

# Princípios para Projeto de Algoritmos Genéticos Competentes

Alexandre Cláudio Botazzo Delbem

Danilo Sipoli Sanches

Helson Mineiro

Hermes Castelo

Marcio Kassouf Crocomo

Marcilyanne Moreira Gois

Tiago Vieira da Silva

**Outros que Venham a Contribuir**

24 de junho de 2010

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Blocos construtivos e armadilhas . . . . .	5
1.2	Relação entre nível de dificuldade dos problemas e blocos construtivos . . . . .	6
1.2.1	Dificuldades intra-BB . . . . .	6
1.2.2	Dificuldades inter-BBs . . . . .	9
1.2.3	Dificuldades extra-BBs . . . . .	12
<b>2</b>	<b>Problemas difíceis: muitos BBs ou BBs grandes?</b>	<b>13</b>
2.1	Um único bloco construtivo . . . . .	13
2.2	Vários blocos construtivos . . . . .	15
2.3	Modelo de tamanho da população . . . . .	16
<b>3</b>	<b>Aumento das instâncias superiores de um bloco construtivo</b>	<b>20</b>
3.1	Teorema dos esquemas . . . . .	22
3.1.1	Efeito da seleção . . . . .	23
3.1.2	Efeito do cruzamento . . . . .	24
3.2	Equação generalizada do crescimento de esquemas . . . . .	25
<b>4</b>	<b>Tempo de domínio para um único bloco</b>	<b>28</b>
4.1	Efeito isolado da pressão de seleção . . . . .	29
4.2	Seleção por truncamento . . . . .	30
4.3	Seleção por torneio . . . . .	31
4.4	Seleção proporcional . . . . .	34
<b>5</b>	<b>Tempo de deriva genética para um único bloco</b>	<b>39</b>
5.1	Conceitos sobre Cadeias de Markov . . . . .	40
5.2	Estimação do tempo de deriva . . . . .	41
<b>6</b>	<b>Tempo de convergência: todos os BBs</b>	<b>46</b>
6.1	Estimativas preliminares . . . . .	47

6.2	O teorema de Fisher . . . . .	48
6.3	Convergência para o problema UmMax . . . . .	50
6.4	Tempo de convergência para o problema BinInt . . . . .	59
6.5	Convergência e deriva genética no BinInt . . . . .	63
<b>7</b>	<b>Competição entre instâncias de um mesmo bloco construtivo</b>	<b>65</b>
7.1	Modelagem da ruína do jogador . . . . .	66
7.2	Complexidade de um algoritmo genético competente . . . . .	69
<b>8</b>	<b>Combinação de instâncias de blocos construtivos diferentes</b>	<b>72</b>
8.1	Mapa de controle . . . . .	74
8.2	Fronteira de deriva genética . . . . .	75
8.3	Fronteira de competição cruzada . . . . .	76
8.4	Tamanho da região de sucesso em problemas difíceis . . . . .	77
8.5	Múltiplos blocos construtivos . . . . .	78

# Lista de Figuras

1.1	Função Trap-5 . . . . .	7
1.2	Distribuição de <i>fitness</i> em 500 indivíduos utilizando a função Trap-5. . . . .	8
1.3	Distribuição dos fitness em uma população de tamanho 500, utilizando uma trap-5. . . . .	9
1.4	Função armadilha tipo Trap-5. . . . .	10
1.5	Função armadilha tipo Trap-5. . . . .	11
2.1	Probabilidade de uma instância de BB ocorrer uma ou mais vezes na população. . . . .	15
2.2	Probabilidade de sucesso para um ou mais instâncias de BBs na população. . . . .	17
2.3	Relação entre tamanho da população e número de BBs. . . . .	18
3.1	Exemplo de um esquema. . . . .	21
3.2	Exemplo de um esquema com $\delta(H) = 6$ . . . . .	24
3.3	Curvas de $p_c$ em função de $s$ para diferentes valores de $\epsilon$ . . . . .	27
4.1	Seleção por truncamento . . . . .	30
4.2	Seleção por torneio . . . . .	32
4.3	Contribuição relativa do fitness de um conjunto de indivíduos - ESCREVER UMA DESCRIÇÃO MELHOR PRA ESSA FIGURA (por Marcio) . . . . .	36
5.1	Deriva genética ilustrada pela proporção de indivíduos 1s ao longo das gerações. . . . .	40
5.2	Diagrama de estados finitos. . . . .	43
5.3	Tempo de deriva genética em função de $l$ e $n$ . . . . .	45
6.1	Fitness resultante da proporção de uns na string. . . . .	50
6.2	Proporção de 1s pelo número de gerações para ... COMPLE- TAR!!!!!!!!!! . . . . .	55
6.3	torneio . . . . .	57

7.1	Comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo <i>generation-wise</i> . Soluções de tamanho 100. . . . .	68
7.2	comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo <i>generation-wise</i> . Soluções de tamanho 500. . . . .	69
7.3	comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo <i>generation-wise</i> . 20 cópias de funções armadilha de tamanho 4. . . . .	70
7.4	comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo <i>generation-wise</i> . 10 cópias de funções armadilha de tamanho 8. . . . .	71
8.1	Distribuição de <i>fitness</i> em diferentes BBs. . . . .	80
8.2	Pressão de Seleção . . . . .	81
8.3	Tamanho BB X $n \ln(n)$ . . . . .	81
8.4	BBs x Tamanho da população . . . . .	82

# Lista de Tabelas

5.1	Probabilidade das possíveis soluções entre os estados. Os estados 0 e 3 são absorvedores . . . . .	42
6.1	Termos que dominam . . . . .	60

# Prefácio

Muitos pesquisadores têm proposto melhorias ou adaptações de algoritmos genéticos para diversos problemas. As novidades ocorrem em diversas áreas, uma vez que tais algoritmos têm sido amplamente aplicados. Porém, várias das novidades são idéias muito similares que se repetem em diferentes áreas sem comunicação direta. A exploração tem sido intensa, mas com a aparência de um processo de tentativa e erro, em que os erros repetem-se por falta de um guia para o próprio processo de investigação.

Muitos pesquisadores têm proposto melhorias ou adaptações de algoritmos genéticos para diversos problemas. As novidades ocorrem em diversas áreas, uma vez que tais algoritmos têm sido amplamente aplicados. Porém, várias das novidades são idéias muito similares que se repetem em diferentes áreas sem comunicação direta. A exploração tem sido intensa, mas com a aparência de um processo de tentativa e erro, em que os erros repetem-se por falta de um guia para o próprio processo de investigação.

Por outro lado, diversos resultados teóricos sobre o funcionamento de algoritmos genéticos, que podem ser guias para investigação, foram obtidos nas últimas décadas. Um conjunto desses foi organizado no livro do pesquisador David E. Goldberg intitulado “Design of Innovation: lessons from and for competent genetic algorithms“. Esses algoritmos são modelados por partes ou aspectos relevantes do seu funcionamento. Para cada aspecto é proposto um modelo simples, mas representativo do seu comportamento principal. O conjunto coordenado desses modelos possibilita planejar como deveriam ser algoritmos genéticos capazes de resolver em tempo polinomial problemas computacionalmente complexos.

De fato, há algoritmos genéticos do tipo algoritmos de estimação de dis-

tribuição que apresentam tal desempenho para diversos problemas complexos. Outros textos tratam das pesquisas sobre tais algoritmos, a proposta desta "Nota Didática" é sintetizar os principais resultados teóricos de forma didática sem com isso faltar base para o entendimento dos modelos ou de suas deduções. Considera-se o caráter multidisciplinar, não somente do assunto, mas principalmente dos leitores interessados que podem possuir formação diversificada, em geral, são Engenheiros, Matemáticos, Físicos, Químicos, Biólogos e Computólogos. Por isso, não se pressupõe que conceitos relativamente básicos para certos profissionais sejam de conhecimento de todos. Assim, bases conceituais são recapituladas de forma simples sempre que necessário.

Em vários pontos deste texto, são apresentados conhecimento, explicações e críticas adicionais em relação à literatura da área. Com isso, busca-se tanto aumentar a clareza quanto despertar a capacidade de análise do interessado no assunto. Busca-se que o interessado seja capaz de, não somente conseguir uma rápida síntese de resultados e da importância deles para o projeto de novos algoritmos, mas também possa utilizar a teoria para projetar novos algoritmos e mesmo estender ou repensar os modelos apresentados. A visão crítica e interpretativa sobre os resultados predomina.

Esse também é o foco da disciplina de pós-graduação "Projeto de Inovação com Algoritmos Genéticos" do ICMC-USP, que é o primeiro propósito desta "Nota Didática". Os autores deste texto, em sua grande parte, foram alunos dedicados e interessados na teoria apresentada nessa disciplina e nas implicações da mesma para suas pesquisas e o desenvolvimento de algoritmos para problemas complexos.

Alexandre C. B. Delbem,  
São Carlos, 04 de janeiro de 2010.



# Capítulo 1

## Introdução

De forma a encontrar soluções mais eficientes para problemas complexos, novos algoritmos de otimização global têm sido proposto [?, ?, ?]. De acordo com [?], esses algoritmos podem ser denominados de Algoritmos Genéticos Competentes (ou em inglês, *Competent Genetic Algorithms*). Para um algoritmo ser caracterizado como competente deve possuir as seguintes propriedades:

1. Resolver problemas computacionalmente difíceis;
2. Rapidamente;
3. Com precisão;
4. Confiavelmente.

Para obter um algoritmo de otimização com tais propriedades, um componente tem mostrado ser fundamental, a construção de modelos probabilísticos sobre os valores das variáveis do problema. Os métodos de otimização que utilizam tais modelos têm sido chamados de Algoritmos de Estimação de Distribuição (EDAs, do inglês *Estimation of Distribution Algorithms*). Vários dos primeiros EDAs são descritos em [?]. As Notas Didáticas “Introdução aos Algoritmos de Estimação de Distribuição” [?] descrevem alguns dos principais EDAs com detalhes suficientes para se implementar adequadamente

tais algoritmos e entender os aspectos fundamentais envolvidos, de forma a habilitar pesquisadores interessados na pesquisa de EDAs.

Por outro lado, a Teoria que descreve o sucesso dos EDAs em relação aos demais Algoritmos Evolutivos da literatura, incluindo nesse contexto meta-heurísticas populacionais de uma forma geral, foi construída para explicar o funcionamento de um Algoritmo Genético (GA, do inglês *Genetic Algorithm*) mais simples que o *simple* GA descrito no relativamente famoso livro de Goldberg de 1989 [?] <sup>1</sup>. Esse GA possui as seguintes características:

1. Uma população de tamanho fixo;
2. Um operador de seleção dos indivíduos que irão reproduzir;
3. A nova população é composta somente pelos filhos gerados pelos indivíduos selecionados, isto é, não há elitismo;
4. A reprodução é realizada somente pelo operador de recombinação.

Esse algoritmo é chamado de GA selectorecombinativo, destacando os nomes dos únicos operadores utilizados: seleção e recombinação (ausência de um operador de mutação). A teoria sintetizada em [?] mostra que se pode construir GAs selectorecombinativos competentes. É importante destacar que essa teoria não afirma que um algoritmo de otimização para ser competente precise ser um GA selectorecombinativo, mas que suas características em conjunto com os sete elementos de projeto de algoritmos descritos a seguir podem ser suficientes para garantir a competência:

1. Entender que o adequado funcionamento de GAs depende da identificação e preservação de valores para subconjuntos de variáveis com forte correlação, que formam os chamados Blocos Construtivos (BBs, do inglês, *Building Blocks*);
2. Conhecer como a presença de BBs afeta o grau de dificuldade dos problemas;

---

<sup>1</sup> Este texto supõe conhecimentos básicos sobre GAs, que podem ser adquiridos estudando por exemplo o livro de De Jong [?] ou um texto mais simples, como as Notas Didáticas No. XY [ ] do ICMC-USP.

3. Suprir na população inicial instâncias de cada BB em número adequado;
4. Garantir que as instâncias superiores de um BB consigam aumentar a fatia de mercado, isto é, aumentem a presença delas nos indivíduos, dominando a população;
5. Conhecer os tempos de deriva genética e de domínio de uma instância de um BB;
6. Tomar decisões corretas em relação às instâncias competidoras de um mesmo BB;
7. Combinar adequadamente as melhores instâncias de BBs diferentes.

Na Seção 1.1, os itens (1) e (2) são melhor explicados, mostrando como relações envolvendo BBs incorrem em problemas com diferentes níveis de dificuldade. Os aspectos de projeto referentes aos itens de (3) a (7) são explorados nos Capítulos ?? a ??.

## 1.1 Blocos construtivos e armadilhas

RECAPITULAR O EXEMPLO DE 3 DIMENSÕES (VER FOTO DA LOUSA DA 1A. AULA) EXPLICANDO QUE AS VARIÁVEIS PODEM TER ALTA DEPENDÊNCIA, ISTO É, UM MÉTODO QUE ANALISE INDIVIDUALMENTE OU EM SEPARDO CADA VARIÁVEL ENCONTRA EM GERAL SOLUÇÕES SUBÓTIMAS. MOSTRAR QUE PARA GARANTIR A SOLUÇÃO ÓTIMA NESSE EXEMPLO É PRECISO UM MÉTODO QUE ANALISE O EFEITO NO VALOR DE FITNESS (EM GERAL A FUNÇÃO OBJETIVO DO PROBLEMA) DAS VARIÁVEIS EM CONJUNTO. COMENTAR QUE AS VARIÁVEIS QUE PRECISAM SER ANALISADAS EM CONJUNTO FORMAM OS BBs. ASSIM, É PRECISO IDENTIFICAR BBs EM PROBLEMAS DE FORMA QUE O MESMO NÃO TOME DECISÕES DE BUSCA EQUIVOCADAS.

## 1.2 Relação entre nível de dificuldade dos problemas e blocos construtivos

Dentre os problemas importantes encontrados no mundo real, vários são muito difíceis para Algoritmos Evolutivos e outras metaheurísticas. Dentre esses, o nível de dificuldade pode ainda variar significativamente. Como forma de medir tais níveis, foi realizada uma classificação baseada nos efeitos de BBs sobre a dificuldade dos problemas. Esses efeitos podem ser decompostos em três categorias:

- Dificuldades intra-BB;
- Dificuldades inter-BB;
- Dificuldades extra-BB.

### 1.2.1 Dificuldades intra-BB

Os Efeitos devido a relações entre as variáveis de um mesmo BB são chamados de dificuldades intra-BB e podem ser representados por problemas do tipo decepção, que podem ser modelados por uma função armadilha associada a variáveis do problema. Um problema é deceptivo se algoritmos gulosos [?], que realizam uma busca local, caminham no espaço de busca em direção a ótimos locais. Seguindo esse conceito, ótimos locais são armadilhas, assim, funções multimodais possuem diversas armadilhas conforme ilustra Figura 1.1.

A Figura 1.1 mostra uma função deceptiva simples de 5 *bits*, em que  $u$  é a quantidade de *bits* (genes) com valor 1 no BB. O BB possui um *fitness* que é um ótimo local (armadilha) quando todos os valores de seus *bits* são iguais a 0. Porém, o ótimo global corresponde a todos os genes com valor 1. Um algoritmo guloso iniciado para a grande maioria das 32 instâncias desse BB (exceto para 11111 e as demais com apenas 0 *bit*: 11110, 11101, ..., 00001), caminhará em direção à solução 00000 com *fitness* igual a 4; enquanto valor de *fitness* ótimo é 5.

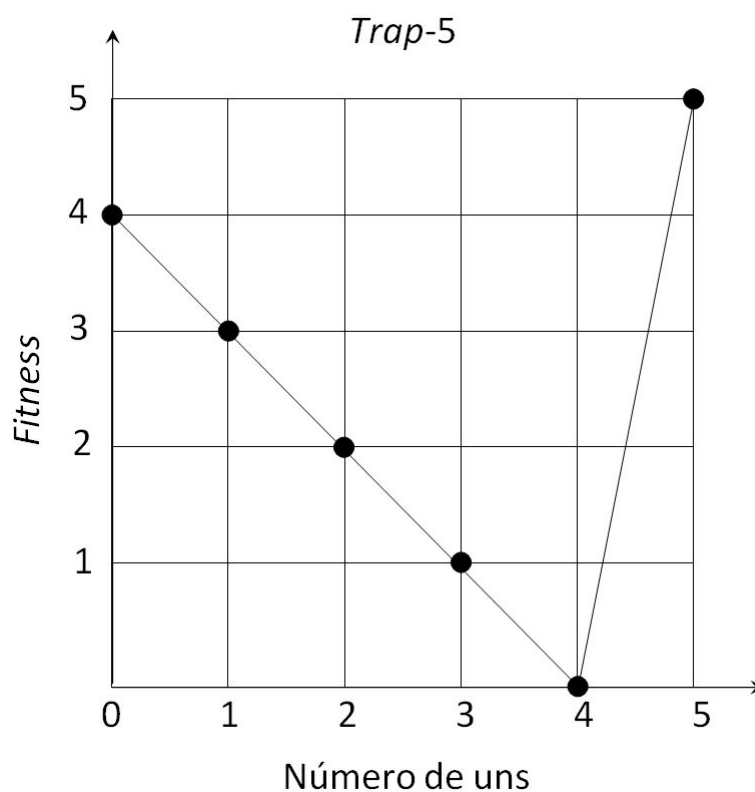


Figura 1.1: Função Trap-5

A escolha de indivíduos com as melhores instâncias de BBs é complicada por dois fatores:

1. A maioria das instâncias contribuem pouco no valor do *fitness*;
2. Dentre as que compõem parte significativa do *fitness* de um indivíduo, estão as armadilhas.

A Figura 1.3 mostra a distribuição dos *fitness*, calculados utilizando a função *Trap-5* para uma população de 500 indivíduos. Para caracterizar a dificuldade de um problema, podem ser utilizados os parâmetros (ver Figura ??):

ADICIONADA A IMAGEM. NO ENTANTO, TAMBEM FOI ADICIONADA UMA OUTRA IMAGEM NA PASTA FIG CHAMADA histgausk20png GERADA A PARTIR DE UMA TRAP-20, ONDE O GRAFICO SE APROXIMA MAIS DE UMA NORMAL. NO ENTANTO, TEMOS QUE

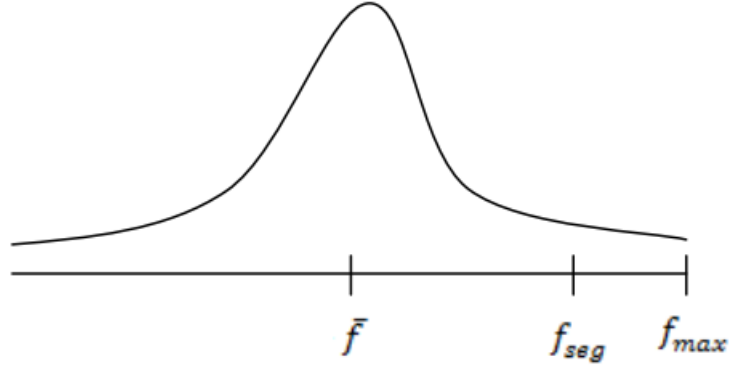


Figura 1.2: Distribuição de *fitness* em 500 indivíduos utilizando a função *Trap-5*.

REPENSAR E REESCREVER O TEXTO PARA SUBSTITUIR ESTA FIGURA (escrito por Marcio)

- $\bar{f}$ , o *fitness* médio na população;
- $f_{max}$ , o maior *fitness*;
- $f_{seg}$ , o segundo maior *fitness*;
- $d$ , calculado por  $f_{max} - f_{seg}$  e chamado de sinal;
- $s$ , calculada por  $f_{max} - \bar{f}$  e chamada de saliência;
- $\sigma^2$ , a variância associada a um BB, calculada por

$$\sigma^2 = \frac{1}{2^k} \sum_{i=0}^{2^k-1} (f_i - \bar{f})^2,$$

em que  $f_i$  é o *fitness* do indivíduo com a instância  $i$  do BB ( $BB_i$ ) e  $k$  é o tamanho (número de *bits*) do BB.

É fácil ver que, quanto menor o sinal ou a saliência, mais difícil é escolher um indivíduo que contenha a melhor instância do BB. Em outras palavras, problemas com funções de objetivo com tais características, tendem a ser mais difíceis.

