

**Guilherme Tadeu Silva Timóteo**

**Desenvolvimento de um Algoritmo Genético para a Resolução do *Timetabling***

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Orientador  
Prof. Joaquim Quinteiro Uchôa

Lavras  
Minas Gerais - Brasil  
2002



**Guilherme Tadeu Silva Timóteo**

**Desenvolvimento de um Algoritmo Genético para a Resolução do *Timetabling***

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 26 de Março de 2002

---

Prof. Jones Oliveira de Albuquerque

---

Prof. Renata Couto Moreira

---

Prof. Joaquim Quinteiro Uchôa  
(Orientador)

Lavras  
Minas Gerais - Brasil



A  
*Deus, aos meus pais  
José Timóteo e Édira,  
aos meus irmãos e a  
minha namorada Josi pelo apoio.*



## **Agradecimentos**

Agradeço a Deus.

Agradeço ao Prof. Joaquim pela orientação neste trabalho.

Agradeço ao Prof. André Zambalde pelas diversas oportunidades concedidas durante minha graduação.

Agradeço ao Prof. Édson e Augusto pela confiança em um trabalho de iniciação científica.

Agradeço ao CNPQ pelo apoio financeiro.

Agradeço ao meu tio José Alberto pelos constantes incentivos e minha prima Michele por sempre estar disponível em me ajudar.

Agradeço a todos os meus colegas pelos vários anos de convivência. Em especial aos colegas: Juliano (Boca), Marcelo (Salviano), Sérgio (Ceboso), Leandro (Flocos), Ari (Arirara), Hesli (Mandrake), Gustavo (Forrest), Rodrigo Marinho (Bright), Rodrigo Felício (Grilo), Rondinelli (Rondinho Benício).

Agradeço especialmente ao meu amigo Sílvio (do Suco de Clorifila). E por fim agradeço ao Awdrey, que estando na frente do CA me apoiou em vários cursos.



## **Resumo**

Os algoritmos genéticos utilizam conceitos provenientes do princípio de seleção natural proposto por Darwin, aliado aos princípios da genética propostos por Mendel, para abordar uma série ampla de problemas, em especial os de otimização. Robustos e facilmente adaptáveis, consistem em uma técnica amplamente estudada e utilizada em diversas áreas. Neste trabalho será realizada uma abordagem dos principais conceitos em relação aos algoritmos genéticos, bem como a implementação de uma aplicação para a resolução do problema da geração de grades horárias, objetivando avaliar e verificar a eficiência de alguns parâmetros genéticos.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Algoritmos Genéticos</b>	<b>3</b>
2.1	Introdução . . . . .	3
2.2	Evolução Histórica . . . . .	3
2.3	Terminologia Biológica . . . . .	4
2.4	Algoritmos Genéticos - Uma Breve Introdução . . . . .	5
2.5	Representação de Soluções . . . . .	6
2.6	Geração da População Inicial . . . . .	7
2.7	Avaliação dos Indivíduos . . . . .	8
2.8	Seleção dos Reprodutores . . . . .	8
2.8.1	Seleção Determinística . . . . .	9
2.8.2	Seleção por Roleta Giratória . . . . .	11
2.8.3	Seleção por Torneio . . . . .	12
2.8.4	Seleção Uniforme . . . . .	13
2.8.5	Seleção por Roleta com Redução . . . . .	13
2.9	Crossover . . . . .	14
2.9.1	Crossover com corte em um ponto (1PX) . . . . .	15
2.9.2	Crossover com corte em dois pontos (2PX) . . . . .	15
2.9.3	Crossover com múltiplos cortes (MPX) . . . . .	16
2.9.4	Crossover segmentado (SX) . . . . .	16
2.10	Mutação . . . . .	17
2.11	Critério de Sobrevivência . . . . .	17
2.12	Condições de término . . . . .	17
2.13	Pseudocódigo de um AG . . . . .	18
2.14	Parâmetros Genéticos . . . . .	19
2.15	Aplicações dos AGs . . . . .	19

<b>3</b>	<b>O Problema da Geração de Horários Escolares</b>	<b>21</b>
3.1	Introdução . . . . .	21
3.2	Formulações do Problema . . . . .	21
3.3	Abordagens do Problema . . . . .	22
3.4	Particularidades e restrições . . . . .	24
3.4.1	Colisão por Professor . . . . .	25
3.4.2	Horários Esparsos . . . . .	25
3.4.3	Blocos de Disciplinas . . . . .	26
3.4.4	Vários Blocos por Dia . . . . .	26
3.4.5	Média de aulas por dia . . . . .	27
3.4.6	Preferências dos professores . . . . .	27
3.5	Considerações Finais . . . . .	28
<b>4</b>	<b>Resolução do Timetabling Utilizando Algoritmos Genéticos</b>	<b>29</b>
4.1	Introdução . . . . .	29
4.2	Representação da Solução . . . . .	29
4.3	Criação da População Inicial . . . . .	30
4.4	Avaliação dos Indivíduos . . . . .	32
4.5	Seleção dos Indivíduos . . . . .	33
4.6	Crossover . . . . .	34
4.6.1	Crossover com corte em um ponto (1PX) . . . . .	34
4.6.2	Crossover com corte em dois pontos (2PX) . . . . .	39
4.6.3	Crossover heurístico com corte em um ponto . . . . .	40
4.7	Mutação . . . . .	43
4.7.1	Mutação heurística . . . . .	44
4.8	Verificação do critério de sobrevivência . . . . .	45
4.9	Critério de Parada . . . . .	45
4.10	Considerações Finais . . . . .	45
<b>5</b>	<b>A Aplicação Desenvolvida</b>	<b>47</b>
5.1	Introdução . . . . .	47
5.2	Dados de Entrada . . . . .	47
5.3	Configuração dos parâmetros genéticos . . . . .	48
5.4	Dados de saída . . . . .	52
<b>6</b>	<b>Resultados e Discussões</b>	<b>55</b>
6.1	Introdução . . . . .	55
6.2	Dados da Escola . . . . .	55

6.3	Seleção por Torneio . . . . .	56
6.4	Seleção por Roleta Giratória . . . . .	61
6.5	Método da Roleta com Redução . . . . .	67
6.6	Considerações Finais . . . . .	70



# Lista de Figuras

2.1	Funcionamento de um AG típico . . . . .	6
2.2	Estrutura para armazenar indivíduos selecionados . . . . .	10
2.3	Troca genética entre os indivíduos . . . . .	15
2.4	Troca genética entre os dois pontos de corte . . . . .	16
4.1	Representação da solução para o Timetabling . . . . .	30
4.2	Representação da população de soluções para o Timetabling . . . . .	31
4.3	Exemplo da geração aleatória de um indivíduo . . . . .	32
4.4	Exemplo da geração heurística de um indivíduo . . . . .	33
4.5	Exemplos de indivíduos representando uma grade horária . . . . .	35
4.6	Realização do crossover . . . . .	36
4.7	Realização do crossover . . . . .	37
4.8	Indivíduos e seus respectivos pontos de corte . . . . .	39
4.9	Crossover com dois pontos de corte . . . . .	40
4.10	Ponto de corte no crossover heurístico . . . . .	41
4.11	Estruturas de dados para o crossover heurístico . . . . .	42
4.12	Mudança aleatória no valor do gene . . . . .	43
4.13	Inversão da posição dos genes . . . . .	44
5.1	Formulário para gerenciamento de turmas, professores e disciplinas . . . . .	48
5.2	Formulário que configura as preferências dos professores . . . . .	49
5.3	Formulário que configura os parâmetros genético do algoritmo . . . . .	50
5.4	Grau de importância dos critérios . . . . .	51
5.5	Configuração do grau de importância dos critérios . . . . .	51
5.6	Gráfico exemplo . . . . .	53
5.7	Formulário com a melhor solução encontrada pelo algoritmo . . . . .	54
6.1	Comportamento do algoritmo para a configuração 1 . . . . .	58

6.2	Comportamento do algoritmo para a configuração 3 . . . . .	59
6.3	Comportamento do algoritmo para a configuração 1 do método da roleta . . . . .	62
6.4	Comportamento do algoritmo para a configuração 2 do método da roleta . . . . .	63
6.5	Comportamento do algoritmo para a configuração 3 do método da roleta . . . . .	64

# Lista de Tabelas

2.1	Tabela de Aptidões dos Indivíduos . . . . .	10
2.2	Intrevalo de Seleção para Roleta Giratória . . . . .	11
2.3	Grau de aptidão para o método de seleção por torneio . . . . .	13
2.4	Intrevalo de Seleção para Roleta com Redução . . . . .	14
2.5	Reajuste do intrevalo de seleção e grau de aptidão do indivíduo selecionado . . . . .	14
3.1	Horários Disponíveis . . . . .	22
3.2	Turmas e Professores . . . . .	23
3.3	Total de Possibilidades para as Turmas . . . . .	24
3.4	Horário gerado para a turma 5SHP . . . . .	25
3.5	Horário gerado para a turma 6SHP . . . . .	25
3.6	Exemplo de janela na grade horária . . . . .	26
3.7	Formação de blocos de disciplinas . . . . .	26
3.8	Exemplo de vários blocos de uma disciplina . . . . .	27
3.9	Verificação da média de aulas . . . . .	27
6.1	Importância dos critérios . . . . .	56
6.2	Parâmetros da configuração 1 para o método de torneio . . . . .	57
6.3	Parâmetros da configuração 2 para o método de torneio . . . . .	57
6.4	Parâmetros da configuração 3 para o método de torneio . . . . .	59
6.5	Parâmetros da configuração 4 para o método de torneio . . . . .	60
6.6	Parâmetros da configuração 5 para o método de torneio . . . . .	60
6.7	Parâmetros da configuração 6 para o método de torneio . . . . .	60
6.8	Parâmetros da configuração 7 para o método de torneio . . . . .	61
6.9	Parâmetros da configuração 8 para o método de torneio . . . . .	61
6.10	Parâmetros da configuração 1 para o método da roleta . . . . .	62
6.11	Parâmetros da configuração 2 para o método da roleta . . . . .	63

6.12	Parâmetros da configuração 3 para o método da roleta . . . . .	64
6.13	Parâmetros da configuração 4 para o método da roleta . . . . .	65
6.14	Parâmetros da configuração 5 para o método da roleta . . . . .	65
6.15	Parâmetros da configuração 6 para o método da roleta . . . . .	65
6.16	Parâmetros da configuração 7 para o método da roleta . . . . .	66
6.17	Parâmetros da configuração 8 para o método da roleta . . . . .	66
6.18	Parâmetros da configuração 9 para o método da roleta . . . . .	66
6.19	Parâmetros da configuração 10 para o método da roleta . . . . .	67
6.20	Parâmetros da configuração 11 para o método da roleta . . . . .	67
6.21	Parâmetros da configuração 12 para o método da roleta . . . . .	67
6.22	Parâmetros da configuração 13 para o método da roleta . . . . .	68
6.23	Parâmetros da configuração 14 para o método da roleta . . . . .	68
6.24	Parâmetros da configuração 1 para método da roleta com redução .	68
6.25	Parâmetros da configuração 2 para o método da roleta com redução	69
6.26	Parâmetros da configuração 3 para o método da roleta com redução	69
6.27	Parâmetros da configuração 4 para o método da roleta com redução	70
6.28	Parâmetros da configuração 5 para o método da roleta com redução	70
6.29	Parâmetros da configuração 6 para o método da roleta com redução	71
6.30	Parâmetros da configuração 7 para o método da roleta com redução	71
6.31	Parâmetros da configuração 8 para o método da roleta com redução	72
6.32	Parâmetros da configuração 9 para o método da roleta com redução	72
6.33	Parâmetros da configuração 10 para o método de roleta com redução	72
6.34	Parâmetros da configuração 11 para o método de roleta com redução	73
6.35	Parâmetros da configuração 12 para o método da roleta com redução	73

# Capítulo 1

## Introdução

A área de otimização combinatória se caracteriza pelo tratamento de problemas cujo conjunto de possíveis soluções é finito e enumerável. Problemas onde a melhor solução pode ser facilmente obtida por um método que a gere e a avalie, como por exemplo, um método de força bruta. No entanto, dependendo das características do problema, o conjunto de soluções possíveis pode ser tão vasto, que se torna impossível avaliar cada um de seus elementos em tempo aceitável. A resolução de problemas desse tipo geralmente requer o uso de computadores e algoritmos especiais.

Entre os algoritmos se destacam dois tipos básicos: os métodos exatos e as heurísticas. Os métodos exatos garantem a obtenção da melhor solução de um determinado problema, e as heurísticas fazem buscas no conjunto de soluções sem garantir a otimalidade da solução final obtida.

Muitos problemas têm características que inviabilizam o uso de métodos exatos, devido à rigidez desses modelos. Nesse caso, a opção é o uso das heurísticas. [Fog91] afirma que existem heurísticas mais gerais, que por meio de adaptações podem ser usadas para a resolução de vários problemas. Elas são referenciadas como meta-heurísticas e como exemplo de uma delas pode-se citar os algoritmos genéticos.

Os algoritmos genéticos constituem uma técnica bastante utilizada em problemas de otimização, e baseiam a lógica de seu funcionamento nas leis de evolução natural propostas por Charles Darwin, aliadas as idéias sobre genética propostas por Mendel.

O objetivo desse trabalho é investigar o comportamento de um algoritmo genético, realizando testes com diversos parâmetros. Entretanto, para que estes testes

sejam realizados, torna-se necessário a abordagem de um problema com grande número de soluções.

O problema adotado neste trabalho foi o da geração de grades horárias, conhecido na literatura como *Timetabling*. Este problema é NP-Hard [Eve76] e vários métodos têm sido utilizados para tentar resolvê-lo. Entre esses métodos estão os algoritmos genéticos, conforme descrito nos trabalhos de [Abr92], [Bur94], [Fil00].

Para abordar a resolução do *Timetabling* utilizando algoritmos genéticos, este trabalho foi dividido em seis capítulos e a estrutura de cada um deles será descrita a seguir:

**Capítulo 2** - introduz os conceitos relativo aos algoritmos genéticos, explorando sua estrutura, os diversos operadores, os métodos de seleção dos indivíduos e os diversos parâmetros genéticos que podem ser configurados.

**Capítulo 3** - trata exclusivamente do problema da geração de grades horárias, explorando suas variantes, particularidades e restrições.

**Capítulo 4** - aborda o desenvolvimento de um algoritmo genético para resolver o problema da geração de grades horárias.

**Capítulo 5** - descreve a aplicação que foi desenvolvida para a realização dos testes.

**Capítulo 6** - mostra os resultados obtidos com diversos parâmetros genéticos, comparando operadores, métodos de seleção, critérios de avaliação e outros quesitos implementados neste trabalho.

## Capítulo 2

# Algoritmos Genéticos

### 2.1 Introdução

Este capítulo visa apresentar a maneira pelo qual os princípios da evolução natural são utilizados na criação de algoritmos para obter soluções de problemas que possuem um espaço de busca muito grande.

O capítulo introduzirá os Algoritmos Genéticos (AGs) realizando uma evolução histórica (amplamente baseada no trabalho de [Fil00]) e em seguida explorará seus parâmetros, operadores, métodos e características.

### 2.2 Evolução Histórica

Até meados do século XIX, os naturalistas acreditavam que cada espécie fosse criada por um ser supremo ou através de geração espontânea. Essa realidade começou a ser mudada a partir do trabalho de classificação biológica dos organismos de Carolus Linnaeus. Este trabalho despertou o interesse pela similaridade entre certas espécies, levando a acreditar numa relação entre elas.

A partir daí, vários outros trabalhos apontaram para o que seria a teoria da evolução natural. Um deles foi o de Jean Baptiste Lamarck, que sugeriu uma teoria evolucionária do "uso e desuso dos órgãos".

Charles Darwin após vários anos de observações e experimentos apresentou em 1858 sua teoria de evolução natural. No ano seguinte, Darwin publica o *On the Origin of Species by Means of Natural Selection* com sua teoria completa.

Por volta de 1900, o trabalho de Gregor Mendel, desenvolvido em 1865, sobre os princípios básicos da genética, foi estudado por vários cientistas e teve grande

influência sobre os futuros trabalhos relacionados a evolução.

A moderna teoria da evolução combina a genética e as idéias de Darwin sobre seleção natural, criando assim o princípio básico da Genética Populacional, onde a variabilidade entre os indivíduos em uma população de organismos que se reproduzem sexualmente é produzida pela mutação e recombinação genética.

Somente nas décadas 50 e 60, os biólogos começam a desenvolver simulações computacionais de sistemas genéticos. Entretanto, foi Jonh Holland quem começou, seriamente, a desenvolver as primeiras pesquisas sobre o tema. Holland foi gradualmente refinando suas idéias e em 1975 publicou o seu livro *Adaptation in Natural and Artificial System*, hoje considerado a "bíblia" dos algoritmos genéticos. A partir daí, os algoritmos genéticos começaram a ser utilizados em várias áreas.

## 2.3 Terminologia Biológica

A terminologia biológica é muito importante para a compreensão do funcionamento dos AGs. Eis os principais termos, segundo [Mit96]:

- **Cromossomo:** textitstring que representa uma determinada característica da solução ou a própria solução;
- **Gene:** característica particular de um cromossomo. O cromossomo é composto por um ou mais genes;
- **Alelo:** valor de determinado gene;
- **Locus:** determinada posição do gene no cromossomo;
- **Genótipo:** estrutura que codifica uma solução. Um genótipo pode ser formado por um ou mais cromossomos;
- **Fenótipo:** decodificação ou o significado da estrutura;
- **Fitness:** significa aptidão. O quanto o indivíduo é apto para determinado ambiente.

## 2.4 Algoritmos Genéticos - Uma Breve Introdução

Após o trabalho de [Hol75], os AGs passaram a ser largamente utilizados em tarefas de busca e otimização, exigidas para a resolução de vários problemas que possuem um elevado número de restrições e um espaço de busca muito grande.

Além dos AGs, existem outros mecanismos conhecidos como tradicionais ou exatos. Estes métodos operam com um único ponto e utilizam recursos matemáticos para tentar encontrar uma solução para o problema. São totalmente voltados para um modelo matemático e por isso tornam-se métodos determinísticos [Gol89].

