

Universidade de São Paulo
Instituto de Ciências Matemáticas e Computação

Arturo Miranda Vera
Valéria de Carvalho Santos

**Previsão da qualidade de convergência de algoritmos
genéticos com base no tamanho da população**

Trabalho apresentado à disciplina
Projeto de Inovação de Algoritmos
Genéticos do Curso de Pós-Graduação
do Instituto de Ciências Matemáticas e
Computação da Universidade de São
Paulo.

Professor Alexandre Delbem

São Carlos, dezembro de 2012.

Previsão da qualidade de convergência de algoritmos genéticos com base no tamanho da população

1. Fornecimento inicial de BBs no algoritmo genético

Antes da seleção e recombinação poder atuar sobre os BBs, o algoritmo genético (AG) deve ter um fornecimento adequado de BBs ótimos. Quando os BBs ótimos são abundantes, é provável que o AG irá escolhê-lo e combiná-los corretamente; por outro lado, quando os BBs ótimos são escassos, as chances de o AG convergir para uma boa solução é pequena.

Considera-se o fornecimento do número de BBs ótimos presentes na população inicial aleatória de um algoritmo genético. A probabilidade que um único BB ótimo de tamanho k é gerado aleatoriamente é $1/2^k$, para domínios binários, e, portanto, o fornecimento inicial de BBs ótimos para tamanho da população n pode ser estimado como:

$$x_0 = \frac{n}{2^k} \quad (1)$$

2. Decisão entre dois BBs

O papel da seleção em AGs é decidir quais os indivíduos sobrevivem para formar a próxima geração. O mecanismo de seleção é suposto escolher esses indivíduos com os BBs ótimos ou corretos e eliminar os outros, mas às vezes os indivíduos errados são escolhidos.

Para entender por que isso pode ocorrer, nesta seção é apresentada a análise dos cálculos por Goldberg et al. (1992), da probabilidade de decidir bem entre um indivíduo com o melhor BB e outro indivíduo com o segundo melhor BB. A ideia é concentrar-se em uma partição e considerar as contribuições de fitness de outras partições, como o ruído que interfere no processo de decisão.

Considere-se uma competição entre um indivíduo i_1 que contém o BB ótimo em um bloco H_1 , e um indivíduo i_2 com o segundo melhor BB H_2 como se ilustra na Figura 1. Uma escolha ideal seria que o mecanismo de seleção deveria escolher i_1 , mas existe a possibilidade de escolher

erroneamente i_2 devido à contribuição das outras partições para o fitness de i_2 que pode ser maior que o fitness de i_1 .

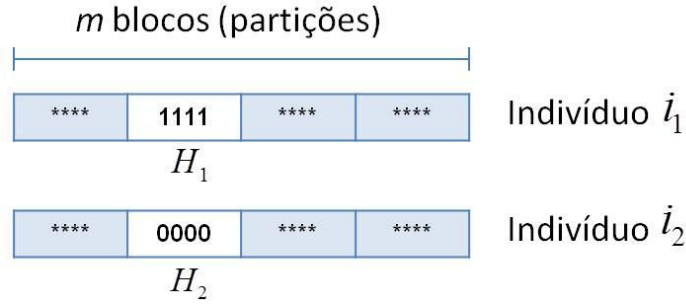
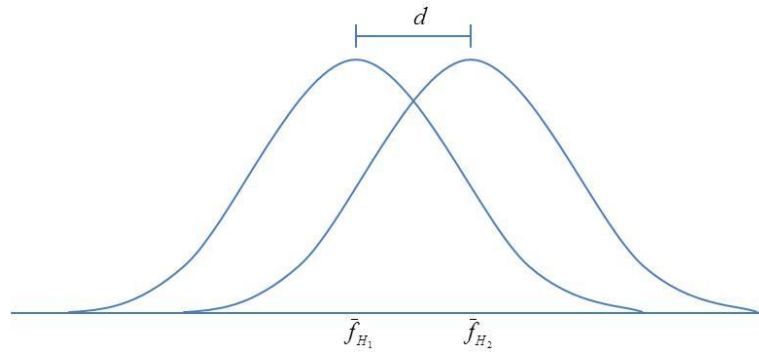


Figura 1: Competição de dois blocos de construção, $k = 4$.

A probabilidade de decidir corretamente entre esses dois indivíduos é a probabilidade de que o fitness do indivíduo i_1 (f_1) é maior do que o fitness do indivíduo i_2 (f_2) ou equivalente à probabilidade de que $f_1 - f_2 > 0$.

Na Figura 2, ilustra-se a distribuição dos fitness dos indivíduos contendo H_1 e H_2 . A *distância* entre a média do fitness dos indivíduos com H_1 ($\overline{f_{H_1}}$) e a média do fitness dos indivíduos com H_2 ($\overline{f_{H_2}}$) é denotado por d . Sabe-se que o fitness é uma função aditiva das contribuições dos fitness de todas as partições no problema, podem-se considerar as distribuições de f_1 e f_2 ser normais pelo teorema central do limite.



A Figura 2: As distribuições de fitness dos indivíduos contendo H_1 e H_2 .