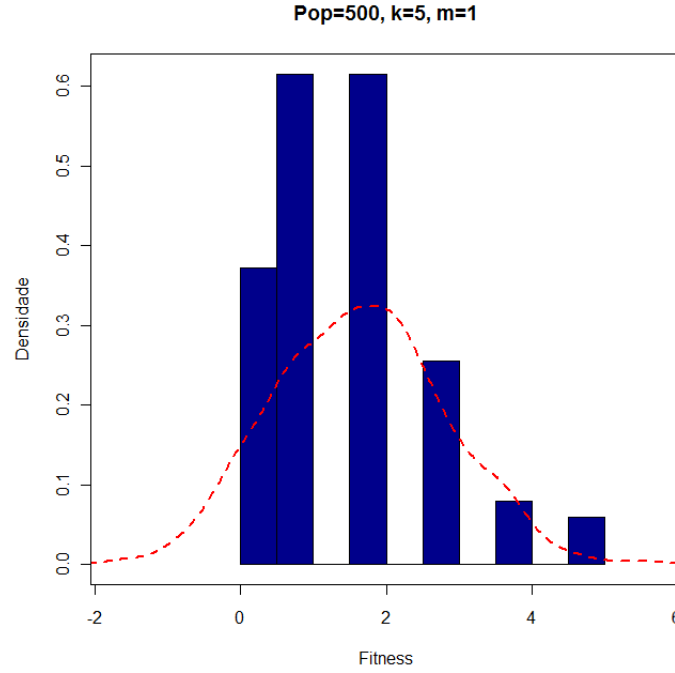


Figura 1.3: Distribuição dos fitness em uma população de tamanho 500, utilizando uma trap-5.

### 1.2.2 Dificuldades inter-BBs

O valor de  $f_i$  na verdade pode corresponder a combinação do efeito de instâncias de mais de um BB. Isso aumenta a variância da distribuição de *fitness*, isto é, a diversidade de valores de *fitness* disponíveis em uma população. Consequentemente, torna-se mais difícil distinguir quais indivíduos possuem as melhores instâncias de cada BB.

A Figura 1.4 mostra a distribuição de *fitness* de uma população de tamanho 500 gerada aleatoriamente utilizando uma função composta por duas funções armadilhas (2 BBs) tipo *Trap* – 5, descrita pela seguinte equação:

$$f_{2-trap-5} = \frac{1}{2}[f_{trap-5}(u_0) + f_{trap-5}(u_1)],$$

em que  $u_0 = \sum_{k=0}^4 x_k$  (número de 1s dos 5 primeiros *bits* de  $x$ ) e  $u_1 =$

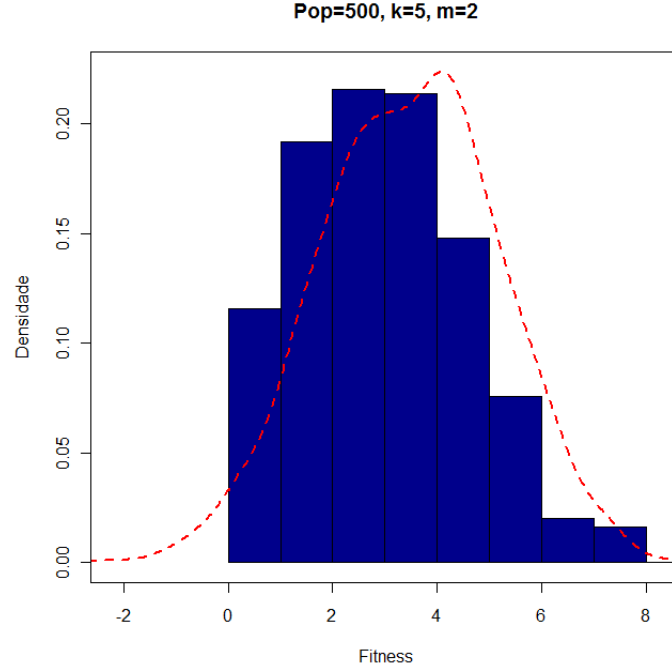


Figura 1.4: Função armadilha tipo *Trap-5*.

$\sum_{k=5}^9 x_k$ ,  $x_k$  é o valor do  $k + 1$ -ésimo *bit* da *string*  $x$  e

$$f_{trap-5}(u) = \begin{cases} 5, & \text{se } u = 5; \\ 4 - u, & \text{caso contrário.} \end{cases}$$

Para a distribuição da Figura ??,  $\sigma^2$  é XY; enquanto que  $\sigma^2$  é KZ para a distribuição de *fitness* dos indivíduos com apenas um BB apresentada na Figura ??.

Outra dificuldade pode ocorrer se diferentes instâncias de BBs possuírem contribuições com intensidades significativamente diferentes no *fitness* de um indivíduo. Com isso, o crescimento das instâncias de cada BB na população pode ser não uniforme, sendo mais difícil preservar as melhores instâncias de BBs com menor contribuição do que dos BBs com maior efeito sobre o *fitness*.

Assim, é necessário um estudo da interação entre instâncias envolvendo BBs diferentes. Esse efeito pode ser estimado pela razão de saliência entre



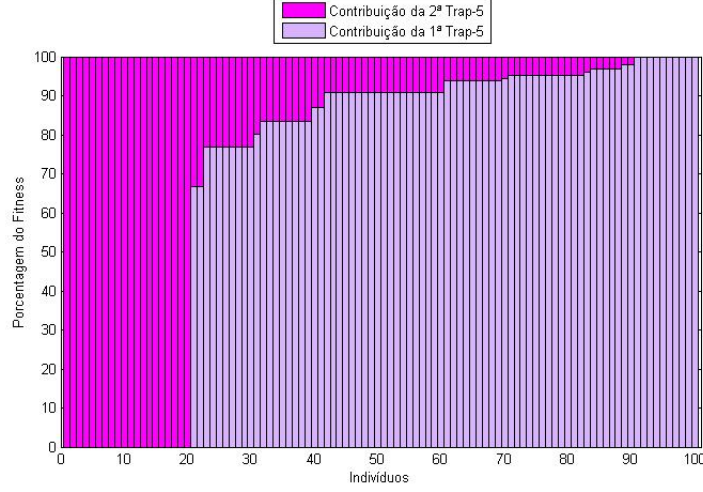


Figura 1.5: Função armadilha tipo *Trap-5*.

um  $BB_i$  e outro  $BB_j$  ( $i \neq j$ ), que pode é dada pela seguinte equação:

$$p_{ij} = \frac{s_i}{s_j}, \quad (1.1)$$

em que  $s_i$  é a saliência do  $BB_i$  e  $s_j$  é a saliência do  $BB_j$ . Com exemplo, considere a  $f'_{2-trap-5} = \frac{1}{11}[10f_{trap-5}(u_0) + f_{trap-5}(u_1)]$ . Nesse caso,  $p_{ij} = 2$  para  $\forall i \in BB_0$  e  $\forall j \in BB_1$ . A distribuição de *fitness* para uma população aleatória de tamanho 100 gerada utilizando a  $f'_{2-trap-5}$  é mostrada na Figura XY2. Observe essa  $\sigma^2$  é menor que a obtida para  $f_{2-trap-5}$ , aproximando-se do valor de variância obtido com a  $f_{trap-5}$ , em que se considerou apenas 1 BB.

A Figura 1.5 mostra a porcentagem de contribuição de cada um dos BBs na composição dos *fitness* da população de 100 indivíduos avaliados pela  $f'_{2-trap-5}$ .

Isso mostra que o efeitos do  $BB_1$  (de saliência menor) é difícil de ser identificado por meio da comparação dos *fitness* dos indivíduos. Portanto, problemas em que os BBs possuem saliências significativamente diferentes tendem a ser mais difíceis.

### Hierarquia entre BBs: BBs de BBs

EXPLICAR A HIERARQUIA QUE APARECE EXPLÍCITA PELA PRIMEIRA VEZ NA PROPOSTA DO H-BOA, ATÉ ONDE SABEMOS, MAS OBSERVAR QUE A TEORIA APRESENTADA NÃO ANALISA O ASPECTO DE NÍVEIS DE BBs SEGUNDO UMA HIERARQUIA

#### 1.2.3 Dificuldades extra-BBs

O *fitness* pode variar também devido a efeitos que independam dos BBs em si. Esses efeitos são fatores externos adicionais, que podem ser separados em:

1. Não-determinísticos (ruídos) e;
2. Não-estacionários.

O ruído é um elemento não-determinístico que altera de forma basicamente aleatória a contribuição de um BB no *fitness*, podendo afetar a escolha dos indivíduos pelo operador de seleção e ocorrer a seleção equivocada de indivíduos para a reprodução. No caso de sistemas não-estacionários, um mesmo indivíduo pode possuir em diferentes momentos, diferenciados valores de *fitness*, pois a função de *fitness* irá variar dinamicamente. Isso também pode gerar a seleção equivocada de indivíduos. Nesse caso, o ideal seria a escolha dos indivíduos mais robustos, que se destacam no mercado em vários períodos diferentes, pois teriam sido melhor avaliados ao longo do tempo.

Em ambos os casos, há em comum um efeito externo modificando de forma imprevisível a contribuição de um BB no *fitness* de um indivíduo. Assim, essas dificuldades presentes em vários problemas do mundo real podem ser modeladas por meio da seguinte alteração na função de *fitness*:

$$f(x) = f'(x) + G(0, \sigma^2(x)), \quad (1.2)$$

em que  $x$  é um BB,  $f'(x)$  representa a contribuição de  $x$  no *fitness* sem os fatores externos,  $\sigma^2(s)$  é variância dos fatores externos sobre  $x$ ,  $G$  é a função Gaussiana e  $f(x)$  a contribuição efetiva do BB no *fitness* do indivíduo.

## Capítulo 2

# Problemas difíceis: muitos BBs ou BBs grandes?

AGs selectorecombinativos requerem que a população inicial contenha toda a diversidade necessária para se encontrar o ótimo global, pois não utiliza um operador de mutação para produção de varibilidade genética ao longo das gerações. Assim, deve-se garantir que a população inicial contenha todas instâncias de BBs possíveis para assegurar que o GA selectorecombinativo possa resolver quaisquer problemas.

Este capítulo apresenta como estimar o tamanho de uma população gerada aleatoriamente de forma que essa amostra adequadamente as instâncias de BBs. A Seção 2.1 calcula o tamanho da população inicial para que essa amostra com alta probabilidade, todas as possíveis instâncias de apenas um BB. A Seção 2.2 generaliza esse resultado considerando vários BBs.

### 2.1 Um único bloco construtivo

Considerando uma população de  $n$  *strings* geradas aleatoriamente, a probabilidade  $p_k$  de que um esquema particular de  $k$  posições ocorra uma ou mais vezes pode ser dada pela Equação (2.1).

$$p_k = 1 - \left[1 - \frac{1}{\chi^k}\right]^n, \quad (2.1)$$

em que:

- $\chi$  é a cardinalidade do alfabeto da *string*;
- $k$  é a ordem do esquema;
- $n$  é o tamanho da população.

A Equação (2.1) pode ser deduzida conforme segue. Seja  $\frac{1}{\chi^k}$  a probabilidade de um esquema específico de  $k$  posições (supondo a independência entre as letras do alfabeto e a cardinalidade igual para cada posição da *string*), então,  $1 - \frac{1}{\chi^k}$  é a probabilidade do esquema não ocorrer e  $(1 - \frac{1}{\chi^k})^n$  é a probabilidade do esquema não ocorrer nenhuma vez na população. Assim,  $p_k$  é a probabilidade da instância ocorrer pelo menos uma vez em  $n$  *strings* da população.

A partir da aproximação  $(1 - \frac{p}{q})^q \approx e^{-p}$ , válida para  $p < q$ , e as seguintes substituições,  $p_k$  pode ser reescrita conforme a Equação (2.2):

$$p_k = 1 - \left[1 - \frac{n}{n\chi^k}\right]^n = 1 - \left[1 - \frac{r}{n}\right]^n,$$

em que  $r = \frac{n}{\chi^k}$ . Assim, utilizando a aproximação,

$$p_k = 1 - e^{-\frac{n}{\chi^k}}.$$

Por fim, fazendo  $n_k = \chi^k$ , obtém-se a equação:

$$p_k = 1 - e^{-\frac{n}{n_k}}. \quad (2.2)$$

Conforme apresentado em [?], esse modelo demonstra ajustar-se bem a resultados empíricos. A Figura 2.1 apresenta resultados teóricos e experimentais da variação da proporção ( $p_k$ ) de uma instância de um BB em uma população gerada aleatoriamente conforme  $k$  varia. O experimento foi gerado usando os seguintes parâmetros:

1.  $n$  variando de  $10^0$  a  $10^3$ ;
2.  $k$  variando de 1 a 8;

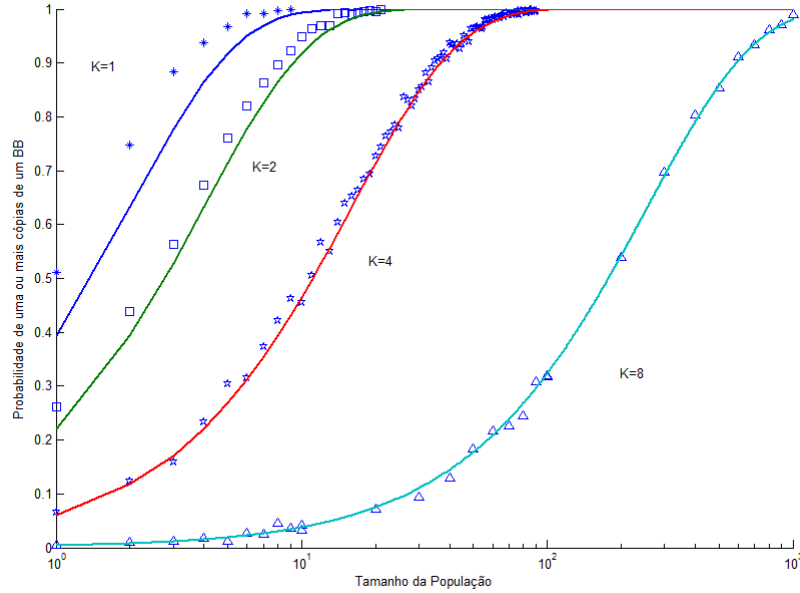


Figura 2.1: Probabilidade de uma instância de BB ocorrer uma ou mais vezes na população.

3.  $\chi = 2$ ;
4. Para cada combinação de parâmetros foram geradas 1000 populações iniciais e calculado o valor médio.

## 2.2 Vários blocos construtivos

Como as melhores instâncias de um BB são desconhecidas, busca-se garantir que todas as alternativas estejam representadas em algum indivíduo da população inicial. Por exemplo, se os primeiros *bits* de uma *string* de 5 *bits* possuem alta correlação, (isto é, o problema requer que esses dois *bits* sejam avaliados em conjunto, ver Capítulo 1) é interessante que todas as instâncias envolvendo os dois primeiros *bits* sejam geradas na população inicial: 00\*\*\*, 01\*\*\*, 10\*\*\*, 11\*\*\* (\* indica indiferença para o valor daquele *bit*).

Supondo que o sucesso de instâncias entre BBs diferentes sejam independentes, então, a probabilidade de ocorrer um ou mais sucessos para cada uma

das  $n_k$  instâncias de um BB é  $p_s = p_k^{n_k}$  (ISSO ESTÁ IGUAL AO DO LIVRO, MAS NÃO É CORRETO (verificar para  $n=2$ ,  $k=1$ ,  $n_k=2$ )). Substituindo  $p_k$  pela expressão da Equação (2.2), obtém-se:

$$p_s = \left(1 - e^{-\frac{n}{n_k}}\right)^{n_k}. \quad (2.3)$$

Utilizando novamente a aproximação  $(1 - \frac{p}{q})^q \approx e^{-p}$ , pode-se escrever:

$$p_s = \left(1 - \frac{n_k e^{-\frac{n}{n_k}}}{n_k}\right)^{n_k} = \left(1 - \frac{r}{n_k}\right)^{n_k}, \quad (2.4)$$

em que  $r = n_k e^{-\frac{n}{n_k}}$ . Com isso, a expressão para  $p_s$  resulta em:

$$p_s = e^{-n_k e^{-\frac{n}{n_k}}}. \quad (2.5)$$

A Figura 2.2 compara resultados empíricos com os obtidos por meio do modelo de  $p_s$ . Deve-se observar que a aproximação é melhor para  $r$  pequeno, assim, conforme  $n$  aumenta e  $n_k$  diminui o modelo de  $p_s$  torna-se mais preciso. Para  $n$  pequeno, o termo  $e^{-\frac{n}{n_k}}$  aproxima-se de 1 e  $p_s$ , de  $e^{-n_k}$ . Para  $n$  muito maior que  $n_k$ , o termo  $e^{-\frac{n}{n_k}}$  aproxima-se de 0, assim,  $p_s$  aproxima-se de 1.

Além disso, a suposição de independência do sucesso entre os BBs é apenas aproximadamente verdadeira. Em [?], é apresentado uma estimativa do  $p_s$  sem tal aproximação.

## 2.3 Modelo de tamanho da população

Da Equação 2.5, basta isolar a variável  $n$  para se estimar o tamanho de uma população que amostra adequadamente as instâncias de todos os BBs. Suponha um problema com BBs de ordem  $k$  sobre um alfabeto de cardinalidade  $\chi$ . Suponha também a probabilidade  $\alpha$  de uma instância de BBs não ter todas as suas instâncias representadas. Assim,  $p_s = 1 - \alpha$ , então,

$$1 - \alpha = e^{-n_k e^{\frac{n}{n_k}}} \quad (2.6)$$

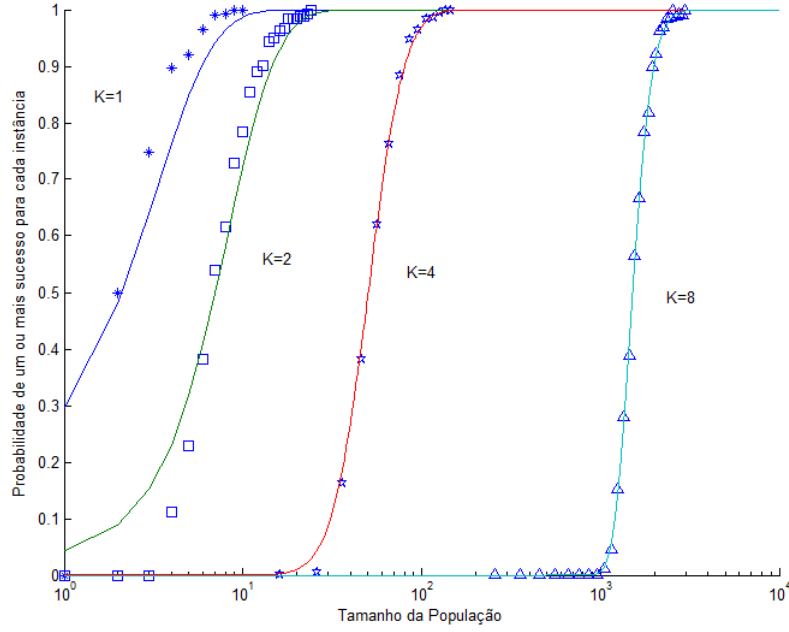


Figura 2.2: Probabilidade de sucesso para um ou mais instâncias de BBs na população.

e, portanto,

$$\ln(1 - \alpha) = -n_k e^{-\frac{n}{n_k}}. \quad (2.7)$$

Sabendo que  $\ln(1 - \chi) \approx -\chi$ , a equação anterior resulta em:

$$-\alpha = -n_k e^{-\frac{n}{n_k}}, \quad (2.8)$$

então,

$$\ln\left(\frac{\alpha}{n_k}\right) = -\frac{n}{n_k} \quad (2.9)$$

$$\ln \alpha - \ln n_k = -\frac{n}{n_k} \quad (2.10)$$

e, portanto,

$$n = n_k(\ln n_k - \ln \alpha) \quad (2.11)$$

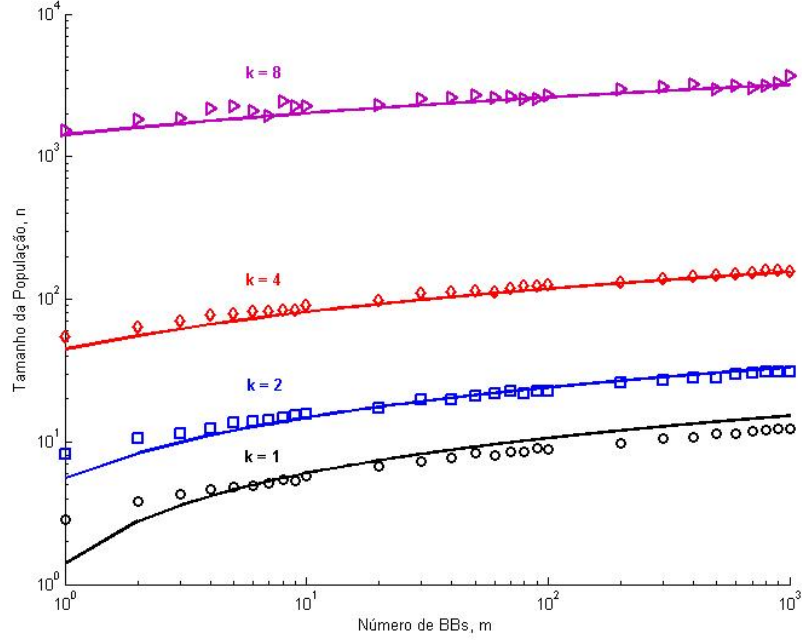


Figura 2.3: Relação entre tamanho da população e número de BBs.