Já os AGs, operam sobre uma população de candidatos a solução para o problema e cada indivíduo é avaliado isoladamente. Devido a essa característica, os AGs possuem um *paralelismo implícito*.

As principais características que diferenciam os AGs de métodos tradicionais são as seguintes [Gol89]:

- **Parâmetros:** Os AGs trabalham com a codificação dos parâmetros e não com os parâmetros propriamente;
- **Número de soluções:** Os AGs trabalham com uma população de indivíduos (representando um conjunto de soluções) e não com um único ponto (representando uma única solução);
- **Avaliação das soluções:** Os AGs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar;
- **Regras:** Os AGs utilizam regras probabilísticas e não determinísticas.

Os AGs são uma forma da estratégia gerar-e-testar, realizando os teste baseados nos parâmetros da evolução biológica. São capazes de encontrar boas soluções ou até mesmo soluções ótimas, porém esse fato não é garantido pelo algoritmo.

Uma desvantagem notável é a variação dos parâmetros genéticos do algoritmo em cada problema. Dessa forma, para resolução de determinado problema torna-se necessário um estudo particular a respeito do mesmo.

Um AG opera sobre uma população fazendo com que esta evolua de acordo com uma função de avaliação. O funcionamento é iterativo, começando com a geração de uma população inicial, seguida do processo de avaliação, seleção, crossover e mutação, que ocorre a cada geração (iteração) até que seja atingido algum critério para o fim do processo. Os passos gerais e o processo iterativo de um AG

são ilustrados na figura 2.1. Cada passo pode ser realizado de várias maneiras e pode variar de problema para problema.

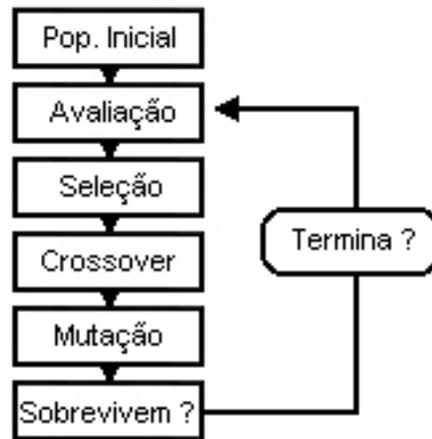


Figura 2.1: Funcionamento de um AG típico

Na maioria dos casos, para a condição de parada pode ser considerado o número de gerações ou tempo de processamento. Porém antes dos detalhes de funcionamento é necessário abordar a representação de soluções.

## 2.5 Representação de Soluções

O primeiro passo ao se implementar um AG é criar uma representação eficiente para as soluções. Na sua forma convencional, proposta por [Hol75], um AG trabalha normalmente com uma representação binária (0 e 1) para associar uma solução ou componentes de uma solução do problema abordado.

Embora esta representação (binária) tenha se mostrado eficiente para vários problemas, observou-se que à medida que foram crescendo as aplicações de AGs, esta passou a não ser a mais adequada. Dessa forma, surgiram alternativas, como por exemplo, a representação por um vetor de números inteiros no caso do caixeiro viajante, onde cada inteiro representa uma cidade e o vetor representa a rota a ser tomada.

A escolha da representação dos indivíduos é um ponto crucial para AGs, sendo um dos fatores mais importantes no seu desenvolvimento e posterior desempenho. Ao optar por uma representação é necessário avaliar, características como [Koz92]:

- a) completude: determina se é possível representar todos os fenótipos possíveis, ou seja, verificar se todas as soluções podem ser decodificadas;
- b) coerência: indica se a partir do esquema de representação é possível gerar um genótipo que codifique um fenótipo não pertencente ao espectro de soluções do problema. No caso do caixeiro viajante seria a geração de uma rota inexistente;
- c) simplicidade: representa o grau de complexidade dos atos de codificação e decodificação das soluções;
- d) localidade: pequenas alterações no genótipo acarretam pequenas alterações em seu fenótipo correspondente.

É importante observar que a utilização de uma representação binária implica numa redução considerável da complexidade de implementação do algoritmo. Isto porque se utilizada uma representação como esta, os vários métodos presentes no algoritmo poderão ser implementados apenas com operações booleanas simples como: AND, NOT e OR. Desta forma, a representação binária é bem desejável em um AG [Mit96].

## 2.6 Geração da População Inicial

A inicialização básica de um algoritmo genético clássico se resume à síntese de uma população inicial, sobre a qual serão aplicadas as ações dos passos subsequentes do processo. Tipicamente, se faz uso de funções aleatórias para gerar os indivíduos, sendo este um recurso simples que visa a fornecer maior biodiversidade <sup>1</sup>, fundamental para garantir uma boa abrangência do espaço de busca.

Além do método randômico, existem outros métodos para gerar a população inicial. Um deles pode ser a utilização de heurísticas. Neste caso, é preciso ter um cuidado especial, pois os indivíduos gerados podem ser semelhantes e isso pode prejudicar fortemente a biodiversidade do AG.

Os operadores de inicialização mais comuns são, segundo [Gol89] :

- a) Inicialização randômica uniforme: cada gene do indivíduo receberá como valor, um elemento do conjunto de alelos sorteado de forma aleatoriamente uniforme;

---

<sup>1</sup> Critério que avalia o quanto às características de cada indivíduo constituinte de uma população são diferentes. Pode parecer estranho, mas uma boa população é aquela formada por indivíduos bons, médios e ruins garantindo assim a diversidade das soluções.

b) Inicialização randômica não uniforme: determinados valores a serem armazenados no gene tendem a ser escolhidos com uma frequência maior do que o restante;

c) Inicialização randômica com "dope": indivíduos otimizados são inseridos em meio à população aleatoriamente gerada. Esta alternativa apresenta o risco de fazer com que um ou mais *superindivíduos* tendam a dominar no processo de evolução e causar o problema de convergência prematura<sup>2</sup>;

d) Inicialização heurística: indivíduos são criados a partir de heurísticas. Essa alternativa apresenta conforme já descrito, a possibilidade de gerar indivíduos semelhantes prejudicando a biodiversidade. Devido a esse fato, torna-se necessário utilizar métodos aleatórios em conjunto com as heurísticas.

## 2.7 Avaliação dos Indivíduos

Nesta etapa cada indivíduo da população sofre um processo de avaliação, visando retornar seu grau de aptidão(fitness), ou seja, o quanto cada indivíduo é apto para determinadas condições. Atualmente, existem várias formas de avaliação utilizadas [Mit96]:

- Em casos de otimização de funções matemáticas, tende a ser escolhido como grau de aptidão de determinado indivíduo, o próprio valor de retorno das funções ao se aplicar como parâmetro a decodificação desse indivíduo.
- Em problemas com muitas restrições, funções baseadas em penalidades são mais comuns, ou seja, quanto menos penalidades, maior grau de aptidão do indivíduo.

A função de avaliação também é chamada de função objetivo ou função de adaptação em um grande número de trabalhos.

## 2.8 Seleção dos Reprodutores

Na etapa de seleção, os indivíduos são escolhidos para posterior cruzamento. Neste ponto, fazendo uso do grau de aptidão de cada um, efetua-se um processo onde os

---

<sup>2</sup> Fato que ocorre quando os indivíduos de uma população possuem muitas características em comum, dificultando uma melhor abrangência do espaço de busca.

mais aptos possuirão uma maior probabilidade de se reproduzirem. O grau de aptidão é calculado a partir da aplicação da função de avaliação em cada indivíduo, determinando o quão apto ele está para a reprodução em relação à população a que pertence. Este é o passo mais importante de um AG, pois uma má escolha dos reprodutores pode acarretar resultados indesejáveis.

Em [Gol89], são citados alguns métodos de seleção:

- Seleção Determinística [Bri79];
- Seleção por Roleta Giratória [Hol75];
- Seleção por Torneio [Wet83];
- Seleção Uniforme [Bac85].

Um método de seleção alternativo foi proposto neste trabalho. Trata-se do método de seleção por Roleta Giratória com Redução, que será detalhado em seções posteriores assim como os outros métodos de seleção citados.

### 2.8.1 Seleção Determinística

Para este caso, os indivíduos são selecionados através do cálculo da expectativa do número de descendentes. Este cálculo é feito através do grau de aptidão de cada indivíduo:

$$ND(x) = \frac{f(x)}{\sum_{x=1}^n f(x)} * n$$

Onde:

- $ND(x)$  - número de descendentes do indivíduo  $x$ .
- $f(x)$  - função que retorna o grau de aptidão do indivíduo  $x$ .
- $n$  - número de indivíduos da população.

Somente para ilustrar o funcionamento deste método de seleção, segue um exemplo com uma população de quatro indivíduos, que pode ser observado na tabela 2.1.

Os indivíduos selecionados ficam armazenados temporariamente em uma estrutura de dados. O número de descendentes que cada indivíduo possui está relacionado na tabela 2.1. Este número é obtido pelo arredondamento de  $ND(x)$ . Para o caso do exemplo, uma possível estrutura está ilustrada na figura 2.2.

**Tabela 2.1:** Tabela de Aptidões dos Indivíduos

Indivíduo	$f(x)$	$ND(x)$	Descendentes
1	169	0.58	1
2	576	1.97	2
3	64	0.22	0
4	361	1.23	1

2	2	1	4
---	---	---	---

**Figura 2.2:** Estrutura para armazenar indivíduos selecionados

É importante observar que conforme a figura 2.2, o indivíduo 2, recebeu duas cópias na estrutura e os indivíduos 1 e 4 receberão uma. O indivíduo 3 por ter um grau de aptidão muito baixo e número de descendentes igual a zero não participará do processo de cruzamento. Baseados na estrutura da figura 2.2, os casais são selecionados na maioria das vezes de forma aleatória. Uma possível escolha poderia ser:

Casal 1: Indivíduo 2 X Indivíduo 4

Casal 2: Indivíduo 1 X Indivíduo 2

Existem outros métodos para a escolha de casais além do aleatório. Alguns deles são citados por [Gol89]:

- a) *In Breeding* - indivíduos semelhantes são combinados;
- b) *Out Breeding* - indivíduos com diferentes características são combinados;
- c) *Self-Fertilization* - um indivíduo isolado é capaz de se reproduzir;
- d) *Clonal Propagation* - réplicas de determinados indivíduos são inseridas na população de descendentes;
- e) *Positive-assortive* - bons indivíduos são combinados com outros bons indivíduos;
- f) *Negative-assortive* - indivíduos não desejados são combinados.

Segundo [Gol89], um AG pode combinar esses métodos num mesmo algoritmo. A porcentagem de uso de cada método deve ser abordada para cada problema especificamente. De uma maneira geral, o método de escolha aleatória é o mais utilizado e também o mais eficiente [Gol89].

### 2.8.2 Seleção por Roleta Giratória

Este método foi proposto por [Hol75] e visa selecionar os indivíduos através do giro aleatório de uma roleta. Entretanto, antes do giro, cada indivíduo adquire uma determinada porção da roleta (calculada baseado no grau de aptidão de cada um). Quanto maior o grau de aptidão, maior a porção na roleta e conseqüentemente maiores as chances de seleção. Observando por uma perspectiva algorítmica, pode-se dizer que o funcionamento deste método é o seguinte:

1. Calcule a soma do grau de aptidão de todos os indivíduos.;
2. Sorteie um número  $s$  entre 1 e a soma das aptidões;
3. Selecione o Indivíduo  $x$  tal que  $s \in [(\sum_{x=1}^{n-1} f(x)) + 1, \sum_{x=1}^n f(x)]$ .

**Tabela 2.2:** Intervalo de Seleção para Roleta Giratória

Indivíduo	$f(x)$	Intervalo de Seleção
1	169	1 - 169
2	576	170 - 745
3	64	746 - 809
4	361	810 - 1171

O intervalo de seleção é dado através dos somatórios. O primeiro valor constituinte do intervalo é obtido através do  $(\sum_{x=1}^{n-1} f(x)) + 1$  e o segundo através do  $\sum_{x=1}^n f(x)$ , lembrando que  $f(x)$  é a função que avalia o grau de aptidão do indivíduo  $x$ .

Conforme o funcionamento da roleta giratória, é necessário realizar o sorteio de um número. Suponha que esse número fosse 712. Nesse caso, o indivíduo selecionado seria o indivíduo 2, pois 712 é um valor que se encontra entre o intervalo de seleção desse indivíduo. Como a população possui quatro indivíduos, seria necessária a realização de quatro sorteios e através destes seriam formados os casais para a reprodução.

O método de seleção por roleta giratória tem um grande problema: *o superindivíduo*. Isto ocorre quando um indivíduo possui um intervalo de seleção bem maior que os outros. Neste caso, como as chances desse indivíduo ser selecionado é bem evidente, a tendência é que este indivíduo tenha muitos descendentes nas próximas gerações, o que poderia levar o algoritmo para uma convergência prematura.

Para resolver este problema, [Gol89] propôs um método para escala do grau de aptidão, chamado de *linear scaling*. O objetivo desse método é reduzir a probabilidade de superindivíduos e fazer com que as chances dos indivíduos médios da população serem sorteados aumente. Para realizar esta tarefa, o método faz com que o valor retornado pela função objetivo sofra uma alteração. Essa alteração é provocada por uma função  $fa$  que recebe como parâmetro o valor retornado pela função de avaliação  $f(x)$ . Segundo [Gol89] essa função seria  $fa(f(x)) = a * f(x) + b$ , onde:

$$a = \frac{(C_{mult} - 1.0) * U_{avg}}{U_{max} - U_{avg}}$$

$$b = \frac{U_{avg} * (U_{max} - C_{mult} * U_{avg})}{U_{max} - U_{avg}}$$

Segue abaixo o significado dos termos:

$U_{avg}$  - média do grau de aptidão dos indivíduos

$U_{max}$  - maior grau de aptidão entre os indivíduos

$C_{mult}$  - fitness multiple

Após a realização de vários experimentos, [Gol89] recomenda que o valor da constante  $C_{mult}$  varie de 1.2 a 2.0 para os problemas onde o número de indivíduos presentes na população varia de 50 a 100. Um outro detalhe é que a função  $fa(x)$  não pode receber como parâmetro valores negativos, pois isso geraria resultados incoerentes.

### 2.8.3 Seleção por Torneio

Este método foi proposto por [Wet83] e funciona de maneira semelhante a um torneio tradicional. Basicamente ocorre um sorteio de dois indivíduos pertencentes a população e vence o torneio aquele que tiver o maior grau de aptidão. Dessa

forma, todos os indivíduos que venceram seus torneios serão selecionados pelo método.

**Tabela 2.3:** Grau de aptidão para o método de seleção por torneio

Indivíduo	$f(x)$
1	169
2	576
3	64
4	361

Baseado na tabela 2.3, suponha que os indivíduos sorteados foram 1 e 2. Nesse caso, quem vence o torneio é o indivíduo 2, pois seu grau de aptidão é o maior. Devido a esse fato, este indivíduo é selecionado para cruzamento.

Esse método possui a grande vantagem da não geração de superindivíduos. Isso porque, a chance do indivíduo com maior grau de aptidão ser selecionado é a mesma, independentemente do quão grande é seu grau de aptidão. No exemplo ilustrado na tabela 2.3, a chance do indivíduo 2 ser selecionado é  $1/4$ , pois se for sorteado, independentemente de quem seja seu rival, ele vencerá o torneio. Se o grau de aptidão fosse 3000 ao invés de 576, as chances de seleção continuariam as mesmas neste método. Já no método da roleta, o intervalo de seleção iria aumentar muito e por isso a chance do indivíduo ser selecionado também iria ser bem maior.

#### 2.8.4 Seleção Uniforme

Este método reduz a seleção a uma escolha aleatória. Cada indivíduo tem a mesma chance de ser selecionado, independentemente do grau de aptidão. Se utilizado sozinho este método reduz o AG a uma busca cega.

Porém, em alguns casos, este método pode complementar outros métodos de seleção. Um AG pode utilizar o método de seleção por roleta e em alguns momentos utilizar a seleção uniforme para diversificar a população, ou seja, selecionar também indivíduos ruins que seriam dificilmente seriam selecionados pela roleta.

#### 2.8.5 Seleção por Roleta com Redução

Este método foi proposto e utilizado neste trabalho, e funciona de maneira semelhante ao método da roleta giratória. A única diferença é que toda vez que um indivíduo for selecionado, o seu grau de aptidão diminui em  $x\%$ , fazendo com que seu intervalo de seleção também diminua.

**Tabela 2.4:** Intervalo de Seleção para Roleta com Redução

Indivíduo	$f(x)$	Intervalo de Seleção
1	169	1 - 169
2	576	170 - 745
3	64	746 - 809
4	361	810 - 1171

É necessário um exemplo para que se possa investigar o comportamento desse método de seleção. Assim como no método da roleta giratória, sorteia-se um número entre 1 e a soma das aptidões dos indivíduos. Suponha que o número sorteado fosse o 1090. Neste caso, o indivíduo 4 será selecionado devido ao seu intervalo de seleção. A diferença desse método para o da roleta giratória é que no próximo sorteio o indivíduo 4 terá seu intervalo de seleção reduzido. Se o fator de redução utilizado fosse 2%, o grau de aptidão e o intervalo de seleção do indivíduo 4 teria que ser reajustado de acordo com este percentual.

**Tabela 2.5:** Reajuste do intervalo de seleção e grau de aptidão do indivíduo selecionado

Indivíduo	$f(x)$	Intervalo de Seleção
1	169	1 - 169
2	576	170 - 745
3	64	746 - 809
4	354*	810 - 1163*

É importante observar que tanto o grau de aptidão quanto o intervalo de seleção do indivíduo 4 diminuam. Com isto, as chances desse indivíduo ser selecionado diminuem e talvez as chances de convergência prematura também, porém esse fato não é garantido.

## 2.9 Crossover

O crossover é a operação que realiza a troca genética entre os cromossomos, ou seja, aquela que irá combinar os genes dos indivíduos selecionados na etapa anterior (seleção) para gerar dois novos indivíduos.

