas de programação de horários, pode-se citar, dentre outras: *Simulated Annealing* (Abramson 1991), Busca Tabu (Costa 1994, Burke et al. 2001, Hertz 1992) e Programação Genética (Ueda et. al. 2001, Collorni et. al. 1998, Erben e Keppler 1996, Rich 1996).

Em geral, as metaheurísticas sofrem grande influência das soluções iniciais, isto é, uma solução inicial de boa qualidade induz a um processo de busca mais rápida, com produção de soluções finais melhores em um menor tempo computacional. A fase de construção da metaheurística GRASP (Feo e Resende 1995) é um dos instrumentos atualmente utilizados em várias aplicações (Souza 2000) para alcançar soluções iniciais de qualidade.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo Geral

O objetivo deste trabalho é desenvolver uma ferramenta capaz de otimizar a resolução de problemas de programação de horários em escolas públicas de ensino médio, através de uma técnica heurística híbrida envolvendo as metaheurísticas GRASP e Busca Tabu. No algoritmo proposto, a solução inicial é gerada pelo procedimento construtivo da técnica GRASP sofrendo, a seguir, um refinamento através do procedimento de Busca Tabu.

### 1.1.2 Objetivos Específicos

O objetivo específico dessa abordagem híbrida é tentar gerar soluções finais de melhor qualidade em um tempo computacional viável em relação aquelas obtidas utilizando-se as técnicas separadamente.

A partir dessa abordagem estudar-se-á a influência da fase de construção do algoritmo GRASP na eficiência do algoritmo Busca Tabu.

Adicionalmente, este trabalho contribui com o estudo e a automação de um problema de programação de horários de uma escola pública do estado de Minas Gerais.

## 1.2 ESTRUTURA DO TRABALHO

Neste trabalho, inicialmente é feita uma breve introdução sobre o problema estudado (seção 1). Logo após, é realizada uma revisão das técnicas propostas (seção 2), seguido de um levantamento das informações relativas à escola (seção 3) e do desenvolvimento do sistema (seção 4). O sistema é executado e os resultados gerados são analisados (seção 5). O trabalho é, então, concluído (seção 6).

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 INTRODUÇÃO

O problema de programação de horários em escolas deve ser resolvido através de um remanejamento de algumas aulas de forma que possibilite a eliminação dos conflitos, que consistem em requisitos da escola para que o quadro tenha qualidade, ou mesmo que seja viável. Pode-se exemplificar alguns requisitos como:

- Não permitir a alocação de mais de um professor em uma mesma turma no mesmo horário;
- Cada professor tem que cumprir sua carga horária semanal;
- Um professor não pode ser alocado a um horário no qual não esteja disponível;
- Evitar quebras de aulas, isto é, aulas não consecutivas de uma matéria para uma turma em um mesmo dia;
- Respeitar o limite diário de aulas de uma mesma matéria para uma mesma turma;
- Eliminar, sempre que possível, as janelas nos horários dos professores;
- Atender ao maior número possível de aulas geminadas, ou seja, aulas realizadas em dois horários consecutivos em um mesmo dia;
- Minimizar a quantidade de dias que cada professor necessita ir à escola;
- Não permitir a falta de aulas em qualquer um dos horários para uma turma.

Como este problema é NP-difícil (Even et al. 1976), sua resolução por técnicas de programação matemática, ditas exatas, é praticamente inviável para os problemas reais devido ao elevado tempo de processamento requerido. Sendo assim, este problema é normalmente abordado por técnicas heurísticas (Schaefer 1999).

As primeiras heurísticas idealizadas para a resolução deste problema eram heurísticas construtivas. A idéia básica era construir uma solução aula por aula procurando atender a todos os requisitos, até que todas as aulas fossem alocadas. O que diferenciava um método de outro, era a forma de escolha da aula a ser inserida. Alguns exemplos são:

- Junginger (1995), que considera como urgente a aula a qual o professor tem pouca disponibilidade de horários e muitas ainda para ministrar. Considera também um horário favorável aquele em que

existe um número reduzido de aulas de outros professores que podem ser alocadas naquele horário.

- Papoulias (1980), que considera um horário mais favorável para uma aula se as outras aulas do professor não estão alocadas a dias consecutivos para uma determinada turma. Assim ele propõe o espalhamento das aulas.
- Mata (1989), formulou uma heurística construtiva para tratar o problema de aulas duplas que é exigido por algumas escolas secundárias brasileiras para um determinado conjunto de matérias.

Posteriormente, surgiram métodos de refinamento, os quais baseavam-se em realocações das aulas de forma a melhorar a qualidade das soluções. Entretanto estes métodos, denominados métodos de descida, ficavam presos no primeiro mínimo local encontrado. A partir do surgimento das metaheurísticas tais como Busca Tabu *Simulated Annealing*, Programação Genética, etc., que têm mecanismos para escapar de ótimos locais, surgiu um novo horizonte para a resolução de problemas combinatoriais, tais como o PPHE. Nesta linha de solução cita-se:

- Colormi et al. (1998), aplicaram Programação Genética ao problema. Quadros inviáveis de horários também são incluídos no espaço de busca do algoritmo e a função de avaliação considera requisitos didáticos, organizacionais e pessoais e incorpora também o número de inviabilidades.
- Abramson (1991), aplicou um método que faz analogia à termodinâmica denominado *Simulated Annealing* ao problema. Onde uma solução inicial qualquer é refinada aceitando movimentos de piora com uma certa probabilidade, a qual está associado a um parâmetro chamado Temperatura. Inicialmente este parâmetro assume um valor elevado e sofre um resfriamento fazendo com que o algoritmo comporte-se como um método de descida.

## 2.2 VIZINHANÇA DE UMA SOLUÇÃO

Entende-se por vizinho de uma solução  $s$  a solução  $s'$  que sofreu a modificação  $m$ . Representa-se essa operação por  $s' \leftarrow s \oplus m$ .

Mais especificamente seja  $S$  o espaço de pesquisa de um problema de otimização. A função  $N$ , a qual depende da estrutura do problema tratado, associa a cada solução viável  $s \in S$ , sua vizinhança  $N(s) \subseteq S$ , ou seja, todas as soluções  $s'$  geradas a partir de um movimento  $m$ . Cada solução  $s' \in N(s)$  é chamada de vizinho de  $s$ .

### 2.3 MÍNIMO LOCAL

O mínimo local é a solução  $s$  onde nenhum de seus vizinhos  $s' \in N(s)$  tem o valor de função de avaliação menor.