Substituindo  $n_k = \chi^k$ , obtém-se a seguinte equação:

$$n = \chi^k (k \ln \chi - \ln \alpha). \quad (2.12)$$

Supondo que o erro do suprimento dos BBs seja tolerável em um nível inversamente proporcional ao número de BBs, isto é,  $\alpha = \frac{1}{m}$ , pode-se reescrever a Equação (2.12) como:

$$n = \chi^k (k \ln \chi - \ln m^{-1})$$

ou

$$n = \chi^k (k \ln \chi + \ln m),$$

em que  $m$  é o número de BBs.

A Figura 2.3, adaptada de [?], ilustra os resultados experimentais para esse modelo teórico de tamanho de população, mostrando a relação entre o tamanho da população ( $n$ ) e o número de BBs ( $m$ ), para diferentes tamanhos



de BBs ( $k$ ).

Observer que o aumento de  $m$  não muda a ordem de grandeza de  $n$  para a faixa de valores analisada. Por outro lado,  $n$  varia exponencialmente com  $k$ . Portanto, problemas com BBs um pouco maiores requerem bem mais recursos computacionais (geração de indivíduos e, principalmente, o cálculo do *fitness* dos mesmos) e, portanto, tendem a ser bem mais difíceis de resolver.

Observe que esse aspecto relevante da dificuldade de um problema expressado pelo tamanho e número dos BBs não contempla efeitos inter-BBs (ver Capítulo 8) e extra-BBs (ver Capítulo ??) que podem gerar problemas mais difíceis.

## Capítulo 3

# Aumento das instâncias superiores de um bloco construtivo

Como forma de encontrar explicações para o funcionamento dos GAs, Holland propôs em 1975 a Teorema dos Esquemas [?]. Um esquema é a forma de representar simultaneamente diversos cromossomos (cadeias de *bits* do alfabeto binário, onde cada posição de uma cadeia representa um gene e seu valor é chamado alelo). A estrutura do esquema é similar a de um cromossomo com seu alfabeto original ampliado pelo caracter asterisco (\*), que representa qualquer um dos elementos do alfabeto original. Por exemplo, um *string* de tamanho  $l$  somente com asteriscos representa todos os possíveis cromossomos. O esquema 1\*\*\*\* representa os cromossomos de tamanho 5 que começam com 1. A Figura 3.1 exemplifica um esquema.

Dessa forma, é fácil verificar que um esquema pode envolver um BB, parte de um BB ou, até mesmo, vários BBs. Assim, o entendimento de como os esquemas crescem sua fatia no mercado (na população), possibilita também entender como uma instância de BB pode dominar uma população.

A teoria proposta por Holland considerava que ao longo das gerações iam aparecendo padrões de alelos na população. Por exemplo, os esquemas 11\*\*\* e \*\*000 poderiam representar toda uma população em que parte tinha os dois

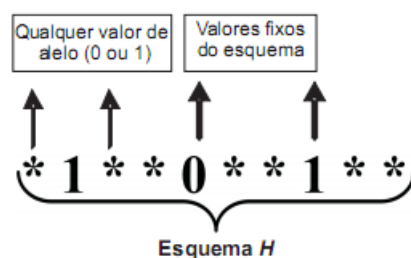


Figura 3.1: Exemplo de um esquema.

primeiros *bits* em 1 e a outra parte os três últimos *bits* em 0. O domínio por esses esquemas de fatias significativas da população indica que os indivíduos que os contém tiveram um aumento relativamente alto de seus *fitness* e que, por isso, puderam espalhar o seu padrão de sucesso na população. Além disso, o cruzamento possibilitaria os melhores esquemas serem combinados e, assim, construir um esquema que dominaria toda a população, encontrando a solução ótima de um problema.

A grande crítica a esse raciocínio e, por consequência, aos seus resultados, é que o cruzamento ao mesmo tempo que pode formar esquemas maiores e com capacidade para dominar uma fatia maior da população, também pode quebrar esquemas destruindo padrões importantes da solução ótima. Assim, os GAs convergiriam para ótimo locais.

A teoria de projeto de GAs competentes retoma o teorema dos esquemas, mostrando que esse é adequado para modelar o crescimento de padrões de sucesso na população e supondo que possa ser adicionado um mecanismo ao GA que evite totalmente a destruição de esquemas ou que pelo menos reduza de forma suficiente a tornar o GA competente. O Capítulo ?? fornece explicações, por meio de um procedimento de análise chamado mapa de controle, mostrando que praticamente não há parâmetros adequados para o GA se a taxa de quebras de esquemas for relativamente alta.

Mecanismos anti-quebra de esquemas têm sido desenvolvidos na literatura nos último dez anos. A idéia central desses mecanismos é a construção de modelos probabilísticos, que representam a distribuição dos alelos na população, possibilitando identificar potenciais esquemas. A eficiência desses

modelos é bastante variável e dependente em geral do tipo de problema. Mais recentemente, versões de EDAs, como o ECGA [?] e o BOA [?], são capazes de resolver problemas relativamente complexos de forma competente.

O modelos probabilísticos existentes e a competência dos respectivos GAs podem ser estudados a partir de outros textos []. A teoria de GAs competentes aqui apresentada supõe que tais mecanismos são possíveis e, com isso, pode-se utilizar o teorema dos esquemas.

### 3.1 Teorema dos esquemas

A Equação (3.1) mostra como ocorre o aumento do número de indivíduos com bons esquemas de uma geração para outra.

$$\langle m(H, t + 1) \rangle \geq m(H, t) \frac{\bar{f}(H, t)}{\bar{f}(t)} \left[ 1 - p_c \frac{\delta(H)}{l - 1} \right], \quad (3.1)$$

em que

- $\langle m(H, t + 1) \rangle$ , o número de indivíduos que representam o esquema  $H$  na geração  $t + 1$ ;
- $m(H, t)$ , o número de indivíduos com o esquema  $H$  na geração  $t$ ;
- $f(H, t)$ , o valor médio do *fitness* dos indivíduos com o esquema  $H$  na geração  $t$ ;
- $\bar{f}(t)$ , o valor médio da aptidão da população na geração  $t$ ;
- $p_c$ , a probabilidade de *crossover*;
- $\delta(H)$ , o número de possíveis pontos de corte que destroem o esquema  $H$ , que é igual a distância entre os dois valores fixos mais externos do esquema (o mais a esquerda e o mais a direita);
- $l$ , o número de *bits* da cadeia ou comprimento do esquema.

O termo  $\frac{\bar{f}(H, t)}{\bar{f}(t)}$  da Equação (3.1) indica o efeito da seleção no número de indivíduos com o esquema  $H$  da geração  $t$  para a  $(t + 1)$ . Supõem-se que

uma seleção proporcional está sendo utilizada, isto é, quanto maior o valor do *fitness* de um indivíduo maior a probabilidade de reproduzir e propagar o esquema que ele representa para as próximas gerações (ver Seção 3.1.1). A roleta [] é um dos operadores de seleção proporcional mais utilizados. O termo  $\left[1 - p_c \frac{\delta(H)}{l-1}\right]$  estima a fatia dos  $\langle m(H, t) \rangle$  indivíduos que geraram novos indivíduos sem destruição do esquema (ver Seção 3.1.2).

### 3.1.1 Efeito da seleção

O número esperado de cópias de um esquema  $H$  para a próxima geração,  $\langle m(H, t+1) \rangle$ , é a probabilidade  $P(H, t)$  de escolha de uma das *strings* com o esquema  $H$  da população corrente multiplicado pelo tamanho da população, conforme é apresentado na Equação (3.2).

$$\langle m(H, t+1) \rangle = P(H, t)n, \quad (3.2)$$

em que  $P(H, t)$  é a proporção de indivíduos com  $H$  que deve gerar novos descendentes. Para operadores de seleção proporcional, como a roleta,  $P(H, t)$  é baseado na proporção do *fitness* total composto pelos indivíduos com  $H$  em relação ao total de *fitness* na população (como se o total de *fitness* fosse um recurso que os indivíduos competiram para obter a maior fatia possível). Assim, pode-se escrever:

$$P(H, t) = \sum_{i \in H} \frac{f_i}{\sum_j f_j}, \quad (3.3)$$

assim,

$$\langle m(H, t+1) \rangle = n \sum_{i \in H} \frac{f_i}{\sum_{j \in n} f_j} \quad (3.4)$$

Desenvolvendo a Equação (3.4), obtém-se:

$$\langle m(H, t+1) \rangle = \frac{m(H, t)}{m(H, t)} n \sum_{i \in H} \frac{f_i}{\sum_{j \in n} f_j} \quad (3.5)$$

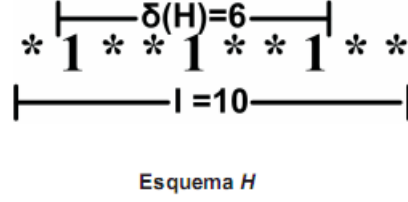


Figura 3.2: Exemplo de um esquema com  $\delta(H) = 6$ .

$$\langle m(H, t + 1) \rangle = m(H, t) \frac{\frac{\sum_{i \in H} f_i}{m(H, t)}}{\frac{\sum_{j \in n} f_j}{n}} \quad (3.6)$$

$$\langle m(H, t + 1) \rangle = m(H, t) \frac{\bar{f}(H, t)}{\bar{f}(t)}. \quad (3.7)$$

Observe que, se  $\langle m(H, t + 1) \rangle$  for maior ou igual a  $n$ , a população ficará saturada com um único esquema. Com isso, o GA converge.

### 3.1.2 Efeito do cruzamento

O cruzamento pode ter um efeito tanto construtivo quanto destrutivo no processo de composição de indivíduos melhores. Esses efeitos podem ser equacionados conforme mostrado a seguir.

Considere o esquema H  $*1**1**1**$  apresentado na Figura 3.2, em que  $\delta(H) = 6$  e  $l = 10$ . Se o ponto de corte para o cruzamento for à esquerda do 1 mais à esquerda, ou à direita do 1 mais à direita do cromossomo, o esquema nunca será quebrado por um cruzamento. Entretanto, se o ponto de corte do cruzamento for entre o 1 mais à esquerda e o 1 mais à direita, o esquema será quebrado. Então, a probabilidade de que um cruzamento de um ponto quebre tal esquema é:  $\frac{\delta(H)}{l-1}$ .

Admitindo uma taxa de cruzamento  $p_c$ , então a probabilidade  $d$  de que o cruzamento não destrua um esquema é:

$$d = \left[ 1 - p_c \frac{\delta(H)}{l-1} \right]. \quad (3.8)$$

Desta forma, dos indivíduos  $H$  selecionados para cruzamento (dados pela Equação (3.7)), apenas a fração  $d$  (dada pela relação (3.8)) irá para a próxima geração. Isso resulta que, em média, a quantidade de indivíduos de um esquema na próxima geração cresce de acordo com a Equação (3.1), como mostrado anteriormente.

Combinando os efeitos de seleção e cruzamento sobre o crescimento do esquema, obtém-se praticamente a Equação 3.1. Falta observar que o símbolo  $>$  em Equação 3.1 deve-se ao fato de que novos indivíduos com o esquema  $H$  podem ser novamente gerados a partir de pais que não tem  $H$ , produzindo um aumento adicional no crescimento da fatia de  $H$ .

## 3.2 Equação generalizada do crescimento de esquemas

A equação para o crescimento dos esquemas foi inicialmente mostrada para o operador de seleção proporcional e cruzamento de um ponto. Essa equação pode ser generalizada para quaisquer operadores de seleção e cruzamento. A generalização é apresentada em [?], da seguinte forma:

$$\langle m(H, t + 1) \rangle \geq m(H, t) \phi(H, m_i, f_i, t) P_s(H, m_i, f_i, t), \quad (3.9)$$

em que  $m_i$  é ?????????,  $f_i$  é ?????????,  $\phi(H, m_i, f_i, t)$  é a função de seleção generalizada ou fator de reprodução;  $P_s(H, m_i, f_i, t)$  é a probabilidade do esquema sobreviver, de acordo com o operador de cruzamento aplicado.

O fator de crescimento ( $\gamma$ ) é definido como segue:

$$\gamma = \phi(H, m_i, f_i, t) P_s(H, m_i, f_i, t). \quad (3.10)$$

Assim, o projeto de um AG deve obter  $\gamma \geq 1$ , possibilitando o aumento do número de indivíduos dos esquemas bem avaliados. Observe que para  $\gamma = 1$ , o crescimento é neutro, a fatia de  $H$  na população permanece constante.

### Analisando o efeito de $\gamma$ e seus parâmetros

Partindo da Equação (3.10) para o fator de crescimento, pode-se escolher  $\varphi$  como simplesmente a quantidade  $s$  indicando os  $s$ -melhores indivíduos que irão reproduzir (a pressão de seleção). Para tornar  $P_s$  independente do tipo de cruzamento e codificação utilizados no GA, pode-se considerar um taxa  $\epsilon$  de destruição de esquemas. Assim, dada a probabilidade de cruzamento  $p_c$ ,  $P_s$  pode ser calculada por  $1 - \epsilon p_c$  e, portanto, o fator de crescimento resulta em:

$$\gamma = s[1 - \epsilon p_c] \geq 1. \quad (3.11)$$

Da Equação (3.11), conclui-se que:

$$p_c \leq \frac{1 - s^{-1}}{\epsilon}. \quad (3.12)$$

Variando os parâmetros  $s$  e  $\epsilon$  da Equação 3.11, pode-se então determinar as faixas de  $p_c$  que garantam o crescimento da fatia de um esquema. A Figura 3.3 apresenta os resultados dessa análise. Observe que o  $p_c$  mínimo satura para  $s$  relativamente baixo, em torno de uma dezena. Uma interpretação é que, para  $s$  não grande o esquema domina rapidamente a população, convergindo o GA prematuramente. Como  $p_c$  não pode ser maior que 1, não há como gerar diversidade (indivíduos diferentes) para evitar convergência.

### Teorema dos esquemas ao longo das gerações

COMENTAR BREVEMENTE SOBRE COMO  $p_c$  E  $\epsilon$  MUDAM AUTOMATAMENTE AO LONGO DAS GERAÇÕES MANTENDO CERTA CAPACIDADE DE RECOMBINAÇÃO DE BBs.



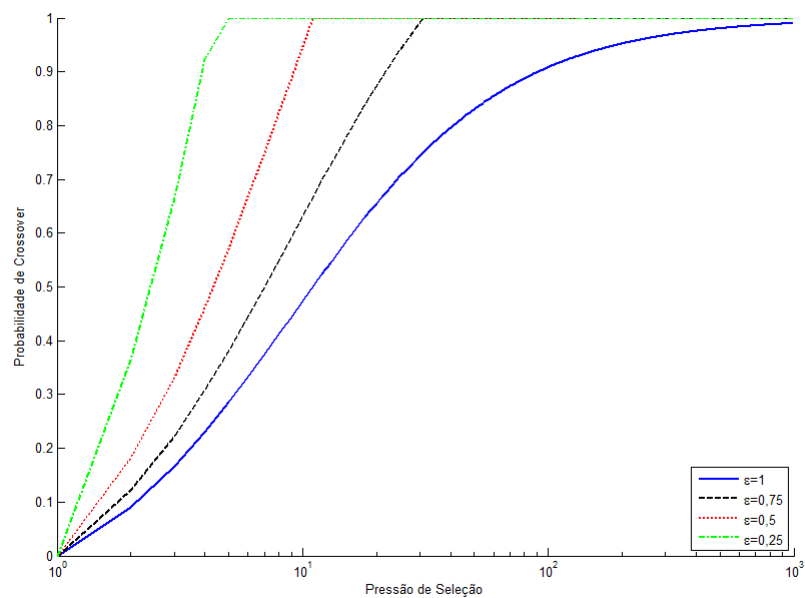


Figura 3.3: Curvas de  $p_c$  em função de  $s$  para diferentes valores de  $\epsilon$ .

## Capítulo 4

# Tempo de domínio para um único bloco

É importante que as melhores instâncias de um BB atinjam uma parcela suficientemente grande da população (uma grande fatia do mercado) para que o cruzamento possa gerar boas soluções. Para isso, busca-se estimar o tempo para que tais combinações ocorram antes da população convergir. A determinação desse tempo envolve, na verdade, dois modelos:

1. Um que estima o tempo que a melhor instância de BB requer para dominar a população;
2. O outro calcula o tempo em que um algoritmo pode convergir sua população se a pressão de seleção for nula. Nesse caso, a convergência deve-se ao acaso, pois não foi guiada pela qualidade das soluções.

O primeiro estimador modela o chamado tempo de domínio (em inglês, *takeover time*) e o segundo modela o tempo de deriva genética (em inglês, *drift time*). Se este tempo for menor que aquele, o algoritmo vai convergir para uma solução qualquer antes que seja possível o AG terminar seu trabalho compondo a solução ótima. Assim, o projeto de um GA deve garantir que o tempo de deriva genética seja maior que o tempo de domínio. As Seções ??, ??, ?? e ?? apresentam a estimativa do tempo de domínio. O tempo de deriva genética é descrito no Capítulo 5.

## 4.1 Efeito isolado da pressão de seleção

O tempo de domínio é o número de gerações necessário para uma instância de um BB (ou esquema no contexto do Capítulo 3) com fatia de mercado  $P_0 = \frac{1}{n}$  atingir a fatia  $P_f = \frac{n-1}{n}$  (quase 100% <sup>1</sup>, lembrando que  $n$  é o tamanho da população e os índices zero e  $f$  referem-se às populações inicial e final. Observe também que  $P_0$  ( $P_f$ ) é  $\frac{\langle m(H, t_0) \rangle}{n}$   $\left( \frac{\langle m(H, t_f) \rangle}{n} \right)$ , veja a Seção 3.1. Assim, usando as Equações 3.9 e 3.10, pode-se escrever que  $P_{t+1} \geq \gamma P_t$ .

As estimativas são calculadas em função da pressão de seleção, supondo a taxa de quebras de BBs igual a zero. Essa simplificação é adequada, uma vez que no desenvolvimento da teoria de GAs competentes, o problema da destruição de BBs pelo cruzamento é supostamente resolvido por mecanismo adicional (ver Capítulo 3). Assim, como  $\epsilon = 0$ , a expressão  $\gamma = s[1 - p_c\epsilon]$  (ver Equação 3.11) resulta em  $\gamma = s$ . Consequentemente,  $P_{t+1} \geq sP_t$ . Sendo conservador, o efeito do surgimento da mesma instância de um BB por cruzamento de pais sem tal BB pode ser desprezado, dessa forma obtém-se

$$P_{t+1} = sP_t. \quad (4.1)$$

O tempo para dominar pode depender do operador de seleção utilizado e não somente da pressão de seleção empregada. Na sequência, são apresentadas separadamente as estimativas desse tempo usando a seleção por truncamento, por torneio e, por fim, a seleção proporcional.

---

<sup>1</sup> EXPLICAR MELHOR O PORQUE DESSE CRITÉRIO DE CONVERGÊNCIA: 1) DÁ QUASE 100%; 2) GARANTIR 100% É IMPOSSÍVEL EM PROCESSOS ESTOCÁSTICOS; 3) MAS SE PODE EXIGIR A PRECISÃO QUE SE QUEIRA, TIPO 99,9999 (FALA 1 EM 1 MILHÃO); 4) OS RECURSOS (TEMPO E MEMÓRIA AUMENTAM EM FUNÇÃO DESSA PRECISÃO; 5) UTILIZANDO  $n-1/n$  OU  $1/m$  PODE-SE CARACTERIZAR O CUSTO DA PRECISÃO EM FUNÇÃO DE  $n$  OU  $m$ , POR EXEMPLO (CUSTO LINEAR, QUADRÁTICO, LOGARÍTMICO, ETC.). OBSERVE QUE ESTE TEXTO PROVÊ 2 FÓRMULAS PARA SE OBTER  $n$  EM FUNÇÃO DA DIFICULDADE DO PROBLEMA, COM ISSO E COM MODELOS DE USO DE RECURSOS EM FUNÇÃO DE  $n$ , PODE-SE OBTER QUAIS OS RECURSOS NECESSÁRIOS EM FUNÇÃO DA DIFICULDADE DO PROBLEMA.

## 4.2 Seleção por truncamento

A seleção por truncamento escolhe os  $s$  melhores indivíduos para reproduzir. Como, para avaliar somente o efeito da seleção não se aplica operador de reprodução de uma geração para a outra, pode-se completar a nova população simplesmente gerando  $\frac{n}{s} - 1$  cópias adicionais dos  $s$  indivíduos selecionados.

É razoável supor que esses  $s$  indivíduos englobem todos os indivíduos com a melhor instância de um certo BB, uma vez que a configuração ótima de um BB deve corresponder a uma contribuição significativa no *fitness*, colocando provavelmente os indivíduos com tal instância entre os  $s$  melhores. Em outras palavras, a seleção por truncamento escolhe as melhores instâncias de sucesso de um BB, conforme ilustrado na Figura ?? . Portanto, pode-se aplicar diretamente a Equação 4.1 para a seleção por truncamento.