Desde que as distribuições de fitness de f_1 e f_2 são normais,

$$f_1 \sim N(\overline{f_{H_1}}, \sigma_{H_1}^2)$$

$$f_2 \sim N(\overline{f_{H_2}}, \sigma_{H_2}^2)$$

a distribuição de $f_1 - f_2$ é normal e tem propriedades conhecidos:

$$f_1 - f_2 \sim N(\overline{f_{H_1}} - \overline{f_{H_2}}, \sigma_{H_1}^2 + \sigma_{H_2}^2) \quad (2)$$

Substituindo $d = \overline{f_{H_1}} - \overline{f_{H_2}}$ na Equação 2 e normalizando, a probabilidade de tomar a decisão correta em um único experimento é:

$$p = \Phi\left(\frac{d}{\sqrt{\sigma_{H_1}^2 + \sigma_{H_2}^2}}\right) \quad (3)$$

onde Φ é a função de distribuição acumulada de uma distribuição normal.

Para calcular $\sigma_{H_1}^2$ e $\sigma_{H_2}^2$, os fitness da função F é a soma das m subfunções independentes F_i , cada uma do mesmo tamanho k . Portanto, a variância total de fitness é calculada como:

$$\sigma_F^2 = \sum_{i=1}^m \sigma_{F_i}^2$$

Portanto, a variância do BB denotado por σ_{bb}^2 é $\sigma_{F_i}^2$. Neste caso, o ruído total proveniente das $m' = m - 1$ partições que não estão competindo diretamente é $\sigma^2 = m' \sigma_{bb}^2$.

Substituindo a variância na Equação 3.

$$p = \Phi\left(\frac{d}{\sqrt{m' \sigma_{bb}^2 + m' \sigma_{bb}^2}}\right)$$

$$p = \Phi\left(\frac{d}{\sqrt{2m' \sigma_{bb}^2}}\right)$$

Da expressão acima, a probabilidade de fazer a escolha certa em um único experimento em um problema com m partições independentes e iguais em escala torna-se:

$$p = \Phi\left(\frac{d}{\sqrt{2m' \sigma_{bb}^2}}\right) \quad (4)$$

a. Modelo de ruína do jogador

No problema da ruína do jogador, o capital de um jogador é representado pela posição x de uma partícula em um espaço unidimensional, como ilustrado na Figura 3. Inicialmente a partícula está posicionada em $x_0 = a$, onde a representa o capital inicial do jogador. O jogador joga contra um adversário que tem um capital inicial de $n - a$.

Existem limites de absorção em $x = 0$, que representa insolvência e em $x = n$, que representa ganhar todo o dinheiro do adversário. Em cada passo do jogo, o jogador tem uma chance p de incrementar seu capital por uma unidade e a probabilidade $q = 1 - p$ de perda de uma unidade. O objetivo do jogo é chegar ao limite em $x = n$ e a probabilidade de sucesso depende do capital inicial e da probabilidade de ganhar um experimento particular.



Figura 3: O problema da ruína do jogador em um espaço unidimensional limitada.

A analogia entre a seleção em AGs e o problema da ruína do jogador aparece naturalmente se assumirmos que as partições são independentes e nos concentrarmos em apenas um deles. A posição da partícula no espaço unidimensional, x , representa o número de cópias dos BBs ótimos na população. As barreiras absorventes em $x = 0$ e $x = n$ representam a convergência para as soluções erradas e corretas, respectivamente. A posição inicial da partícula x_0 é o número esperado de cópias do BB numa população aleatoriamente inicializada, que em um domínio binário, e considerando BBs de ordem k , é $x_0 = \frac{n}{2^k}$, como já foi mencionado na seção 1 (Equação 1).

Da abordagem acima, para usar problema da ruína do jogador para prever a qualidade das soluções do AG, supõe-se que o modelo da ruína do jogador considera que as decisões em um AG ocorrem uma de cada vez até que todos os n indivíduos na sua população converjam para o mesmo valor. Em outras palavras, no modelo não há noção explícita de gerações, e o resultado de cada decisão é ganhar ou perder uma cópia do BB ótima.

Assumindo conservadoramente que todas as competições ocorrem entre as strings que representam o melhor BB e o segundo melhor BB em uma partição, a probabilidade de ganhar uma cópia do BB global é dada pela probabilidade p de tomada de decisão correta, que foi calculada na seção anterior (ver Equação 4).

Como foi discutido acima, o AG tem sucesso quando existem n cópias do BB correto na partição de interesse. Um resultado bem conhecido na literatura de passeio aleatório é que a probabilidade de que a partícula irá eventualmente ser capturada pela barreira absorvente em $x = n$ é:

$$P_{bb} = \frac{1 - \left(\frac{q}{p}\right)^{x_0}}{1 - \left(\frac{q}{p}\right)^n} \quad (5)$$

Onde $q = 1 - p$ é a probabilidade de perder uma cópia do BB em uma competição específica (Feller, 1966).