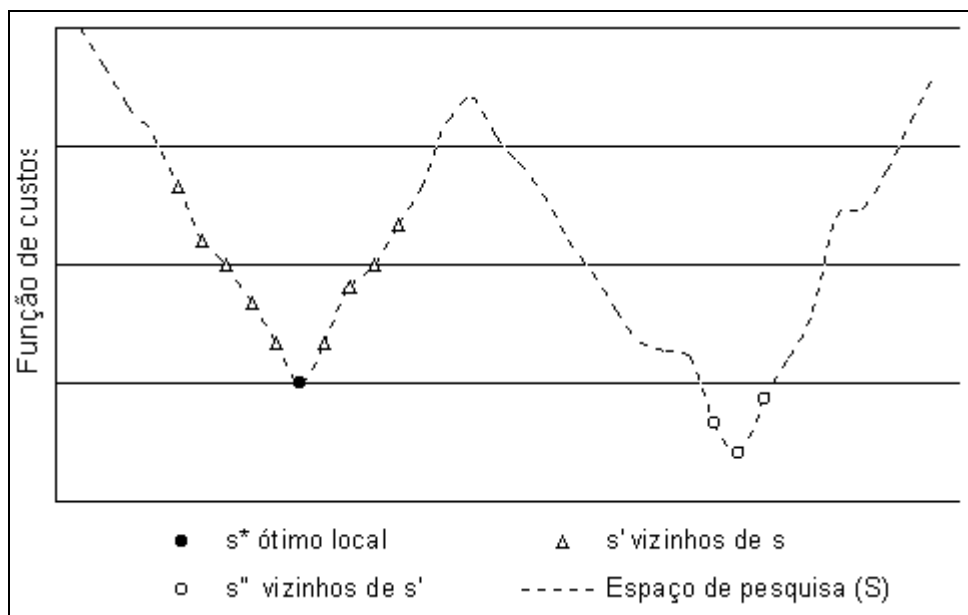


Figura 1 Representação do espaço de soluções

Pode-se observar na figura 1 que  $s^*$  é um ótimo local, pois, nenhum de seus vizinhos  $s'$  tem valor de função de avaliação menor. As soluções  $s''$  apresentam valores menores de função de avaliação, porém, não fazem parte do conjunto de vizinhos de  $s^*$ .

### 2.4 BUSCA LOCAL

Métodos de busca local em problemas de otimização constituem uma família de técnicas baseadas na noção de vizinhança, ou seja, são métodos que percorrem o espaço de pesquisa passando, iterativamente, de uma solução para outra que seja sua vizinha.

### 2.4.1 Método de Descida

É um método de busca local que analisa todos os possíveis vizinhos de uma solução  $s$  em sua vizinhança  $N(s)$ , escolhendo, a cada passo, aquele que tem menor valor para a função de avaliação, ele também é chamado de método guloso. É importante observar, que para efetivar a mudança o vizinho candidato deve melhorar estritamente o valor da melhor solução obtida até o momento. O critério de parada se dá quando um mínimo local é encontrado.

### 2.4.2 Método Randômico de Descida

Consiste em analisar um vizinho qualquer e o aceitar se ele for estritamente menor que a solução corrente, caso isso não ocorra, a solução corrente permanece inalterada e gera-se aleatoriamente outro vizinho. O procedimento é interrompido após um número fixo de iterações sem melhora no valor da melhor solução obtida até então. Esse método também fica preso no primeiro mínimo local encontrado.

## 2.5 METAHEURÍSTICAS UTILIZADAS

A utilização de técnicas heurísticas visando a solução de problemas de programação de horários é justificada pela complexidade do problema em seu caso geral, o que inviabiliza a utilização de técnicas exatas para resolvê-lo. Dentre tais técnicas, as metaheurísticas têm um papel importante, pois, ao contrário das heurísticas convencionais, elas têm capacidade de escapar de ótimos locais, percorrendo o espaço de pesquisa amplamente. A grande vantagem de utilizar uma técnica híbrida, como a proposta, é que se explora as características positivas de cada metaheurística.

### 2.5.1 GRASP (Greedy Randomized Adaptive Search Procedure)

O Processo de busca adaptativa gulosa e randomizada é um método iterativo, proposto por Feo e Resende (1995) que consiste de duas fases:

- Uma fase de construção, na qual uma solução é gerada, elemento a elemento;
- Uma fase de busca local, na qual um ótimo local na vizinhança da solução construída é pesquisado.

A melhor solução encontrada ao longo de todas as iterações GRASP realizadas é retornada como resultado. Na fase de construção do algoritmo GRASP, uma solução viável é construída iterativamente, um elemento da solução por vez, até que a solução esteja completa. Os elementos candidatos que compõem a solução são ordenados em uma lista, chamada de lista de candidatos, a qual contém todos os candidatos. Esta lista é ordenada por uma função gulosa que mede o benefício que o mais recente elemento escolhido concede à parte da solução já construída. Um subconjunto denominado lista restrita de candidatos (LRC) é formado pelos melhores elementos que compõem a lista de candidatos, o tamanho da lista restrita de candidatos é controlado por um parâmetro  $\alpha \in [0,1]$ , onde para  $\alpha = 0$  tem-se um comportamento puramente guloso do algoritmo e para  $\alpha = 1$  um comportamento aleatório. A componente probabilística do método é devida à escolha aleatória de um elemento da lista restrita de candidatos. Este procedimento permite que diferentes soluções de boa qualidade sejam geradas. O pseudocódigo do algoritmo GRASP é apresentado na figura 2.

<p><b>Procedimento GRASP</b> (<math>f()</math>, <math>g()</math>, <math>N()</math>, <math>GRASP_{max}</math>, <math>s</math>)</p> <ol style="list-style-type: none"> <li>1 <math>f^* \leftarrow \infty</math>;</li> <li>2 Para (<math>iter = 1, 2, \dots, GRASP_{max}</math>) faça;</li> <li>3.     Construção (<math>g()</math>, <math>\alpha</math>, <math>s</math>);</li> <li>4.     BuscaLocal (<math>f()</math>, <math>N()</math>, <math>s</math>);</li> <li>5.     Se (<math>f(s) &lt; f^*</math>) então</li> <li>6.         <math>s^2 \leftarrow s</math>;</li> <li>7.         <math>f^* \leftarrow f(s)</math>;</li> <li>8.     <u>fim-se</u>;</li> <li>9. <u>fim-para</u>;</li> <li>10. <math>s \leftarrow s^2</math>;</li> <li>11. Retorne <math>s</math>;</li> </ol> <p><b>Fim GRASP</b></p>
---

Figura 2 Algoritmo GRASP





### 3 LEVANTAMENTO DAS CARACTERÍSTICAS DA ESCOLA

A fase de levantamento dos dados relativos a escola é de grande importância para a construção do modelo. É necessário entender o funcionamento da escola e conhecer sua estrutura, com esse intuito foi aplicado um questionário de levantamento de características da escola.

#### 3.1 DESCRIÇÃO DA ESCOLA

A instituição de ensino considerada para análise de programação de horários é a Escola Estadual Dom Velloso situada à rua Clodomiro Oliveira, 45, bairro Pilar, Ouro Preto, Minas Gerais. Esta escola pertence à rede pública de ensino, atendendo ao 1º e 2º ciclo do ensino fundamental (1º ao 8º ano), possuindo 11 turmas no turno matutino e 11 no vespertino.

Para este trabalho analisou-se apenas o turno matutino, onde lecionam cerca de 23 professores para as turmas do 6º ao 8º ano.