Existem vários operadores de crossover, dentro os quais citados por [Fil00]:

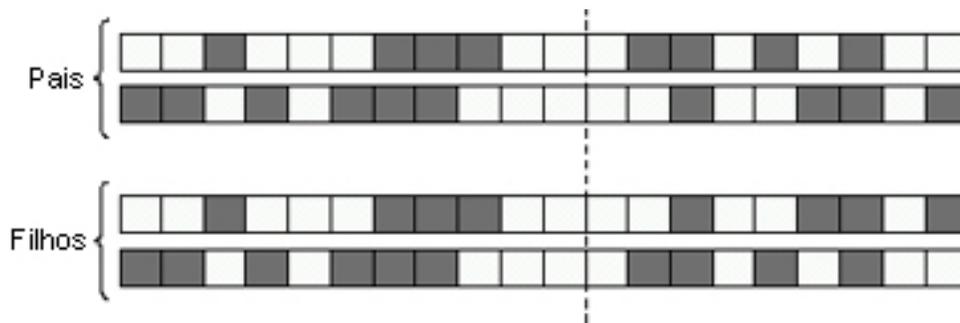
- Crossover com corte em 1 ponto (1PX);

- Crossover com corte em 2 pontos (2PX);
- Crossover com múltiplos pontos (MPX);
- Crossover segmentado (SX);
- Crossover uniforme.

Visando fins didáticos, um operador de crossover heurístico foi desenvolvido para este trabalho. Seu funcionamento será descrito detalhadamente no Capítulo 4.

### 2.9.1 Crossover com corte em um ponto (1PX)

Considerado padrão para um algoritmo genético, este operador sorteia aleatoriamente um ponto no cromossomo (chamado ponto de corte) e partir desse ponto realiza a troca genética gene a gene entre os indivíduos relacionados.



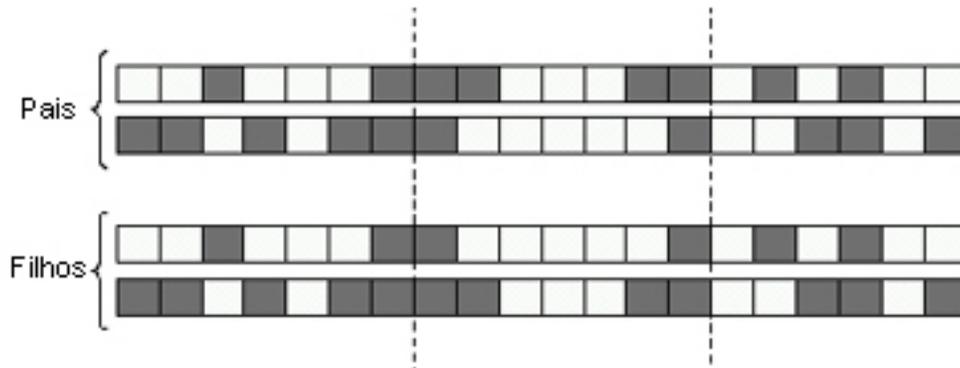
**Figura 2.3:** Troca genética entre os indivíduos

Cada bloco na figura 2.3 representa um gene e a linha tracejada representa o ponto de corte. É importante observar que a partir desse ponto, todos os genes são trocados entre os indivíduos ancestrais (pais), originando dois novos indivíduos chamados descendentes (filhos). Dessa forma, os indivíduos descendentes (filhos) serão a combinação dos seus ancestrais.

### 2.9.2 Crossover com corte em dois pontos (2PX)

Este operador funciona de maneira semelhante ao operador de crossover 1PX. A única diferença é o uso de dois pontos de corte ao invés de apenas um. Portanto,

os indivíduos ancestrais realizarão a troca genética entre esses pontos.



**Figura 2.4:** Troca genética entre os dois pontos de corte

Repare na figura 2.4 que os indivíduos descendentes (filhos) serão originados a partir da combinação genética dos ancestrais (pais) na região situada entre os pontos de corte.

### 2.9.3 Crossover com múltiplos cortes (MPX)

O cruzamento multi-ponto é uma generalização dos operadores apresentados anteriormente. Para este operador, será sorteado um número fixo ( $n$ ) de pontos de corte. Ao invés de um ou dois pontos, serão vários pontos de corte.

Na maioria dos casos, este operador não funciona de maneira eficiente. Em geral, os operadores de crossover 1PX são os que apresentam melhores resultados [Gol89].

### 2.9.4 Crossover segmentado (SX)

Este operador é apenas uma variação do operador de crossover MPX. Ao invés de ser sorteado um número fixo de pontos, serão sorteados números variáveis de pontos. Cada vez que o crossover for realizado, um novo número de pontos será sorteado.

Tanto esse quanto o operador anterior são mais complexos de serem implementados e na maioria dos casos não apresentam bons resultados [Gol89].

## 2.10 Mutação

A mutação é um recurso que permite a aleatoriedade em um AG. Trata-se de mudar o valor de um gene de forma aleatória. É utilizado para que o algoritmo tenha uma maior abrangência do espaço de busca, não ficando restrito aos indivíduos gerados pelo cruzamento genético.

A mutação opera sobre os indivíduos resultantes do processo de cruzamento com uma probabilidade pré-determinada efetuando algum tipo de alteração em sua estrutura. Geralmente a probabilidade é bem pequena para não fazer do processo uma busca cega. Os principais operadores de mutação são descritos a seguir [Gol89]:

- a) Mutação aleatória (Flip Mutation): cada gene a ser mutado recebe um valor sorteado do alfabeto válido;
- b) Mutação por troca (Swap Mutation): são sorteados  $n$  pares de genes, e os elementos do par trocam os valores desses genes entre si;
- c) Mutação creep: um valor aleatório é somado ou subtraído do valor do gene.

## 2.11 Critério de Sobrevivência

Nesta etapa, os indivíduos resultantes do processo de cruzamento e mutação (descendentes) formarão a nova população segundo a política adotada pelo AG. As formas de políticas mais comuns segundo [Gol89], são:

- a) Os descendentes (filhos) sempre substituem os ancestrais (pais);
- b) Os descendentes (filhos) substituem os ancestrais (pais) somente se a média do grau de aptidão dos filhos for maior que a média de aptidão dos pais.

A maioria dos algoritmos utiliza o método de sempre substituir os ancestrais pelos descendentes, devido ao fato deste método ajudar a manter a biodiversidade dos indivíduos [Gol89].

## 2.12 Condições de término

São as condições que determinam o fim do processo iterativo. Existem algumas maneiras de terminar o processamento de um AG. [Fil00] cita:

- a) Tempo;
- b) Número de Gerações;
- c) Convergência (95 % dos genes iguais entre os indivíduos).

Adotar o critério de convergência pode ser arriscado em alguns casos. Podem ocorrer situações em que os indivíduos demorem um tempo relativamente grande para se convergirem. O tempo pode ser abstrato, pois no mesmo intervalo de tempo máquinas diferentes podem realizar diferentes processamentos. O número de gerações, na maioria dos casos é a melhor escolha para terminar o processamento de um AG, pois independe da máquina e torna o critério de término mais realístico.

## 2.13 Pseudocódigo de um AG

O primeiro passo é criar uma população inicial e logo em seguida calcular o grau de aptidão dos indivíduos gerados. Após isso, se inicia o processo iterativo até a condição de término ser atingida.

Após a seleção dos casais, é necessário realizar um crossover (cruzamento) com probabilidade  $P_c$  (segundo [Mit96],  $P_c = 0.7$  na maioria dos casos). O resultado do crossover são dois novos descendentes resultantes da troca genética entre os ancestrais.

```
Início
  Crie uma população inicial
  Calcule o grau de aptidão dos indivíduos
  Enquanto não atingiu a condição de término
    Selecione os indivíduos e forme os casais
    Realize crossover com probabilidade  $P_c$ 
    Para cada gene dos descendentes
      Realize uma mutação com probabilidade  $P_m$ 
    Se utiliza critério de sobrevivência
      Verifique os sobreviventes
    Calcule grau de aptidão dos sobreviventes
  Fim do Enquanto
Fim
```

A mutação deve ser aplicada com probabilidade  $P_m$  em cada gene dos descendentes. Segundo [Mit96], geralmente  $P_m = 0.001$ . Se o algoritmo levar

em conta critérios de sobrevivência é necessário avaliar e decidir quem formará a população para a próxima geração: descendentes ou ancestrais. Se o critério de sobrevivência não for utilizado, os descendentes formarão a nova população independente de qualquer fator.

## 2.14 Parâmetros Genéticos

Ao se estudar um AG, é importante analisar a influência de alguns parâmetros no seu comportamento, para que se possa configurá-los de acordo com as necessidades do problema e dos recursos disponíveis. Pode-se citar como tais parâmetros:

- **Tamanho da População:** o tamanho da população afeta o desempenho global e a eficiência dos AGs. Uma população formada por poucos indivíduos pode fazer com que o desempenho do AG caia. Isso porque a população pode fornecer uma pequena cobertura do espaço de busca para o problema. Uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, populações muito grande exigem maiores recursos computacionais e também pode acarretar o problema da convergência muito lenta.
- **Taxa de cruzamento:** quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. No entanto, se esta taxa for muito alta, as boas estruturas podem ser quebradas mais facilmente. Se esta taxa for baixa, o algoritmo pode se tornar muito lento.
- **Taxa de mutação:** uma baixa taxa de mutação previne que determinada posição do cromossomo fique estagnada em um valor. Uma alta taxa faz com que a busca se torna essencialmente aleatória.

## 2.15 Aplicações dos AGs

Os algoritmos genéticos podem ser utilizados em diversas áreas com bastante sucesso. [Mit96] cita os seguintes:

- **Otimização:** usados em uma grande variedade de problemas de otimização, incluindo otimização de problemas numéricos e combinatoriais, otimização do *layout* de circuitos eletrônicos entre uma série de outros problemas.

- **Aprendizagem de máquina:** usados para muitas aplicações em aprendizagem, incluindo classificação e predição de tarefas, tais como: previsão do tempo ou estrutura de proteínas. Usados também para desenvolver aspectos particulares de sistema de aprendizagem, tais como: pesos para redes neurais e sensores para robôs.
- **Economia:** usados para modelar processos de inovação, o desenvolvimento de estratégias de lances e na predição de mercados econômicos.
- **Ecologia:** utilizados para modelar fenômenos ecológicos como competição entre nichos ecológicos, co-evolução parasita-hospedeiro, simbiose e fluxo de recursos.
- **Genética:** utilizados para estudar questões da genética de populações, tais como a viabilidade das condições de um gene em termos evolucionários.
- **Sistemas sociais:** usados para estudar aspectos evolucionários de sistemas sociais, tais como a evolução do comportamento social em colônias de insetos, e mais genericamente, a evolução da cooperação e comunicação em sistemas multi-agentes.

## Capítulo 3

# O Problema da Geração de Horários Escolares

### 3.1 Introdução

O problema da geração de horários escolares, conhecido na literatura como *Timetabling*, consiste em arranjar encontros entre professores e alunos em um período de tempo previamente fixado (tipicamente uma semana) de modo a satisfazer um conjunto de restrições.

A solução manual do problema além de ser demorada, pode ocasionar grades horárias não ideais, como por exemplo, uma grade onde um professor leciona duas aulas ao mesmo tempo.

Vários métodos de otimização foram tentados, mas a maioria encontra problemas ao lidar com o grande número de restrições e a alta variedade de seu grau de importância [Fil00].

### 3.2 Formulações do Problema

A literatura cita muitas variantes do problema que ocorrem devido as necessidades específicas de algumas instituições. [Sch99] cita três classes para esse problema:

- **School Timetabling**: seqüenciamento semanal das aulas de uma escola, evitando que professores e alunos tenham mais de uma aula ao mesmo tempo;
- **Course Timetabling**: seqüenciamento semestral das aulas de um conjunto

de cursos de uma universidade, evitando a simultaneidade de cursos com estudantes em comum;

- **Examination Timetabling:** seqüenciamento de exames de um conjunto de cursos em uma universidade, evitando exames simultâneos de cursos com estudantes em comum, e espalhando os exames o máximo possível.

É importante lembrar que cada divisão não é restrita e pode haver situações com características particulares.

### 3.3 Abordagens do Problema

O problema *Timetabling* é NP-Hard [Eve76] e várias técnicas têm sido experimentadas para automação deste problema. Historicamente, o estudo do problema começou em 1963 por Gotieb e a partir daí vários artigos têm sido publicados em conferências e vários sistemas têm sido desenvolvidos [Sch99].

O número de possíveis soluções para esse problema é bastante grande. Considerando um exemplo real, ou seja, a grade horária do Colégio Nossa Senhora de Lourdes, uma escola de ensino fundamental e médio, localizada em Lavras-MG, pode-se verificar o quão grande é esse número de soluções.

Devido ao número de alunos e espaço físico dessa escola, as aulas são realizadas sempre na parte da manhã, sendo 6 horários disponíveis. A tabela 3.1 ilustra os horários disponíveis. Cada horário (ex: segunda às 8:00 horas) é considerado um *slot*. No caso da tabela 3.1, cada turma terá 30 *slots* disponíveis ( 5 dias X 6 horários).

**Tabela 3.1:** Horários Disponíveis

--	Segunda	Terça	Quarta	Quinta	Sexta
7:00	—	—	—	—	—
7:50	—	—	—	—	—
8:55	—	—	—	—	—
9:45	—	—	—	—	—
10:45	—	—	—	—	—
11:25	—	—	—	—	—

Para o cálculo do número de soluções possíveis é necessário levar em conta o número de professores para cada turma. Devido a este fato, o total de possibilidades deve ser calculado separadamente para cada turma. Uma relação que mostra a

turma e o respectivo número de professores para o caso do colégio adotado como exemplo é ilustrada na tabela 3.2 :

**Tabela 3.2:** Turmas e professores para o Colégio Nossa Senhora de Lourdes

Turma	Número de Professores
5SHP	8
6SHP	9
7SHP	8
8SHP	9
1M	12
2M	13
3M	14

O número de soluções possíveis pode ser calculado através de um arranjo combinatório, envolvendo o número de professores e também o número de textslots disponíveis.

$$\text{Total de Possibilidades} = A_{n,p} :$$

$A_{n,p}$  - arranjo de  $n$  termos combinados  $p$  a  $p$ .  
 $n$  - número de slots disponíveis.  
 $p$  - número de professores que a turma possui.

Baseado nessa fórmula, o número de possibilidades para cada turma pode ser observado na tabela 3.3. Para calcular o total de soluções possíveis é preciso realizar uma multiplicação entre o número de possibilidades de cada turma.

$$\text{Total de Possibilidades} = 8,1 * 10^{100}$$

Note que o número de soluções possíveis é realmente muito grande. Se um método de força bruta <sup>1</sup> fosse utilizado (considerando que a máquina onde o algoritmo será processado consiga gerar e avaliar uma solução em 1 *ms*), o tempo de processamento seria:

$$\text{Tempo total} = 8,1 * 10^{100} \text{ms} .$$

<sup>1</sup>Um método que apenas gerasse uma solução e a testasse, comparando com a melhor solução encontrada até o momento.

Se este tempo fosse medido em anos:

$$\text{Tempo total} = 2,6 * 10^{90} \text{ anos} .$$

**Tabela 3.3:** Total de Possibilidades para as Turmas

Turma	Número de Possibilidades
5SHP	$2,3 * 10^{11}$
6SHP	$5,2 * 10^{12}$
7SHP	$2,3 * 10^{11}$
8SHP	$5,2 * 10^{12}$
1M	$2,2 * 10^{16}$
2M	$3,9 * 10^{17}$
3M	$6,6 * 10^{18}$

Nesse caso, aplicar um método de força bruta é totalmente inviável. É importante lembrar que existem instituições que possuem muito mais professores e horários disponíveis, tornando este fato ainda mais considerável.

### 3.4 Particularidades e restrições

O enfoque principal deste trabalho será a variante citada como *School Timetabling*, já que ela se adapta de uma forma coerente á realidade da maioria das escolas de ensino fundamental e médio.

Tipicamente, essas escolas possuem um número de turmas e alunos determinado pelas suas limitações físicas. Em geral, existem mais turmas do que salas, pois trabalham em períodos matutinos, vespertinos e em alguns casos em períodos noturnos.

Cada turma possui uma lista de disciplinas obrigatórias que variam de acordo com o currículo escolar. Geralmente, as aulas de determinada turma são em apenas um período (matutino, vespertino ou noturno).

As aulas são ministradas por um conjunto de professores que trabalham na escola e as vezes em outras também. Cada professor tem seu próprio número de aulas.

Existem várias restrições que devem ser consideradas na tentativa de encontrar uma solução para o problema. Elas variam de acordo com necessidades específicas das escolas. No entanto, serão mencionadas somente as implementadas neste trabalho.

Cada restrição tem uma determinada importância, que deve ser quantificada pela instituição de ensino. Para que o algoritmo seja mais dinâmico é necessário permitir a alteração dos pesos de cada critério.

### 3.4.1 Colisão por Professor

Essa restrição avalia se um professor lecionará mais de uma aula simultaneamente. As tabelas 3.4 e 3.5 ilustram o fato. Nesse caso, ocorreram duas colisões, pois tanto a professora Maione como a professora Rosa estão ministrando aulas no mesmo horário para turmas diferentes.

**Tabela 3.4:** Horário gerado para a turma 5SHP

5SHP	Segunda	Terça	Quarta	Quinta	Sexta
7:00	Inglês/Maione	—	—	—	—
7:50	—	Redação/Rosa	—	—	—
8:55	—	—	—	—	—
9:45	—	—	—	—	—
10:45	—	—	—	—	—
11:25	—	—	—	—	—

**Tabela 3.5:** Horário gerado para a turma 6SHP

6SHP	Segunda	Terça	Quarta	Quinta	Sexta
7:00	Francês/Maione	—	—	—	—
7:50	—	Português/Rosa	—	—	—
8:55	—	—	—	—	—
9:45	—	—	—	—	—
10:45	—	—	—	—	—
11:25	—	—	—	—	—

### 3.4.2 Horários Esparsos

Essa restrição verifica se a grade horária formada possui alguma "janela", ou seja, se existem horários vagos em um determinado dia. A tabela 3.6 mostra um dia da semana contendo uma janela.

**Tabela 3.6:** Exemplo de janela na grade horária

6SHP	Segunda
7:00	Francês/Maione
7:50	Redação/Rosa
8:55	Redação/Rosa
9:45	—
10:45	Inglês/Maione
11:25	História/Carlos

### 3.4.3 Blocos de Disciplinas

Esse critério avalia uma condição didática baseada na forma pela qual as disciplinas estão dispostas. Didaticamente, convém que as disciplinas estejam dispostas em blocos de duas. Se a carga horária de uma disciplina é cinco aulas semanais, o ideal é que elas estejam dispostas em dois blocos de duas aulas e uma aula isolada.