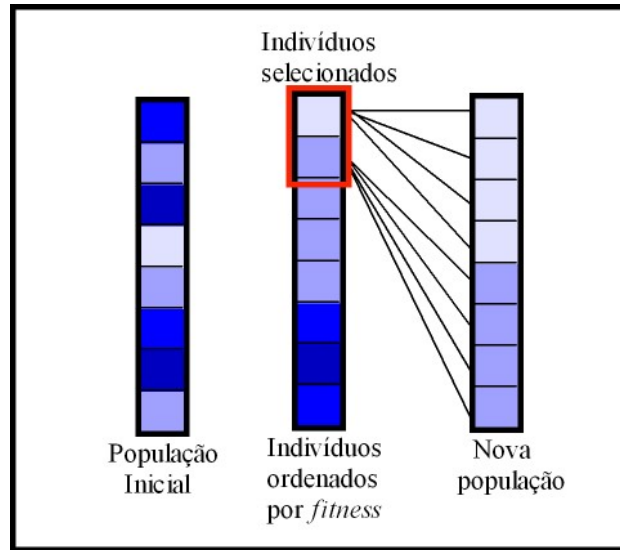


Figura 4.1: Seleção por truncamento

Aplicando sucessivamente a Equação 4.1 da geração inicial até a geração  $t$ , obtém-se:

$$P_1 = sP_0,$$

$$P_2 = sP_1 = ssP_0 = s^2P_0$$

$$\vdots$$

$$P_t = sP_{t-1} = ssP_{t-2} = \dots = s^t P_0.$$

Portanto,  $P_t = s^t P_0$ . Substituindo os valores de  $P_t$  ( $P_f$ ) e  $P_0$ , obtém-se:

$$\frac{n-1}{n} = s^t \frac{1}{n}$$

$$n-1 = s^t$$

$$\ln(n-1) = t \ln s$$

e, portanto,

$$t = \frac{\ln(n-1)}{\ln s}, \quad (4.2)$$

que obviamente é adequado aproximar para

$$t = \frac{\ln n}{\ln s}. \quad (4.3)$$

O domínio da fatia de mercado por uma instância do BB requer então  $O(\ln n)$  gerações. O processo é rápido, no entanto, esse ponto positivo pode atrapalhar se as inovações (os cruzamentos combinando as melhores instâncias de BBs diferentes) não ocorrem rapidamente. Estariam sendo selecionados indivíduos com melhores instâncias de BBs, porém ainda não haveria um indivíduo com todas as melhores instâncias de todos os BBs, compondo o ótimo global.

### 4.3 Seleção por torneio

No torneio, sorteia-se aleatoriamente  $s$  indivíduos, o melhor desses é separado para reproduzir. Na seleção por torneio, há a possibilidade de indivíduos com a melhor instância do BB não competirem em nenhum ou em parte significativa dos torneios. Assim, a seleção por torneio não garante as melhores instâncias de sucesso de um BB ganhem sempre. A Figura ?? ilustra o efeito do torneio de uma população para outra sem considerar a aplicação de oper-

ador de reprodução<sup>2</sup>.

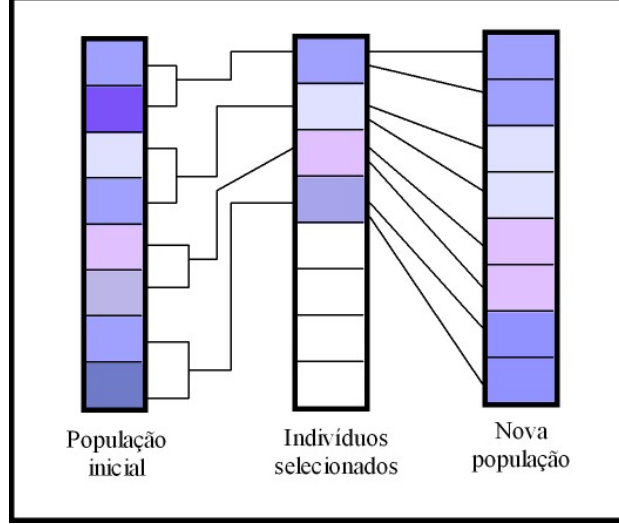


Figura 4.2: Seleção por torneio

A cada sorteio de um indivíduo para competir no torneio, a probabilidade de tal instância não estar no indivíduo sorteado é  $(1 - P_t)$ . A probabilidade de não estar em nenhum dos  $s$  indivíduos que participam do torneio é de  $(1 - P_t)^s$ . Assim, a chance de ter entre esses pelo menos um indivíduo com a referida instância é  $1 - (1 - P_t)^s$ , que é a probabilidade da mesma instância ser a vencedora do torneio supondo que a contribuição dela no *fitness* significativa para isso. O número esperado de tais instâncias entre  $n$  selecionadas<sup>3</sup> utilizando o torneio é, então,  $n[1 - (1 - P_t)^k] = \langle m(H, t + 1) \rangle$  ou

$$P_{t+1} = 1 - (1 - P_t)^s. \quad (4.4)$$

Fazendo  $Q_t = 1 - P_t$  (fatia de indivíduos com instâncias ruins do BB), a Equação 4.4 resulta em  $1 - Q_{t+1} = Q_t$ . Assim, pode-se escrever:  $Q_{t+1} = Q_t^s$ .

<sup>2</sup> Observe que, ao não se aplicar operador de reprodução para essas estimativas, obtém-se exclusivamente o efeito de certo tipo de seleção sobre a “velocidade” com que a uma instância de sucesso de um BB domina a população a longo das gerações, independentemente do efeito de um operador de reprodução aplicado a um conjunto de indivíduos selecionados para reproduzir.

<sup>3</sup> Como não se está aplicando operador de reprodução para essas estimativas, é preciso de alguma forma gerar um total de  $n$  indivíduos a partir dos escolhidos pelo operador de seleção em análise, construindo a próxima população.

Usando essa equação da geração inicial até a geração  $t$ , obtém-se:

$$\begin{aligned} Q_1 &= Q_0^s, \\ Q_2 &= Q_1^s = (Q_0^s)^s = Q_0^{s^2} \\ &\vdots \\ Q_t &= Q_{t-1}^s = (Q_{t-2}^s)^{s^2} = \dots = Q_0^{s^t} \end{aligned}$$

Portanto,

$$Q_t = Q_0^{s^t}. \quad (4.5)$$

Como  $P_0 = \frac{1}{n}$  e  $P_t = \frac{n-1}{n}$ , então  $Q_0 = \frac{n-1}{n}$  e  $Q_t = \frac{1}{n}$ . Assim, a Equação 4.5 resulta em:

$$\frac{1}{n} = \left( \frac{n-1}{n} \right)^{s^t}. \quad (4.6)$$

A aplicação dupla da função logarítmica  $\ln$  e algumas substituições produzem:

$$\begin{aligned} \ln \frac{1}{n} &= s^t \ln \frac{n-1}{n} \\ \ln \ln \frac{1}{n} &= t \ln s + \ln \ln \left( 1 - \frac{1}{n} \right) \\ t \ln s &= \ln \ln \frac{1}{n} - \ln \ln \left( 1 - \frac{1}{n} \right), \end{aligned}$$

usando a aproximação  $\ln(1-x) \approx -x$ , para  $x$  pequeno<sup>4</sup>,

$$t \ln s = \ln \ln \frac{1}{n} - \ln \frac{-1}{n}$$

ou

$$t \ln s = \ln \ln n^{-1} - \ln(-n^{-1})$$

$$t \ln s = \ln(-\ln n) + \ln(-n),$$

---

<sup>4</sup> Com  $x < 0,1$ , a precisão na aproximação é de  $10^{-1}$ . Esse nível de aproximação ocorre por exemplo para população com tamanho  $n = 10$ :  $\ln(1 - \frac{1}{10}) = -0,10536$  e  $-\frac{1}{10} = -0,10000$ .

como a parte real de  $\ln(-x)$  é  $\ln x$ <sup>5</sup>, pode-se escrever:

$$t = \frac{\ln n + \ln \ln n}{\ln s}. \quad (4.7)$$

O termo  $\ln \ln n$  é dominado por  $\ln n$  e, então, pode ser ignorado para  $n$  não pequeno. Assim, pode-se escrever:

$$t = \frac{\ln n}{\ln s}. \quad (4.8)$$

Esse resultado é igual ao obtido para a seleção por truncamento, em que o domínio da população pela instância também ocorre em  $O(\ln n)$  gerações para uma pressão de seleção  $s$ .

## 4.4 Seleção proporcional

A relação da pressão de seleção  $s$  com os indivíduos efetivamente escolhidos é de certa forma indireta ao ser utilizada a seleção proporcional. A Equação 4.1 não pode ser empregada diretamente, pois a função  $\varphi$  da Equação 3.9 não é mais constante e igual a  $s$ .

Operadores de seleção proporcional têm como base a seguinte relação:

$$P_{i,t+1} = \frac{f_i^t P_{i,t}}{\sum_{j \in J} f_j^t}, \quad (4.9)$$

em que  $P_{i,t}$  é a fatia de mercado do indivíduo  $i$  na população da geração  $t$ ,  $J$  é um conjunto de indexadores dos indivíduos e  $f_j^t$  é o *fitness* do indivíduo  $j$  na geração  $t$ .

Como esse efeito depende do problema, ou seja, da função de *fitness*, é preciso escolher uma função para determinar uma expressão explícita para  $P_{i,t}$  em função de  $t$ . Em [?], é sugerido a função  $f(x) = x^c$ , em que  $c$  é uma constante positiva e  $x$  é o intervalo contínuo  $[0, 1]$  que é particionado em  $n$  segmentos, um para cada indivíduo da população. Assim, mesmo que

---

<sup>5</sup> Observe que  $\ln(-x) = \ln x(-1) = \ln x + \ln(-1) = \ln x + i\pi$ , em que  $i$  é a unidade imaginária. Assim  $\Re\{\ln(-x)\} = \ln x$ .



a diferença de genótipo entre dois indivíduos seja pequena, o *fitness* deles será significativamente diferente, possivelmente distinguindo bem instâncias melhores de um BB. Essa propriedade ajuda no sentido de evitar a seleção equivocada de um indivíduo inferior em relação a outro.

Suponha  $P_0(x) = p_0$ , isto é, todos os indivíduos da população inicial com a mesma fatia de mercado  $p_0$ . A fatia de um indivíduo selecionado  $x$  na segunda geração será proporcional a  $x^c p_0$ . Na terceira geração, será  $(x^c)^2 p_0$  e assim sucessivamente, de forma que sua fatia na geração  $t + 1$  seja proporcional a  $(x^c)^t p_0$ .

Considere os indivíduos ordenados na população segundo  $f(x) = x^c$  e que, então, o melhor indivíduo corresponda ao intervalo (ver Figura 4.3)

$$B = [x : 1 - n^{-1} \leq x \leq 1].$$

Assim, partindo da Equação (4.9), pode-se escrever a Equação (4.10), que aproxima (substituindo somatórias por integrais) a soma dos *fitness* dos indivíduos selecionados (que estão em  $B$ ):

$$P_{B,t+1} = \frac{\int_{1-n^{-1}}^1 (x^c)^t p_0 dx}{\int_0^1 (x^c)^t p_0 dx}, \quad (4.10)$$

em que o denominador estima a *fitness* total da população da geração  $t$ , o numerador aproxima a soma dos *fitness* dos indivíduos em  $B$  e  $P_{B,t+1}$  é a fatia da população dos melhores indivíduos em termos de distribuição de *fitness*. Resolvendo as integrais conforme segue:

$$P_{B,t+1} = \frac{\int_{1-n^{-1}}^1 (x^c)^t p_0 dx}{\int_0^1 (x^c)^t p_0 dx} = \frac{\left. \frac{x^{ct+1}}{ct+1} \right|_{1-n^{-1}}^n}{\left. \frac{x^{ct+1}}{ct+1} \right|_0^1} = \frac{\frac{1^{ct+1}}{ct+1} p_0 - \frac{(1-n^{-1})^{ct+1}}{ct+1} p_0}{\frac{1^{ct+1}}{ct+1} p_0 - \frac{0^{ct+1}}{ct+1} p_0} =$$

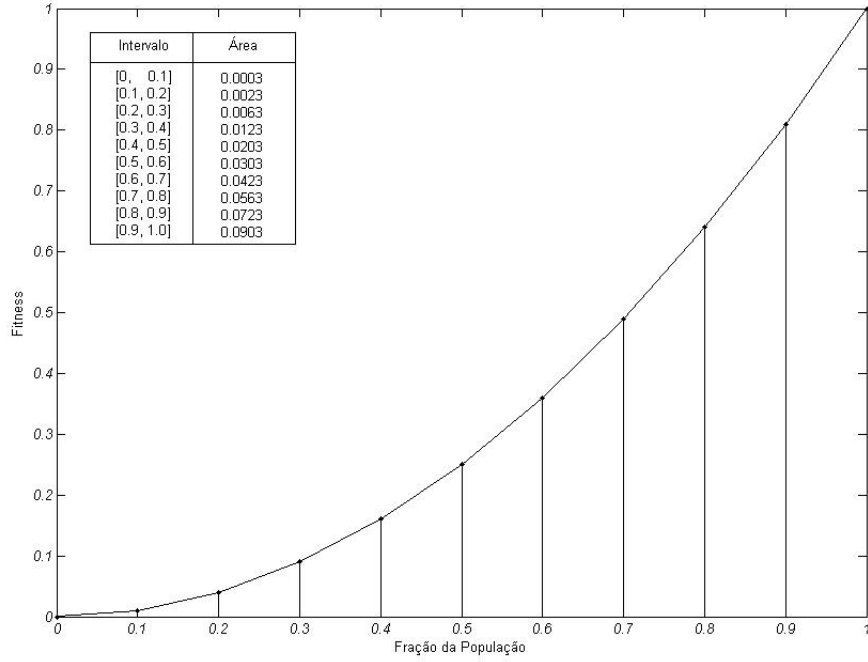


Figura 4.3: Contribuição relativa do fitness de um conjunto de indivíduos - ESCREVER UMA DESCRIÇÃO MELHOR PRA ESSA FIGURA (por Marcio)

$$\frac{\frac{p_0}{ct+1}[1 - (1 - n^{-1})^{ct+1}]}{\frac{p_0}{ct+1}} = 1 - (1 - n^{-1})^{ct+1} = 1 - \left(\frac{n-1}{n}\right)^{ct+1},$$

obtem-se:

$$P_{B,t+1} = 1 - \left(\frac{n-1}{n}\right)^{ct+1}. \quad (4.11)$$

Supondo que a fatia final seja  $P_f = \frac{n-1}{n}$ , a Equação (4.11) resulta em:

$$\frac{n-1}{n} = 1 - \left(\frac{n-1}{n}\right)^{ct+1}, \quad (4.12)$$

então, isolando a variável  $t$ ,

$$1 - \left(\frac{n-1}{n}\right) = \left(\frac{n-1}{n}\right)^{ct+1}$$

$$\ln \left(1 - \frac{n-1}{n}\right) = (ct+1) \ln \frac{n-1}{n}$$

$$t = \frac{1}{c} \frac{\ln(1 - \frac{n-1}{n})}{\ln(\frac{n-1}{n})}$$

ou

$$t = \frac{1}{c} \frac{\ln \frac{1}{n}}{\ln(1 - \frac{1}{n})}$$

$$t = \frac{1}{c} \frac{-\ln n}{\ln(1 - \frac{1}{n})}.$$

Utilizando a aproximação  $\ln(1-x) \approx -x$ , para  $x$  pequeno, pode-se escrever:

$$t = \frac{1}{c} \frac{1 - \ln n}{-\frac{1}{n}}$$

e, portanto,

$$t = \frac{1}{c} n \ln n. \quad (4.13)$$

Comparando esse resultado com o tempo para dominar obtido para as seleções por truncamento e torneio, verifica-se que, a menos de uma constante  $c$ , o tempo de domínio por seleção proporcional é um fator de  $n$  mais lenta. O efeito da seleção proporcional pode ser pior se a função de *fitness* não distinguir adequadamente entre os indivíduos. Para verificar isso, por exemplo, refaça os cálculos de estimativa de  $t$  utilizando a função linear  $f(x) = dx$ . O tempo de domínio torna-se  $t = n \ln n$ ,  $c$  vezes maior que a primeira estimativa. Recalcule utilizando  $f(x) = dx^e$ , para  $0 < e < 1$ . Então,  $t = \frac{1}{e} n \ln n$ , como  $e < 1$ , esse tempo é  $\frac{1}{e}$  vezes maior que a segunda estimativa e  $c \frac{1}{e}$  vezes maior que a primeira estimativa (Equação 4.13).

Assim, quando menos significativa é a diferença entre o *fitness* dos indivíduos, mais tempo é preciso para ocorrer o domínio por uma instância de BB. Além disso, a má distinção entre indivíduos pode provocar a convergên-

cia equivocada para uma solução subótima. Por essas razões, usuários de GA utilizando seleção proporcional, em geral, empregam algum mecanismo adicional, do tipo fator de escala ou ranqueamento [?, ?], buscando reduzir os efeitos negativos da seleção proporcional para certo problema.

## Capítulo 5

# Tempo de deriva genética para um único bloco

Quando não há diferenças de *fitness* significativas, não deveria existir nenhuma razão aparente para ocorrer a convergência, no entanto, o AG pode convergir mesmo com uma pressão de seleção nula. Esse fenômeno ocorre também na natureza [?], uma vez que erros estocásticos em uma população finita podem ser acumulados, resultando em flutuações significativas na proporção de espécies diferentes sem nenhuma pressão de seleção para isso. Esse é um fenômeno importante chamado de deriva ou afastamento genético.

A Figura 5.1 ilustra um caso de deriva genética, mostrando a oscilação da proporção entre indivíduos, ou formados com todos os *bits* iguais a zero (0s) ou com todos os *bits* iguais a um (1s). Inicialmente é gerada uma população com 50 indivíduos de 0s e 50 de 1s. Ambos os indivíduos possuem o mesmo *fitness*:  $f_{1s} = f_{0s}$ . Foi utilizado somente o operador de seleção. Esse experimento foi repetido 100 vezes, produzindo praticamente com a mesma frequência convergências tanto para indivíduos de 0s quanto de 1s.

O modelo de tempo de deriva explicado neste Capítulo é baseado em estimativas por Cadeias de Markov []. Outro modelo, mais simples e menos preciso, pode ser encontrado em [] apud [?]. A Seção 5.1 apresenta conceitos básicos sobre Cadeias de Markov. A Seção 5.2 aplica essa técnica para a estimação do tempo de deriva para um BB.

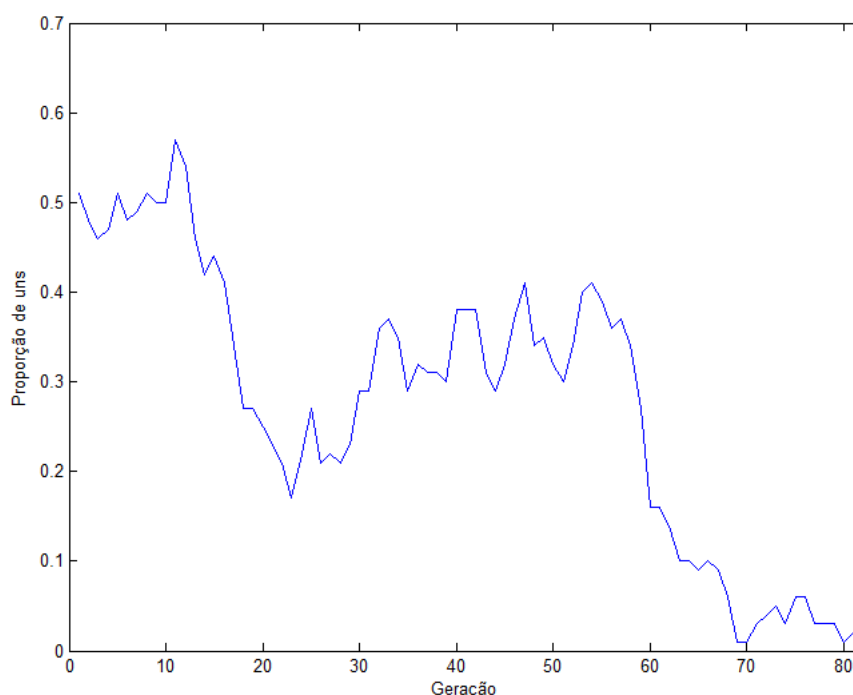


Figura 5.1: Deriva genética ilustrada pela proporção de indivíduos 1s ao longo das gerações.

## 5.1 Conceitos sobre Cadeias de Markov

Muitos processos físicos e biológicos que ocorrem na natureza envolvem seqüências de observações. Cadeias de Markov são modelos matemáticos usados para estudar processos resultantes dessas seqüências, em que cada observação pode ser vista como um estado discreto. Nessas seqüências, assume-se que um determinado estado depende exclusivamente do estado anterior. Em outras palavras, essa propriedade, denominada propriedade Markoviana, diz que os estados anteriores são irrelevantes para a predição dos estados seguintes, desde que o estado atual seja conhecido [?].