Para cada dia da semana são reservados 5 aulas para cada turma o que totaliza durante a semana (segunda à sexta) 25 hora-aulas por turno. Sendo cada hora-aula de 50 minutos com um intervalo de 20 minutos entre o terceiro e o quarto horário.

Atualmente a instituição conta com uma média de 23 professores para o segundo ciclo e 9 professores para o primeiro ciclo. Neste quadro estão incluídos os efetivos e os contratados. A cada professor efetivo está associado um “cargo”, que corresponde no mínimo a 18 hora-aulas que cada professor deve lecionar. Na rede estadual o professor pode assumir apenas dois cargos. Pode acontecer que algum professor não consiga lecionar a matéria de sua competência em um turno, então ele completa o cargo em outro turno e caso não consiga completar as horas aulas do cargo, ele pode assumir outras matérias afins para atingir o limite estipulado.

Geralmente os professores costumam concentrar suas atividades em apenas um turno. E para isso escolhem as turmas que estão concentradas no turno de sua preferência. Não completando a carga de trabalho escolhem o restante em outro turno. Existem aqueles que necessitam trabalhar os três turnos, pois a matéria que

lecionam tem uma pequena quantidade de aulas por turma. Pode-se citar como exemplo a disciplina Educação Religiosa.

A distribuição dos professores às turmas é feita respeitando as preferências dos professores efetivos com maior tempo de serviço na instituição. Cada professor efetivo tem um dia de folga, sendo que a escolha do dia de folga também obedece ao “tempo de casa”.

Alguns professores trabalham em mais de uma instituição de ensino, ficando assim indisponíveis em determinados horários. Esses professores necessitam de uma agenda de trabalho compacta e se possível em apenas um turno. Os contratados também não têm disponibilidade integral, portanto requerem o mesmo tratamento anterior.

Para a programação do quadro de horário nesta escola é feito um levantamento da carga horário desejada de cada professor e da matéria que iram lecionar durante o ano. Os planejadores do quadro de horários da escola procuram atender sempre que possível às exigências destes professores. Entretanto, por questões pedagógicas, uma turma não pode ter mais que duas hora-aulas da mesma matéria por dia, com exceção das disciplina Educação física, Educação artística e Educação religiosa que devem ter apenas uma hora-aula por dia.

### 3.2 LEVANTAMENTO DE DADOS

Para o levantamento das informações referentes à escola, foi elaborado e aplicado o questionário mostrado na figura 4.

QUANTO À ESTRUTURA DA ESCOLA
1.Quantas salas têm? 2.Tem laboratórios ? Quantos? 3.Qual é a hora do recreio? 4.Qual é o número de funcionários? 5.Quais são os horários de início e de termino das aulas?
QUANTO ÀS TURMAS
1.Quantas aulas cada turma tem ? 2.Quantos horários são disponibilizados para cada turma? 3.Quantas turmas têm por turno? 4.Quais turmas utilizam o(s) laboratório(s)? 5.A cada turma está associado um professor por matéria? 6.Qual é o número máximo de aulas de uma mesma matéria para uma mesma turma?
QUANTO AOS PROFESSORES
1.Como é feita a escolha de professor por matéria ?(Tempo de casa) 2.Aulas geminadas é critério do professor ou da escola? 3.Quantos professores são efetivos e quantos são contratados? 4.Qual o número de horas/aulas por professor ? 5.Quantos professores têm um cargo? E quantos têm dois cargos? 6.Como é tratada a folga dos professores? 7.Como é tratada as preferências e as indisponibilidades dos professores por horário? 8.Existem casos de professores que necessitam dar aulas em um mesmo horário para turmas distintas? (Educação Física para duas turmas com dois professores). 9.É desejável compactar a agenda dos professores?
QUANTO ÀS AULAS
1.É desejável aula geminada? 2.Uma aula que se inicia antes do recreio e outra que continua após o recreio é considerada geminada? 3.Existem aulas que são pré-alocadas? 4.Aulas disjuntas é admitido? 5.Há relação de precedência entre as matérias? 6.Qual é o número de aulas por matéria ?

Figura 4 Questionário de levantamento das características da escola



## 4 METODOLOGIA

### 4.1 REPRESENTAÇÃO DO PROBLEMA

O problema de programação de horários em escolas (PPHE) abordado consiste em um conjunto de  $m$  professores,  $n$  turmas,  $s$  matérias e  $p$  horários semanais reservados para a realização das aulas. Os  $p$  horários semanais são distribuídos em  $d$  dias da semana de  $h$  horários diários realizados em um único turno. Pode-se considerar que as turmas estão sempre disponíveis e são definidas como um grupo de estudantes que cursam a mesma série, sendo assim destinados a cursar as mesmas matérias. A cada matéria de uma turma está associado um único professor, previamente fixado. Além disso, a carga horária de cada turma é exatamente  $p$ .

A representação adotada para o problema utiliza a definição de tipos simples e naturais de movimentos e torna mais fácil o levantamento de algumas características da função de avaliação, à qual está baseada a alocação das aulas dos professores. Esta é a forma com a qual os responsáveis pela elaboração manual de quadro de horário trabalham com a resolução do problema.

A figura 5 mostra um fragmento de um quadro de horários.

<i>prof</i> \ <i>hor</i>	0	1	2	3	4	...	$p$
0	1	1	2	2	4	...	7
1	X	4	4	3	3	...	5
2	F	F	F	F	F	...	1
3	5	5	X	4	2	...	2
...	...	...	...	...	...	...	...
$M$	6	6	5	1	1	...	3

Figura 5 Fragmento da representação de um quadro de horários

Nesta figura as colunas representam os horários e as linhas representam os professores, de tal forma que, se um professor  $i$  em um horário  $j$  está:

- Disponível, recebe o valor X;
- Indisponível, recebe o valor F;
- Lecionando, recebe o valor referente à turma (0,1,2,3...n).

## 4.2 AVALIAÇÃO DE UM QUADRO DE HORÁRIOS

### 4.2.1 Função de Avaliação

A avaliação de um quadro de horários é realizada por uma função  $f(s)$ , que leva em consideração uma variedade de fatores relativos aos professores, às turmas e às exigências de cada instituição de ensino.

Para avaliar um quadro de horários com relação à qualidade e à viabilidade, os diferentes requisitos do problema são inicialmente organizados em:

- (i) **Requisitos essenciais:** são aqueles que se não forem satisfeitos, gerarão alocações inviáveis, como por exemplo:
- Alocar mais de um professor para uma mesma turma, em um mesmo horário;
  - Não alocar aula para uma turma, em um dado horário;
  - Não respeitar o limite máximo de aulas.
- (ii) **Requisitos não-essenciais:** são aqueles cujo atendimento é desejável, porém, quando não satisfeitos, não geram programações inviáveis, como por exemplo:
- Minimizar o número de dias que um professor vai à escola;
  - Evitar quebras de aulas;
  - Maximizar o número de aulas geminadas;
  - Evitar janelas na agenda do professor.