**Tabela 3.7:** Formação de blocos de disciplinas

6SHP	Segunda	Terça	Quarta	Quinta	Sexta
7:00	Mat./Salette	Por/Rosa	Cie/Salette	Mat/Salette	Por/Rosa
7:50	Mat/Salette	Por/Rosa	Cie/Salette	Mat/Salette	Ing/Maione
8:55	Por/Rosa	Geo/James	Geo/James	His/Fátima	Ing/Maione
9:45	Por/Rosa	Geo/James	Por/Rosa	His/Fátima	Ed.Fís/Sandra
10:45	Cie/Salette	His/Fátima	—	—	Ed. Fís/Sandra
11:25	—	—	—	—	—

Na tabela 3.7, a disciplina de Português (Port.) não está disposta da forma didaticamente correta, pois existem duas disciplinas isoladas que poderiam formar um bloco. No entanto, a disciplina de Matemática (Mat.) está disposta de acordo com este critério.

### 3.4.4 Vários Blocos por Dia

Um problema pode ocorrer durante a formação dos blocos. Trata-se do fato de dois blocos de uma mesma disciplina ficarem alocados em um mesmo dia, constituindo assim uma atitude didaticamente incorreta. É preciso evitar, por exemplo, quatro aulas de matemáticas em um determinado dia, pois os alunos e professores ficariam insatisfeitos com o horário gerado. A tabela 3.8 mostra um exemplo de um dia

de uma determinada grade horária contendo mais de um bloco de uma mesma disciplina.

**Tabela 3.8:** Exemplo de vários blocos de uma disciplina no mesmo dia

6SHP	Segunda
7:00	Mat/Salete
7:50	Mat/Salete
8:55	Mat/Salete
9:45	Mat/Salete
10:45	Port/Rosa
11:25	-

### 3.4.5 Média de aulas por dia

Esse critério avalia se o número de horários por dia está de acordo com a média de aulas. Essa média é calculada baseada na carga horária semanal de uma determinada turma. Se uma turma tem vinte e cinco aulas semanais, o ideal é que ela tenha cinco aulas por dia. A avaliação desse critério evita que uma turma tenha muitos horários em alguns dias e poucos horários em outros, fato este exemplificado na tabela 3.9.

**Tabela 3.9:** Verificação da média de aulas

6SHP	Segunda	Terça	Quarta	Quinta	Sexta
7:00	Mat./Salete	Por/Rosa	Cie/Salete	Mat/Salete	Por/Rosa
7:50	Mat/Salete	Por/Rosa	Geo/James	Mat/Salete	Ing/Maione
8:55	Por/Rosa	Geo/James	Geo/James	His/Fátima	Ing/Maione
9:45	Por/Rosa	Geo/James	-	-	Ed.Fís/Sandra
10:45	Cie/Salete	His/Fátima	-	-	Ed. Fís/Sandra
11:25	Cie/Salete	His/Fátima	-	-	-

### 3.4.6 Preferências dos professores

Os professores geralmente trabalham em mais de uma escola e têm suas preferências de horário. Por isso, além de avaliar todos os critérios anteriores é preciso verificar se o professor está lecionando num horário de sua preferência. O ideal é que se tenha uma tabela de preferência para cada professor, indicando os horários

que o professor prefere, os horários que o professor pode lecionar mas não quer e por fim os horários que o professor não pode.

### **3.5 Considerações Finais**

Os critérios citados neste capítulo são os mais comuns entre as instituições de ensino fundamental e médio. Como cada uma dessas instituições possui um regimento interno, pode haver critérios particulares.

Uma vantagem da utilização de um algoritmo genético é que esses critérios serão definidos na função de avaliação. Desta forma, se houver qualquer modificação em relação aos critérios, somente a função de avaliação será modificada e não o algoritmo inteiro.

## Capítulo 4

# Resolução do Timetabling Utilizando Algoritmos Genéticos

### 4.1 Introdução

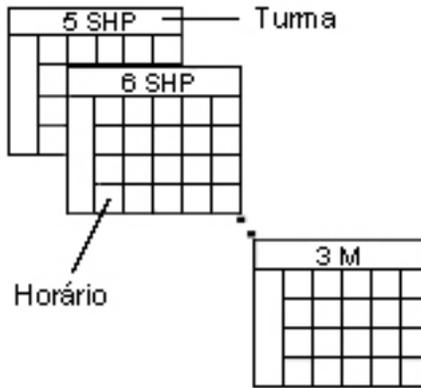
O objetivo deste capítulo é abordar a produção de um algoritmo genético para tratar especificamente a variante citada como *School Timetabling* (contendo algumas características particulares).

A estrutura pelo qual o algoritmo será abordado segue os mesmos passos descritos no Capítulo 2, iniciando com a representação da solução e logo em seguida com o processo iterativo do AG.

### 4.2 Representação da Solução

Para representar a solução no problema da geração de grades horárias, foi escolhida uma forma intuitiva, onde cada gene representa um *slot* (ex: segunda-feira às 7:00). Um cromossomo seria então uma matriz de 3 dimensões (Número de turmas X [Número de Dias X Número de Horários]). Essa representação apresenta a vantagem de não permitir, graças a sua codificação, que duas disciplinas referentes a mesma turma ocupem o mesmo horário. Porém, não garante por si só a não ocorrência de colisões de professores e outras restrições.

Cada indivíduo é formado por uma matriz de três dimensões conforme mostrado na figura 4.1. É importante observar que dependendo do número de turmas e *slots* disponíveis, a representação de uma solução pode ser muito grande, requisitando assim maiores recursos computacionais.



**Figura 4.1:** Representação da solução para o Timetabling

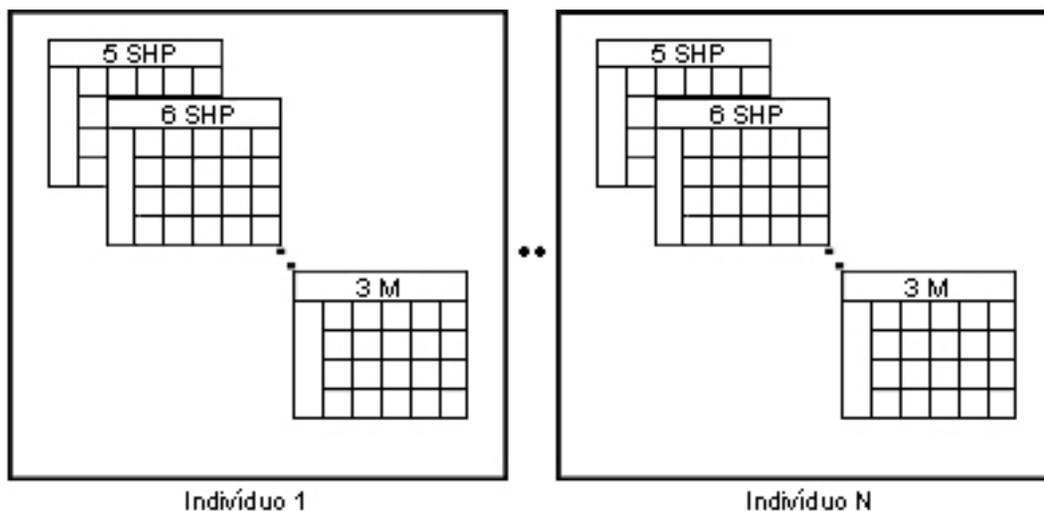
Na realidade, um AG trabalha com uma população de indivíduos, e por isso a representação desta população seria, um vetor de matrizes, conforme ilustrado na figura 4.2. Como uma população ideal é de 50 a 100 indivíduos segundo [Gol89], pode-se afirmar que esta representação de soluções para o Timetabling é realmente grande. Entretanto, existem outras maneiras de representar a solução para o Timetabling:

[Abr92]: cada dia é descrito como uma lista encadeada (representando o cromossomo) e o genótipo é o conjunto das listas de todos os dias. De acordo com [Abr92] este tipo de representação facilita uma implementação em paralelo;

[Bur94]: cita a representação por matriz, porém cada turma é representada por um cromossomo. O conjunto da representação de todas as turmas formam o genótipo.

### 4.3 Criação da População Inicial

Após a representação é necessário gerar uma população inicial. Neste trabalho foram desenvolvidos dois operadores com este propósito:

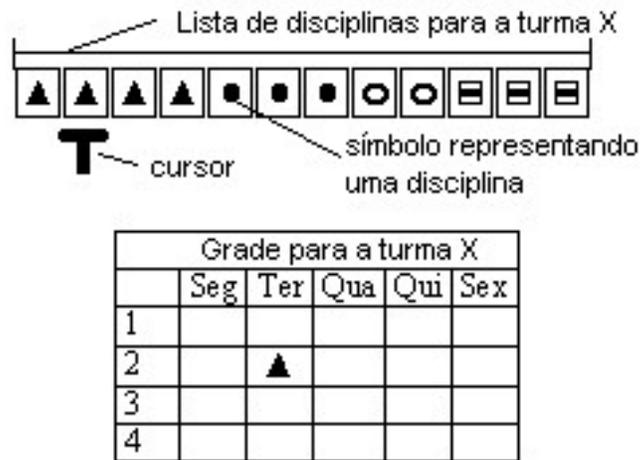


**Figura 4.2:** Representação da população de soluções para o Timetabling

a) Geração aleatória: este operador percorre uma lista disciplinas de determinada turma e para cada uma das disciplinas, sorteia um slot disponível na grade horária. No caso ilustrado na figura 4.3, a primeira disciplina da lista de uma determinada turma, já está inserida no *slot* sorteado para ela. O processo continua até que a listas de disciplinas de todas as turmas sejam percorridas.

b) Geração Heurística - funciona de maneira semelhante ao operador anterior. A única diferença é que ao invés de sortear uma disciplina isolada, serão sorteados blocos de duas disciplinas. A lista de disciplinas de uma determinada turma é dividida em duas outras listas: *LBlocos* e *LIso-ladas*. A lista *LBlocos* contém os blocos disponíveis para sorteio e a lista *LIso-ladas* contém as disciplinas que não puderam formar blocos e desta forma precisam ser sorteadas isoladamente.

Primeiramente serão sorteados os blocos de disciplinas. Na figura 4.4, o primeiro bloco de disciplinas já foi sorteado (repare que o cursor aponta para o segundo bloco). Ao terminar a lista *LBlocos* o algoritmo percorre a lista *LIso-ladas* sorteadando também um *slot* para cada disciplina dessa lista.



**Figura 4.3:** Exemplo da geração aleatória de um indivíduo

A população gerada pelo método heurístico geralmente é mais evoluída que a gerada pelo método aleatório, pois preserva blocos ideais de disciplinas.

#### 4.4 Avaliação dos Indivíduos

Os indivíduos são avaliados baseado nos critérios de avaliação que foram descritos no Capítulo 3. São eles:

- Colisão por professor;
- Horários esparsos;
- Blocos de disciplinas;
- Vários blocos por dia;
- Média de aulas por dia;
- Preferências dos professores.

Uma função de avaliação faz o uso do cálculo de penalidades associadas a um indivíduo. Cada vez que a função de avaliação encontra uma colisão por professor,

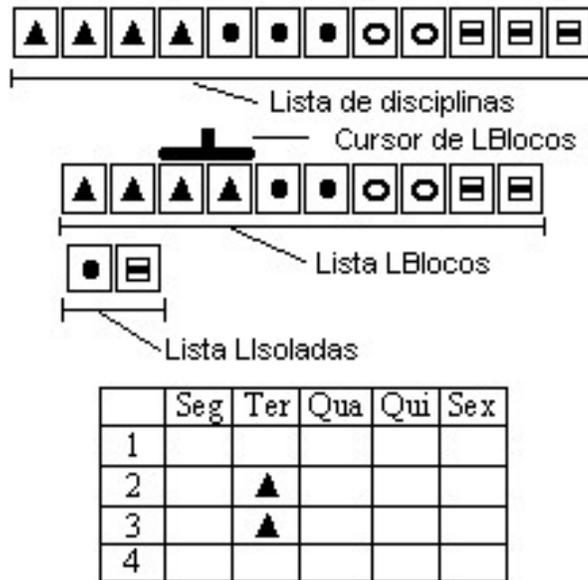


Figura 4.4: Exemplo da geração heurística de um indivíduo

um horário esparsa ou qualquer outra restrição, uma penalidade é indicada e o valor da mesma varia de acordo com a sua importância. A função de avaliação retorna um valor numérico de acordo com a fórmula a seguir:

$$f(x) = \frac{1}{1 + \text{penalidades}}$$

Através dessa fórmula é possível perceber que o valor de  $f(x)$  varia de 0 a 1. O indivíduo ótimo é aquele que não possui nenhuma penalidade. Desta forma, o valor de  $f(x)$  se torna igual a 1.

## 4.5 Seleção dos Indivíduos

Os métodos de seleção implementados foram o da roleta giratória, método de seleção por torneio e por fim o método da roleta com redução (proposto neste trabalho). Antes da seleção, os indivíduos são avaliados separadamente e o resultado é a quantificação do seu grau de aptidão. Basicamente, um método de seleção é uma

função que recebe como parâmetro o grau de aptidão de cada indivíduo e retorna apenas os indivíduos selecionados.

O casamento ocorre entre os indivíduos selecionados pelo método de seleção. Na maioria dos casos, o casamento é feito de forma aleatória, mas é bom ressaltar que existem outros métodos para realizar esse casamento, como, por exemplo, o cruzamento dos melhores indivíduos [Gol89].

## 4.6 Crossover

O crossover conforme descrito é o cruzamento genético entre os indivíduos selecionados. Em geral, os operadores de crossover buscam trocar determinados genes e a partir daí gerar novos indivíduos. Os operadores de crossover implementados foram:

1. Crossover com corte em um ponto;
2. Crossover com corte em dois pontos;
3. Crossover heurístico com corte em um ponto.

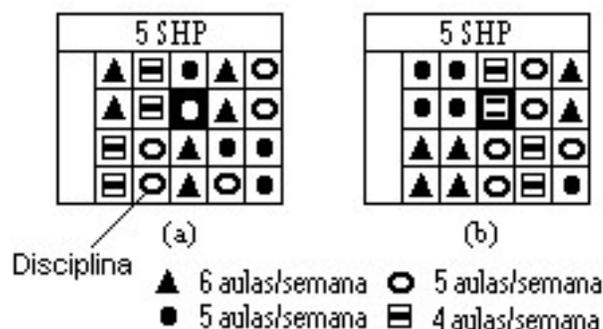
O operadores de crossover com corte em um ponto e o de crossover com corte em dois pontos foram implementados por serem os mais tradicionais entre os operadores. O operador de crossover heurístico foi uma tentativa de realizar a troca genética com alguma heurística envolvida. No caso, a heurística utilizada foi a preservação de blocos de duas disciplinas.

### 4.6.1 Crossover com corte em um ponto (1PX)

Para este operador um ponto é escolhido e a partir desse serão realizadas as trocas de genes entre os indivíduos. No caso do problema da geração de horários escolares, existem alguns cuidados que devem ser tomados para que os novos indivíduos gerados não sejam incoerentes.

Para simplificar, suponha que exista apenas uma turma, no caso a 5 SHP, conforme figura 4.5. Nesta figura existem dois indivíduos, representando duas soluções para o problema. Abaixo das grades horárias existe a carga semanal de cada disciplina. O *slot* destacado indica o ponto de corte.

O problema do horário incoerente ocorre se a troca genética for feita gene a gene sem se preocupar em nenhum momento com a incoerência da solução. A figura 4.6 mostra o resultado de uma troca como esta. Note que nos descendentes



**Figura 4.5:** Exemplos de indivíduos representando uma grade horária

indicados como (c) e (d), as cargas horárias das disciplinas estão incoerentes de acordo com a legenda.

Para resolver este problema, basta que haja troca apenas entre genes compatíveis, ou seja, somente genes que possuam como valor a mesma disciplina. A abordagem da operação de crossover será dividida em quatro partes. Todo o processo pode ser acompanhado na figura 4.7.

### Criação da relação entre os genes dos indivíduos

O primeiro passo da realização do crossover é a criação de duas listas que contêm os genes dos indivíduos a partir do ponto de corte. A figura 4.7 apresenta dois indivíduos (a) e (b) e duas listas representadas por (c) e (d). A lista (c) representa os genes do indivíduo (a) que possivelmente irão realizar crossover e a lista (d) representa os genes do indivíduo (b). Cada nó dessas listas são compostos pelos campos:

- **Disc:** representando a disciplina;
- **Pgene:** posição do gene no cromossomo;
- **Pcross:** indica a posição de crossover;
- **Disponível:** indica se o gene está disponível para crossover.