A partir da Equação 5, pode-se encontrar uma expressão para o tamanho da população. Observa-se que $p > 1 - p$ (porque a média de fitness do melhor BB é maior do que a média de fitness do segundo melhor), e x_0 é geralmente pequena em comparação ao tamanho da população. Assim, quando o valor de n cresce, o denominador na Equação 5 se aproxima de 1 muito rapidamente e pode ser ignorado nos cálculos. Substituindo o fornecimento inicial de BBs ($x_0 = n/2^k$), P_{bb} pode ser aproximada como:

$$P_{bb} = 1 - \left(\frac{1-p}{p}\right)^{n/2^k} \quad (5)$$

Uma vez que assumimos que os BBs são independentes uns dos outros, o número esperado (média, esperança) de partições com o BB ótimo no final de uma execução é $E(BBs) = mP_{bb} = \hat{Q}$. Se assumirmos que estamos interessados em encontrar uma solução com uma média de \hat{Q} BBs ótimos, podemos resolver $P_{bb} = \frac{\hat{Q}}{m}$ para obter n o seguinte equação de dimensionamento da população:

Da Equação 5 se tem

$$\left(\frac{1-p}{p}\right)^{n/2^k} = 1 - P_{bb}$$

Aplicando logaritmo:

$$\frac{n}{2^k} \ln\left(\frac{1-p}{p}\right) = \ln(1 - P_{bb})$$

$$n = \frac{2^k \ln(1 - P_{bb})}{\ln\left(\frac{1-p}{p}\right)} \quad (6)$$

Substituindo P_{bb} por $\frac{\hat{Q}}{m}$

$$n = \frac{2^k \ln\left(1 - \frac{\hat{Q}}{m}\right)}{\ln\left(\frac{1-p}{p}\right)}$$

$$n = \frac{2^k \ln(\alpha)}{\ln\left(\frac{1-p}{p}\right)}$$

(7)

Onde $\alpha = 1 - \frac{\hat{Q}}{m}$ é a probabilidade de fracasso do AG.

Para observar mais claramente as relações entre o tamanho da população e as variáveis dependentes do domínio envolvidos, podemos expandir p e escrever a Equação 7 em termos da distância (d), do ruído (σ_{bb}), e do número de partições do problema (m).

Primeiro, aproxima-se p (Equação 4) usando os dois primeiros termos da expansão em série de potências para a distribuição normal (Abramowitz and Stegun, 1972) como:

$$p = \Phi\left(\frac{d}{\sqrt{2m'\sigma_{bb}}}\right)$$

$$p = \frac{1}{2} + \frac{1}{2} \frac{d}{\sqrt{2m'\sigma_{bb}}}$$

Substituindo esta aproximação para p na Equação 7, tem-se

$$n = \frac{2^k \ln(\alpha)}{\ln\left(\frac{1 - \frac{1}{2} - \frac{1}{2} \frac{d}{\sqrt{2m'\sigma_{bb}}}}{\frac{1}{2} + \frac{1}{2} \frac{d}{\sqrt{2m'\sigma_{bb}}}}\right)}$$

$$n = \frac{2^k \ln(\alpha)}{\ln\left(\frac{\frac{1}{2} - \frac{1}{2} \frac{d}{\sqrt{2m'\sigma_{bb}}}}{\frac{1}{2} + \frac{1}{2} \frac{d}{\sqrt{2m'\sigma_{bb}}}}\right)}$$

$$n = \frac{2^k \ln(\alpha)}{\ln\left(\frac{\frac{1}{2} \left(1 - \frac{d}{\sqrt{2m'\sigma_{bb}}}\right)}{\frac{1}{2} \left(1 + \frac{d}{\sqrt{2m'\sigma_{bb}}}\right)}\right)}$$

$$n = \frac{2^k \ln(\alpha)}{\ln\left(\frac{1 - \frac{d}{\sqrt{2m'}\sigma_{bb}}}{1 + \frac{d}{\sqrt{2m'}\sigma_{bb}}}\right)}$$

Como $\frac{d}{\sqrt{2m'}\sigma_{bb}}$ tende a ser um número pequeno, $\ln\left(1 \pm \frac{d}{\sqrt{2m'}\sigma_{bb}}\right)$ pode ser aproximado como $\pm \frac{d}{\sqrt{2m'}\sigma_{bb}}$. Portanto, da expressão acima temos:

$$n = \frac{2^k \ln(\alpha)}{\ln(1 - \frac{d}{\sqrt{2m'}\sigma_{bb}}) - \ln(1 + \frac{d}{\sqrt{2m'}\sigma_{bb}})}$$

$$n = \frac{2^k \ln(\alpha)}{-\frac{d}{\sqrt{2m'}\sigma_{bb}} - \frac{d}{\sqrt{2m'}\sigma_{bb}}}$$

$$n = \frac{2^k \ln(\alpha)}{-2 \frac{d}{\sqrt{2m'}\sigma_{bb}}}$$

$$n = -\frac{\ln(\alpha)2^{k-1}\sigma_{bb}\sqrt{2m'}}{d}$$

3. Experimentos

Esta seção verifica o modelo da ruína do jogador para a qualidade de soluções encontradas por um GA simples. Os problemas tratados aqui são para as funções OneMax e BinInt.

Os tamanhos da população necessários para resolver os problemas de teste variam de 2 a 100.

Os resultados deste trabalho são a média de 100 execuções independentes de um AG simples. O AG usa a seleção do torneio de pares sem substituição. O operador de crossover foi escolhido. Em todos os experimentos, a probabilidade de mutação é definida como zero, porque a abordagem considera que a única fonte de diversidade é a população inicial aleatória. Ou seja, a mutação tornaria possível afastar dos limites de absorção do passeio aleatório. Cada execução foi encerrada quando a população tinha convergido completamente, ou seja, quando todos os indivíduos possuíam o mesmo fitness. Isso é

possível, porque a taxa de mutação é zero. Os resultados apresentam a porcentagem de partições que convergiram para o valor ótimo.