Desse modo, uma solução  $s$  pode ser medida com base em duas componentes, uma de inviabilidade  $g(s)$ , a qual mede o não atendimento aos requisitos essenciais, e outra de qualidade  $h(s)$ , a qual mede o não atendimento aos requisitos considerados não-essenciais. A função de avaliação de uma solução  $s$ , a qual deve ser minimizada, pode ser calculada, portanto, pela expressão 1.

$$f(s) = g(s) + h(s) \quad (1)$$

#### 4.2.2 Hierarquia da Função de Avaliação

A função de avaliação é formada por funções que medem o valor de cada requisito, com base no atendimento das restrições impostas por eles. Cada uma destas funções tem peso diferenciado, com o intuito de dar maior prioridade aos requisitos considerados mais importantes.

A parcela  $g(s)$ , que mensura o nível de inviabilidade de uma solução  $s$ , é avaliada com base na expressão 2.

$$g(s) = \sum_{k=1}^K \alpha_k I_k \quad (2)$$

Onde:  $k$  é número de medidas de inviabilidade,

$I_k$  o valor da  $k$ -ésima medida de inviabilidade e,

$\alpha_k$  o peso associado à  $k$ -ésima medida.

A parcela  $h(s)$ , que mensura a qualidade de uma solução  $s$ , é avaliada com base na expressão 3.

$$h(s) = \sum_{l=1}^L \beta_l Q_l \quad (3)$$

Onde:  $L$  representa o número de medidas de qualidade,

$Q_l$  o valor da  $l$ -ésima medida de qualidade,

$\beta_l$  o peso associado a essa  $l$ -ésima medida.

Deve ser observado que uma solução  $s$  é viável se e somente se  $g(s) = 0$ . Nas componentes da função  $f(s)$  os pesos dados às diversas medidas refletem a importância relativa de cada uma delas e, sendo assim, deve-se tomar  $\alpha_k \gg \beta_l \quad \forall k, l$ , de forma a privilegiar a eliminação das soluções inviáveis.



### 4.3 ESTRUTURA DE VIZINHANÇA

No problema estudado, dada uma solução  $s$ , sua vizinhança  $N(s)$  é constituída por todas as soluções  $s'$  que são obtidas de  $s$  a partir de um movimento que consiste na simples troca de duas atividades (aulas) distintas, não se incluindo as indisponibilidades, da programação de um professor. Tal movimento é identificado pela tripla  $\langle i, k_1, k_2 \rangle$ , onde  $k_1$  e  $k_2$  representam os horários nos quais as atividades do professor  $i$  serão permutadas.

	<b>0</b>	<b>1</b>	<b>2</b>	...	<b>p</b>			<b>0</b>	<b>1</b>	<b>2</b>	...	<b>P</b>	
<b>0</b>	1	1	2	...	7	→	<b>0</b>	1	1	2	...	7	
<b>1</b>	<b>X</b>	<b>4</b>	4	...	5		<b>1</b>	<b>4</b>	<b>X</b>	4	...	5	
...	...	...	...	...	...		...	...	...	...	...	...	...
<b>M</b>	6	6	5	...	3		<b>M</b>	6	6	5	...	3	
<b>S</b>							<b>S'</b>						

Figura 6 Representação de um movimento

### 4.4 GERAÇÃO DE UMA SOLUÇÃO INICIAL

Para gerar uma solução inicial pode-se utilizar vários procedimentos, desde os aleatórios até os heurísticos. A geração de soluções aleatórias diversifica o espaço de soluções, no entanto, uma solução aleatória é, em geral, de baixa qualidade e exige um tempo de processamento muito elevado para ser melhorada. Por outro lado, soluções gulosas, apesar de serem de boa qualidade, apresentam baixa diversidade. Recentemente, Feo e Resende (1995) idealizaram uma heurística que representa uma alternativa a esses métodos puramente aleatórios ou puramente gulosos. Várias aplicações bem sucedidas dessa heurística, chamada GRASP, têm sido relatadas na literatura (Souza 2000). Desta forma, optou-se por gerar uma população inicial através da aplicação sucessiva da fase de construção do algoritmo GRASP.

No problema de programação de horários, uma solução inicial é gerada conforme a seguir se descreve. Inicialmente, determinam-se os horários mais críticos, isto é, aqueles que têm o menor número de professores disponíveis. A seguir enumeram-se e ordenam-se todas as aulas, priorizando aquelas mais difíceis de serem

alocadas, ou seja, aquelas que têm os professores com maior carga horária e maior número de horários indisponíveis. Forma-se uma lista de candidatos (C) destas aulas, um subconjunto de C denominado lista restrita de candidatos (LRC) é formado com tamanho  $|LRC|$ . Seleciona-se aleatoriamente um elemento da LRC, a seguir procura-se alocá-lo (usando a ordem dos horários críticos) de forma que todas as restrições essenciais sejam satisfeitas. Toda vez que uma aula é alocada, os horários críticos são atualizados, bem como as aulas mais difíceis remanescentes. Quando todas as aulas são alocadas o procedimento chega ao final, retornando um quadro de horários completo.

#### 4.5 REFINAMENTO DE UMA SOLUÇÃO

Construída uma solução, esta é refinada pelo método de Busca Tabu. A cada iteração deste procura-se selecionar o melhor vizinho  $s'$  da vizinhança  $N(s)$  da solução corrente  $s$ . Independentemente de  $f(s')$  ser melhor ou pior que  $f(s)$ ,  $s'$  será sempre a nova solução corrente.

Se o melhor vizinho  $s'$  de um quadro de horários  $s$  for obtido a partir deste pelo movimento de troca entre a aula do professor  $i$  do horário  $k_1$  e a aula do horário  $k_2$ , então o movimento que envolve a troca das aulas do professor  $i$  entre os horários  $k_1$  e  $k_2$  é armazenado em uma lista tabu durante  $|T|$  iterações. Tais atributos foram escolhidos por possuírem características fáceis de detectar, ao . Esta lista é do tipo *FIFO* (*first in first out*), ou seja, o primeiro elemento a entrar é o primeiro elemento a sair. O tempo tabu é definido de modo a prevenir o retorno a soluções visitadas recentemente. Tais movimentos são proibidos a menos que a solução satisfaça a um critério de aspiração, no caso, o de aspiração por objetivo. Isto é, um movimento mesmo estando na lista tabu é aceito se e somente se a solução encontrada for melhor que a melhor solução encontrada até então, pois, representa que esta solução ainda não foi pesquisada. O procedimento chega ao fim quando alcança um certo número de iterações sem melhora, conhecido como BTMAX. O número de iterações sem melhora é dado pela diferença entre a iteração corrente e a iteração onde foi encontrada a última melhor solução.

#### 4.6 O ALGORITMO HÍBRIDO

O algoritmo híbrido proposto é um método de duas fases. Na primeira fase constrói-se uma solução inicial via um procedimento GRASP (seção 2.5.1). Na segunda, essa solução construída é submetida a um refinamento pelo método de Busca Tabu (seção 2.5.2). O pseudocódigo do algoritmo híbrido GRASP + Busca Tabu (BT) é apresentado na figura 7.