Inicialmente as listas são preenchidas com a disciplina e a posição absoluta dessa disciplina no cromossomo (slot referente na grade horária), setando assim o

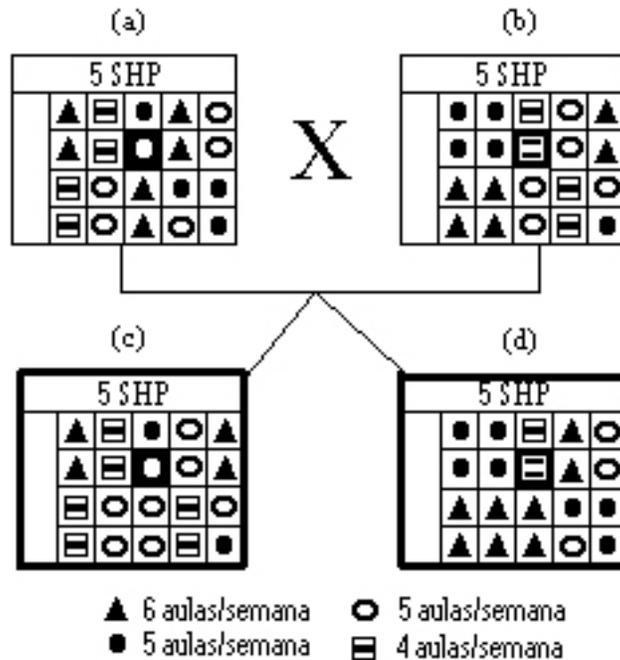


Figura 4.6: Realização do crossover

campo *Disc* e *Pgene* respectivamente. O campo relativo á posição de crossover é iniciado como -1 e o campo *Disponivel* configurado para VERDADEIRO. Observe na figura 4.7 que existe uma relação entre as listas (c) e (d). A relação entre as listas é obtida através do campo *Pcross*, como será abordado no pseudocódigo a seguir. Neste pseudocódigo as listas (c) e (d) serão referenciadas como ListaC e ListaD respectivamente.

```

Para cada gene i da ListaC faça
  posgene:= ProcuraGene(ListaC[i].Disc,ListaD)
  Se posgene <> -1 então
    ListaC[i].Pcross:= ListaD[posgene].Pgene
    ListaD[posgene].Pcross:= ListaC[i].Pgene
    ListaC[i].Disponivel:= FALSO
    ListaD[posgene].Disponivel:= FALSO
  Fim do Se
Fim do Para

```

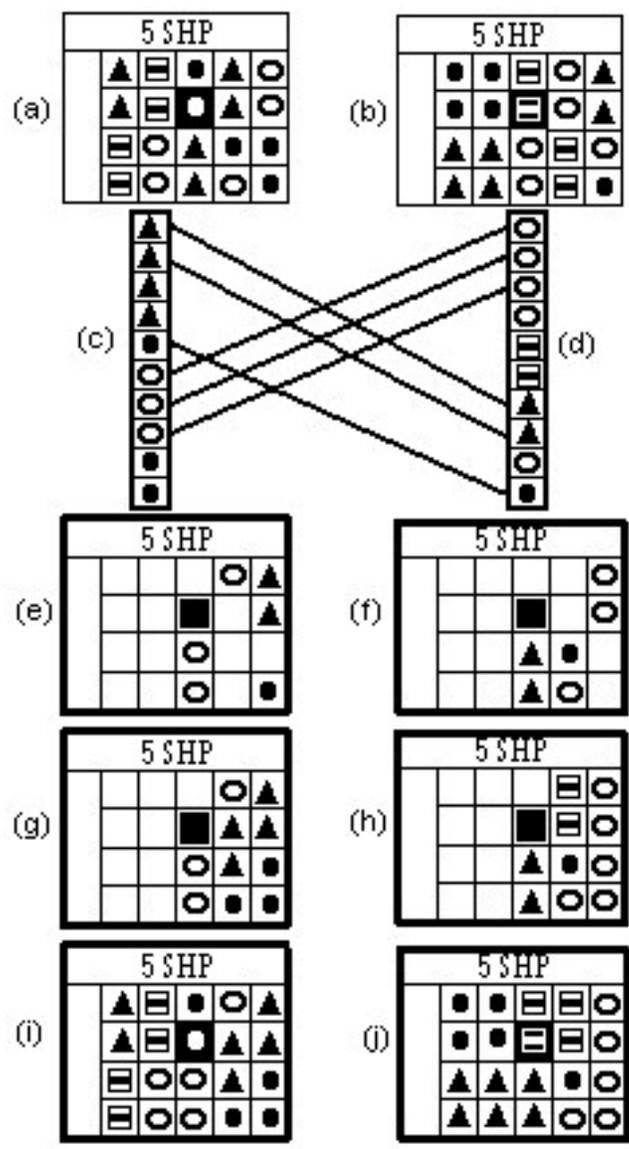


Figura 4.7: Realização do crossover

A função *ProcuraGene* descrita no pseudocódigo, recebe como parâmetro uma disciplina e uma lista, retornando a posição do gene que tem como valor essa

disciplina. Um detalhe é que essa função retorna -1 quando não encontra um gene compatível na lista. Também é importante ressaltar que a função só retorna o gene compatível se ele estiver disponível (*Disponível* = VERDADEIRO), pois se não preocupasse com isso, dois ou mais genes poderiam ser relacionados a uma mesma posição de crossover.

### **Troca dos genes compatíveis nos indivíduos**

Neste caso, é necessário percorrer a lista de genes e para cada gene com posição de crossover diferente de -1 realizar a troca. Na figura 4.7, os indivíduos indicados por (e) e (f) são os resultantes da troca de genes compatíveis. Repare como houve uma troca na posição das disciplinas em relação aos seus ancestrais. Considere por exemplo a disciplina representada por um triângulo. Note que ela está disposta no indivíduo (e) da mesma maneira que ela estava disposta no indivíduo (b) e está disposta no indivíduo (f) da mesma maneira que se encontrava em (a).

A troca é possível justamente pelo campo *Pcross* das listas. Esse campo guarda a posição de um gene compatível no cromossomo do outro indivíduo participante do crossover, fazendo com que haja troca da posição onde as disciplinas estão alocadas.

### **Inserção dos genes não compatíveis**

Os genes não compatíveis são aqueles sem relação nas listas de genes. A posição de crossover desses é configurada para -1. Como não há possibilidade de troca, eles são alocados de forma aleatória nos slots vagos restantes (a partir do ponto de corte) da grade horária. Na figura 4.7 os indivíduos indicados como (g) e (h) mostram uma maneira pela qual os genes não compatíveis foram alocados na grade horária. É importante ressaltar existem várias formas de alocar os genes não compatíveis (pois para cada gene desses um slot é sorteado) e que a forma mostrada é apenas um exemplo.

### **Inserção dos genes dos ancestrais**

A partir do ponto de corte os genes já foram trocados, restando portanto os genes anteriores a esse ponto. Como estes genes serão os mesmos dos ancestrais, basta copiá-los. O indivíduo indicado na figura 4.7(i) recebeu os genes que antecedem o ponto de corte relativo ao indivíduo (a) da mesma figura e o indivíduo indicado como (j) recebeu os genes que antecedem o ponto de corte relativos ao indivíduo (b).

Após essa operação dois novos descendentes foram criados, sendo representados na figura 4.7 como (i) e (j). Note que os novos indivíduos além de diferentes dos seus ancestrais, são coerentes em relação as cargas horárias das disciplinas.

#### 4.6.2 Crossover com corte em dois pontos (2PX)

O operador difere do apresentado anteriormente somente na escolha dos pontos de corte. Nesse caso, dois pontos serão escolhidos e os genes que se localizam entre esses serão trocados.

Assim como o operador anterior, é necessário a preocupação com a geração de grades horárias incoerentes. Os esquemas de soluções são os mesmos adotados para o operador de crossover com corte em um ponto. A única diferença é o corte em dois pontos ao invés de apenas um, fato este ilustrado na figura 4.8.

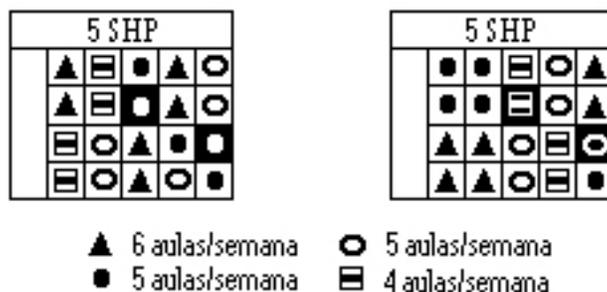


Figura 4.8: Indivíduos e seus respectivos pontos de corte

Os dois pontos de corte podem ser observados pelos *slots* destacados. A idéia é realizar a troca genética entre esses pontos. Somente os genes compatíveis são trocados, conforme figura 4.9. Note que nesse esquema, o número de genes que serão compartilhados diminuiu em relação ao operador de crossover com um ponto de corte.

O processo de troca genética a partir desse ponto é idêntico ao operador de crossover anteriormente apresentado. Entretanto, esse operador geralmente obtém um resultado inferior ao operador de crossover com um único ponto de corte [Fil00].

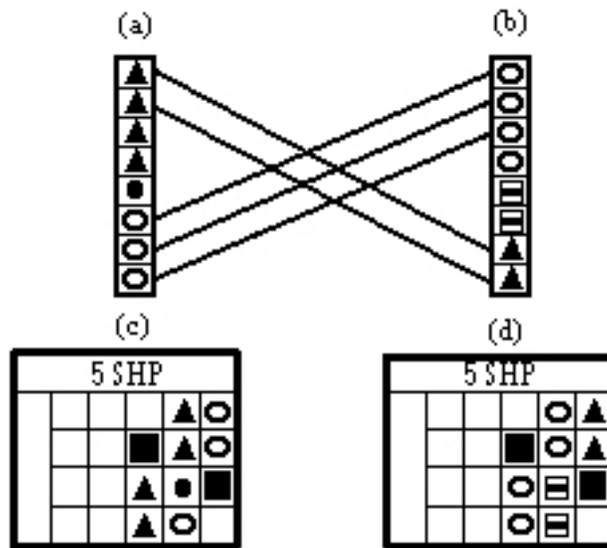


Figura 4.9: Crossover com dois pontos de corte

### 4.6.3 Crossover heurístico com corte em um ponto

Como o próprio nome diz existe alguma heurística envolvida com este operador e os genes serão trocados a partir de um ponto de corte. A heurística se refere ao agrupamento de disciplinas em blocos de duas.

Didaticamente, o rendimento do aluno e do professor é maior quando a disciplina lecionada está agrupada de duas em duas. É preferível ter um dia com duas aulas de português seguido de duas aulas de matemática do que um dia com as aulas na seguinte ordem: português, matemática, português, matemática.

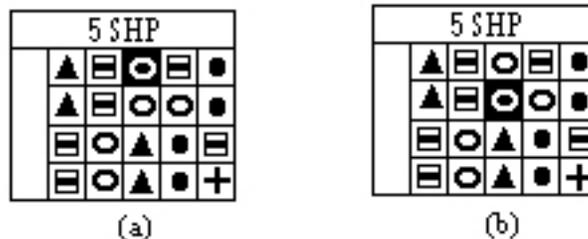
Esse operador difere dos outros apresentados no fato de preservar os blocos de disciplinas já formados, sendo esta a única diferença desse para o operador de crossover com corte em um ponto. Serão descritos apenas os passos que diferem este do operador de crossover com corte em um ponto convencional:

1. Ao selecionar o gene que será o ponto de corte, verifique se este corte não desfaz um bloco de disciplinas. Se isso acontecer selecione o próximo gene.
2. Após formada a lista com os genes a partir do ponto de corte, verifique qual desses estão agrupados em blocos de duas disciplinas e crie uma nova lista

*LBlocos*. Para as disciplinas que não estão agrupadas em blocos crie uma lista *LIsoladas*.

3. Verifique quais os blocos na lista *LBlocos* podem ser trocados.
4. Verifique quais genes podem ser trocados na lista *LIsoladas*.
5. Troque os blocos compatíveis, copiando o resultado para os indivíduos descendentes. Para os blocos que não puderam ser trocados, procure um local (dois *slots* vagos consecutivos) disponível na grade horária. Se não houver, os genes do bloco são adicionados no fim da lista *LIsoladas*.
6. Se novos genes foram inseridos na lista *LIsoladas*, é preciso verificar se esses genes podem ser trocados.
7. Troque os genes compatíveis inseridos em *LIsoladas*, copiando o resultado para os indivíduos descendentes.
8. Dos genes que não puderam ser trocados (se houver algum), sorteie aleatoriamente um *slot* disponível para ele na grade horária da turma.

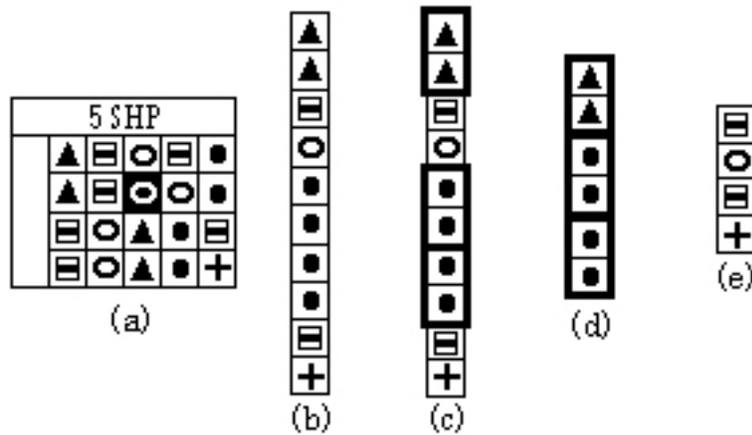
O primeiro passo a ser detalhado é o que verifica se o ponto de corte interrompe um bloco de disciplina já formado. A figura 4.10 detalha esse fato.



**Figura 4.10:** Ponto de corte no crossover heurístico

O indivíduo (a) da figura 4.10, mostra um suposto ponto de corte para a realização da troca genética. Note que se esse gene for considerado o ponto de corte, possivelmente um bloco de disciplinas será quebrado. Para evitar essa quebra, basta selecionar o próximo gene. Essa operação é ilustrada no indivíduo (b) da mesma figura.

Após escolhido o ponto de corte correto, o próximo passo que difere da operação de crossover convencional é o fato da criação de duas listas : *LBlocos*, *LIsoladas*.



**Figura 4.11:** Estruturas de dados para o crossover heurístico

A figura 4.11 ilustra todo o processo que o algoritmo realiza antes da criação das duas listas. A figura 4.11(a) mostra onde será o ponto de corte a ser considerado. Como o corte se dá em apenas um ponto, todos os genes a partir desse ponto serão armazenados numa lista temporária mostrada pela figura 4.11(b). O próximo passo é verificar quais as disciplinas que estão agrupadas em blocos de duas. A figura 4.11(c) representa os blocos de disciplinas que estão formados para esta grade horária. A figura 4.11(d) representa justamente a lista *LBlocos* e a figura 4.11(e) representa a lista *LIsoladas*, ou seja, lista das disciplinas que não estão agrupadas em bloco.

Um importante fato a ser observado é que na figura 4.11 estão mostradas apenas as operações de um indivíduo. É importante lembrar que o crossover se realiza entre dois indivíduos. Sendo assim, essa operação tem que ser realizada em todos os indivíduos participantes do crossover.

Com as listas de blocos (*LBlocos*) preenchida, é necessário verificar quais blocos podem ser trocados entre os indivíduos participantes do crossover. Note que somente blocos serão trocados e não um gene isolado. Para os blocos não compatíveis, ou seja, aqueles que não puderam ser trocados, resta sortear um local para encaixá-lo dentro da grade horária. Se não houver esse local, ou seja, dois

slots vagos consecutivos, os dois genes do bloco serão inseridos no final da lista de disciplinas isoladas e a partir daí não serão mais considerados como um bloco.

Após a inserção na lista *LIsoladas*, é necessário verificar se existe algum gene compatível na lista de disciplinas isoladas do outro indivíduo. Se houver, a troca poderá ser realizada. Caso contrário, um slot vago será sorteado para esse gene.

O desempenho desse operador depende da importância do critério de agrupamento de blocos de disciplinas, ou seja, se é realmente importante agrupar disciplinas em blocos de duas. Quanto menor a importância do critério pior o desempenho do operador. Em escolas onde o critério de blocos de disciplinas não é considerado, o desempenho desse operador pode ser péssimo.

## 4.7 Mutação

Para realizar uma mutação num algoritmo genético onde a representação dos indivíduos é binária, basta realizar a inversão de um bit. Isso gera uma mudança aleatória nesse indivíduo e é justamente essa que permite ao algoritmo uma maior abrangência de soluções.

No caso do problema da geração de grades horárias não há como inverter determinado gene, nem atribuir a ele um valor aleatório, pois soluções inconsistentes poderiam ser geradas.

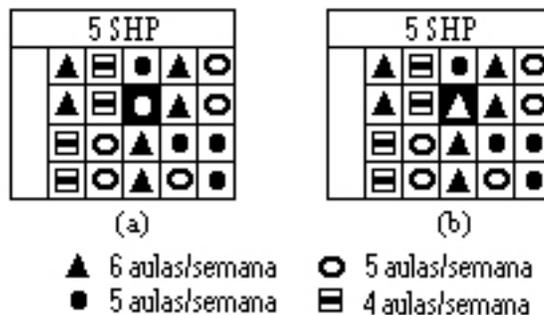
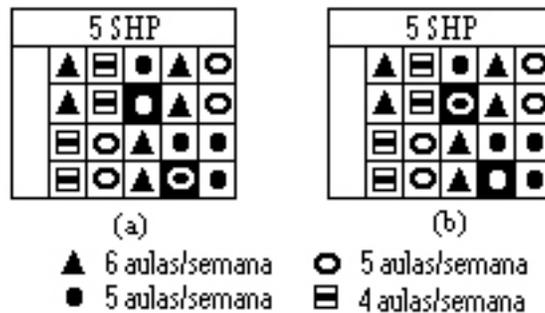


Figura 4.12: Mudança aleatória no valor do gene

Uma possível solução seria atribuir a um determinado gene, um valor pertencente ao conjunto de disciplinas válidas para determinada turma. No caso da figura 4.12, um gene foi sorteado (em destaque) e o indivíduo original está representado

como (a). Para realizar a mutação atribui-se ao gene sorteado um outro valor (valor válido no conjunto de disciplinas para a turma), resultando o indivíduo (b). O problema é a inconsistência da solução, bastando observar o número de aulas semanais do indivíduo (b).

A solução proposta neste trabalho foi inverter a posição de determinados genes ao invés de alterar seu valor, não perdendo assim a consistência. Para realizar esse processo basta selecionar dois genes aleatoriamente e invertê-los de posição.



**Figura 4.13:** Inversão da posição dos genes

Na figura 4.13(a) está ilustrado um indivíduo original. Os genes que foram sorteados para realizar a mutação estão em destaque. A figura 4.13(b) mostra um novo indivíduo resultante da inversão da posição dos genes.

#### 4.7.1 Mutação heurística

Quando o critério de alocação de disciplinas em bloco é importante, pode ser interessante um operador de mutação que leve esse fator em consideração. Baseado nisso, uma proposta para um operador de mutação foi elaborada.

A idéia básica é "enxergar" o conjunto de disciplinas pertencentes a uma determinada turma como um conjunto de blocos formados. Ao invés de sortear dois genes isolados, sorteiam-se dois blocos e em seguida é feita a inversão de posição. Caso não haja blocos disponíveis, esse operador utiliza a mesma idéia do operador de mutação descrito no tópico anterior.