As previsões teóricas da abordagem da ruína do jogador foram calculadas usando a Equação 5.

a. OneMax

O experimento da função OneMax considera strings com 100 bits, de modo que $m = 100$, a ordem dos BBs é $k = 1$.

Como a população foi inicializada aleatoriamente, tem-se em média 50% dos BBs ótimos.

A diferença de fitness ou a distância é:

$$d = f_{max} - f_{min} = 1 - 0 = 1$$

onde f_{max} é máximo valor de bloco e f_{min} é o valor mínimo.

Considerando que metade das strings possui valor f_{min} e a outra metade possui f_{max} , a variância pode ser calculada como (Goldberg et al., 1992):

$$\sigma_{bb}^2 = \frac{(f_{max} - f_{min})^2}{4} = \frac{(1 - 0)^2}{4}$$

$$\sigma_{bb}^2 = 0,25$$

Assim, o desvio é $\sigma_{bb} = 0,5$

Substituindo esse valores na Equação 4, determinamos a probabilidade de escolher corretamente entre dois BBs que competem:

$$p = N\left(\frac{d}{\sqrt{2m} \sigma_{bb}}\right) = N\left(\frac{1}{\sqrt{2 * 99} * 0,5}\right) = N(0,1421338)$$

O valor de p pode ser obtido pela tabela de distribuição normal. Usando interpolação, tem-se:

$$0,14 \rightarrow 0,5557$$

$$0,15 \rightarrow 0,5596$$

$$0,1421338 \rightarrow p$$

$$\frac{0,14 - 0,15}{0,14 - 0,1421338} = \frac{0,5557 - 0,5596}{0,5557 - p}$$

$$p = 0,55653$$

Calculando a proporção dos BBs ótimos pela Equação 5:

$$P_{bb} = \frac{1 - \left(\frac{q}{p}\right)^a}{1 - \left(\frac{q}{p}\right)^n} = \frac{1 - \left(\frac{1-p}{p}\right)^{n/2^k}}{1 - \left(\frac{1-p}{p}\right)^n} = \frac{1 - \left(\frac{1-0,55653}{0,55653}\right)^{n/2^1}}{1 - \left(\frac{1-0,55653}{0,55653}\right)^n}$$

Uma vez que o comprimento dos BBs é um ($k = 1$), o crossover não pode interrompê-los, sendo assim utilizado o crossover uniforme para esta função. A probabilidade de troca de cada bit foi definida para 0,5.

Na Figura 4 são ilustrados os resultados do experimento. A curva em cor vermelha representa a previsão teórica do modelo de ruína do jogador (Equação 5) e a curva em cor azul são os resultados experimentais para uma função OneMax de 100 bits. A curva experimental está bem próxima da teórica, validando o modelo.

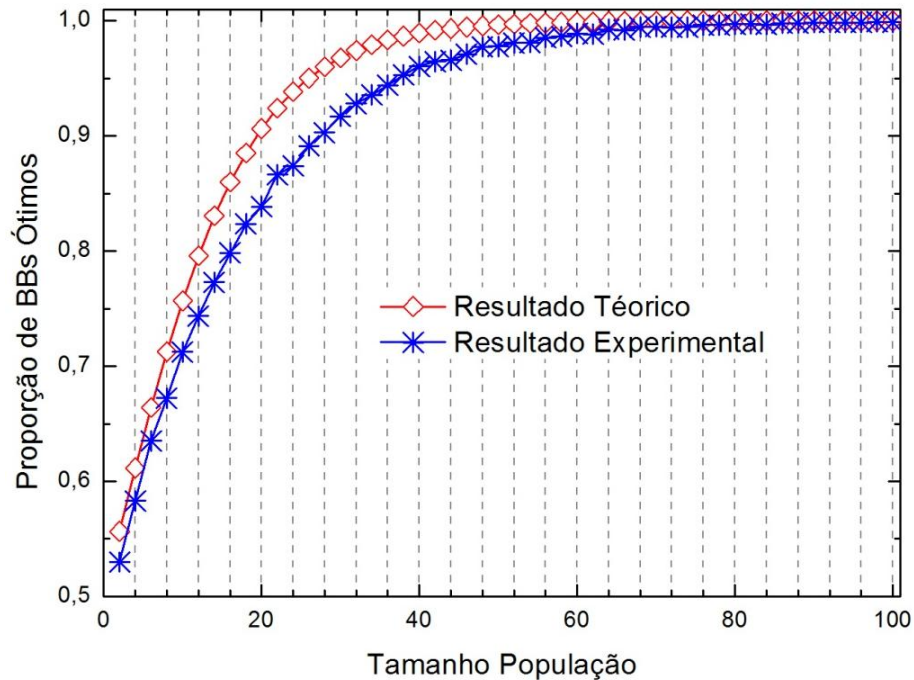


Figura 4: Resultados teórico e experimental da proporção de BBs ótimos em uma função OneMax de 100 bits.

b. BinInt

O experimento da função BinInt considera strings com 50 bits, de modo que $m = 50$, a ordem dos BBs é $k = 1$.