**Procedimento GRASP + BT**

1.  $s^0 \leftarrow \text{ConstruaSolucaoGRASP}();$

2.  $s^1 \leftarrow \text{BT}(s^0);$

**Fim GRASP + BT**

Figura 7 Algoritmo híbrido GRASP + Busca Tabu

## 5 RESULTADOS COMPUTACIONAIS

O algoritmo proposto foi implementado na linguagem C++ usando o compilador Borland C++ Builder 5.0 e testado em um microcomputador PC Pentium IV, 1.6 MHz, com 512 MB de RAM.

Para testá-lo foram utilizados dados relativos à programação de horários da Escola Estadual Dom Velloso de Ouro Preto do ano de 2001 no período matutino. As restrições do sistema foram levantadas conjuntamente com os responsáveis pela elaboração manual dos quadros de horário e a diretoria da escola.

O algoritmo foi submetido a uma bateria preliminar de testes para calibrar os diversos parâmetros existentes, bem como para escolher os pesos mais adequados às componentes da função de avaliação. Estes valores são mostrados respectivamente nas tabelas 3 e 4.

Tabela 3 Parâmetros do sistema

<b>Parâmetro</b>	<b>Valor</b>
BTMAX	70
T	30

Tabela 4 Pesos associado às restrições

<b>Peso</b>	<b>Valor</b>	<b>Categoria</b>
PS	40	Inviabilidade
PL	25	Inviabilidade
PV	7	Qualidade
PQ	6	Qualidade
PD	5	Qualidade
PJ	1	Qualidade

Onde:

PS - Peso associado à função que mede a quantidade de sobreposição de aulas em um quadro de horários

PL - Peso associado à função que mede a quantidade de aulas que excedem o limite máximo de aulas por dia

PV - Peso associado à função que mede a quantidade de dias que um professor vai à escola além do necessário.

PQ - Peso associado à função que mede a quantidade de aulas quebradas, ou seja, aquela que em um mesmo dia acontece mais de uma vez em horários não contíguos.

PD - Peso associado à função que mede a quantidade de aulas geminadas não atendidas.

PJ - Peso associado à função que mede a quantidade de buracos ou janelas na programação de um professor.

Vale lembrar que os pesos referentes à qualidade do quadro são parâmetros associados a cada um dos professores, podendo ser alterados individualmente conforme o grau de importância dado.

Inicialmente estudou-se o desempenho do procedimento GRASP para a geração de soluções iniciais em relação ao procedimento aleatório. Para tanto, foram realizados testes gerando um conjunto de soluções iniciais por cada um dos procedimentos.

Numa segunda etapa as soluções iniciais geradas passaram por um processo de refinamento pelo procedimento Busca Tabu. Para isso foram implementados os seguintes procedimentos:

- ABT: onde a solução inicial é gerada de forma aleatória e refinada pelo procedimento de Busca Tabu.
- GBT: onde a solução inicial é gerada pelo procedimento GRASP e, refinada pelo procedimento de Busca Tabu.

Os gráficos 1 e 2 mostram, respectivamente, os comportamentos típicos dos procedimentos ABT e GBT. As linhas contínuas indicam os melhores valores da função de avaliação obtidos a cada iteração e as linhas pontilhadas indicam a evolução da melhor solução gerada ao longo de todas as iterações. O comportamento do procedimento ABT demonstra uma fase inicial de refinamento onde se procura alcançar soluções de qualidade para, a partir de então, iniciar o refinamento mais intenso o que no procedimento GBT ocorre já em uma fase inicial.