Uma forma simples de visualizar um tipo específico de Cadeia de Markov é por meio de um diagrama de estados finitos, isto é, um grafo orientado [?], em que os vértices são os estados e cada aresta representa uma transição entre estados. As arestas são rotuladas com as respectivas probabilidades de transição de estado.

Se você está no estado  $x$  no instante  $t$ , então a probabilidade de que você mover-se para o estado  $y$  no instante  $t + 1$  não depende de  $t$ , mas somente do estado atual  $x$  em que você está. Assim, em qualquer  $t$ , uma Cadeia de Markov finita pode ser caracterizada por uma matriz de probabilidades, em que cada elemento  $(i, j)$  da matrix é a probabilidade da transição do estado  $i$  para o estado  $j$  conforme a Equação (5.1).

$$P_{ij} = P(X_{t+1} = j \mid X_t = i), \quad (5.1)$$

em que  $X_t$  é o estado do instante  $t$  e  $P$  é a probabilidade do estado  $X_{t+1}$  ser  $j$  se o estado  $X_t = i$ .

Para um espaço de estados discretos, as integrações na probabilidade de transição de  $t$  passos (instantes de tempos consecutivos) podem ser calculadas como a  $t$ -ésima potência da matriz de transição. Se  $P$  é a matriz de transição para certo instante, então  $P^t$  é a matriz de transição para a transição após  $t$  passos.

Modelos desse tipo são utilizados em estudos de diversos fenômenos que ocorrem no mundo real, incluindo aplicações dos mesmos na Genética, que são cada vez mais frequentes. A referência [?] explora as Cadeias de Markov no estudo do fenômeno da deriva genética, que pode ser definido como a imprevisibilidade das frequências alélicas no decorrer das gerações devido a efeitos de amostragem.

## 5.2 Estimação do tempo de deriva

Primeiramente, constrói-se um modelo simples (diagrama de estados) para se exemplificar como o fenômeno de deriva genética em um GA pode ser representando utilizando Cadeias de Markov. Para isso, empregam-se os seguintes parâmetros:

- Tamanho do BB  $k = 1$ ;
- Tamanho da população  $n = 2$ ;
- Cardinalidade do alfabeto  $\chi = 2$ .

	0	1	2	3
0	1	0	0	0
1	0.25	0.25	0.25	0.25
2	0.25	0.25	0.25	0.25
3	0	0	0	1

Tabela 5.1: Probabilidade das possíveis soluções entre os estados. Os estados 0 e 3 são absorvedores

Dessa forma, há 4 estados (indexador como 0, 1, 2 e 3) para essa população, em que  $c_0$  ( $c_1$ ) é o cromossomo do indivíduo 0 (1):

- Estado 0: Indivíduo  $c_A = 0$  e  $c_B = 0$ ;
- Estado 1: Indivíduo  $c_A = 0$  e  $c_B = 1$ ;
- Estado 2: Indivíduo  $c_A = 1$  e  $c_B = 0$ ;
- Estado 3: Indivíduo  $c_A = 1$  e  $c_B = 1$ .

Probabilidades das possíveis transições entre esses 4 estados são apresentadas na Tabela 5.1

A partir dessas probabilidades, é possível montar o diagrama de estados finitos ilustrado na Figura 5.2. Observe que há estados em que a probabilidade de transição "para fora dele" é zero. Isso significa que uma vez atingido um desses estados, o sistema não "sai" mais dele, isto é, o sistema converge. Esses estados são chamados de estados absorvedores. Observe também que há uma submatriz  $Q$  composta somente de estados não-absorvedores. Se houver transição com probabilidade não nula desses estados para um estado absorvedor, o sistema irá convergir para tal estado para algum número de iterações.

Usando esse fato, é possível determinar também a quantidade de iterações  $t$  necessárias para ocorrer a convergência do sistema, isto é, atingir probabilidade praticamente 1 para as transições para estados absorvedores. Isso ocorre quando a probabilidade de transição entre estados não-absorvedores é zero. Dessa forma, pode-se determinar tal convergência por



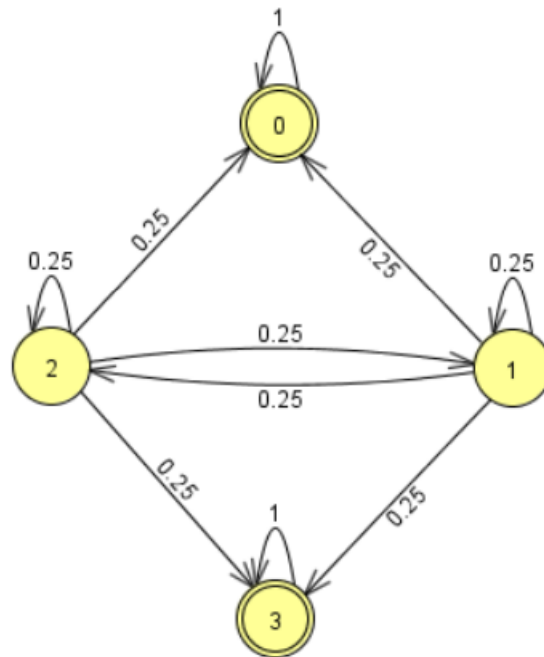


Figura 5.2: Diagrama de estados finitos.

análise somente da submatrix  $Q$ . Assim, esse problema pode ser formulado da seguinte maneira: encontrar  $t$ , tal que  $I - Q^t \approx I$ <sup>1</sup>.

Por exemplo, utilizando  $Q$  da Tabela ??, pode-se escrever:

$$Q = \begin{bmatrix} 0.2500 & 0.2500 \\ 0.2500 & 0.2500 \end{bmatrix}$$

$$I - Q = \begin{bmatrix} 0.7500 & -0.2500 \\ -0.2500 & 0.7500 \end{bmatrix}$$

Iteração 1:

$$Q^2 = \begin{bmatrix} 0.1250 & 0.1250 \\ 0.1250 & 0.1250 \end{bmatrix}$$

---

<sup>1</sup> O menor  $t$  tal que cada elemento da diagonal de  $I - Q^t$  aproxima-se significativamente de 1.

$$I - Q^2 = \begin{bmatrix} 0.8750 & -0.1250 \\ -0.1250 & 0.8750 \end{bmatrix}$$

Iteração 2:

$$Q^3 = \begin{bmatrix} 0.0625 & 0.0625 \\ 0.0625 & 0.0625 \end{bmatrix}$$

$$I - Q^3 = \begin{bmatrix} 0.9375 & -0.0625 \\ -0.0625 & 0.9375 \end{bmatrix}.$$

Portanto, em 2 iterações, o sistema deve convergir, pois  $I - Q^3$  aproxima-se significativamente da matriz identidade.

Esse mesmo resultado pode ser obtido por meio da propriedade de que a soma de uma das linhas da matriz  $N = (I - Q)^{-1}$  corresponde ao número esperado de iterações para o sistema convergir para um estado absorvedor []. Para o sistema do exemplo anterior, tem-se:

$$N = \begin{bmatrix} 1.5000 & 0.5000 \\ 0.5000 & 1.5000 \end{bmatrix}.$$

Calculando a soma para todas as linhas de  $N$ , geram-se os seguintes resultados:

- Linha 1:  $N_{0,1} + N_{0,1} = 2,0000$ ;
- Linha 2:  $N_{1,1} + N_{1,1} = 2,0000$ .

Assim, espera-se que o sistema convirja em 2 iterações.

Então, o número de iterações  $t$  é dado por  $\sum_j N_{i,j}$ . Esse número é diretamente proporcional ao número de estados não-absorvedores do sistema ( $|N_i|$ , cardinalidade de uma linha de  $N$ ), isto é,  $t = c_1 |N_i|$ , em que  $c_1$  é uma constante positiva. O valor de  $c_1$  depende das probabilidades de transição em  $Q$ . Por exemplo, se um dos estados absorvedores tiver probabilidades das transições para ele relativamente altas, o sistema possuirá uma tendência em “caminhar” para esse estado, atrasando a convergência para um estado absorvedor.

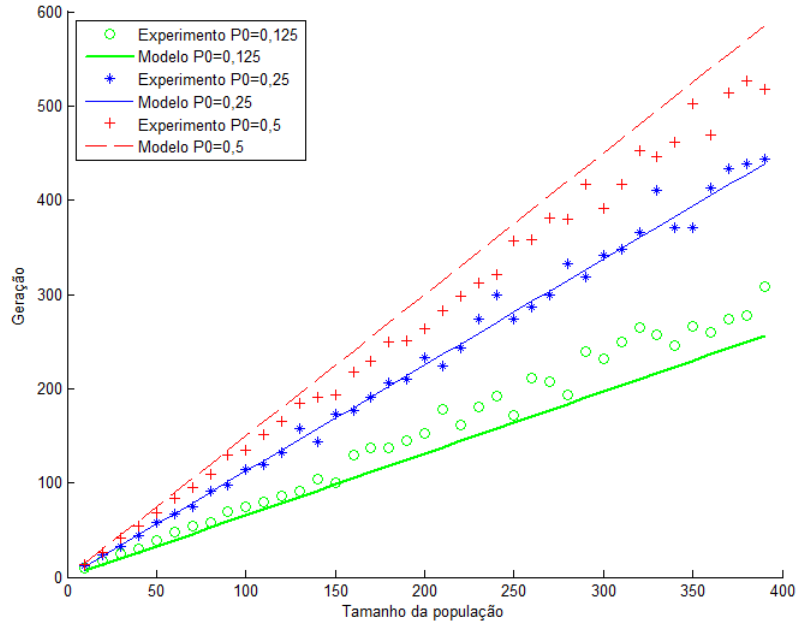


Figura 5.3: Tempo de deriva genética em função de  $l$  e  $n$ .

É fácil verificar que o número de estados não-absorvedores é proporcional ao número de indivíduos da população  $n$  e ao tamanho do cromossomo  $l$ . Assim,  $|N_i| = c_2 nl$ . Portanto,

$$t = c_3 nl, \quad (5.2)$$

em que  $c_3 = c_1 c_2$ .

A Figura 5.3 mostra resultados experimentais comparando tempos de deriva genética estimados com tempos obtido pelo modelo baseado em Cadeias de Markov. Por inspeção visual, conclui-se que os resultados do modelo teórico são relativamente precisos.

## Capítulo 6

# Tempo de convergência: todos os BBs

As estimativas sobre tempo de domínio (ver Capítulo 4) analisam o comportamento do GA apenas para um BB. Porém, o tempo de domínio de um BB pode ser afetado por efeitos de outro BB. Por exemplo, certa instância de um BB pode resultar em um aumento de *fitness* tão grande que somente os indivíduos com tal instância cresçam sua fatia de mercado por algumas gerações, deixando o crescimento das melhores instâncias de outros BBs estagnado. Esse tipo de fenômeno pode aumentar o tempo de domínio dos demais BBs. Dessa forma, o tempo de convergência do GA pode ser significativamente diferente do tempo de domínio de um BB.

Para se obter uma noção da variação do tempo de convergência de um GA em função de efeitos inter-BBs (Seção 1.2.2), a Seção 6.1 apresenta estimativas preliminares considerando dois casos extremos em que:

1. Todos os BBs dominam o mercado simultaneamente, isto é, de forma independente;
2. Os BBs dominam a fatia de mercado um por vez, isto é, a dependência é alta resultando em atrasos sucessivos no domínio dos BBs remanescentes.

Ao invés de utilizar o tempo de domínio de um BB para estimar o tempo de convergência do GA, nas Seções 6.2, 6.4 e 6.5 apresentam-se estimativas

que avaliam o domínio dos vários BBs em conjunto. Essa análise é baseada no Teorema sobre Seleção Natural [?, ?, ?, ?, ?] do estatístico e biólogo R. A. Fisher.

## 6.1 Estimativas preliminares

Com base no tempo de domínio da melhor instância de um BB, pode-se estimar valores mínimos e máximos para o tempo de convergência (de todos as melhores instâncias de todos os BBs dominarem a população) supondo que os BBs são processados todos:

1. Simultaneamente;
2. Sequencialmente.

No primeiro caso, assume-se que a melhor instância de cada BB irá dominar o mercado de forma independente. Com isso, o processo de domínio dessas instâncias iniciam-se com fatias de mercado similares, se para criação da população inicial utilizou-se a distribuição uniforme. Assim, o processo para cada BB ocorre de forma paralela sem acréscimo de tempo de computação. Portanto, o tempo de convergência  $t_c$  é igual ao tempo de domínio  $t$ , isto é:

$$t_c = t. \quad (6.1)$$

No caso de processamento sequencial, o *fitness* de um BB torna praticamente desprezível o *fitness* dos outros BBs (de menor saliência, ver Seção 1.2). Com isso, aqueles BBs que ainda não se destacaram, possuem fatias de mercado mais ou menos iguais. No pior caso, um BB de cada vez domina o mercado partindo de uma fatia de tamanho similares. Com isso, o processamento torna-se completamente sequencial e o tempo de convergência é o produto entre número de BBs do problema ( $m$ ) e o tempo de domínio:

$$t_c = mt. \quad (6.2)$$

Utilizando seleção por truncamento (ver Seção 4.3),  $t = \frac{\ln n}{\ln s}$  e assumindo o processamento totalmente simultâneo dos BBs, obtém-se:

$$t_c = \frac{\ln n}{\ln s}. \quad (6.3)$$

Por outro lado, assumindo o processamento sequencial dos BBs e seleção por truncamento, pode-se escrever:

$$t_c = m \frac{\ln n}{\ln s}. \quad (6.4)$$

As seleções por truncamento e torneio são independentes da saliência dos BBs. Por outro lado, o processo de seleção proporcional é dependente, pois as diferenças de saliência de um BB para o outro geram contribuições que podem ser significativamente diferentes no *fitness*, afetando o processo de seleção.

Dessa forma, as Equações 6.3 e 6.4 não considerarem efeitos inter-BBs (ver Capítulo 1) que são relevantes para processos evolutivos. Na Seção 6.2 é apresentado uma estimativa de  $t_c$  que modela as relações inter-BBs utilizando o Teorema de Fisher sobre Seleção Natural. Com isso, o tempo de domínio de cada BB (baseada em informações intra-BB) não é utilizado e estimativas mais plausíveis são obtidas.

## 6.2 O teorema de Fisher

O Teorema de Fisher [?] estabelece uma relação entre o crescimento do *fitness* médio de uma geração para outra com a variância do *fitness* dos indivíduos. Edwards, A.W.F. em [?] apresenta a versão atualizada desse teorema.

A seguir mostra-se que essa relação pode ser deduzida de uma forma simples. Para isso, considera-se a seleção proporcional (ver Equação 4.9), isto é:

$$P_{i,t+1} = \frac{f_i}{f_t} P_{i,t}, \quad (6.5)$$

em que  $f_i$  é o *fitness* do indivíduo  $i$ ,  $P_{i,t}$  é a proporção em que o indivíduo  $i$

aparece na população da geração  $t$  e  $f_t$  é o *fitness* médio da população da geração  $t$ . Observe que  $f_{t+1}$  pode ser calculado como segue:

$$f_{t+1} = \sum_{i=1}^n P_{i,t+1} f_i. \quad (6.6)$$

Assim, substituindo a Equação (6.5) em (6.6), obtém-se:

$$f_{t+1} = \sum_{i=1}^n \frac{f_i}{\bar{f}_t} P_{i,t} f_i,$$

ou

$$f_{t+1} = \sum_{i=1}^n \frac{f_i^2}{\bar{f}_t} P_{i,t}.$$

Assim,

$$\begin{aligned} f_{t+1} - f_t &= \sum_{i=1}^n \frac{f_i^2}{\bar{f}_t} P_{i,t} - \sum_{i=1}^n f_i P_{i,t} \\ f_{t+1} - f_t &= \frac{1}{\bar{f}_t} \sum_{i=1}^n f_i^2 P_{i,t} - f_t \sum_{i=1}^n f_i P_{i,t} \\ f_{t+1} - f_t &= \frac{1}{\bar{f}_t} \left[ \sum_{i=1}^n (f_i^2 P_{i,t}) - f_t^2 \right]. \end{aligned} \quad (6.7)$$

A variância de uma população com distribuição uniforme (fatia da população igual para todos os indivíduos:  $p_i = \frac{1}{n}$ ) é dada por  $\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$ , que pode ser escrita como  $\sigma^2 = \sum_{i=1}^n p_i x_i^2 - \bar{x}^2$ .<sup>1</sup> Como essa relação mantém-se para  $p_i$  não uniforme, a Equação 6.7 pode ser reescrita como segue:

$$f_{t+1} - f_t = \frac{\sigma_t^2}{f_t}, \quad (6.8)$$

em que  $\sigma_t^2$  é a variância na geração  $t$ .

---

<sup>1</sup> Em outras palavras, a variância pode ser calculada por  $E(x^2) - E(x)^2$ , em que  $E(x)$  é o valor esperado de  $x$ .

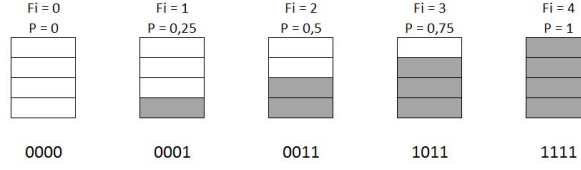


Figura 6.1: Fitness resultante da proporção de uns na string.

### 6.3 Convergência para o problema UmMax

O problema UmMax (em inglês, *OneMax problem*) consiste em, a partir de um conjunto de soluções iniciais aleatórias, atingir uma solução pré-especificada do domínio do problema (o máximo). Para isso, deve-se minimizar a função objetivo  $f_o(x) = \sum_i^l |x_i^k - x_i^O|$ , em que  $x_i^k$  ( $x_i^O$ ) é o valor do *bit*  $i$  da solução  $k$  (da solução ótima  $x^O$ ) e  $l$  é o tamanho de  $x$ ,

O problema UmMax pode ser resolvido rapidamente por um algoritmo de busca local, pois basta passar por cada *bit* de forma independente (BBs de tamanho 1) verificando qual o valor que diminui  $f(x)$ . Com isso, tal tipo de algoritmo encontra o ótimo em tempo linear.

Para simplificar, considere  $x_O = (1, \dots, 1)$ . Assim, o *fitness*  $f$  conforme o número de 1s aumenta e  $f(x)$  pode ser escrita como  $\sum_{i=1}^l x_i$ . Assim, a fatia  $o_t$  de indivíduos na geração  $t$  com *fitness*  $v$  é a proporção de indivíduos com  $v$  1s. A Figura 6.1 ilustra indivíduos com as respectivas proporções  $p$  de 1s usadas para avaliá-los.

A média de  $f(x)$  ( $f_t$ ) e a variância ( $\sigma_t^2$ ) na geração  $t$  podem ser obtidos pelos respectivos momentos de uma distribuição de Benoulli []. Em outras palavras, a *string*  $x$  pode ser vista como um conjunto de  $l$  experimentos com resultado aleatório 0 ou 1 (chamado um conjunto de experimentos de Bernoulli), isto é, o valor do *bit*  $x_i$  de  $x$  só pode ser 0 ou 1. Dado que  $p$  é a probabilidade de  $x_i$  ser 1, então  $(1 - p)$  é a probabilidade de ser 0. A distribuição de probabilidade de um conjunto de *strings*  $x$  é uma distribuição de Bernoulli [].

A média de  $x$  no conjunto é o valor esperado  $E(x) = 1p + 0(1 - p) = p$ . A



variância  $VAR[x]$  pode ser calculada por  $E[x^2] - E[x]^2$ , como  $E[x]^2 = p^2$  e  $E[x^2] = p$ .<sup>3</sup>

Verificando que  $f_t$  é a média de  $f(x)$ , então  $f_t = \frac{\sum_{j=1}^n f_j}{n} = \frac{\sum_{j=1}^n \sum_{i=1}^l x_i}{n}$ , em que  $x_j = \{0, 1\}$ , na geração  $t$ ). Assim,  $\frac{\sum_{j=1}^n l p_t}{n} = l p_t \frac{\sum_{j=1}^n 1}{n} = l p_t \frac{n}{n} = l p_t$ . Portanto,  $f_t = l p_t$  ou  $f_t = l E[x]$ . Então, a variância resulta em  $E[f^2] - E[f]^2 = E[(l E[x])^2] - E[l E[x]]^2 = E[l^2 E[x]^2] - E[l E[x]]^2 = E[l^2 p_t] - E[l p_t]^2 = l^2 p_t - l^2 p_t^2 = l^2 p_t (1 - p_t)$ .<sup>4</sup>

Em resumo,  $f_t = l p_t$  e  $\sigma_t^2 = l^2 p_t (1 - p_t)$ . Se tentar utilizá-la como estimador de  $t_c$  substituindo  $f_t$  e  $\sigma_t^2$  na Equação 6.8, você deve obter que  $t_c$  diminui com  $l$ . Esse resultado é incompatível com a realidade, pois obviamente problemas maiores de mesma complexidade requerem mais tempo computacional.