Como no caso da função OneMax o fornecimento de BBs ótimos é uma população inicializada aleatoriamente que tem em média 50% dos BBs ótimos.

A diferença de fitness ou a distância é

$$d = f_{max} - f_{min} = 1 * 2^{49} - 0 * 2^0 = 2^{49}$$

Aqui foi considerada a maior distância possível, que está relacionada ao bit mais significativo.

Considerando que metade das strings possui valor f_{min} e a outra metade possui f_{max} , a variância pode ser calculada como (Goldberg et al., 1992):

$$\sigma_{bb}^2 = \frac{(f_{max} - f_{min})^2}{4} = \frac{(2^{49} - 0)^2}{4} = \frac{(2^{49})^2}{4}$$

$$\sigma_{bb}^2 = 7.922816e + 28$$

Assim, o desvio é $\sigma_{bb} = 2.81475e + 14$

Substituindo esses valores na Equação 4, determinamos a probabilidade de escolher corretamente entre dois BBs que competem:

$$p = N\left(\frac{d}{\sqrt{2m'\sigma_{bb}}}\right) = N\left(\frac{2^{49}}{\sqrt{2 * 49 * (2.81475e + 14)}}\right) = 0,5800536$$

Calculando a proporção dos BBs ótimos pela Equação 5:

$$P_{BB} = \frac{1 - \left(\frac{q}{p}\right)^a}{1 - \left(\frac{q}{p}\right)^n} = \frac{1 - \left(\frac{1-p}{p}\right)^{n/2^k}}{1 - \left(\frac{1-p}{p}\right)^n}$$

$$P_{BB} = \frac{1 - \left(\frac{1-0,5800536}{0,5800536}\right)^{n/2^1}}{1 - \left(\frac{1-0,5800536}{0,5800536}\right)^n}$$

Foi utilizado crossover uniforme. A probabilidade de troca de cada bit foi definida em 0,5.

Na Figura 5 são ilustrados os resultados do experimento. A curva em cor vermelha representa a previsão teórica do modelo de ruína do jogador (Equação 5), a curva em cor azul são os resultados experimentais.

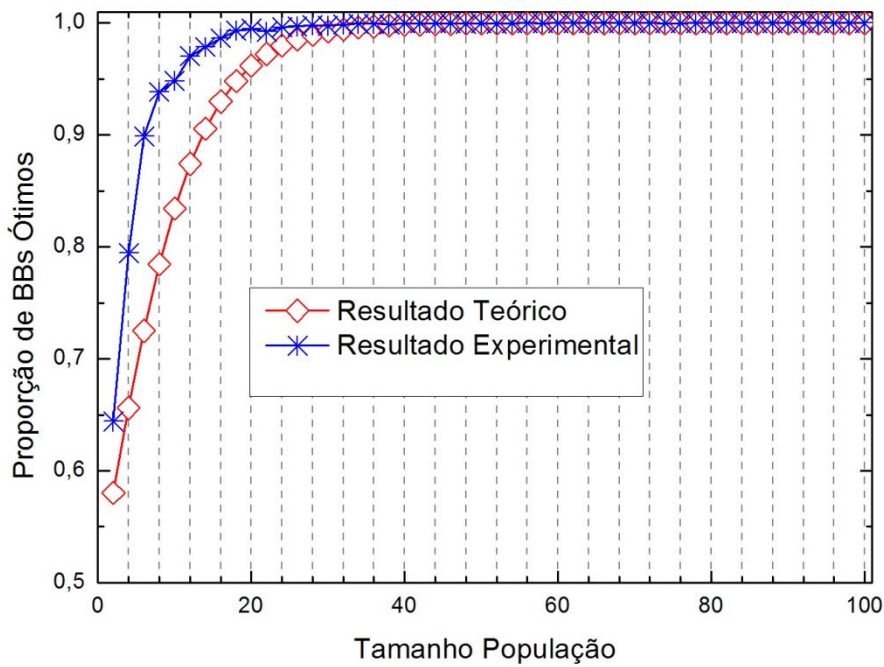


Figura 5: Resultados teórico e experimental da proporção de BBs ótimos em uma função BinInt de 50 bits.

Enquanto o OneMax é mais fácil de ser tratado ao se aumentar a escala do problema, o BinInt apresenta uma dificuldade maior, uma vez que bits mais significativos no cálculo da função de fitness direcionam a seleção das melhores soluções. Por esse motivo, quando os parâmetros de entrada do algoritmo não são adequados, bits menos significativos frequentemente convergem prematuramente para valores aleatórios. Para essa convergência prematura, dá-se o nome de convergência à deriva (drift, em inglês).

No último experimento realizado, foi analisada a influência de σ_{bb} e de d , no tamanho da população, de acordo com a fórmula:

$$n = - \frac{\ln(\alpha) 2^{k-1} \sigma_{bb} \sqrt{2m'}}{d}$$

Para mostrar a relação entre o tamanho da população (n) e a proporção de blocos ótimos, substituímos nessa equação α por $1 - P_{bb}$.

A Tabela 1 abaixo mostra os resultados obtidos. Na primeira coluna estão os valores de P_{bb} gerados, entre 0 e 1.