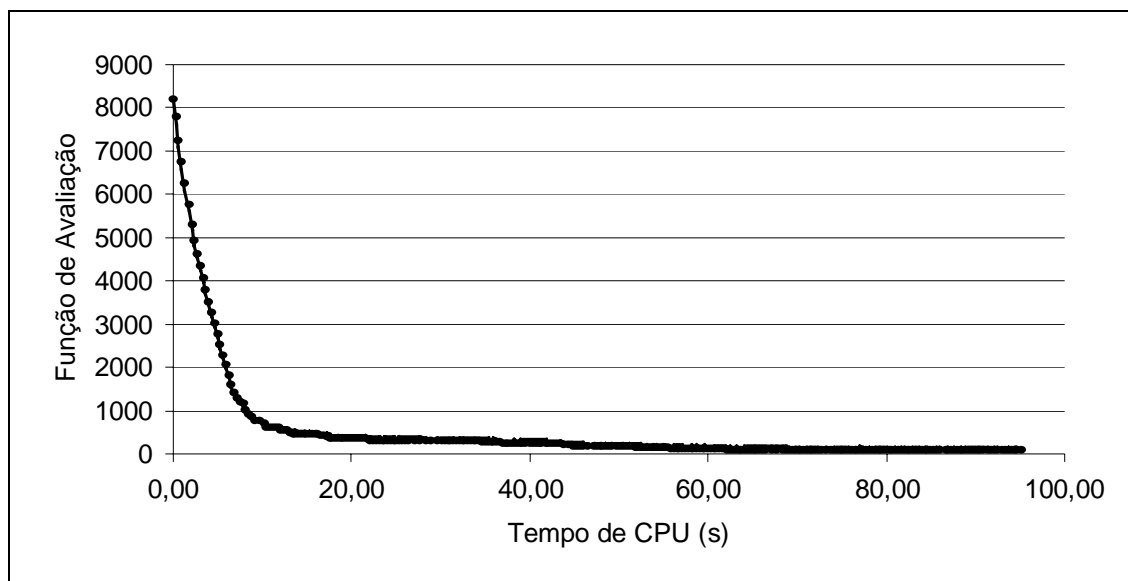


Gráfico 1 Comportamento típico do procedimento ABT

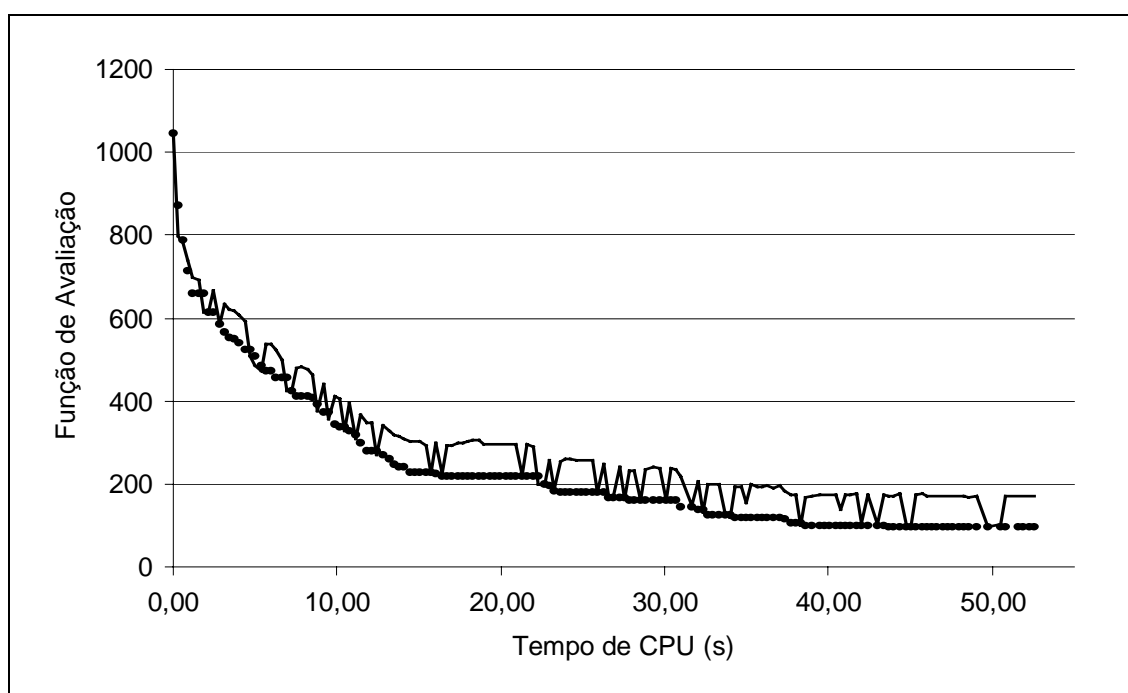


Gráfico 2 Comportamento típico do procedimento GBT

Pode-se observar no gráfico 2 que os valores de função de avaliação de cada iteração podem assumir valores de piora, ou seja, maiores que o da melhor solução encontrada. Isto ocorre pelo fato do procedimento de Busca Tabu aceitar movimentos de piora para conseguir escapar de mínimos locais.

## 5.1 COMPARAÇÃO ENTRE OS PROCEDIMENTOS TESTADOS

Para cada um dos procedimentos de geração de solução inicial foram realizadas 10 execuções com sementes diferentes de números aleatórios. Calcularam-se os valores mínimos, médios, o desvio médio percentual em relação à melhor solução encontrada e a amplitude. Os valores apresentados nas tabelas foram calculados com base nas expressões 4, 5, 6 e 7.

$$mínimo = \min_{1 \leq j \leq 10} \{fo_j\} \quad (4)$$

$$média = \frac{\sum_{i=1}^{10} fo_i}{10} \quad (5)$$

$$Desvio = \frac{média - mínimo^*}{mínimo^*} \times 100 \quad (6)$$

Onde  $mínimo^*$  é a melhor solução encontrada em todos os testes.

$$amplitude = \max_{1 \leq j \leq 10} \{fo_j\} - \min_{1 \leq j \leq 10} \{fo_j\} \quad (7)$$

Tabela 5 Resultados da geração de soluções iniciais

<b>Procedimento</b>	<b>Mínimo</b>	<b>Média</b>	<b>Desvio Médio</b>	<b>Amplitude</b>
<b>Aleatório</b>	7720	8532,4	10.565,5%	1345
<b>Grasp</b>	1044	1273,1	1.491,4%	574

Pode-se observar pela tabela 5 que o procedimento aleatório gera soluções mais diversificadas, enquanto o procedimento GRASP se mostra mais eficiente, gerando soluções com melhor qualidade. Porém, ambos procedimentos não geram soluções viáveis, surgindo daí a necessidade de refinamento das soluções geradas procurando superar as inviabilidades.

Pela tabela 6, observa-se que o procedimento ABT gerou, na média, quadros de qualidade um pouco melhores que os gerados pelo procedimento GBT. Entretanto, os quadros gerados pelo procedimento GBT exigiram um tempo computacional significativamente inferior àquele exigido pelo procedimento ABT. Além disso, muitas das soluções finais geradas pelo procedimento ABT eram inviáveis, ao contrário do procedimento GBT, que não produziu soluções inviáveis.

Adicionalmente, a amplitude dos valores finais das soluções produzidas pelo procedimento GBT foi bem inferior aos dos valores produzidos pelo procedimento ABT. Sendo assim, o procedimento GBT se mostrou o mais eficiente e mais robusto dos procedimentos implementados, ou seja, a partir de soluções iniciais geradas pelo método GRASP, os resultados finais diferem pouco da melhor solução encontrada e têm menor possibilidade de serem inviáveis. Vale a pena ressaltar que o procedimento GBT ainda pode ser melhorado por um melhor ajuste do parâmetro  $\alpha$ , bem como dos parâmetros do procedimento Busca Tabu.

Tabela 6 Resultados do refinamento

Procedimento	Mínimo	Média	Desvio Médio	Amplitude	Tempo médio de CPU (s)
<b>ABT</b>	80	123,6	54,5%	66	95,08
<b>GBT</b>	95	127,1	58,9%	46	52,54

O gráfico 3 demonstra que partindo de soluções iniciais geradas via GRASP, quadros de melhor qualidade são produzidas mais rapidamente. Verifica-se, ainda, que tal comportamento gera um intervalo que representa um tempo computacional adicional para o refinamento da solução.

A linha contínua representa o comportamento da melhor solução encontrada pelo procedimento ABT e a linha pontilhada o comportamento da melhor solução encontrada pelo procedimento GBT.

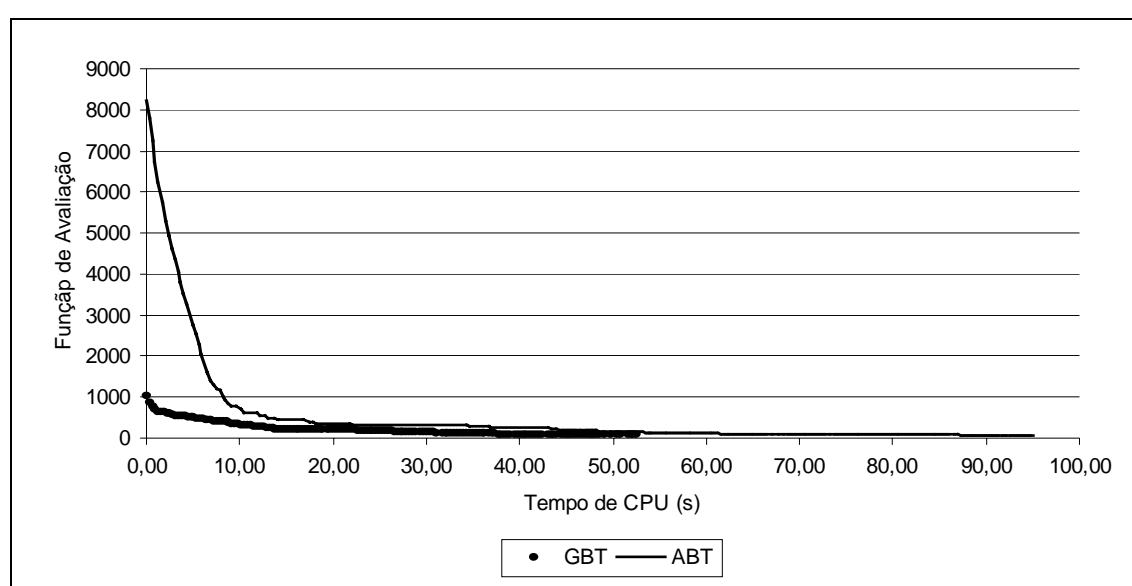


Gráfico 3 Comportamento das melhores soluções geradas



Os resultados mostram que no procedimento ABT partindo de soluções iniciais com valores de função de avaliação próximos a 8532 chega-se a soluções com valores de função de avaliação próximos a 123 em um tempo médio de 90 segundos. Já no procedimento GBT, partindo de soluções iniciais com valores de função de avaliação próximos a 1273 chega-se a soluções com valores de função de avaliação próximos a 127 em um tempo médio de 50 segundos. Isto demonstra a capacidade do procedimento GBT em obter uma boa solução mais rapidamente.