O fato é que a expressão  $\sigma_t^2 = l^2 p_t (1 - p_t)$  é válida somente para a primeira geração, por esse motivo será denominada  $\sigma_0^2$ . Na primeira geração, é adequado supor que a função de distribuição de probabilidade possui forma de sino como a função distribuição de probabilidade de Bernoulli. Na geração seguinte, essa função está truncada pela esquerda devida à aplicação do operador de seleção, conforme mostra a parte verde da Figura ?? . As demais gerações basicamente deslocam o limite esquerdo para a direita e, com isso, o formato da área verde (de função distribuição truncada) é mantido.

Em outras palavras, independentemente do tamanho  $l$  da string, o desvio padrão dos *fitness* após a primeira geração é bastante restrito. Assim, removendo a variável  $l$  de  $\sigma_t^2 = l^2 p_t (1 - p_t)$ , obtém-se um estimador  $\sigma_t^2$  razoável para as próximas gerações:  $\sigma_t^2 = p_t (1 - p_t)$ , para  $t \geq 1$ .

Deve-se observar que a distribuição de probabilidade diferenciada entre a

<sup>2</sup> Por definição,  $VAR[x] = \frac{\sum_{i=1}^l (x_i - \bar{x})^2}{l}$ , em que  $\bar{x} = E[x]$ . Assim,  $VAR[s] = \frac{\sum_{i=1}^l x_i^2}{l} - 2 \frac{\sum_{i=1}^l x_i \bar{x}}{l} + \frac{\sum_{i=1}^l \bar{x}^2}{l} = \frac{\sum_{i=1}^l x_i^2}{l} - 2 \bar{x} \frac{\sum_{i=1}^l x_i}{l} + \bar{x}^2 \frac{\sum_{i=1}^l 1}{l} = \frac{\sum_{i=1}^l x_i^2}{l} - 2 \bar{x} \bar{x} + \bar{x}^2 \frac{l}{l} = \frac{\sum_{i=1}^l x_i^2}{l} - \bar{x}^2 = E(x^2) - E(x)^2$ .

<sup>3</sup> Observe que  $E[x^2] = \frac{\sum_{i=1}^l x_i^2}{l}$ , como  $x_i = \{0, 1\}$ ,  $\sum_{i=1}^l x_i^2 = \sum_{i=1}^l x_i = l p$ . Portanto,  $E[x^2] = p$ .

<sup>4</sup> Observe que a probabilidade de ser ter *strings* com  $k$  1s em um conjunto de  $n$  ( $n \geq k$ ) (por exemplo, em uma população inicial de indivíduos de um GA de cromossomos binários) aproxima-se de uma distribuição de probabilidade binomial []:  $\Pr(K = k) = \binom{n}{k} p^k (1 - p)^{n-k}$ , em que  $\Pr$  é a probabilidade de ocorrer a *string*  $x$  com  $k$  1s entre  $n$  *strings*.

primeira e as demais gerações é um fator muito importante para a modelagem teórica de GAs. Esse aspecto é melhor explorado no Capítulo ?? que apresenta modelos mais precisos para estimação do tamanho da população.

Substituindo  $f_t = lp_t$  e  $\sigma_t^2 = l^2 p_t(1 - p_t)$ , para  $t \geq 1$ , na Equação 6.8, obtém-se:

$$p_{t+1} - p_t = \frac{p_t(1 - p_t)}{lp_t}$$

ou

$$\begin{aligned} p_{t+1} &= p_t + \frac{1 - p_t}{l} \\ 1 - p_{t+1} &= 1 - p_t - \frac{1}{l}(1 - p_t) \\ 1 - p_{t+1} &= (1 - p_t) \left(1 - \frac{1}{l}\right). \end{aligned} \tag{6.9}$$

Aplicando essa relação para as gerações de 2 e t, obtém-se:

$$1 - p_2 = (1 - p_1) \left(1 - \frac{1}{l}\right) \tag{6.10}$$

$$1 - p_3 = (1 - p_2) \left(1 - \frac{1}{l}\right) \tag{6.11}$$

⋮

$$1 - p_t = (1 - p_{t-1}) \left(1 - \frac{1}{l}\right). \tag{6.12}$$

Substituindo a Equação 6.10 em 6.11:

$$\begin{aligned} 1 - p_3 &= \left[ (1 - p_1) \left(1 - \frac{1}{l}\right) \right] \left(1 - \frac{1}{l}\right) \\ 1 - p_3 &= (1 - p_1) \left(1 - \frac{1}{l}\right)^2. \end{aligned} \tag{6.13}$$

Realizando essas substituições sucessivamente para as expressões da geração 4 a t (ver Equações 6.10 e 6.13), obtém-se:

$$1 - p_t = (1 - p_1) \left(1 - \frac{1}{l}\right)^t$$

ou

$$p_t = 1 - (1 - p_1) \left(1 - \frac{1}{l}\right)^t. \quad (6.14)$$

Para a geração 0, deve-se utilizar  $\sigma_0^2$ , assim pelo Teorema de Fisher, obtém-se:

$$p_1 - p_0 = \frac{l^2 p_0 (1 - p_0)}{l p_0}$$

ou

$$p_1 = p_0 + l(1 - p_0)$$

$$p_1 = p_0(1 - l) + l.$$

A substituição de  $p_1$  na Equação 6.14, produz:

$$p_t = 1 - [1 - p_0(1 - l) + l] \left(1 - \frac{1}{l}\right)^t$$

ou

$$1 - p_t = [1 - p_0(1 - l) + l] \left(1 - \frac{1}{l}\right)^t.$$

Sabendo que  $\left(1 - \frac{p}{q}\right)^q \approx e^{-p}$  para  $p < q$  e  $p_t = \frac{n-1}{n}$  (convergência do GA), pode-se escrever:

$$1 - \frac{n-1}{n} = [1 - p_0(1 - l) + l] e^{-\frac{t}{l}}$$

$$\frac{1}{n} = [1 - p_0(1 - l) + l] e^{-\frac{t}{l}}$$

$$\ln \frac{1}{n} = \ln[1 - p_0(1 - l) + l] - \frac{t}{l}$$

$$\frac{t}{l} = -\ln n^{-1} + \ln[1 - p_0(1 - l) + l]$$

$$t = l(\ln n + \ln[1 - p_0(1 - l) + l]).$$

Para  $l$  não pequeno,  $(1 - l) \approx -l$ , assim:

$$t = l(\ln n + \ln(1 - p_0(-l) + l)).$$

$$t = l(\ln n + \ln(1 + lp_0 + l)).$$

$$t = l(\ln n + \ln[1 + l(1 + p_0)]).$$

Portanto,  $t_c = O(l \ln l)$ .

Esse resultado difere, pelo termo  $\ln l$ , do estimador  $t_c = O(l)$ <sup>5</sup> obtido em ???. Nesse trabalho, considera-se para todas as gerações  $\sigma_t^2$  sem diferenciar a primeira geração utilizando  $\sigma_0^2$ .

A Figura 6.2 mostra que os modelos teóricos aproximam razoavelmente bem tempos de convergência obtidos em testes. Os dados experimentais foram gerados utilizando ...

COMPLETAR COM A DESCRIÇÃO DO EXPERIMENTO

TRAÇAR NESSE GRÁFICO AS CURVAS DOS 2 MODELOS:  $O(l)$  e  $O(l \ln l)$

Um ajuste diferente entre o modelo  $t_c = O(l)$  e resultados experimentais foi obtido utilizando o método de seleção não usual, o XYZ [], conforme mostra o artigo [] apud [?].

É importante destacar que o resultado  $t_c = O(l \ln l)$  para o tempo de convergência do GA (de todos os BBs) é quase linear com o tamanho do problema. Isso para um problema fácil como o UmMax. Em outras palavras, o GA obtém um desempenho pouco inferior ao de um algoritmo de busca local para esse problema.

Deve-se observar que os GAs são desenvolvido com o propósito de serem algoritmos otimização de caixa-preta (em inglês, *blackbox optimization*), em que o método não conhece a função objetivo, mas somente tem acesso ao valor de  $f(x)$  para cada  $x$ . Assim, o algoritmo caminha pelo espaço de busca utilizando sem utilizar nenhuma propriedade de  $f(s)$  que pudesse ser útil para orientar o processo de busca. Com isso, pode-se dizer que um GA, mesmo

<sup>5</sup> Observe que considerando  $\sigma_t^2 = p_t(1 - p_t)$  para todas as gerações, pode-se utilizar a Equação 6.14 a partir de  $t = 0$ , assim,  $p_t = 1 - (1 - p_0) \left(1 - \frac{1}{l}\right)^t$ . Então:  $\frac{n-1}{n} = 1 - (1 - p_0)e^{-\frac{t}{l}} \Rightarrow \frac{1}{n} = (1 - p_0)e^{-\frac{t}{l}} \Rightarrow -\frac{t}{l} = \ln \frac{1}{1 - p_0} \Rightarrow t = -l \ln \frac{1}{1 - p_0} \Rightarrow t = l \ln \frac{n}{1 - p_0}$ . Portanto,  $t_c$  seria  $O(l)$ .

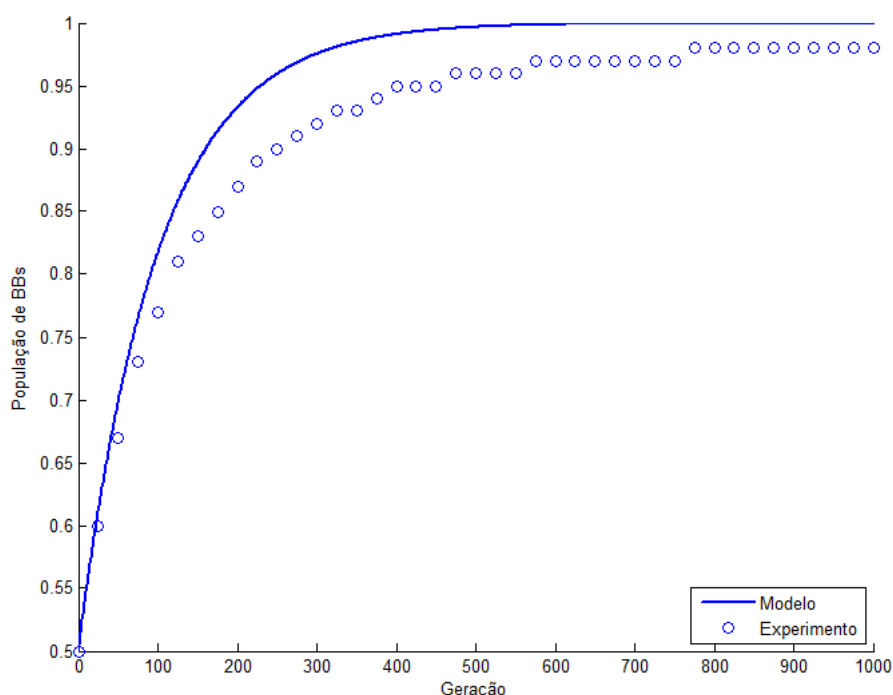


Figura 6.2: Proporção de 1s pelo número de gerações para ... COMPLETAR!!!!!!!!!!!!.

não sabendo que o problema UmMax é fácil, consegue um desempenho tão quase tão bom quanto um algoritmo de busca local, projetado para encontrar a solução ótima de problemas mais fáceis que, na linguagem proposta (ver Capítulo [?]), correspondem a problemas com BBs de tamanho 1.

Resta saber se o GA pode manter desempenho desse nível para problemas mais difíceis em que uma busca local falha ou outros algoritmos de otimização requerem tempo de processamento maior. Uma análise preliminar sobre esse aspecto é apresentada na Seção 6.4. Na sequência, mostra-se uma análise do efeito das seleções por torneio e truncamento sobre esse modelo teórico de  $t_c$ .

### Seleção por torneio e truncamento

As seleções por torneio e por truncamento não escolhem soluções baseando-se diretamente na magnitude do *fitness*, ao invés disso, o *fitness* é utilizado apenas para ordenar (extamente no caso do truncamento e aproximadamente

no torneio) de forma decrescente para se escolher as  $s$  que deverão reproduzir.

Para obter  $t_c$ , isto é, a o tempo de todos os BBs convergirem sem depender de hipótese sobre convergência sequencial ou simultânea de BBs (ver Seção 6.1), foi utilizando o Teorema de Fisher (ver Seção 6.8) que é fundamentado na seleção proporcional. O que se propõe a seguir é uma adaptação de um resultado relevante baseado em seleção proporcional que é ajustado para modelar efeitos das seleções por torneio e truncamento. Com isso, busca-se preservar a propriedade do modelo de  $t_c$  (Equação ??) de considerar o domínio de todos os BBs de forma simples e, ao mesmo tempo, os efeitos adicionados pelas seleções por torneio e truncamento.

Para calcular o tempo de convergência nesses casos, é suposto que o *fitness* da população possui uma distribuição normal na geração  $t$ . A Figura 6.3 ilustra tal distribuição, cuja média é  $f_t$  e a faixa da distribuição destacada em verde mostra a proporção de indivíduos (e os respectivos *fitness*) que seriam selecionados para uma pressão de seleção  $s$ . Suponha para isso, truncamento, em que os  $s$  primeiros (maiores *fitness*) são escolhidos. A distribuição truncada à esquerda (em verde) possui média ( $f_{t+1}$ ). O efeito do torneio é similar, conforme ilustrado na Figura 6.3.

Pode-se observar que a distância entre  $f_{t+1}$  e  $f_t$  ( $f_{t+1} - f_t$ ) depende da abertura da Gaussiana, que pode ser medida pelo  $\sigma_t$  da distribuição. A Figura ?? ilustra o efeito de  $\sigma_t$  sobre  $f_{t+1} - f_t$ . Essa distância depende também de  $s$ , que determina o valor que trunca a Gaussiana pela esquerda.

A Figura ?? mostra como a pressão de seleção ?? altera a distância  $f_{t+1} - f_t$ . O  $f$  mas à esquerda ( $f_{esq}$ ) da área (a fatia de mercado  $p$  de indivíduos selecionados) da parte verde (truncada) aumenta se  $s$  diminui. A relação entre a área  $1 - p$  (em que  $p$  é intensidade da pressão de seleção  $I$ , nesse contexto) de uma Gaussiana (aproximando a distribuição de Bernoulli por uma Gaussiana) e a abcissa  $f$  é  $1 - p = \frac{1}{2} + k \ln f$  [], para  $f > 0$ . Como  $p = \frac{s}{n}$ ,  $1 - \frac{s}{n} = \frac{1}{2} + k \ln f_{esq}$ , assim,  $\ln f_{esq} = \frac{1}{k}(\frac{-1}{2} + \frac{s}{n})$ , portanto,  $f_{esq} = e^{\frac{1}{k}(\frac{-1}{2} + \frac{s}{n})}$ .

$$p = k \ln f_{esq}. \text{ Como } p = k \frac{s}{n}, \frac{s}{n} = k \ln f_{esq} \Rightarrow f_{esq} = e^{k \frac{s}{n}}.$$

Assim, há dois efeitos que combinados :

FAZER AS FIGURAS CORRESPONDENTES

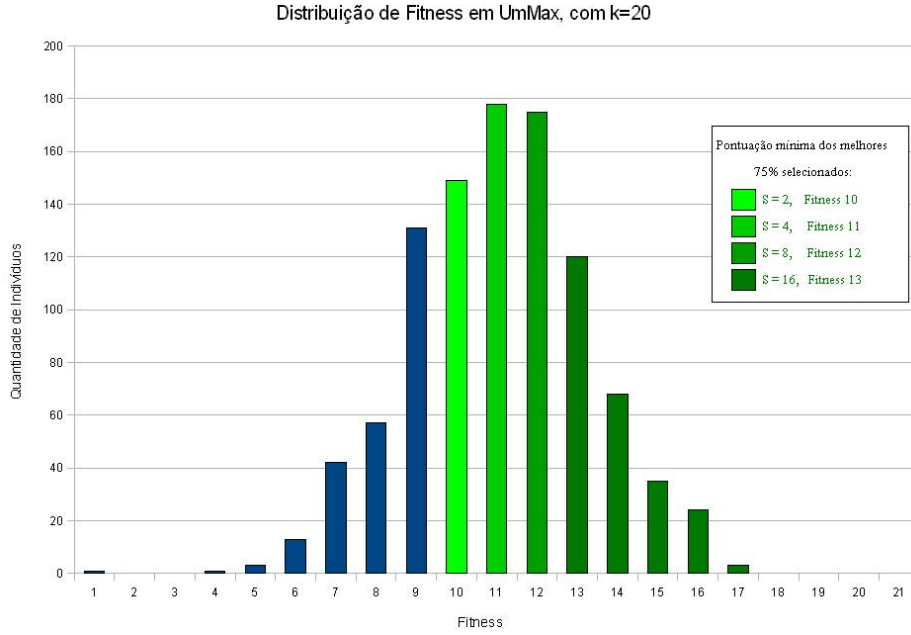


Figura 6.3: torneio

$$f_{t+1} - f_t = I\sigma_t, \quad (6.15)$$

em que  $I$  é a intensidade da pressão de seleção ( $I = f(s)$ ). Seu efeito é modelado como uma alteração de  $\sigma_t$ . Por exemplo, dobrar  $I$  é equivalente a uma distribuição com  $2\sigma_t$ .

Para completar a adaptação do Teorema de Fisher para seleção de truncamento, falta determinar  $f(s)$ . [?] obtiveram empiricamente (((ONFIRMAR!!!!!!!!!!!!!!!!!!!!))) a seguinte expressão para  $f(s)$ :

$$I = \sqrt{2(\ln s - \ln \sqrt{4, 14 \ln s})}. \quad (6.16)$$

Essa expressão é válida para seleção por torneio.

Conforme mostra-se a seguir, pode-se estimar o  $t_c$  para um problema sem precisar especificar  $f(s)$ . É fácil verificar que para o problema UmMax a Equação 6.15 resulta em:

$$p_{t+1} - p_t = I\sqrt{p_t(1 - p_t)}. \quad (6.17)$$

ERRO OU APROXIMAÇÃO RUIM: SUPOR QUE A MÉDIA E O DESVIO PADRÃO APÓS A PRIMEIRA GERAÇÃO CONTINUAM SENDO  $lp_t$  e  $p_t(1 - p_t)$  RESPECTIVAMENTE. A FUNÇÃO DISTRIBUIÇÃO DE PROBABILIDADE PROVAVELMENTE DEIXA DE SER UMA FUNÇÃO DISTRIBUIÇÃO DE PROBABILIDADE DE BERNOULLI OU MESMO UMA FUNÇÃO TIPO SINO. ISSO É FÁCIL DE VERIFICAR, VEJA A FUNÇÃO RESULTOU NA PARTE VERDE DA CURVA DA FIGURA EM QUE SE APRESENTA O CORTE À ESQUERDA NA GAUSSIANA FEITO PELA SELEÇÃO POR TRUNCAMENTO.

Fazendo a soma das diferenças de  $p_0$  a  $p_t$  e calculando a média, obtém-se a Equação (6.18).

É PRECISO EXPLICAR EM MAIS DETALHES ESSA DEDUÇÃO  
!!!!!!!!!!!!!!

$$p_t = \frac{1}{2}[1 + \sin(\omega t + \theta_0)], \quad (6.18)$$

em que  $\omega = \frac{I}{\sqrt{l}}$  e  $\theta_0 = \arcsin(1 - 2p_0)$ . Na convergência do GA,  $p_t = \frac{n-1}{n} = 1 - \frac{1}{n}$ . Assim,

$$1 - \frac{1}{n} = \frac{1}{2}[1 + \sin(\omega t + \theta_0)]$$

ou

$$1 - \frac{2}{n} = \sin(\omega t + \theta_0).$$

Para  $n$  não pequeno, o termo à esquerda desta equação é próximo de 1. Como  $\text{seno}(x) \approx x$  para  $x$  próximo de zero,  $1 - \text{seno}(x) = 1 - x$  para  $1 - x$  próximo de 1. Então, pode-se escrever:

$$1 - \left(1 - \frac{2}{n}\right) = 1 - \text{seno}(\omega t + \theta_0 + \pi k),$$

$$1 - \left(1 - \frac{2}{n}\right) = 1 - (\omega t + \theta_0 + \pi k),$$



em que  $k = 0, 1, 2, 3, \dots$ , então,

$$t = \frac{1 - \frac{2}{n} + \theta_0 + \pi k}{w}.$$

Substituindo  $w$ :

$$t = \left(1 - \frac{2}{n} + \theta_0 + \pi k\right) \frac{\sqrt{l}}{I}.$$

Supondo  $p_0 = 0,5$ , aproximação válida para uma população inicial gerada aleatoriamente, então  $\theta_0 = 0$  e, portanto,

$$t = \left(1 - \frac{2}{n} + \pi k\right) \frac{\sqrt{l}}{I}.$$

Considerando a variação de  $k$ , o menor  $t$  ocorre para  $k = 0$ , assim, obtém-se:

$$t_c = \left(1 - \frac{2}{n}\right) \frac{\sqrt{l}}{I}. \quad (6.19)$$

Dessa forma, para um GA com parâmetros especificados ( $I$  e  $n$  constantes),  $t_c = O(\sqrt{l})$ . Para seleção proporcional havia sido obtido  $O(l)$ . A convergência por seleção ordinal (como seleção por truncamento ou torneio) é  $O(\sqrt{l})$  mais lenta para o problema UmMax. Portanto, o GA pode-se significativamente mais eficiente que um algoritmo de busca local para um problema fácil como o UmMax.