Na segunda coluna, caso 1, foi calculado o desvio padrão e a distância para o bit mais significativo.

Caso 1:

Quando a distância é

$$d = f_{max} - f_{min} = 1 * 2^{49} - 0 * 2^{49} = 2^{49}$$

Determinamos fitness médio:

$$\bar{f} = \frac{f_{max} + f_{min}}{2} = \frac{1 * 2^{49} + 0 * 2^{49}}{2} = \frac{2^{49}}{2}$$

$$\bar{f} = 2^{48}$$

A variância é:

$$\sigma_{bb}^2 = \frac{(f_{max} - \bar{f})^2 + (f_{min} - \bar{f})^2}{2}$$

$$\sigma_{bb}^2 = \frac{(2^{49} - 2^{48})^2 + (0 - 2^{48})^2}{2}$$

$$\sigma_{bb}^2 = 7,92282e + 28$$

O desvio é : $\sigma_{bb} = 2,81475e + 14$

Na terceira coluna, caso 2, o desvio (σ_{bb}) utilizado foi calculado pela média aritmética dos desvios de todos os blocos. O valor de d também foi calculado como a média aritmética do valor de d de cada bloco.

Caso 2:

Quando a distância média é

$$\bar{d} = \frac{(f_{max_1} - f_{min_1}) + (f_{max_2} - f_{min_2}) + \dots + (f_{max_m} - f_{min_m})}{m}$$

$$\bar{d} = \frac{(2^{49} - 0) + (2^{48} - 0) + \dots + (2^0 - 0)}{50}$$

$$\bar{d} = 2,2518e + 13$$

Determinamos fitness médio de cada bloco como:

$$\bar{f}_1 = \frac{f_{max_1} + f_{min_1}}{2} = \frac{2^{49} + 0}{2} = 2^{48}$$

$$\bar{f}_2 = \frac{f_{max_2} + f_{min_2}}{2} = \frac{2^{48} + 0}{2} = 2^{47}$$

...

$$\bar{f}_m = \frac{f_{max_m} + f_{min_m}}{2} = \frac{2^0 + 0}{2} = 2^{-1}$$

A variância de cada bloco é:

$$\sigma_{bb_1}^2 = \frac{(f_{max_1} - \bar{f})^2 + (f_{min_1} - \bar{f})^2}{2} = \frac{(2^{49} - 2^{48})^2 + (0 - 2^{48})^2}{2}$$

$$\sigma_{bb_2}^2 = \frac{(f_{max_2} - \bar{f})^2 + (f_{min_2} - \bar{f})^2}{2} = \frac{(2^{48} - 2^{47})^2 + (0 - 2^{47})^2}{2}$$

...

$$\sigma_{bb_m}^2 = \frac{(f_{max_m} - \bar{f})^2 + (f_{min_m} - \bar{f})^2}{2} = \frac{(1 - 0,5)^2 + (0 - 0,5)^2}{2}$$

$$\sigma_{bb}^2 = \frac{\sigma_{bb_1}^2 + \sigma_{bb_2}^2 + \dots + \sigma_{bb_m}^2}{50}$$

O desvio é : $\sigma_{bb} = 1.1259e+13$

Na quarta coluna, caso 3, calculou-se o desvio σ_{bb} e o d para o bit menos significativo.

Caso 3:

Quando a distância é

$$d = f_{max} - f_{min} = 1 * 2^0 - 0 * 2^0$$

$$d = 1$$

Determinamos fitness médio:

$$\bar{f} = \frac{f_{max} + f_{min}}{2} = \frac{1 * 2^0 + 0 * 2^0}{2}$$

$$\bar{f} = 0,5$$

A variância é:

$$\sigma_{bb}^2 = \frac{(f_{max} - \bar{f})^2 + (f_{min} - \bar{f})^2}{2}$$

$$\sigma_{bb}^2 = \frac{(1 - 0,5)^2 + (0 - 0,5)^2}{2}$$

$$\sigma_{bb}^2 = 0,25$$

O desvio é : $\sigma_{bb} = 0,5$

Nas duas últimas colunas, analisou-se os casos em que o desvio σ_{bb} é alto e a distância d é baixa, e vice-versa. Assim, na penúltima coluna, utilizou o desvio do bit mais significativo e a distância do menos significativo (caso 4), e na última coluna, o contrário (caso 5).

Tabela 1: Resultados obtidos.