Uma desvantagem notável é que dependendo do nível de importância, é preferível quebrar um bloco do que entrar em conflito com uma preferência de um professor ou outro critério qualquer.

## 4.8 Verificação do critério de sobrevivência

Este passo do algoritmo genético tem como objetivo verificar os critérios de substituição da população ancestral pela população descendente, ou seja, avaliar quando a substituição é interessante para o desempenho do algoritmo. Neste trabalho foram abordadas duas formas para substituição da população:

- a) Sempre substituir os ancestrais pelos descendentes;
- b) Substituir os ancestrais pelos descendentes somente se a média dos descendentes for maior ou igual a média dos ancestrais.

A substituição dos indivíduos de acordo com a média pode fazer com que o algoritmo perca alguns indivíduos ruins. Apesar de estranho, indivíduos ruins contribuem para a diversidade de soluções. Um algoritmo genético tende a obter um bom desempenho quando sua população é formada por indivíduos ruins, médios e também por indivíduos bons.

## 4.9 Critério de Parada

O critério para finalizar o algoritmo genético adotado neste trabalho é o número de gerações. O critério de tempo também poderia ser utilizado, mas nesse caso o número de gerações processadas iria ser determinado pela velocidade do hardware utilizado.

## 4.10 Considerações Finais

A implementação dos métodos de seleção, dos operadores de crossover e mutação, da geração da população inicial e outros passos de um algoritmo genético, não garante por si só, a geração de boas soluções. Após a etapa de implementação, torna-se necessário "calibrar" o algoritmo, configurando os parâmetros genéticos de acordo com as necessidades do problema. O Capítulo 5 introduz a aplicação desenvolvida neste trabalho. Já o Capítulo 6 trata dos resultados obtidos utilizando essa aplicação e demonstra algumas maneiras para realizar a "calibragem" do algoritmo.



## Capítulo 5

# A Aplicação Desenvolvida

### 5.1 Introdução

Após elaborar todos os passos necessários para desenvolver um algoritmo genético específico para a resolução do problema das grades escolares, tornou-se necessário criar uma aplicação para que os testes pudessem ser executados.

Este capítulo visa abordar a aplicação desenvolvida, explorando os dados de entrada e saída, a configuração dos parâmetros genéticos do algoritmo e geração de gráficos que demonstram a evolução das gerações.

### 5.2 Dados de Entrada

Para que a aplicação funcione de maneira correta, o usuário precisa fornecer os seguintes dados de entrada :

- o número de horários disponíveis;
- as turmas;
- as disciplinas;
- os professores e o número de aulas semanais;
- as preferências de horário para cada professor;

A figura 5.1 mostra o formulário para cadastro de turmas, professores e disciplinas presente na aplicação. Já a figura 5.2, mostra o formulário onde as preferências dos professores serão configuradas.

The image shows a software window titled "Cadastro de Professores". It is divided into three main sections. The first section, "Dados", has a dropdown menu for "Nome" currently showing "Professor X". The second section, "Dados Adicionais", contains three fields: "Disciplina" (dropdown showing "Matemática"), "Turma" (dropdown showing "1 M"), and "Num. Aulas" (text input showing "1" with up and down arrow buttons). The third section, "Disciplinas e Turmas do Professor", is a table with three columns: "Disciplina", "Turma", and "Num. Aulas". It contains one row with the values "Matemática", "1 M", and "1". At the bottom of the window are three buttons: "Remover" (with a trash icon), "Salvar" (with a green checkmark icon), and "Fechar" (with a window icon).

**Figura 5.1:** Formulário para gerenciamento de turmas, professores e disciplinas

### 5.3 Configuração dos parâmetros genéticos

É desejável que o usuário " calibre" o algoritmo de acordo com alguns parâmetros genéticos, tais como: taxa de crossover, taxa de mutação, método de seleção de indivíduos e uma série de outros. Pensando nisso, a aplicação dispõe de um formulário para que o usuário configure estes parâmetros, conforme a figura 5.3.

O frame *Avaliação dos Indivíduos* presente na figura 5.3 se refere aos critérios de avaliação dos indivíduos. Note que seis critérios estão sendo avaliados na função de avaliação:

1. Blocos de Disciplinas - avalia se as disciplinas encontram-se agrupadas em blocos de duas;
2. Colisão por Professor - avalia se um professor está lecionando mais de uma aula ao mesmo tempo;
3. Horários Esparsos - avalia a existência de janelas na grade horária formada;

Preferências dos Professores

Legenda

- Horários que o professor não pode
- Horários que o professor pode, mas não quer
- Horários preferenciais para o professor

Professor e preferência em questão

Camila

● Preferência Atual

Salvar Fechar

Professor e preferência em questão

<<<<<>>>>>	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:00   07:50	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>
07:50   08:40	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>
08:40   09:50	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>
09:50   10:40	<span style="color: red;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: green;">●</span>	<span style="color: red;">●</span>
10:40   11:30	<span style="color: red;">●</span>	<span style="color: red;">●</span>	<span style="color: yellow;">●</span>	<span style="color: yellow;">●</span>	<span style="color: red;">●</span>
11:30   12:20	<span style="color: yellow;">●</span>	<span style="color: red;">●</span>	<span style="color: yellow;">●</span>	<span style="color: yellow;">●</span>	<span style="color: red;">●</span>

**Figura 5.2:** Formulário que configura as preferências dos professores

- Médias Aulas/Dia - avalia se o número de aulas por dia está de acordo com a média de aulas para a turma;
- Preferências dos Professores - avalia se uma determinada disciplina está alocada de acordo com a preferência do professor;
- Vários Blocos por Dia - avalia se em um dia existem mais de 2 aulas de uma mesma disciplina.

Observe que ao lado da descrição do critério aparece o grau de importância para o mesmo, conforme figura 5.4. Esse grau de importância pode ser definido da seguinte forma:

**A+** - nível de prioridade altíssima;

**Parâmetros**

**Avaliação dos Indivíduos**

- Blocos de disciplinas >> B+
- Colisão por Professor >> A+
- Horários Esparsos >> A
- Média de Aulas/Dia >> M
- Preferências dos Professores >> B+
- Vários Blocos por Dia >> A

Modificar

**Crossover**

- Heurístico 1PX
- Crossover 1PX
- Crossover 2PX

70 chances em 100

**Mutaçao**

- Heurística
- Comum

20 chances em 10000

**Populaçao Inicial**

- Heurística
- Aleatória

Número de Indivíduos: 100

**Métodos para seleçao**

- Torneo
- Roleta Giratória
- Roleta Di Jong- Reduçao: 10 %

**Critérios para substituiçao**

- Nao utilizar este critério
- Média dos Filhos > Pais

**Escalagem**

- Nao utilizar este critério
- Linear Scaling - SuperInd. 20

Salvar Configurações  Fechar

Figura 5.3: Formulário que configura os parâmetros genético do algoritmo

**A** - nível de prioridade alta;

**M+** - nível de prioridade variando entre média e alta;

**M** - nível de prioridade média;

**B+** - nível de prioridade baixa;

**B** - nível de prioridade baixíssima.

Como a importância desses critérios varia de problema para problema, o usuá-

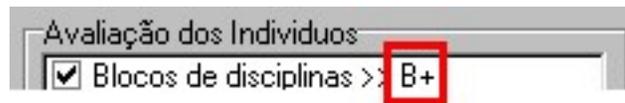


Figura 5.4: Grau de importância dos critérios

rio pode alterar cada importância de acordo com sua necessidade, fato esse ilustrado na figura 5.5.

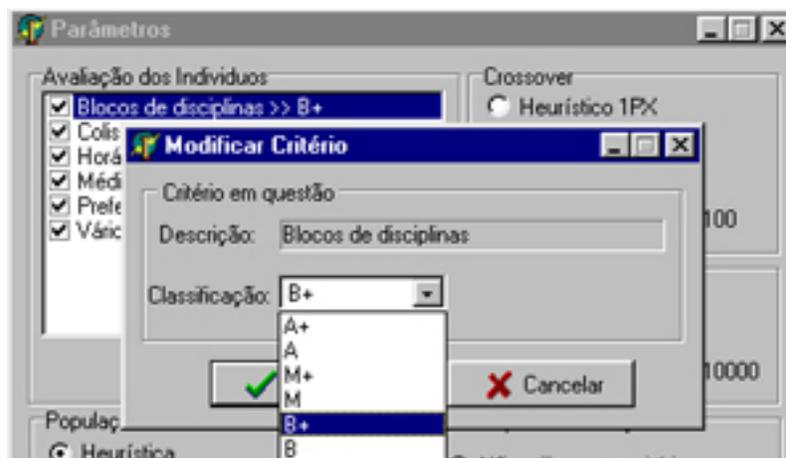


Figura 5.5: Configuração do grau de importância dos critérios

O frame *População Inicial* da figura 5.3 diz respeito à formação da população inicial, que pode ser obtida a partir de uma forma heurística (levando em conta o critério de alocação de disciplinas em blocos), ou de forma aleatória.

Os métodos de seleção considerados importantíssimos para o desempenho do algoritmo, também podem ser configurados através da aplicação. Foram implementados três métodos de seleção, conforme descrito no Capítulo 4. O único detalhe é que se o método de seleção escolhido for o da Roleta com Redução a taxa de redução do textitfitness precisar ser informada.

Para o operador de crossover são 3 opções disponíveis: Crossover 1PX (Crossover com corte em um ponto), Crossover 2PX (Crossover com corte em dois pontos) e por último o Crossover Heurístico (Crossover com corte em um ponto, que leva em consideração os blocos de disciplinas já formados). O usuário também

pode configurar o percentual para a realização de crossover.

Assim como no operador de crossover, o usuário pode escolher o percentual de mutação a ser realizada. Os dois tipos de operadores de mutação disponíveis na aplicação são: Mutação Comum e Mutação Heurística (operador que leva em consideração os blocos de disciplinas já formados).

Os dois últimos parâmetros que podem ser configurados pelo usuário são a escalagem (textitLinear Scaling) e o critério de substituição da população.

## 5.4 Dados de saída

Após cadastrar professores, turmas, disciplinas, horários e todos os requisitos relacionados em tópicos anteriores, o usuário pode resolver o problema da alocação de horários. Quando o processamento do algoritmo se encerrar, o usuário pode realizar as seguintes tarefas:

- salvar a grade horária gerada como resultado;
- imprimir a grande horária;
- verificar as colisões, ou seja, a avaliação daquela solução;
- gerar um gráfico exemplificado na figura 5.6, que demonstra a evolução da população no decorrer das gerações.

O gráfico gerado fornece informações completas que ajudam a entender o comportamento global do algoritmo. Cada informação é apontada na figura 5.6.

- Figura 5.6(a) significa o grau de aptidão de cada indivíduo constituinte da população final. Note que existem vários desses pontos na parte superior do gráfico;
- Figura 5.6(b) significa a evolução da média do grau de aptidão da população ao longo das gerações;
- Figura 5.6(c) indica o indivíduo com melhor grau de aptidão encontrado durante todo o processamento;
- Figura 5.6(d) indica uma noção aproximada da diversidade biológica existente entre os indivíduos constituintes de uma determinada geração;

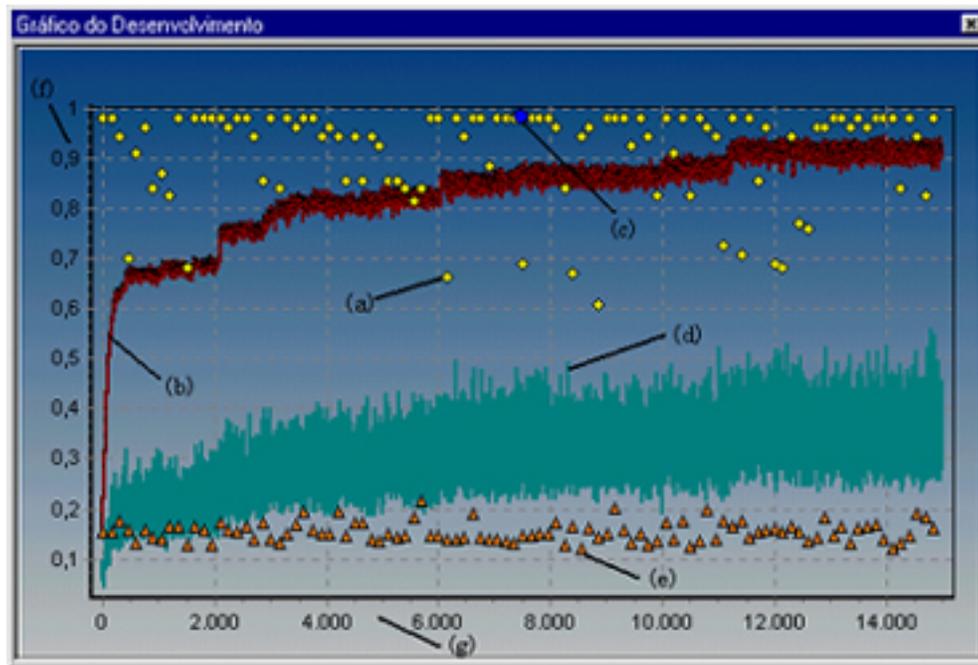


Figura 5.6: Gráfico exemplo

- Figura 5.6(e) indica o grau de aptidão da população inicial gerada. É importante observar que neste caso, ao final do processo, a população final - figura 5.6(a) - está bem mais evoluída que a população inicial - figura 5.6(e);
- Figura 5.6(f) indica o grau de aptidão dos indivíduos que varia de 0 a 1;
- Figura 5.6(g) indica o número de gerações.

E por fim, a figura 5.7 exemplifica um formulário contendo uma grade horária gerada pela aplicação. Note que existe também um outro formulário contendo as penalidades para aquela solução.

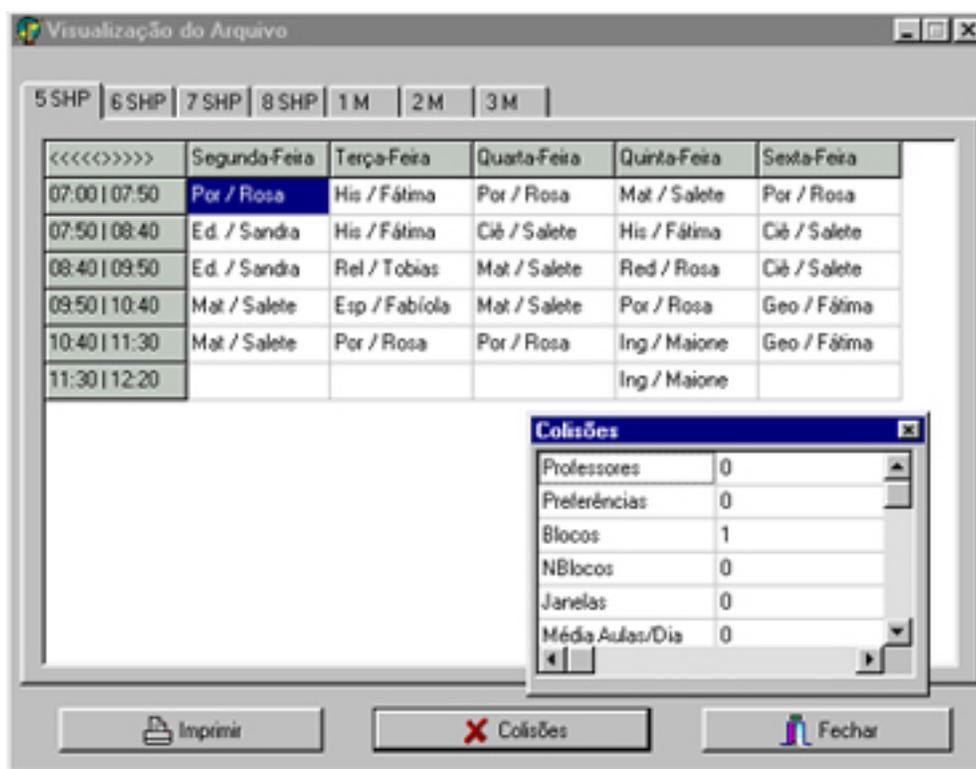


Figura 5.7: Formulário com a melhor solução encontrada pelo algoritmo

## Capítulo 6

# Resultados e Discussões

### 6.1 Introdução

Os testes foram realizados baseando-se em um problema real. Trata-se da geração de horários escolares para uma instituição de ensino fundamental e médio, chamada Colégio Nossa Senhora de Lourdes, que fica localizada no município de Lavras - MG. O interesse principal consistiu em gerar boas grades horárias de acordo com os parâmetros estabelecidos pela instituição.

### 6.2 Dados da Escola

Os dados de entrada adotados nos testes foram:

- cadastro de 7 turmas;
- cadastro de 19 professores, suas preferências de horário, suas disciplinas e também as turmas onde cada professor leciona determinada disciplina;
- foram disponibilizados 30 horários semanais em 5 dias da semana, sendo 6 em cada dia.

A importância dos critérios foi configurada de acordo com a tabela 6.1. Os testes foram divididos de maneira que cada método de seleção fosse testado separadamente e combinados com os operadores de crossover e mutação implementados.

**Tabela 6.1:** Importância dos critérios

Critério	Importância
Colisão por Professor	A+
Blocos de Disciplinas	B+
Média de Aulas	M
Vários Blocos por Dia	A
Horários Esparsos	A
Preferências dos Professores	B+

Um detalhe importante é que a quantificação do grau de aptidão dos indivíduos foi modificada para se obter uma melhor comparação das soluções. A função adotada pelo algoritmo foi descrita no capítulo 4:

$$f(x) = \frac{1}{1 + penalidades}$$

Essa função retorna valores muito pequenos, comprometendo assim a compreensão do usuário em relação a solução encontrada. Para criar uma visão mais realística da solução, os resultados serão exibidos de acordo com a função:

$$f(x) = \frac{1}{100 + penalidades} * 100$$

Suponha que o indivíduo que esteja sendo avaliado tenha uma penalidade com peso (importância) 2. Na função descrita no capítulo 4, o seu grau de aptidão seria 0,33 ( $1/1+2$ ). Na função citada neste capítulo, o grau de aptidão seria 0,98 ( $(1/100+2) * 100$ ). Como os indivíduos avaliados variam de 0 a 1, o valor 0,98 é mais realístico, já que o indivíduo avaliado é quase um indivíduo ótimo.