## 5.2 COMPARAÇÃO COM O QUADRO MANUAL

A comparação entre o quadro gerado de forma manual e os melhores quadros obtidos pelos procedimentos testados demonstra a menor importância quanto à qualidade pedagógica da confecção manual dos quadros de horário em detrimento ao atendimento das necessidades dos professores. Pode-se observar que o não atendimento a aulas geminadas chega a ser 228% superior nos resultados obtidos de forma manual, o que prejudica a qualidade do ensino. A tabela 7 mostra os valores de avaliação do quadro manual e os valores dos quadros obtidos com a execução dos procedimentos.

Tabela 7 Comparação entre os resultados

	<i>Veze</i> s ()	<i>Quebras</i> ()	<i>Duplas</i> ()	<i>Janelas</i> ()	<b>Total</b>
<b>Manual</b>	0	0	115	12	127
<b>ABT</b>	21	0	35	24	80
<b>GBT</b>	28	0	50	17	95

Na tabela 7, cada coluna tem o seguinte significado:

*Veze*s (), é a função que mede quantos dias um professor vai à escola além do necessário.

*Quebras* (), é a função que mede a quantidade de aulas quebradas, ou seja, aquela que em um mesmo dia acontece mais de uma vez em horários não contíguos.

*Duplas* (), é a função que mede a quantidade de aulas geminadas não atendidas.

*Janelas* (), é a função que mede a quantidade de buracos ou janelas na programação de um professor.

Os quadros confeccionados pelo método manual e o automatizado podem ser vistos nas figuras 8 e 9, respectivamente.

	Segunda					Terça					Quarta					Quinta					Sexta					
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
0	F	F	F	F	F	0	0	1	2	3	1	1	2	0	0	3	3	1	2	2	2	1	0	3	3	
1	x	x	x	x	x	7	7	5	5	4	5	5	4	7	6	6	6	7	4	4	7	6	6	4	5	
2	3	x	2	4	6	x	x	x	x	x	7	6	6	5	4	7	7	2	3	3	5	5	2	x	4	
3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	8	8	9	10	10	9	9	8	x	10	
4	x	x	x	x	x	3	5	7	1	2	0	8	3	x	x	5	4	6	7	8	0	2	4	6	1	
5	F	F	F	F	F	2	2	3	3	1	3	0	0	1	2	x	x	x	x	x	x	x	x	1	0	
6	x	x	x	x	x	x	x	x	x	x	6	9	8	4	7	0	1	3	x	5	x	x	x	10	2	
7	x	x	x	x	x	6	6	x	7	7	x	x	x	x	x	x	x	x	x	x	6	7	x	x	x	
8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	4	5	x	6	7	x	x	x	x	x	
9	x	2	3	x	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	3	2	x	
10	10	9	x	x	x	x	x	x	x	x	x	x	x	9	10	x	x	x	x	x	x	x	x	x	x	
11	6	4	7	5	0	F	F	F	F	F	9	10	x	x	x	F	F	F	F	F	10	0	9	8	8	
12	x	x	x	7	7	F	F	F	F	F	x	x	x	x	x	x	x	x	x	x	x	x	x	x	7	
13	1	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	x	
14	7	7	x	3	3	x	x	x	x	x	x	7	7	3	3	x	x	x	x	x	3	3	7	7	x	
15	x	10	8	9	9	8	8	9	10	10	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	
16	5	5	6	6	4	5	4	4	6	6	F	F	F	F	F	x	x	4	5	6	4	4	5	5	6	
17	2	3	4	1	5	4	3	0	x	x	4	3	5	2	1	1	2	5	0	0	F	F	F	F	F	
18	8	8	9	10	10	9	9	8	8	x	8	x	10	10	9	10	10	8	9	9	F	F	F	F	F	
19	0	0	1	2	2	1	1	2	0	0	2	2	1	x	x	2	0	0	1	1	F	F	F	F	F	
20	4	6	5	8	8	10	x	6	4	5	10	4	9	6	5	9	9	10	8	x	F	F	F	F	F	
21	9	x	10	x	x	x	10	10	9	9	x	x	x	x	x	x	x	x	x	x	x	10	10	9	9	
22	x	x	x	x	x	x	x	x	x	8	x	x	x	8	8	x	x	x	x	x	8	8	x	x	x	
<b>Função de avaliação</b>						<b>127</b>																				
<b>Função de qualidade</b>						<b>127</b>																				
<b>Função de inviabilidade</b>						<b>0</b>																				

Figura 8 Quadro gerado manualmente

	Segunda					Terça					Quarta					Quinta					Sexta																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24															
0	F	F	F	F	F	1	1	0	3	3	2	0	3	1	1	2	2	0	3	3	0	0	2	2	1															
1	6	6	7	4	4	6	6	x	7	7	6	5	5	x	4	7	7	x	5	5	x	4	4	5	x															
2	7	3	3	x	2	x	x	x	x	x	x	3	4	5	5	5	6	6	7	7	2	2	6	4	4															
3	x	x	x	x	x	x	10	10	x	9	x	x	x	x	x	10	x	x	8	8	9	9	8	x	x															
4	x	1	4	3	0	7	8	6	4	2	5	6	0	x	7	x	x	x	x	x	5	8	3	1	2															
5	F	F	F	F	F	2	2	1	0	0	3	2	x	x	x	3	3	1	1	0	x	x	x	x	x															
6	x	9	1	0	6	x	x	x	2	4	x	x	x	x	x	x	x	x	x	x	8	3	10	7	5															
7	x	x	x	x	x	x	x	x	x	x	7	7	6	6	x	x	x	x	x	x	6	x	7	x	x															
8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	6	5	7	x	4	x	x	x	x	x															
9	2	2	x	x	3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	3	1	1	x	x															
10	x	x	x	x	x	x	x	x	x	x	x	9	10	x	x	x	x	x	x	x	x	x	x	10	9															
11	5	7	8	9	9	F	F	F	F	F	0	10	x	x	x	F	F	F	F	F	4	6	0	8	10															
12	x	x	x	x	x	F	F	F	F	F	x	x	7	7	x	x	x	x	x	x	x	x	x	x	7															
13	x	x	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	0	0															
14	3	x	x	7	7	3	7	7	x	x	x	x	x	3	3	x	x	x	x	x	7	7	x	3	3															
15	8	8	9	10	10	10	x	9	9	8	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F															
16	4	4	5	5	x	5	4	4	6	6	F	F	F	F	F	4	4	5	6	6	x	5	5	6	6															
17	0	0	x	x	5	4	3	3	5	5	4	4	1	2	2	1	1	3	0	2	F	F	F	F	F															
18	9	10	10	8	8	9	9	8	8	x	8	x	9	10	10	8	10	10	9	9	F	F	F	F	F															
19	1	x	2	2	x	0	0	2	1	1	1	1	2	0	0	0	0	2	2	1	F	F	F	F	F															
20	10	5	6	6	x	8	5	5	10	10	9	8	8	4	6	9	9	4	4	x	F	F	F	F	F															
21	x	x	x	x	x	x	x	x	x	x	10	x	x	9	9	x	x	9	10	10	10	10	9	9	x															
22	x	x	x	x	x	x	x	x	x	x	x	x	x	8	8	x	8	8	x	x	x	x	x	x	8															
<b>Função de avaliação</b>							<b>80</b>																																	
<b>Função de qualidade</b>							<b>80</b>																																	
<b>Função de inviabilidade</b>							<b>0</b>																																	