P_{bb}	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5
0,05	0,253888854	0,253888854	0,253888854	1,42927E+14	4,50997E-16
0,1	0,521507946	0,521507946	0,521507946	2,93583E+14	9,26384E-16
0,15	0,80442766	0,80442766	0,80442766	4,52853E+14	1,42895E-15
0,2	1,104504228	1,104504228	1,104504228	6,21781E+14	1,96199E-15
0,25	1,42395361	1,42395361	1,42395361	8,01615E+14	2,52945E-15
0,3	1,765450901	1,765450901	1,765450901	9,93861E+14	3,13607E-15
0,35	2,132266648	2,132266648	2,132266648	1,20036E+15	3,78767E-15
0,4	2,528457838	2,528457838	2,528457838	1,4234E+15	4,49144E-15
0,45	2,959142181	2,959142181	2,959142181	1,66585E+15	5,25649E-15
0,5	3,430903502	3,430903502	3,430903502	1,93143E+15	6,09451E-15
0,55	3,952411448	3,952411448	3,952411448	2,22501E+15	7,02089E-15
0,6	4,53540773	4,53540773	4,53540773	2,55321E+15	8,0565E-15
0,625	4,854857112	4,854857112	4,854857112	2,73304E+15	8,62396E-15
0,65	5,196354403	5,196354403	5,196354403	2,92529E+15	9,23058E-15
0,675	5,56317015	5,56317015	5,56317015	3,13179E+15	9,88218E-15
0,7	5,95936134	5,95936134	5,95936134	3,35482E+15	1,0586E-14
0,725	6,390045683	6,390045683	6,390045683	3,59728E+15	1,1351E-14
0,75	6,861807004	6,861807004	6,861807004	3,86285E+15	1,2189E-14
0,775	7,38331495	7,38331495	7,38331495	4,15644E+15	1,31154E-14
0,8	7,966311232	7,966311232	7,966311232	4,48463E+15	1,4151E-14
0,82	8,487819178	8,487819178	8,487819178	4,77822E+15	1,50774E-14
0,84	9,070815461	9,070815461	9,070815461	5,10642E+15	1,6113E-14
0,86	9,731762133	9,731762133	9,731762133	5,4785E+15	1,72871E-14
0,88	10,49476907	10,49476907	10,49476907	5,90803E+15	1,86425E-14
0,9	11,39721473	11,39721473	11,39721473	6,41606E+15	2,02455E-14
0,91	11,91872268	11,91872268	11,91872268	6,70964E+15	2,11719E-14
0,92	12,50171896	12,50171896	12,50171896	7,03784E+15	2,22075E-14
0,93	13,16266564	13,16266564	13,16266564	7,40992E+15	2,33816E-14
0,94	13,92567257	13,92567257	13,92567257	7,83946E+15	2,4737E-14
0,95	14,82811824	14,82811824	14,82811824	8,34749E+15	2,634E-14
0,955	15,34962618	15,34962618	15,34962618	8,64107E+15	2,72664E-14
0,96	15,93262246	15,93262246	15,93262246	8,96927E+15	2,8302E-14
0,965	16,59356914	16,59356914	16,59356914	9,34135E+15	2,94761E-14
0,97	17,35657607	17,35657607	17,35657607	9,77088E+15	3,08315E-14
0,975	18,25902174	18,25902174	18,25902174	1,02789E+16	3,24345E-14
0,98	19,36352597	19,36352597	19,36352597	1,09007E+16	3,43965E-14
0,985	20,78747958	20,78747958	20,78747958	1,17023E+16	3,6926E-14
0,99	22,79442947	22,79442947	22,79442947	1,28321E+16	4,0491E-14
0,995	26,22533297	26,22533297	26,22533297	1,47635E+16	4,65855E-14

Observa-se que nos casos 1, 2 e 3, os valores do tamanho da população obtidos são iguais. Isso é possível porque quando varia-se a distância, o desvio varia proporcionalmente.

Para os casos 4 e 5, obteve-se os valores esperados, pois de acordo com a fórmula, quando o σ_{bb} possui valor alto e a distância é pequena,

espera-se que o tamanho da população seja alto. De forma contrária, quando o σ_{bb} possui valor baixo e a distância é grande, espera-se que o tamanho da população seja pequena.

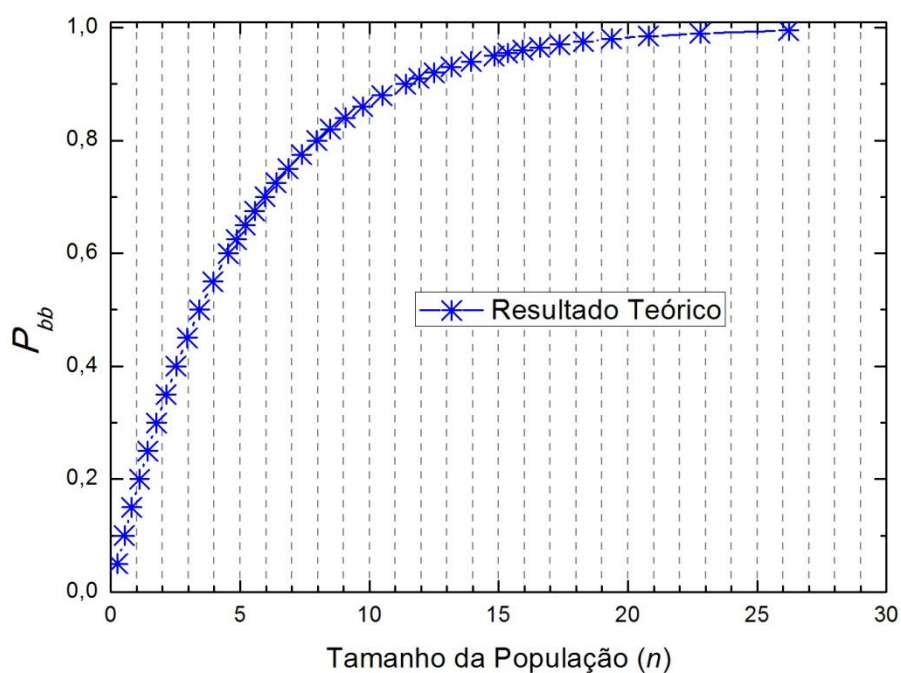


Figura 6: Resultado teórico para os casos 1, 2 e 3 em uma função BinInt de 50 bits.