## 6.4 Tempo de convergência para o problema BinInt

A análise anterior é útil para estimar o tempo de convergência em problemas com BBs com escala uniforme ou aproximadamente uniforme. Agora analisaremos problemas com escalas ruins entre BBs. Suponha o problema de maximizar uma função inteira.

j	Termo que domina
1	$2^0 x_1$
2	$2^1 x_1$
...	...
$\lambda$	$2^{\lambda-1} x_\lambda$

Tabela 6.1: Termos que dominam

$$f(x_j) = \sum_{j=1}^l 2^{j-1} x_j \quad (6.20)$$

O conhecimento a partir do *One Max Problem* nos levou a concluir que:

- Os genes convergiam uniformemente
- i) Uma única variável de proporção  $p_t$  seria suficiente para se conseguir determinar a intensidade da pressão de seleção.

Suponha, o que é chamado de domínio de convergência, os BBs mais salientes convergem primeiro. Seja  $\lambda_t$  a linha limite entre os *bits* configurados por convergência devido a pressão de seleção e os por convergência aleatória. Inicialmente a população é aleatória, determinada pela notação RRRRR, então  $\lambda=0$ . Para 1RRRR,  $\lambda = 1$

Para 11111,  $\lambda = l = 5$ .

Conforme  $\lambda$  aumenta, o termo que denomina a função  $f(x_j) = \sum_{j=1}^l 2^{j-1} x_j$  é o coeficiente de  $x_\lambda (j = \lambda)$ .

Usando  $\lambda$ , pode-se calcular o valor de *fitness* esperado, sendo 1 os valores dos loci que convergiram é 1/2 o valor dos loci que não convergiram, resultado na expressão para  $\bar{f}$

Pode ser mostrado que a variância é dada por:

$$\sigma^2 = \frac{(f_{max} - f_{min})^2}{4}$$

$f_{max}$  é obtido para  $\lambda = l$   $f_{max}$  para  $\lambda = 0$

Assim,

$$\sigma^2(\lambda) = \frac{2^{2(l-\lambda)}}{12}$$

A variância inicialmente é de todos os bits, conforme os mais salientes convergem com o crescimento de  $\lambda$ , a contribuição de sua variância vai para zero. Usando o Teorema de Fisher para seleção baseada em *rank*:

$$\bar{f}_{t+1} - \bar{f}_i = I\sigma_t$$

$$\bar{f}(\lambda) = 2^{l-1} (2 - 2^{-\lambda} - 2^{-l})$$

$$\sigma^2(\lambda) = \frac{2^{2(l-\lambda)} - 1}{12}$$

$$\sigma(\lambda) = \frac{2^{l-\lambda-1}}{\sqrt{3}}$$

$$\bar{f}_{t+1} - \bar{f}_t = I\sigma_t$$

$$2^{l-1} (2 - 2^{-\lambda_{t+1}} - 2^{-l}) - 2^{l-1} (2 - 2^{-\lambda_t} - 2^{-l}) = \frac{I}{\sqrt{3}} 2^{-\lambda_t-1}$$

$$2^{l-1} (2^{\lambda_t} - 2^{-\lambda_{t+1}}) = 2^{l-1} \frac{I}{\sqrt{3}} 2^{-\lambda_t}$$

Fazendo  $\mu = 2^{-\lambda}$ , obtém-se uma equação que diferença:

$$\mu_{t+1} - \mu_t = -c\mu_t$$

onde

$$c = \frac{I}{\sqrt{3}}$$

Resolvendo para  $\mu_t$ , obtém-se uma equação da diferença

$$\mu_{t+1} - \mu_t = -c\mu_t$$

onde

$$c = \frac{I}{\sqrt{3}}$$

Resolvendo para  $\mu_t$ , obtém-se  $\mu_t = \mu_0 (n - c)^t$

$\mu_0 = 2^0 = 1$ , assim,

$$2^{-\lambda t} = (1 - c)^t \rightarrow \ln(2^{-\lambda t}) = t \ln(1 - c)$$

$$t_c = \frac{-\lambda t \ln 2}{\ln(1 - c)} = \frac{l \ln 2}{\ln(1 - c)}$$

Usando a aproximação  $\ln(1 - x) \cong -x$ , para  $x$  pequeno, resulta na fórmula  $t_c = \frac{l \ln 2}{-c}$  como  $c = \frac{I}{\sqrt{3}}$  Obtém-se a equação do *Domino Model*:

$$t_c = -\frac{\sqrt{3} \ln 2}{I} l$$

Assim, para estratégias de seleção com  $I$  constante, o tempo de convergência para o *Bit Int Problem* é  $O(l)$ . Lembrando que para o *One Max Problem* o tempo é  $O(\sqrt{l})$ .

### Seleção proporcional

Como *bits* com escalas ruins entre si têm sinal no *fitness* muito pequeno, abaixo da média da população, essa estratégia não guia o AG para uma convergência continuada.

Usando:

$$f_{t+1} - f_t = \frac{\sigma_t^2}{\bar{f}_t}$$

e

$$\bar{f}(\lambda) = 2^{l-1} (2 - 2^{-\lambda} - 2^{-l})$$

Obtém-se:

$$2^{-\lambda t+1} - 2^{-\lambda t} = \frac{-2^{-2\lambda t}}{2 - 2^{-\lambda t}}$$

Fazendo  $\mu = 2^{-\lambda}$  e elevando ao limite, obtém-se:

$$\frac{3du}{dt} = \frac{-\mu^2}{2 - \mu}$$

Integrando e reconhecendo que  $\mu = 1$  com  $t_0 = 0$ .

$$t(\lambda) = 3(2^{\lambda+1} + \ln 2 - 2)$$

Desconsiderando os termos lineares e constantes, tem-se:

$$t_c = 62^l$$

Portanto para problemas com escalas ruins entre bits, a seleção proporcional leva a um tempo exponencial em  $l$ . Este, é mais um argumento forte contra a seleção proporcional.

## 6.5 Convergência e deriva genética no BinInt

Experimentos com o *Domino Model* mostra que BBs salientes convergem primeiro para o *Bin Int Problem*.

$l = 200$ ,  $n = 10, 20, 30, 40$  e  $60$ , seleção por torneio com reposição e  $S = 2$ .

O gráfico mostra que os genes mais salientes (posições baixas) convergem seguindo o *Domino Model* (quanto maior a contribuição do gene para o *fitness*, menor é o número de gerações até a convergência do gene). Os demais genes convergem aleatoriamente, (não esperam o tempo necessário para convergir corretamente). Isso sugere que estão convergindo via *drift*.

Solução: usar modelo combinado *Domino-Drift* Parece que há um locus crítico, a partir do qual o comportamento do algoritmo muda de convergência por causa da seleção para convergência devido à *drift*.

Normalmente, o locus crítico deve ocorrer quando  $t_c = t_d$ .

Para a seleção proporcional  $t_d = C_d n$  e  $t_c = 62^\lambda$ , assim:

$$\lambda = \frac{\ln C_d - \ln 6}{\ln 2} + \frac{\ln n}{\ln 2} \rightarrow \lambda_c = C_1 + C_2 \ln n$$

Para estratégia de seleção com  $I$  constante e com mesmo  $t_d$ :

$$t_c = C_c \lambda \text{ ou } t_c = C_c l,$$

onde,

$$C_c = \frac{-\ln 2}{\ln(1 - \frac{I}{\sqrt{3}})}$$

então,

$$\lambda_c = \frac{C_d}{C_d} n$$

Obs:

1) Na seleção proporcional, o crescimento de  $n$  aumenta pouco o número de bits que convergem por causa de seleção.

2) Na seleção baseada em *rank*, o crescimento da população aumenta a probabilidade de resolver corretamente o problema.

O coeficiente  $C_d$  varia muito com o tipo de *crossover* e valor de  $S$  utilizados.

## Capítulo 7

# Competição entre instâncias de um mesmo bloco construtivo

Até agora, procurou-se garantir um suprimento inicial suficientemente grande de BBs, para que os melhores Bbs conquistem uma parcela suficientemente grande do mercado. No entanto, sempre partimos do princípio que a função de seleção escolhe o esquema correto. Neste capítulo, procura-se verificar se os esquemas que podem encontrar as melhores soluções, são os que realmente são selecionados.

### Por que a tomada de decisão é um problema?

A princípio, parece que a decisão entre selecionar uma solução ou outra é simples. Suponha que queremos otimizar uma função determinística  $f(x)$ . Para cada solução  $x$ , a função retorna uma dada pontuação. Para selecionar a melhor solução, basta selecionar  $x$  que tenha a melhor pontuação. No entanto, sabemos que isto não é possível, a menos que nosso espaço de busca seja pequeno o suficiente para que nossa amostra contenha todas as soluções possíveis. O que buscamos selecionar a cada geração, são subestruturas de nossa solução, os Bbs. Isto torna a tomada de decisão de nossa seleção, um problema não trivial, como parecia a princípio.

Para entender melhor este problema, considere que a solução ótima a ser encontrada é dada pela *string* 11111. Considere também um esquema que

contenha essa *string*: 1####. O *fitness* deste esquema é dado pela média dos *fitness* de todas as nossas amostras que representadas por este esquema. É possível, que o *fitness* do esquema 0#### seja maior do que do esquema 1####. Isto faz com que o esquema selecionado para a próxima geração, não esteja conduzindo a solução ótima de nosso problema.

Nosso problema então, é o de seleção de esquemas. Diferente do problema de selecionar uma melhor solução, ao selecionar um melhor esquema, estamos olhando não para a seleção de valores pontuais, mas para a seleção de uma distribuição de valores.

## 7.1 Modelagem da ruína do jogador

A equação obtida com o método anterior possui uma limitação: Para encontra-la, foi admitido que a decisão de selecionar o melhor esquema é realizada uma vez. Na prática, sabemos que o AG continua fazendo este tipo de decisão a cada geração. Para representar essas decisões cumulativas, buscou-se encontrar uma equação baseando-se em um conhecido modelo de problema: Ruína do Jogador ( do inglês, *Gambler's Ruin*).

### Descrição do problema

Um jogador de azar, com uma quantidade inicial  $a$  de recursos, decide apostar para aumentar seus recursos até que uma quantidade desejada  $n$  seja alcançada. Cada aposta possui uma probabilidade  $p$  de vencer, e uma probabilidade  $q = 1-p$  de perder. Ao perder a aposta, a quantidade de recursos do jogador diminui em uma unidade, e ao vencer, a quantidade de recursos do jogador aumenta em uma unidade. O jogador para de apostar quando:

1. Sua quantidade de recursos chega a 0
2. Atinge a quantidade de recursos  $n$  desejada.

Queremos saber qual é a probabilidade  $P_n$  de que o jogador vença (segunda opção listada acima). A solução para este problema é conhecida, e dada pela



Equação (7.1)

$$P_n = \frac{1 - (q/p)^a}{1 - (q/p)^n} \quad (7.1)$$

Considerando nosso problema original, onde dois esquemas (o melhor e o segundo melhor) estão competindo para serem selecionados. Temos que a probabilidade  $p$ , de que o melhor esquema seja selecionado é dada pela Equação (7.2):

$$p = \mathbb{N}\left(\frac{d}{\sigma\sqrt{2m}}\right) \quad (7.2)$$

sendo,

- $\mathbb{N}$  a função de distribuição acumulada para uma distribuição normal com média 0.

Juntando as Equações (7.1) e (7.2), temos uma equação que define o tamanho da população que queremos utilizar. Uma forma mais interessante dessa função pode ser encontrada, quando percebemos que o denominador da função 10.3 tende a 1 mais rapidamente do que o observador, com esta simplificação de (7.1) e também levando em consideração que  $a = 2^k$  (estimativa do número de soluções em nossa população que possuem o valor correto do *building block*) encontramos a Equação (7.3).

$$P_n = 1 - \left(\frac{1-p}{p}\right)^{\frac{n}{2^k}} \quad (7.3)$$

Sendo

- $k$ : o tamanho do *building block*

Considerando  $\alpha = 1 - P_n$  e resolvendo para  $n$ , é dado a Equação (7.4).

$$n = -\ln(\alpha)2^k / \ln \frac{1-p}{p} \quad (7.4)$$

Usando a probabilidade  $p$  para os dois primeiros termos da série de potência para função de distribuição acumulada para a distribuição normal e reconhecendo que  $\ln 1 - x = -x$  para pequenos valores de  $x$ , chegamos a Equação (7.5).

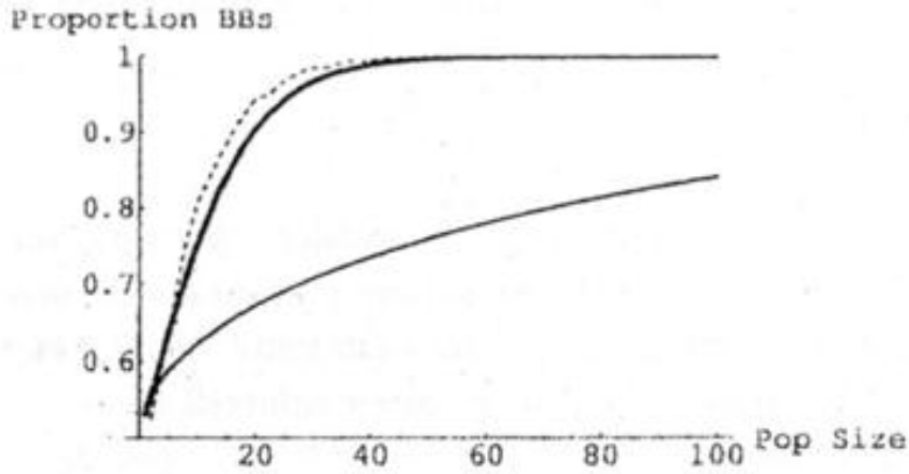


Figura 7.1: Comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo *generation-wise*. Soluções de tamanho 100.

$$n = -\ln(\alpha)2^{k-1}\frac{\sigma}{d}\sqrt{\pi m} \quad (7.5)$$

Sendo,

- m: o número de BBs ( $l/k$ )

### Verificação das equações

Os Gráficos das Figuras 7.1 e 7.2 ilustram uma comparação de resultados obtidos experimentalmente (linha pontilhada) representando a média de 100 testes e também com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo *generation-wise*. O experimento demonstrado pela Figura 7.1 utilizou soluções de tamanho 100 (100 bits), enquanto o da Figura ?? utilizou soluções de 500 *bits*.

Em ambos os experimentos, percebemos que as funções obtidas fornecem limitadores inferiores para o tamanho da população. Sendo que a função obtida pela ruína do jogador oferece uma melhor aproximação, como esperado. O mesmo experimento foi realizado utilizando um problema mais

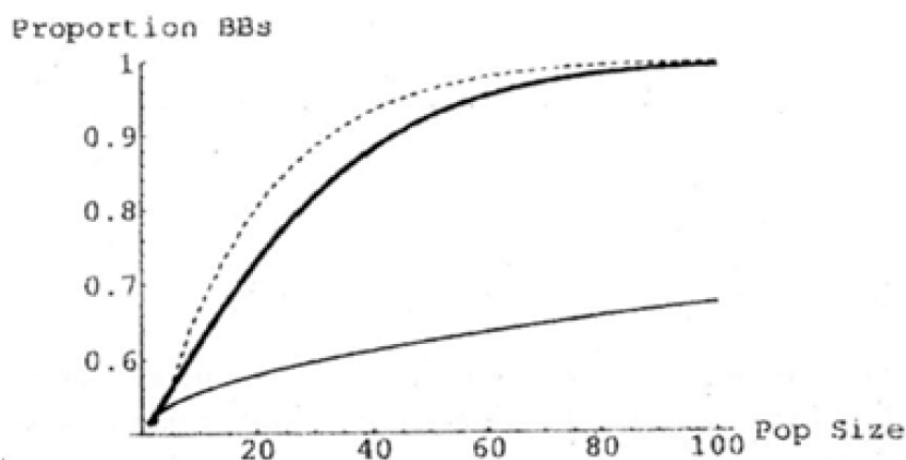


Figura 7.2: comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo *generation-wise*. Soluções de tamanho 500.

complexo, utilizando funções armadilha de tamanho 4 (Resultados no gráfico da Figura (7.3) e tamanho 8 (7.4).

Examinando os resultados das funções armadilha, percebemos que a equação obtida pelo modelo *generation-wise* não resulta em boas estimativas para valores de  $n$  pequeno. Isto se deve ao fato deste modelo levar em consideração apenas duas opções a serem escolhidas (melhor e segunda melhor). Isto faz com que o menor valor de proporção de BBs previsto por este modelo, seja 0.5.

## 7.2 Complexidade de um algoritmo genético competente

Com base nos resultados obtidos até este ponto, parece ser possível determinar a complexidade de tempo de computação de um AG para qualquer problema. Considere o modelo da ruína do jogador e também que:

- O sinal e a variância dos BBs permanecem constantes durante a evolução.

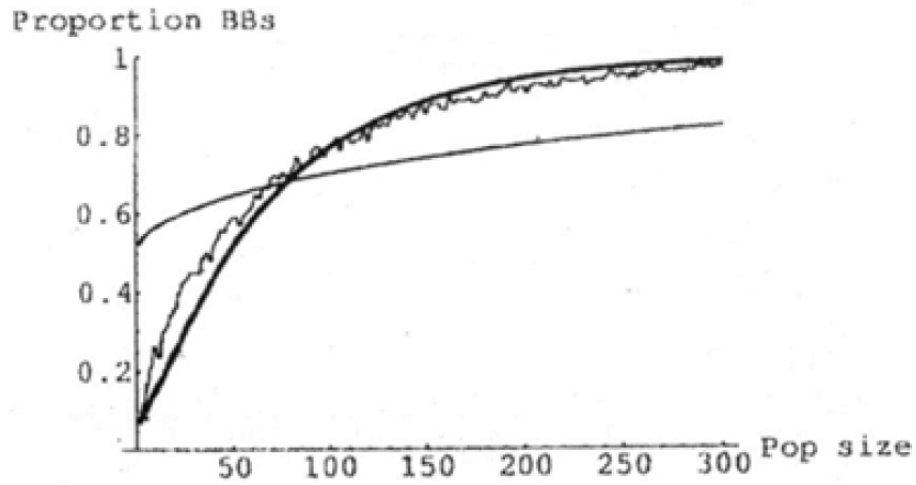


Figura 7.3: comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo *generation-wise*. 20 cópias de funções armadilha de tamanho 4.

- Supondo erro constante por BB ou erro em não mais que um número constante de BBs, seja,  $\alpha$  constante, ( $\alpha = c l^{-1}$ )

Assim como  $km = l, n = \ln(\alpha) 2^{2k-1} (\sigma/d) \sqrt{\pi m}$  Na Seção 1.3 segue que o número de gerações necessárias para o AG convergir é:

- $O(\sqrt{l})$  no caso do problema escalado uniformemente.
- $O(l)$  no problema com escalar entre BBs exponencial.

Multiplicando o número de gerações (Seção 1.3) pelo tamanho da população (Seção 1.4), obtem-se o número de avaliações necessárias para se convergir corretamente.

Da Seção 1.3:

- $O(\sqrt{l})$
- $O(l)$

Da Seção 1.4:

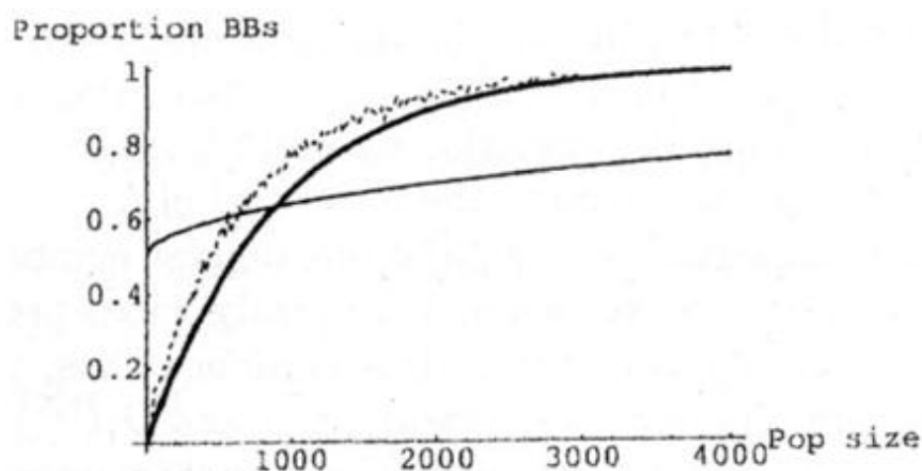


Figura 7.4: comparação de resultados obtidos experimentalmente (linha pontilhada) com resultados previstos usando a função obtida pelo modelo da ruína do jogador (linha mais espessa) e a função obtida pelo modelo *generation-wise*. 10 cópias de funções armadilha de tamanho 8.