Figura 9 Quadro gerado pelo procedimento proposto

## 6 CONCLUSÕES

O problema de programação de quadros de horários foi mais bem compreendido após o acompanhamento da sua realização feita de forma manual. Assim foi possível verificar as suas características básicas. O tratamento manual do problema é árduo e necessita de experiência do programador, além de demandar um elevado tempo. Para buscar a automação da resolução do problema foi necessário estudar as técnicas descritas na seção 2 e assim aplicá-las verificando o seu desempenho diante do problema abordado.

Como contribuição deste trabalho, desenvolveu-se um algoritmo híbrido utilizando as metaheurísticas GRASP e Busca Tabu para o problema de programação de horários em escolas. Conseguiu-se produzir soluções finais exequíveis e com qualidade, em um tempo computacional viável. Após o desenvolvimento do sistema, baseado no algoritmo híbrido, vários quadros de horários puderam ser gerados em um curto espaço de tempo. Assim, os responsáveis pela programação dos quadros de horário podem escolher a partir de um determinado número de soluções aquela que será a melhor para a instituição.

Foi possível ainda, observar a importância da correta calibração dos parâmetros nos algoritmos implementados, pois, parâmetros mal dimensionados têm influência negativa no desempenho do algoritmo.

Vale ressaltar, no entanto, que tal procedimento ainda pode ser melhorado, com a inclusão do procedimento intraturmas-interturmas proposto em Souza (2000), de forma a alcançar melhores soluções mais rapidamente.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

- Abramson, D. (1991). "Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms", *Management Science*, 37:98-113.
- Back, T., Fogel, D.B., Michalewicz, Z. (1996) Handbook of Evolutionary Computation, Univ. Oxford Press, NY.
- Bardadym, V. A (1996) "Computer-Aided School and University Timetabling: The New Wave", In Burke, E.K. and Ross, P. (eds), Lecture Notes in Computer Science, 1153:22-45, Springer-Verlag.
- Burke, E.K., Cowling, P., Landa Silva, J.D. and McC, B. (2001) "Three Methods to Automate the Space Allocation Process in UK Universities", Lecture Notes in Computer Science, 2079: 254-276.
- Carter, M.W. and Laporte, G. (1998). "Recent Developments in Practical Course Timetabling", Lecture Notes in Computer Science, 1408: 3-19.
- Collorni, A., Dorigo, M. and Maniezzo, V. (1998) "Metaheuristics for High School Timetabling", *Computational Optimization and Applications*, 9:275-298.
- Costa, D. (1994). "A tabu search algorithm for computing an operational timetable". *European Journal of Operational Research*, 76:98-110.
- de Werra, D. (1995) "An introduction to timetabling", *European Journal of Operational Research*, 19:151-162.
- Even, S., et al. (1976) "On the complexity of timetabling and multicommodity flow problems", *SIAM Journal of Computation*, 5:691-703.
- Feo, T.A. and Resende, M.G.C. (1992) "Greedy randomized adaptive search procedures", *Journal of Global Optimization*, 6:109-133.
- Glover, F. (1986) Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research*, 5: 553-549.
- Glover, F. and Laguna, M. (1997) "Tabu Search", *Kluwer academic Publishers*, Boston.
- Hertz, A. (1992) "Tabu search for large scale timetabling problems", *European Journal of Operational Research*, 54:39-47.
- Junginger, W. (1986) "Timetabling in Germany – a survey", *Interfaces*, v. 16, pp 66-74.
- Mata, S.S. (1989) O problema de horário na escola de Segundo grau: modelagem e implementação. Dissertação de mestrado, Programa de Eng. De Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro.
- Papoulias, D.B. (1980) "The assignment-to-days problem in a school time-table, a heuristic approach", *European Journal of Operational Research Society*, v. 4, pp 31-41.
- Schaefer, A. (1999) "A survey of automated timetabling", *Artificial Intelligence Review*, 13:87-127.
- Souza, M.J.F. (2000) "Programação de horários em escolas: uma aproximação por metaheurísticas", Tese de Doutorado, Programa de Engenharia de Sistemas e Computação, UFRJ, Rio de Janeiro.
- Ueda, H., Ouchi, D., Takahashi, K. and Miyahara, T. (2001) "A Co-evolving Timeslot/Room Assignment Genetic Algorithm Technique for Universities Timetabling", Lecture Notes in Computer Science, 2079: 48-63.



UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP  
ESCOLA DE MINAS - EM  
COLEGIADO DO CURSO DE ENGENHARIA DE PRODUÇÃO - CEPRO

CERTIDÃO

Certifico que, aos 27 dias do mês de fevereiro de 2003, às 15:00 horas, na Universidade Federal de Ouro Preto, reuniu-se a Comissão Avaliadora designada para julgar a Monografia do graduando de Engenharia de Produção, intitulada “**Programação de Horários Em Escolas Via GRASP E Busca Tabu**”, sendo a referida Comissão composta pelos professores Marcone Jamilson Freitas Souza (orientador), Gustavo Peixoto Silva e Tatiana Alves Costa.

A Comissão Avaliadora resolveu considerar o trabalho aprovado e por ter atendido as exigências para defesa e recomendações da banca examinadora e, por verdade, eu, \_\_\_\_\_, Secretário (a) do Colegiado do Curso de Engenharia de Produção, assino a presente Certidão.

Ouro Preto, de            de            .

Secretário do CEPRO

Visto:

\_\_\_\_\_  
Marcone Jamilson Freitas Souza

\_\_\_\_\_  
Jorge Luiz Bréscia Murta