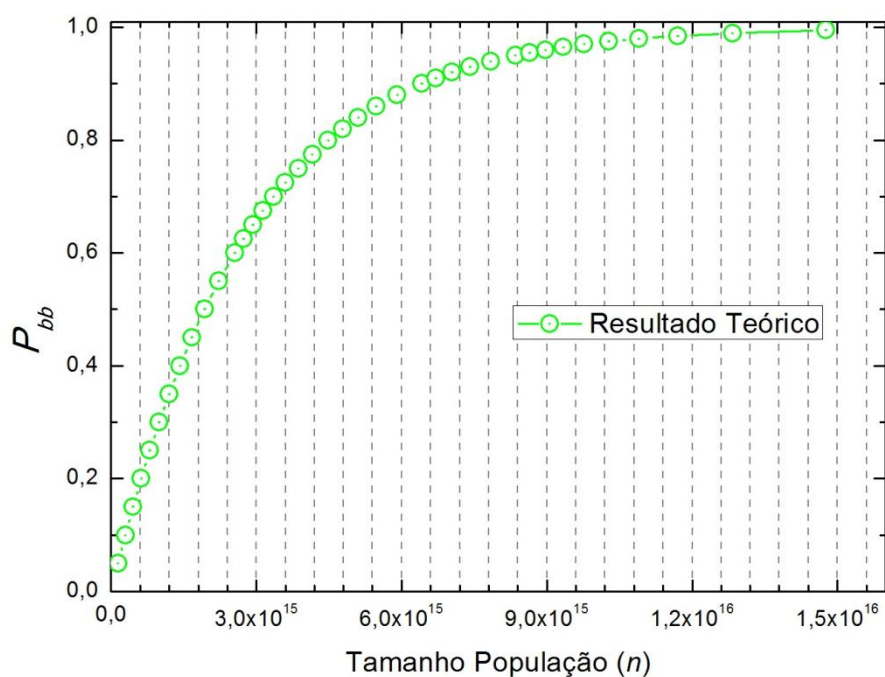


Figura 7: Resultado teórico para o caso 4 em uma função BinInt de 50 bits.

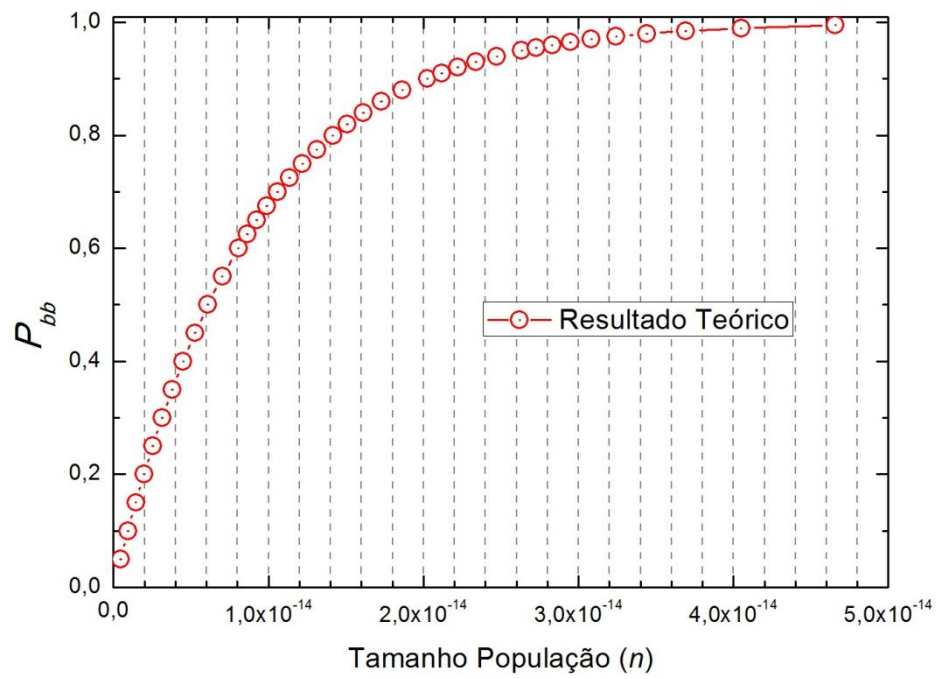


Figura 7: Resultado teórico para o caso 5 em uma função BinInt de 50 bits.

4. Bibliografia

Abramowitz, M. and Stegun, I. Handbook of mathematical functions with formulas, graphs, and mathematical tables. Dover Publications, Mineola, New York, (1972).

Feller, W. An introduction to probability theory and its applications (2nd ed.), Volume 1, John Wiley and Sons, New York, New York. (1966).

Goldberg, D. E., Deb, K. and Clark, J. H. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362. (1992).

Crocomo, Marcio K. Algoritmo de otimização bayseano com detecção de comunidades. Tese de doutorado – Universidade de São Paulo. São Carlos, Setembro de 2012.