- $O(\sqrt{l})$
- $O(\sqrt{l} \ln(l))$

Assim, a complexidade dos AGs seria subquadrática se todos os requisitos da decomposição em modelos pequenos forem respeitados. Vários destes requisitos foram vistos até este ponto, no entanto, ainda falta analisar a mistura dos BBs. Isto será feito na próxima seção.

## Capítulo 8

# Combinação de instâncias de blocos construtivos diferentes

Na seção anterior, temos que soluções precisas e confiáveis são encontradas controlando-se somente o tamanho da população ( $n$ ), que é um problema sub-quadrático ( $O(\sqrt{l})$ ) ou aproximadamente linear ( $O(l)$ ). Essa situação parece ocorrer tanto para problemas fáceis quanto para difíceis. No entanto, nos cálculos de  $n$  foi suposta a ligação correta dos genes. Na prática, essa suposição não é razoável, pois o usuário não tem idéia dos BBs até o AG começar a rodar.

O propósito geral de um AG é justamente a partir do problema como foi codificado, resolver o problema sem conhecimento prévio sobre o mesmo. Para resolver esse problema deve-se focar na combinação ou mistura dos BBs.

Com isso, é necessário construir os mapas de controle, no plano  $s \times p_c$ , que dão as fronteiras entre o sucesso e o fracasso, tanto para AGs da primeira geração, como também para projetos de AGs que escalam bem, independente do crescimento da dificuldade do problema ou do tamanho do problema.

### Aprendizagem das ligações

A tese De Jong em 1975, ao propor o AG simples, desprezou aspectos importantes que se tornaram cruciais para enfrentar problemas mais complexos. Holland, em 1975, já conhecia que:

- BB, com ligação forte tem vantagem relativa sobre aqueles com comprimento mais longos;
- Poderia ser necessário incluir operadores para adaptar as ligações e escolher as ligações de alelos

Essas idéias e a recomendação de Holland para o uso de um operador de "Inversão" são explorados na ultima seção.

### O teorema dos esquemas não é suficiente

Recordando a teoria do esquemas:  $s[1 - \epsilon p_c] \geq 1$ , onde:

- $s$  é o número de cópias associadas pela seleção para indivíduos contendo um bom esquema.
- $p_c$  é a probabilidade de crossover.
- $\epsilon$  é a taxa de perda de BBs.

Segundo esse teorema:

- Para fazer o melhor esquema crescer, pode-se usar  $s$  alto, para vencer as perdas, pelo melhor esquema, por cruzamento.
- Pode-se simplesmente usar  $p_c$  baixo para vencer o problema de quebra de BBs. Mas para  $\epsilon=1$ , fazer  $p_c$  baixo ou  $s$  alto, não ajuda muito, pois evita a mistura ou troca de BBs, com outros bons esquemas.

Isso é claramente indesejado, mas o raciocínio com base no Teorema dos Esquemas foca em um único BB, enquanto o sucesso de um AG (capaz de inovar pela recombinação dos melhores BBs) conta com a probabilidade de diferentes BBs unirem-se.

## 8.1 Mapa de controle

A idéia básica é colocar num plano  $s \times p_c$  a região de alta-qualidade da performance de um AG, assumindo que é possível estimar uma série de fenômenos em relação ao tempo como: tempo de convergência sobre seleção, sobre *drift* e tempo de inovação sobre crossover.

### Fronteira de mistura

Na construção do mapa de controle, primeiro deve-se determinar a fronteira de mistura ou fronteira de inovação. A capacidade de mistura de BBs é perdida se o tempo de seleção é menor que o tempo de mistura, assim não há tempo para que os BBs se recombinem.

$$t_s < t_x$$

onde,

- $t_s$  é o tempo de convergência devido a seleção.
- $t_x$  é o tempo de mistura ou inovação, causando convergência prematura.

Quando o tempo de seleção é maior que o tempo de mistura:  $t_s > t_x$ , pode-se atingir o estado de inovação permanente, pois:

- Novas descobertas entraram no sistema antes da convergência da população;
- As descobertas serão elevadas pela seleção induzindo mais e melhores descobertas;
- Esse processo irá continuar.

Utilizando  $t_s = \frac{\ln(n)}{\ln(s)}$  e sabendo que o número esperado de novos e melhores indivíduos em uma dada geração é igual ao produto do tamanho da população ( $n$ ), a probabilidade de crossover ( $p_c$ ) e a probabilidade de sucesso de mistura ( $p_x = 1 - \epsilon$ ) Assim;

$$t_x = \frac{1}{np_cp_x} \quad (8.1)$$



Um AG selectorecombinativo irá executar bem quando o numero de inovações  $I = \frac{t_s}{t_x}$  for maior que 1, assim tem-se os limites de máximo e mínimo para  $p_c$ :

$$I = \frac{t_s}{t_x} = \frac{\frac{\ln(n)}{\ln(s)}}{\frac{1}{np_c p_x}} > 1 \rightarrow p_c > \frac{\ln(s)}{p_x n \ln(n)}$$

sendo,  $p_c$  mínimo.

$$s[1 - p_c \epsilon] \geq 1 \rightarrow p_c = \frac{1 - s^{-1}}{\epsilon}$$

sendo,  $p_c$  máximo.

## 8.2 Fronteira de deriva genética

A fronteira de *drift* aparece quando há convergência para alelos quando pouca ou nenhuma pressão de seleção existe. Neste caso, a razão crítica é dada por:

$$D = \frac{t_s}{t_d}$$

$$t_d = C_d * n$$

$$D = \frac{\frac{\ln(n)}{\ln(s)}}{C_d * n}$$

Nominalmente, o *drift* ocorre se  $D > 1$ . Deseja-se então  $D \leq 1$ .

$$\frac{\ln(n)}{C_d * n \ln(s)} \leq 1 \rightarrow \ln(s) \geq \frac{\ln(n)}{C_d * n}$$

Ou seja, existe uma pressão de seleção mínima para não ocorrer o *drift*.

Assim, no plano  $s$  x  $p_c$  o *drift* ocorre para algum valor abaixo da pressão de seleção mínima, que é independente de  $p_c$ .

$$\ln(s) \geq \frac{\ln(n)}{C_d * n} \text{ mnimo}$$

Resultando em uma reta vertical no valor mínimo no plano  $p_c$  x  $s$

### 8.3 Fronteira de competição cruzada

Na seção anterior obtemos que a condição de inovação define uma fronteira de probabilidade de crossover mínima, que cresce com o logaritmo da pressão de seleção  $s$  ou linearmente com a intensidade da pressão de seleção  $\ln(s)$ .

$$p_c > \frac{\ln(s)}{p_x n \ln(n)}$$

Se aumenta a população reduz o crossover.

Supondo  $p_x$  independente da pressão de seleção a partir do qual o AG é indiferente. Por exemplo, em seleção por truncamento ( $\frac{1}{s}$  melhores indivíduos receberão  $S$  cópias), se  $S = n$ , o melhor indivíduo ranqueado receberá  $n$  cópias e os demais nenhuma. A menos que esse indivíduo seja ótimo, a probabilidade dele carregar os BBs necessários é zero. Para casos em que  $S$  é próximo de  $n$ , essa probabilidade é próxima de zero.

Como nem todos os BBs estarão ou poderão não estar, sendo representados na próxima geração, isso significa que os BBs de uma posição estão competindo com BBs de outras posições. Assim, é importante manter a pressão de seleção baixa para que “BBs de maçãs não compitam com BBs de laranjas”.

Suponha a seleção por truncamento ou torneio, após a primeira geração, há  $\frac{n}{s}$  indivíduos independentes, a partir dos quais pode-se construir indivíduos melhores.

Suponha uma proporção  $p_0$  de posições de *bits* fixadas corretamente na população, então há  $[1 - (1 - p_0)^{\frac{n}{s}}] * l$  alelos cobertos pelos  $\frac{n}{s}$  *strings* independentes. Falhas ocorrem quando  $\frac{\alpha}{l}$  *bits* corretos são perdidos quando:

$$(1 - \alpha)l < [1 - (1 - p_0)^{\frac{n}{s}}]l \rightarrow \alpha > (1 - p_0)^{\frac{n}{s}} \rightarrow \alpha > \frac{n}{s} \ln(1 - p_0)$$

$$s > \frac{n \ln(1 - p_0)}{\ln \alpha}$$

sendo  $s$  máximo.

Portanto, a falha por competição cruzada é esperada, quando a pressão de seleção ( $s$ ) é uma fração significativa do tamanho da população ( $n$ ). No plano  $p_cxs$ , isso significa que  $s$  tem um limite superior independente de  $p_c$ , uma reta vertical à direita do plano.

A Figura 8.1 mostra um mapa de controle na teoria, com suas fronteiras definidas como explicado, e ao lado apresenta um mapa para um experimento do *One Max Problem*, nos resultados experimentais não foi possível determinar uma relação linear entre a probabilidade de crossover e pressão de seleção, mas foi possível estabelecer uma região de sucesso.

## 8.4 Tamanho da região de sucesso em problemas difíceis

O mapa de controle apresenta uma região de sucesso (em inglês, *Sweet Spot*) grande que garante o sucesso do AG. Considerando suposições anteriores, o tempo de convergência do AG no pior caso seria sub-quadrático  $t_c = O(l)$  (para BBs escaláveis linearmente e seleção por *rank*) e tamanho de população para decisão correta sobre BBs era da ordem de  $O(\sqrt{l} \ln l)$ .

Assim, o AG seria  $O(l^{1.5} \log l)$ , com esse tempo e uma boa faixa de valores para  $p_c$  e  $s$ , poderiam ser resolvidos quaisquer problemas com AG simples. A questão é saber se o *Sweet Spot* mantém-se grande para problemas difíceis, lembrando que os resultados experimentais foram para o *One Max Problem*. Os resultados experimentais comprovam que não é possível.

O *One Max Problem* é *bitwise* otimizável, BBs de  $k=1$ ! A chance de distribuir um BB por causa do crossover zero. Para problemas *bitwise* otimizável, o Teorema dos Esquemas não restringe o *Sweet Spot* do AG. Conforme o problema fica mais difícil com o BBs maiores, o Teorema tem efeito, o *Sweet Spot* encolhe! A Figura 8.2 mostra os resultados para  $k = 4$  e  $m = 2$ ,  $n = 600$  e  $l = 8$ .

(Para  $s < 2$  não foi calculado, não se determinou a fronteira de *drift*). Observe que para um problema minúsculo ( $l = 8$  e  $m = 2$ ) é preciso uma população relativamente grande ( $n = 600$ ). Esse é um comportamento que já

tinhamos verificado com um modelo pequeno:

$$n = \chi^k(k \ln(\chi) + \ln(m))$$

, a população requer tamanho exponencial em relação a  $k$  devido ao efeito da mistura. De fato, o estudo mostrado em (Thierens e Goldberg, 1993) revela que:

$$t_x = \frac{2^{\mu*k}}{n * p_c}$$

Já tínhamos usado que:

$$t_x = \frac{1}{n * p_c * p_x}$$

Aumentar  $k$  ou  $m$  complica o problema Usando

$$t_s = \frac{\ln(n)}{\ln(s)}$$

e fazendo  $I = \frac{t_s}{t_x} = 1$

$$\frac{\ln(n)}{\ln(s)} = \frac{2^{\mu*k}}{n * p_c} \rightarrow n \ln(n) = 2^{\mu*k} * \frac{\ln(s)}{p_c} \rightarrow \ln(n \ln(n)) = \mu k n(2) k * \frac{\ln(s)}{p_c}$$

Experimentalmente, obtém-se os resultados mostrados na Figura 8.3 para  $m = 2$ ,  $s = 4$  e  $p_c = 0,5$  e obtém-se  $\mu = 1,6$  (Thierens e Goldberg, 1993).

Para mais de 2 BBs, obtém-se o seguinte resultado experimental mostrados na Figura 8.4,  $n$  é a população mínima para convergir 49 e 50 execuções.

## 8.5 Múltiplos blocos construtivos

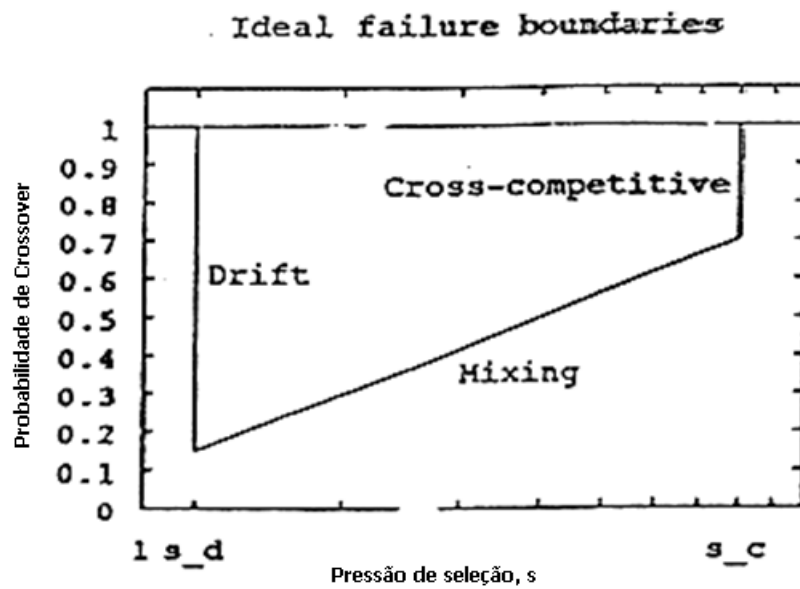
. O estudo experimental de (Thierens e Goldberg, 1993) para mais de 2 BBs, mostra que o tempo de mistura é dado pela Equação (8.2).

$$t_x - C * \frac{2^{m+\mu*k}}{p_c * m^{\frac{s}{2}}} \quad (8.2)$$

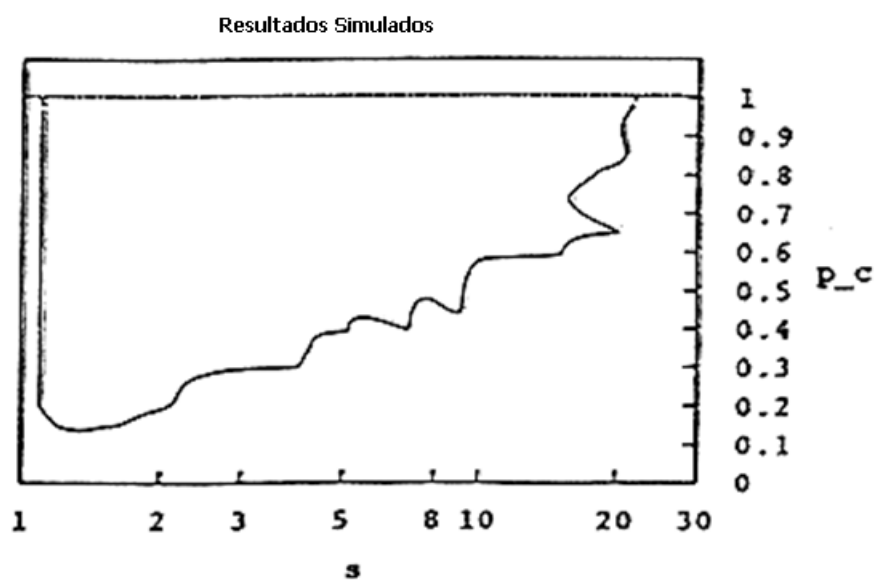
Para  $I = \frac{t_s}{t_x} = 1$  então, obtém-se a Equação (8.3)

$$n \ln(n) = c * \frac{2^{m+\mu*k} * \ln(s)}{p_c * m^{\frac{s}{2}}} \quad (8.3)$$

Assim, a população cresce exponencialmente com  $k$  e  $m$ . O mapa de controle obtido com base em  $p_c, s, n, k$  para tomada de decisão garantir que os BBs correlatos sejam escolhidos ficará infactível para  $k$  e  $m$  relativamente pequenos. Para garantir boas soluções, tamanhos de populações exponencialmente grandes devem ser escolhidos. Obs: Ou aumenta  $n$  exponencialmente ou determina os BBs de problemas difíceis ( $k$  e  $m$  não pequenos), pois para não quebrar BBs,  $s$  tem que ser tão pequeno que ocorra *drift*.



(a) Probabilidade de Crossover X Pressão de Seleção.



(b) Resultados Simulados.

Figura 8.1: Distribuição de *fitness* em diferentes BBs.

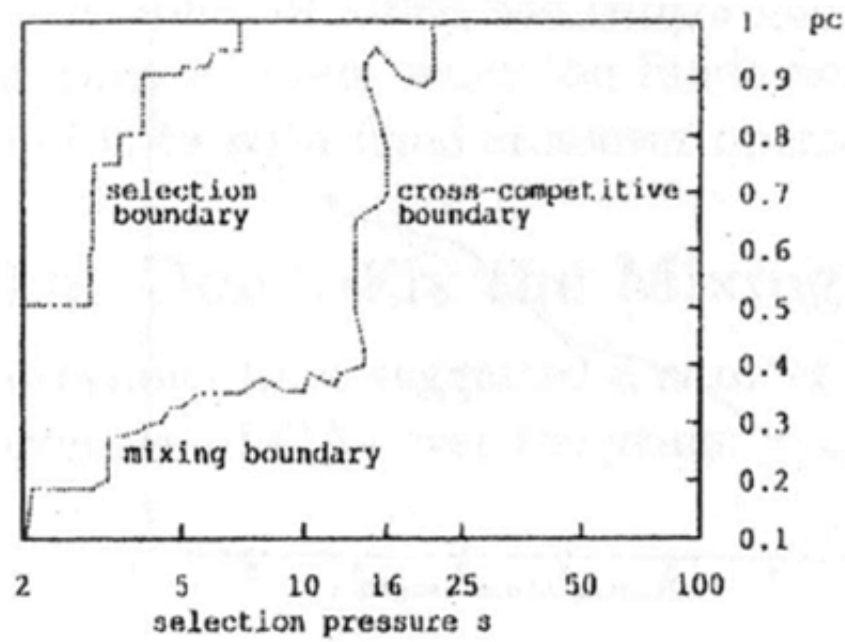


Figura 8.2: Pressão de Seleção

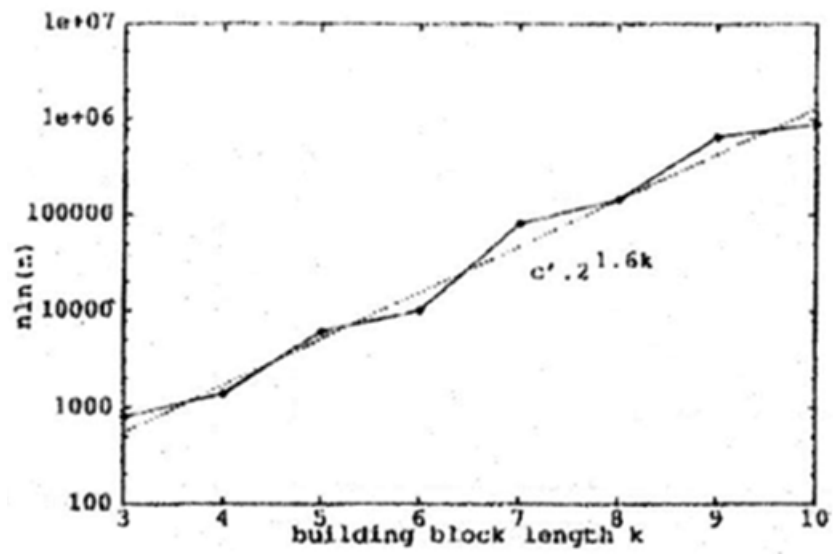


Figura 8.3: Tamanho BB X  $n \ln(n)$

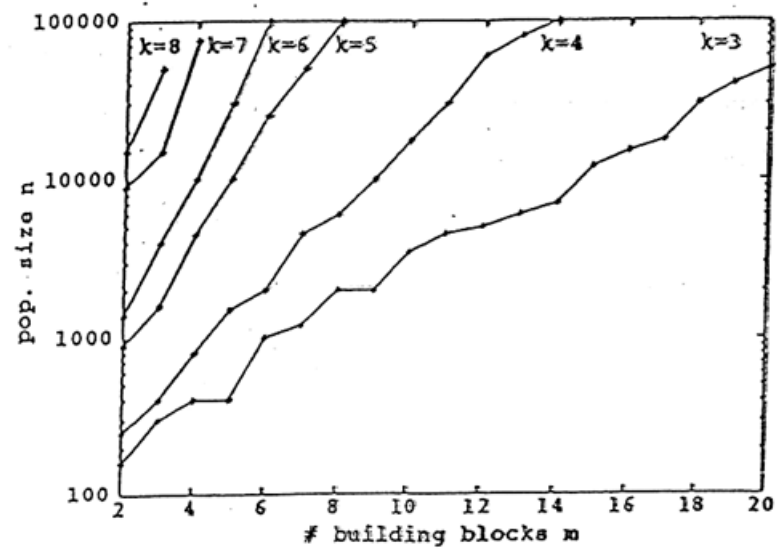


Figura 8.4: BBs x Tamanho da população