### 6.3 Seleção por Torneio

Para um primeiro teste foram adotados os operadores de crossover e mutação padrão em um algoritmo genético. Neste caso, os operadores a serem testados são o crossover com corte em um ponto e mutação comum. As probabilidades de crossover e mutação foram adotadas de acordo com [Mit96], sendo 0.7 e 0.001 respectivamente. O tamanho da população foi de 100 indivíduos, pois [Gol89] indica uma população de 50 a 100 indivíduos para a maioria dos problemas.

Para cada configuração dos parâmetros foram realizados três testes, devido aos diferentes resultados que podem ser obtidos. O número de gerações foi 15000,

tempo o suficiente para demonstrar o comportamento do algoritmo nas condições propostas. Os parâmetros para a primeira configuração são citados na tabela 6.2.

**Tabela 6.2:** Parâmetros da configuração 1 para o método de torneio

Configuração 1	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b> Teste 1 - 0.84 Teste 2 - 0.80 Teste 3 - 0.82
	Crossover IPX	0.7	
	Mutação Comum	0.001	
	<b>População Inicial</b>	<b>Tamanho</b>	
	Aleatória	100	

Os resultados dos testes mostram o grau de aptidão do melhor indivíduo encontrado durante o processo das 15000 gerações. Apesar desses resultados serem razoáveis, foi possível notar que a população final estava muito convergida, ou seja, os indivíduos estavam na sua maioria numa faixa restrita, conforme figura 6.1.

Uma maneira de aumentar a diversidade dos indivíduos é aumentar a probabilidade da mutação ocorrer. Baseado nessa proposta, uma nova configuração foi designada, conforme a tabela 6.3.

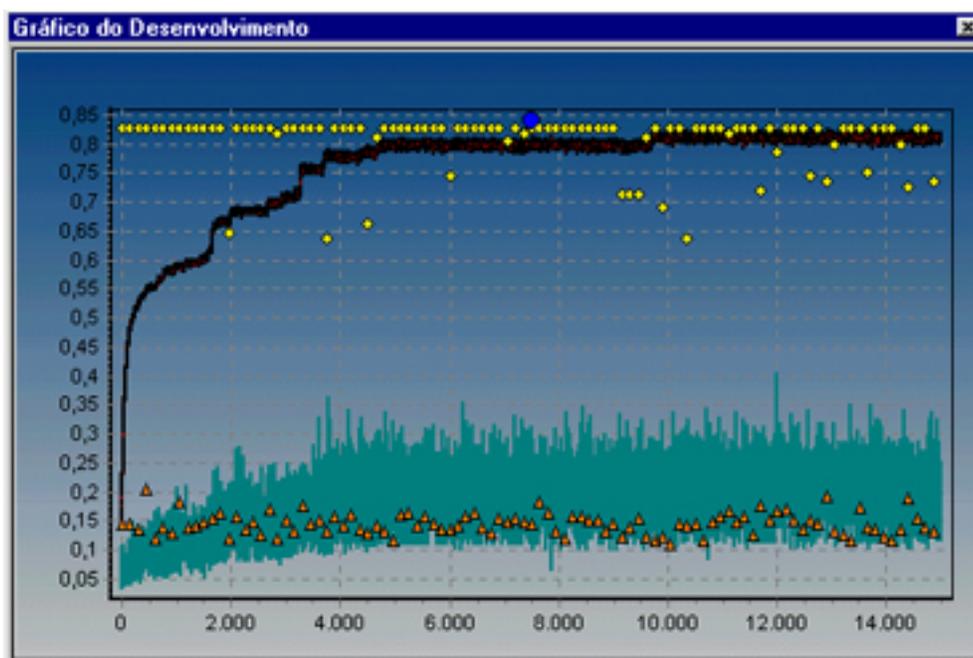
**Tabela 6.3:** Parâmetros da configuração 2 para o método de torneio

Configuração 2	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b> Teste 1 - 0.94 Teste 2 - 0.96 Teste 3 - 0.98
	Crossover IPX	0.7	
	Mutação Comum	0.002	
	<b>População Inicial</b>	<b>Tamanho</b>	
	Aleatória	100	

A configuração 2 apresentou resultados excelentes, com destaque para o indivíduo retornado pelo Teste 3. É importante observar que este indivíduo (com grau de aptidão 0.98) se aproxima muito do indivíduo ótimo (com grau de aptidão 1.00), ou seja, aquele sem nenhuma penalidade.

A mudança no percentual de mutação aumentou consideravelmente os resultados. Talvez um novo aumento desse percentual possa acarretar resultados ainda melhores. Na configuração 3, descrita na tabela 6.4, o percentual de mutação foi aumentado para 0.003.

Os gráficos mostraram que a população se dispersou demasiadamente, o que é



**Figura 6.1:** Comportamento do algoritmo para a configuração 1

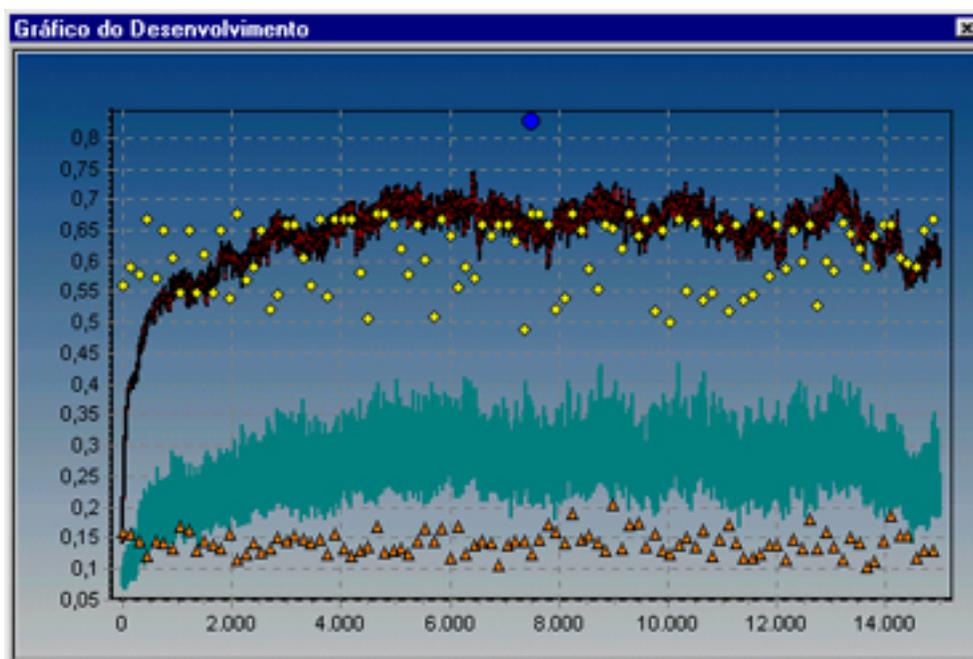
tão ruim ou pior quanto uma população convergida. Um alto percentual de mutação pode fazer a busca se tornar quase aleatória. A figura 6.2 mostra o gráfico de um resultado obtido a partir da configuração 3.

Uma outra configuração a ser considerada é a mudança no operador de crossover. É preciso testar cada operador de crossover para verificar sua eficiência. Apenas um operador foi testado até o momento. Para os próximos operadores, os parâmetros serão configurados de acordo com os melhores resultados obtidos utilizando operador crossover com corte em um ponto. O segundo operador de crossover a ser testado será o Crossover com corte em dois pontos, conforme tabela 6.5. Os resultados foram muito bons, entretanto o operador de Crossover Convencional (Crossover 1PX) apresentou melhores resultados.

O operador de crossover heurístico precisa ser testado com o operador de mutação heurística, pois ambos precisam preservar os blocos de disciplinas formados. Não haveria sentido em realizar um teste com o operador de crossover heurístico e o operador de mutação comum, já que o operador de mutação não iria tomar

**Tabela 6.4:** Parâmetros da configuração 3 para o método de torneio

Configuração 3	Operador	Probalibilidade	Resultados
	Crossover 1PX	0.7	
Mutação Comum	0.003	Teste 1 - 0.82	
População Inicial	Tamanho	Teste 2 - 0.79	
Aleatória	100	Teste 3 - 0.75	



**Figura 6.2:** Comportamento do algoritmo para a configuração 3

nenhum cuidado em quebrar os blocos formados. A tabela 6.6 mostra uma configuração com os operadores heurísticos.

Os resultados apresentados com operadores heurísticos não foram nada favoráveis. Provavelmente, o fracasso desses operadores se deve ao fato da pequena importância dada ao critério de Blocos de Disciplinas (B+). Talvez com uma importância maior para este critério os testes poderão ser melhores.

Em todos os testes realizados anteriormente, o critério de sobrevivência dos in-

**Tabela 6.5:** Parâmetros da configuração 4 para o método de torneio

Configuração 4	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover 2PX	0.7	
	Mutação Comum	0.002	Teste 2 - 0.90
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.92
	Aleatória	100	

**Tabela 6.6:** Parâmetros da configuração 5 para o método de torneio

Configuração 5	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover Heurístico	0.7	
	Mutação Heurística	0.002	Teste 2 - 0.56
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.52
	Heurística	100	

díviduos não foi considerado. Isso significa que os descendentes formaram a nova população independentemente de qualquer que seja seu grau de aptidão. Um critério de sobrevivência seria avaliar a média do grau de aptidão dos descendentes em relação a média do grau de aptidão dos ancestrais. Se fosse menor os descendentes seriam eliminados e os ancestrais formariam a nova população. Caso contrário, os descendentes seriam a nova população. Esta configuração é descrita na tabela 6.7.

**Tabela 6.7:** Parâmetros da configuração 6 para o método de torneio

Configuração 6	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover 1PX	0.7	
	Mutação Comum	0.002	Teste 1 - 0.45
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 2 - 0.31
	Aleatória	100	Teste 3 - 0.37

Um outro parâmetro que pode ser avaliado é o tamanho da população. Com este propósito foram desenvolvidos duas novas configurações apresentadas nas tabelas 6.8 e 6.9. Na configuração representada pela 6.8, o número de indivíduos presentes na população foi aumentado de 100 para 120. Na configuração representada pela tabela 6.9, o número de indivíduos da população foi diminuído de

100 para 80.

**Tabela 6.8:** Parâmetros da configuração 7 para o método de torneio

		<b>Operador</b>	<b>Probalibilidade</b>		
Configuração 7		Crossover IPX	0.7	Resultados	
		Mutação Comum	0.002	Teste 1 - 0.70	
				Teste 2 - 0.81	
				Teste 3 - 0.75	
		<b>População Inicial</b>	<b>Tamanho</b>		
		Aleatória	120		

**Tabela 6.9:** Parâmetros da configuração 8 para o método de torneio

		<b>Operador</b>	<b>Probalibilidade</b>		
Configuração 8		Crossover IPX	0.7	Resultados	
		Mutação Comum	0.002	Teste 1 - 0.83	
				Teste 2 - 0.81	
				Teste 3 - 0.81	
		<b>População Inicial</b>	<b>Tamanho</b>		
		Aleatória	80		

Os resultados apresentados pela mudança (aumento ou diminuição da população) foram piores em relação a configuração cujo o número de indivíduos era 100. Baseado em todos estes testes, pode-se concluir que a melhor configuração utilizando o método de seleção por torneio para o problema abordado é a Configuração 2.

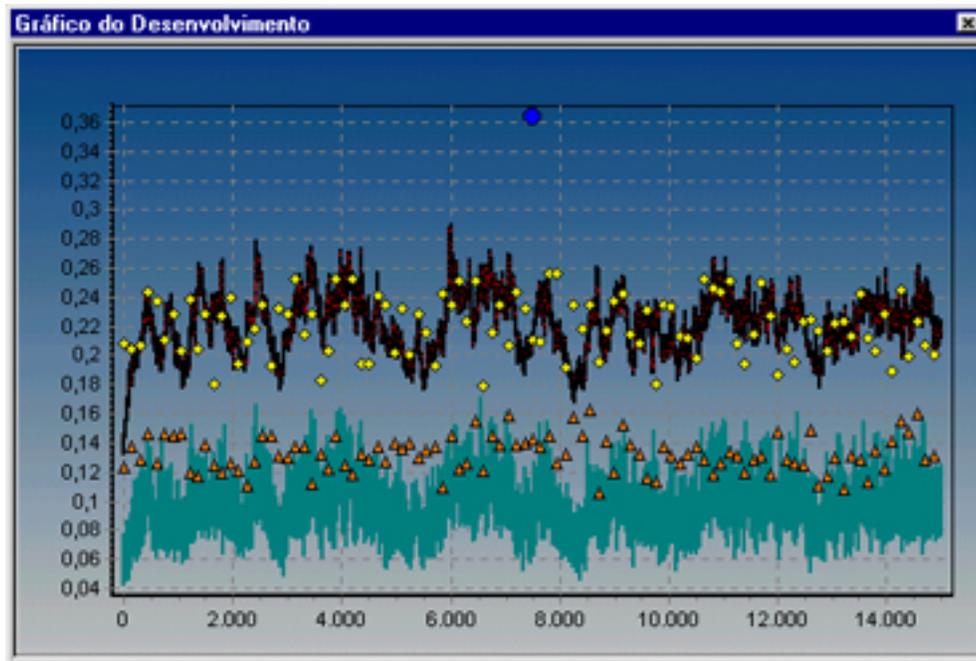
## 6.4 Seleção por Roleta Giratória

Um ponto de partida seria utilizar os parâmetros que melhor se adequaram ao método de seleção por torneio para realizar algumas comparações. A primeira configuração para o método da roleta é descrita na tabela 6.10.

Os resultados foram péssimos e um dos problemas se deve justamente a maneira pela qual os parâmetros foram configurados. A população se mostra muito dispersa, como se pode observar na figura 6.3. Neste caso, a busca tende a ser mais aleatória do que orientada. Um fato que pode minimizar este problema é a redução da probabilidade de mutação. A configuração 2 terá a probabilidade de mutação reduzida e será mostrada na tabela 6.11.

**Tabela 6.10:** Parâmetros da configuração 1 para o método da roleta

Configuração 1	Operador	Probalibilidade	Resultados
	Crossover 1PX	0.7	
Mutação Comum	0.002	Teste 1 - 0.37	
População Inicial	Tamanho	Teste 2 - 0.36	
Aleatória	100	Teste 3 - 0.30	

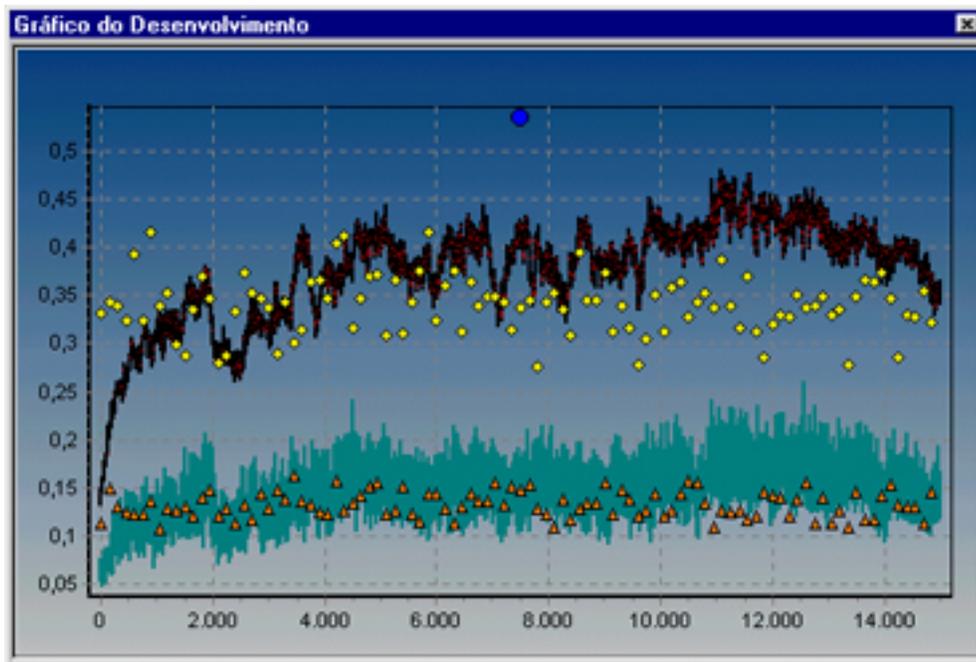


**Figura 6.3:** Comportamento do algoritmo para a configuração 1 do método da roleta

Através da figura 6.4, observa-se uma melhora significativa dos resultados, porém a população ainda continua dispersa. Um outro fato que também deve ser observado, é que a evolução da média do grau de aptidão dos indivíduos é muito inconstante, pois em alguns pontos ela cresce e em outros decresce. O ideal é que ela seja crescente. Isso pode ocorrer pelo incorreto uso da probabilidade de crossover, que dificulta a permanência dos bons indivíduos na população. Para

**Tabela 6.11:** Parâmetros da configuração 2 para o método da roleta

		<b>Operador</b>	<b>Probalibilidade</b>		
Configuração 2		Crossover 1PX	0.7	Resultados	
		Mutação Comum	0.001		
		<b>População Inicial</b>	<b>Tamanho</b>	Teste 1 - 0.52	
		Aleatória	100	Teste 2 - 0.50	
				Teste 3 - 0.55	



**Figura 6.4:** Comportamento do algoritmo para a configuração 2 do método da roleta

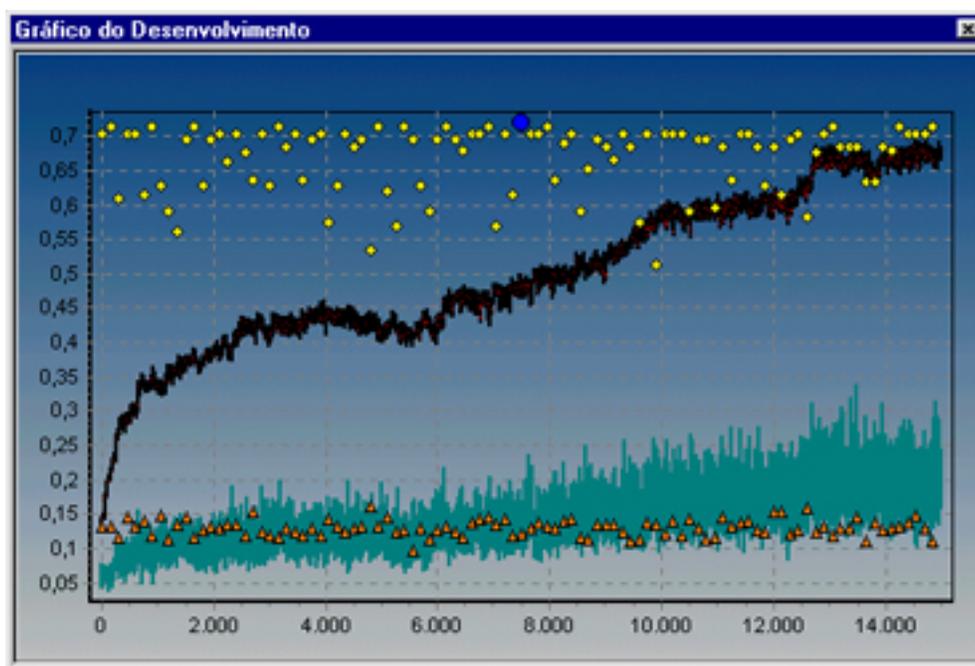
tentar minimizar estes dois problemas uma nova configuração foi criada. Esta configuração será descrita na tabela 6.12.

A configuração 3 fez com que a dispersão diminuísse e que a evolução ficasse um pouco mais constante, fatos estes observados na figura 6.5. O objetivo agora é investigar qual a probabilidade correta de crossover e de mutação.

Na configuração 4, tanto a probabilidade de crossover quanto a probabilidade de

**Tabela 6.12:** Parâmetros da configuração 3 para o método da roleta

		<b>Operador</b>	<b>Probalibilidade</b>		
Configuração 3		Crossover 1PX	0.6	<b>Resultados</b> Teste 1 - 0.68 Teste 2 - 0.72 Teste 3 - 0.70	
		Mutação Comum	0.0005		
		<b>População Inicial</b>	<b>Tamanho</b>		
		Aleatória	100		



**Figura 6.5:** Comportamento do algoritmo para a configuração 3 do método da roleta

mutação foram reduzidas. A tabela 6.13 apresenta os resultados obtidos por esta configuração.

Os resultados obtidos pela configuração 4 foram piores em relação a configuração 3. Uma tentativa que talvez traga melhores resultados é possibilidade de aumentar a taxa de crossover, já que a diminuição não foi favorável. Esta é justamente a proposta da configuração 5, descrita na tabela 6.14.

**Tabela 6.13:** Parâmetros da configuração 4 para o método da roleta

Configuração 4	<b>Operador</b>	<b>Probalibilidade</b>	Resultados
	Crossover 1PX	0.55	
	Mutação Comum	0.0005	Teste 1 - 0.60
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 2 - 0.66
	Aleatória	100	Teste 3 - 0.61

**Tabela 6.14:** Parâmetros da configuração 5 para o método da roleta

Configuração 5	<b>Operador</b>	<b>Probalibilidade</b>	Resultados
	Crossover 1PX	0.7	
	Mutação Comum	0.0005	Teste 1 - 0.76
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 2 - 0.71
	Aleatória	100	Teste 3 - 0.74

Como o nível dos resultados (obtidos na configuração 5) melhorou com o aumento da taxa de crossover, seria coerente aumentar ainda mais esta taxa e observar os resultados. Na configuração 6, a probabilidade de crossover é aumentada para 0.8 e os resultados dessa configuração são exibidos na tabela 6.15.

**Tabela 6.15:** Parâmetros da configuração 6 para o método da roleta

Configuração 6	<b>Operador</b>	<b>Probalibilidade</b>	Resultados
	Crossover 1PX	0.8	
	Mutação Comum	0.0005	Teste 1 - 0.60
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 2 - 0.55
	Aleatória	100	Teste 3 - 0.64

Note que está complicado determinar uma taxa correta de crossover e mutação. Devido a este fato, o próximo passo é variar o critério do tamanho da população. Nas configurações 7, 8, 9, 10 e 11 representadas pelas tabelas 6.16, 6.17, 6.18, 6.19, 6.20, o critério de tamanho da população é modificado.

O tamanho da população influenciou muito nos resultados. É importante observar que a Configuração 9 obteve um indivíduo muito bom (0.93). Configurado dessa maneira, o método da roleta aproxima-se dos resultados obtidos com a melhor con-

**Tabela 6.16:** Parâmetros da configuração 7 para o método da roleta

Configuração 7	<b>Operador</b>	<b>Probalibilidade</b>	Resultados
	Crossover 1PX	0.7	
	Mutação Comum	0.0005	Teste 1 - 0.66
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 2 - 0.60
Aleatória	80	Teste 3 - 0.58	

**Tabela 6.17:** Parâmetros da configuração 8 para o método da roleta

Configuração 8	<b>Operador</b>	<b>Probalibilidade</b>	Resultados
	Crossover 1PX	0.7	
	Mutação Comum	0.0005	Teste 1 - 0.77
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 2 - 0.80
Aleatória	120	Teste 3 - 0.79	

figuração relativa ao método de seleção por torneio. Entretanto como a população é maior, o tempo de processamento também será.

Um outro fator que também pode ser considerado é a possibilidade da utilização de um método para escalagem (Linear Scaling) do grau de aptidão. O único método implementado foi o de escalagem linear proposto por [Gol89]. A configuração 12 é criada baseado neste propósito. Esta configuração é mostrada na tabela 6.21. Os resultados apresentados pela configuração 12 não foram satisfatórios.

Assim como nos testes realizados com o método de seleção por torneio é conveniente testar o método da roleta utilizando os outros operadores de crossover e mutação implementados. A configuração 13 indicada pela tabela 6.22, realiza o teste com o operador de crossover com corte em dois pontos e a configuração 14,

**Tabela 6.18:** Parâmetros da configuração 9 para o método da roleta

Configuração 9	<b>Operador</b>	<b>Probalibilidade</b>	Resultados
	Crossover 1PX	0.7	
	Mutação Comum	0.0005	Teste 1 - 0.88
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 2 - 0.93
Aleatória	140	Teste 3 - 0.91	

**Tabela 6.19:** Parâmetros da configuração 10 para o método da roleta

Configuração 10	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover 1PX	0.7	Teste 1 - 0.75
	Mutação Comum	0.0005	Teste 2 - 0.81
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.71
	Aleatória	160	

**Tabela 6.20:** Parâmetros da configuração 11 para o método da roleta

Configuração 11	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover 1PX	0.7	Teste 1 - 0.81
	Mutação Comum	0.0005	Teste 2 - 0.71
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.86
	Aleatória	150	

indicada pela tabela 6.23, realiza o teste com os operadores heurísticos.

## 6.5 Método da Roleta com Redução

Conforme abordado anteriormente, o método da roleta com redução foi proposto neste trabalho e o objetivo era um pouco mais de justiça na distribuição das porções da roleta. Pelo fato da semelhança, os primeiros testes serão realizados com os parâmetros que melhor se adequaram a roleta giratória. A primeira configuração para este método é descrita na tabela 6.24.

Como a redução foi um valor aleatório escolhido, talvez seja este fator que esteja influenciando nos resultados. As configurações 2 e 3 representadas pelas tabelas

**Tabela 6.21:** Parâmetros da configuração 12 para o método da roleta

Configuração 12	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover 1PX	0.7	Teste 1 - 0.45
	Mutação Comum	0.0005	Teste 2 - 0.37
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.39
	Aleatória	140	

**Tabela 6.22:** Parâmetros da configuração 13 para o método da roleta

Configuração 13	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover 2PX	0.7	Teste 1 - 0.85
	Mutação Comum	0.0005	Teste 2 - 0.81
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.78
	Aleatória	140	

**Tabela 6.23:** Parâmetros da configuração 14 para o método da roleta

Configuração 14	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover Heurístico	0.7	Teste 1 - 0.31
	Mutação Heurística	0.0005	Teste 2 - 0.46
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.40
	Heurístico	140	

6.25 e 6.26 levam em conta o critério da redução.

Com fator de redução 0.8%, os resultados foram os melhores até o momento. O objetivo a partir de agora é investigar o fator tamanho da população. Na configuração 4 indicada pela tabela 6.27, a população foi reduzida de 140 para 120. Por outro lado, na configuração 5 indicada pela tabela 6.28, a população foi aumentada de 140 para 160.

Os resultados mostraram que variando o critério do tamanho da população não foi possível alcançar melhores resultados. As próximas configurações serão baseadas nas alterações das taxas de crossover e mutação. As configurações 6,7,8,9

**Tabela 6.24:** Parâmetros da configuração 1 para método da roleta com redução

Configuração 1	<b>Operador</b>	<b>Probalibilidade</b>	<b>Resultados</b>
	Crossover 1PX	0.7	Teste 1 - 0.60
	Mutação Comum	0.0005	Teste 2 - 0.70
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 3 - 0.68
	Aleatória	140	
	<b>Redução</b>	<b>Percentual</b>	
	Sim	1%	

**Tabela 6.25:** Parâmetros da configuração 2 para o método da roleta com redução

Configuração 2	<b>Operador</b>	<b>Probalibilidade</b>	Resultados Teste 1 - 0.70 Teste 2 - 0.71 Teste 3 - 0.66
	Crossover 1PX	0.7	
	Mutação Comum	0.0005	
	<b>População Inicial</b>	<b>Tamanho</b>	
	Aleatória	140	
	<b>Redução</b>	<b>Percentual</b>	
	Sim	0.6%	

**Tabela 6.26:** Parâmetros da configuração 3 para o método da roleta com redução

Configuração 3	<b>Operador</b>	<b>Probalibilidade</b>	Resultados Teste 1 - 0.76 Teste 2 - 0.71 Teste 3 - 0.70
	Crossover 1PX	0.7	
	Mutação Comum	0.0005	
	<b>População Inicial</b>	<b>Tamanho</b>	
	Aleatória	140	
	<b>Redução</b>	<b>Percentual</b>	
	Sim	0.8%	

foram elaboradas com este propósito e são indicadas pelas tabelas 6.29, 6.30, 6.31, 6.32 respectivamente.

Note que a alteração das taxas de crossover e mutação não trouxeram resultados mais significativos. Entretanto, assim como no método da roleta é necessário realizar teste com diferentes operadores de crossover e mutação. Na configuração 10 indicada pela tabela 6.33, o operador de crossover testado será o Crossover com corte em dois pontos e a configuração 11 representada pela tabela 6.34, realizará os testes com operadores heurísticos.

O último teste a ser realizado é verificar se o critério de sobrevivência dos indivíduos produz melhores resultados. Conforme relatado anteriormente, o critério implementado avalia a média dos descendentes em relação a média dos ancestrais. A configuração 12 indica pela tabela 6.35, realiza os testes utilizando critérios de sobrevivência.

**Tabela 6.27:** Parâmetros da configuração 4 para o método da roleta com redução

	<b>Operador</b>	<b>Probalibilidade</b>	
Configuração 4	Crossover 1PX	0.7	<b>Resultados</b>
	Mutação Comum	0.0005	
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 1 - 0.60
	Aleatória	120	Teste 2 - 0.65
	<b>Redução</b>	<b>Percentual</b>	Teste 3 - 0.58
	Sim	0.8%	

**Tabela 6.28:** Parâmetros da configuração 5 para o método da roleta com redução

	<b>Operador</b>	<b>Probalibilidade</b>	
Configuração 5	Crossover 1PX	0.7	<b>Resultados</b>
	Mutação Comum	0.0005	
	<b>População Inicial</b>	<b>Tamanho</b>	Teste 1 - 0.59
	Aleatória	160	Teste 2 - 0.65
	<b>Redução</b>	<b>Percentual</b>	Teste 3 - 0.57
	Sim	0.8%	

## 6.6 Considerações Finais

Os testes foram realizados baseados em apenas um problema. Para que o nível de confiança nos parâmetros aumente é necessário abordar vários problemas e realizar uma comparação entre eles. O problema abordado pode estar favorecendo um método de seleção, um operador de crossover ou qualquer outro parâmetro.

As várias formas de configurações são uma desvantagem muito grande para um algoritmo genético, pois é complicado explicar ao usuário final o que significa cada parâmetro. Este fato se agrava, quando a configuração do AG depende da entrada do problema. Desta forma, podem existir várias configurações para o mesmo problema.

Por outro lado, se o algoritmo for bem "calibrado", ele pode gerar bons resultados. Nos testes realizados neste trabalho, observou-se que de uma maneira geral o método de seleção por torneio, o operador de crossover com corte em um ponto e o operador de mutação comum apresentaram os melhores resultados. Baseado nessa idéia, estes parâmetros são candidatos fortes na tentativa de uma "calibragem" ideal

**Tabela 6.29:** Parâmetros da configuração 6 para o método da roleta com redução

		<b>Operador</b>	<b>Probalibilidade</b>		
		Crossover 1PX	0.7	<b>Resultados</b>	
		Mutação Comum	0.001		
Configuração 6	<b>População Inicial</b>	<b>Tamanho</b>	Teste 1 - 0.60 Teste 2 - 0.70 Teste 3 - 0.71		
	Aleatória	140			
	<b>Redução</b>	<b>Percentual</b>			
	Sim	0.8%			

**Tabela 6.30:** Parâmetros da configuração 7 para o método da roleta com redução

		<b>Operador</b>	<b>Probalibilidade</b>		
		Crossover 1PX	0.7	<b>Resultados</b>	
		Mutação Comum	0.0001		
Configuração 7	<b>População Inicial</b>	<b>Tamanho</b>	Teste 1 - 0.50 Teste 2 - 0.54 Teste 3 - 0.61		
	Aleatória	140			
	<b>Redução</b>	<b>Percentual</b>			
	Sim	0.8%			

do algoritmo.

**Tabela 6.31:** Parâmetros da configuração 8 para o método da roleta com redução

		<b>Operador</b>	<b>Probalibilidade</b>		
		Crossover 1PX	0.6		
		Mutação Comum	0.0005	<b>Resultados</b>	
Configuração 8	<b>População Inicial</b>	<b>Tamanho</b>			
	Aleatória	140	Teste 1 - 0.62		
	<b>Redução</b>	<b>Percentual</b>	Teste 2 - 0.71		
	Sim	0.8%	Teste 3 - 0.51		

**Tabela 6.32:** Parâmetros da configuração 9 para o método da roleta com redução

		<b>Operador</b>	<b>Probalibilidade</b>		
		Crossover 1PX	0.8		
		Mutação Comum	0.0005	<b>Resultados</b>	
Configuração 9	<b>População Inicial</b>	<b>Tamanho</b>			
	Aleatória	140	Teste 1 - 0.58		
	<b>Redução</b>	<b>Percentual</b>	Teste 2 - 0.45		
	Sim	0.8%	Teste 3 - 0.40		

**Tabela 6.33:** Parâmetros da configuração 10 para o método de roleta com redução

		<b>Operador</b>	<b>Probalibilidade</b>		
		Crossover 2PX	0.7		
		Mutação Comum	0.0005	<b>Resultados</b>	
Configuração 10	<b>População Inicial</b>	<b>Tamanho</b>			
	Aleatória	140	Teste 1 - 0.71		
	<b>Redução</b>	<b>Percentual</b>	Teste 2 - 0.66		
	Sim	0.8%	Teste 3 - 0.64		

**Tabela 6.34:** Parâmetros da configuração 11 para o método de roleta com redução

	<b>Operador</b>	<b>Probalibilidade</b>	
	Crossover Heurístico	0.7	
	Mutação Heurística	0.0005	
Configuração 11	<b>População Inicial</b>	<b>Tamanho</b>	Resultados Teste 1 - 0.31 Teste 2 - 0.40 Teste 3 - 0.38
	Aleatória	140	
	<b>Redução</b>	<b>Percentual</b>	
	Sim	0.8%	

**Tabela 6.35:** Parâmetros da configuração 12 para o método da roleta com redução

	<b>Operador</b>	<b>Probalibilidade</b>	
	Crossover Heurístico	0.7	
	Mutação Heurística	0.0005	
Configuração 12	<b>População Inicial</b>	<b>Tamanho</b>	Resultados Teste 1 - 0.33 Teste 2 - 0.41 Teste 3 - 0.29
	Aleatória	140	
	<b>Redução</b>	<b>Percentual</b>	
	Sim	0.8%	



# Referências Bibliográficas

- [Abr92] D. Abramsom. A parallel genetic algorithm for solving the school timetabling problem, 1992. Australian Computer Science Conference, Hobart.
- [Bac85] J. E. Backer. Adaptive selection methods for genetic algorithm, 1985. Internacional Conference on Genetic Algorithms and Their Applications.
- [Bri79] A. Brindle. Genetic algorithms for function optimization, 1979. University of Alberta.
- [Bur94] Edmund Burke. A genetic algorithm based university timetabling system, 1994. University of Nottingham.
- [Eve76] S. Even. On the complexity of timetabling and multicommodity flow problems. *SIAM Journal of Computation*, (5), 1976.
- [Fil00] Geraldo Ribeiro Filho. *Melhoramentos do Algoritmo Genético Construtivo e Novas Aplicações em Problemas de Agrupamento*. PhD thesis, INPE, São José dos Campos-SP, Sep 2000.
- [Fog91] D. B. Fogel. *System identification through simulated evolution: A Machine Learning Approach to Modeling*. Ginn Press, 1991.
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. The University of Alabama, 1989.
- [Hol75] J. H. Holland. *Adaptation in Natural and Artificial System*. University of Michigan Press, 1975.
- [Koz92] J. R. Koza. *Genetic Programming*. MIT Press, 1992.
- [Mit96] Melanie Mitchell. *An Introduction to Genetic Algorithms*. Massachusetts Institute of Tecnology, 1996.

- [Pac96] Marco Aurélio Pacheco. Fluxo de caixa inteligente: Planejamento e otimização por algoritmos genéticos, 1996. PUC-RJ.
- [Sch99] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, (13), 1999.
- [Wet83] A. Wetzel. Evaluation of the effectiveness of genetic algorithms in combinatorial optimization, 1983. University of Pittsburg.