

5 ALGORITMOS GENÉTICOS

5.1-INTRODUÇÃO

Algoritmos Genéticos são algoritmos de otimização global, baseados nos mecanismos da seleção natural e da genética. Eles empregam uma estratégia de busca paralela e estruturada, que é voltada em direção ao reforço da busca de pontos de "alta aptidão", ou seja, pontos nos quais a função a ser maximizada tenha valores relativamente altos.

A busca de uma solução não é uma caminhada aleatória ou não direcionada, pois explora informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos. Isto é feito através de processos iterativos, onde cada iteração é chamada de geração.

Em cada iteração, os princípios de seleção e reprodução são aplicados a uma população de candidatos cujo tamanho varia conforme a complexidade do problema e os recursos computacionais disponíveis. Através da seleção, se determina quais indivíduos (que são as soluções candidatas do problema) conseguirão se reproduzir, gerando um número determinado de descendentes para a próxima geração, com uma probabilidade determinada pelo seu índice de aptidão. Ou seja, os indivíduos com maior aptidão têm maiores chances de se reproduzir.

5.2-A INSPIRAÇÃO ORIGINAL

Segundo a teoria da evolução das espécies, de Charles Darwin, na natureza os indivíduos mais capacitados para a competição por recursos são os que conseguem sobreviver em ambientes hostis. A capacidade de adaptação a um ambiente permanentemente em mutação é essencial à sobrevivência das espécies, sendo que o conjunto de características de um indivíduo determina a sua capacidade de sobrevivência. Estas características, em geral, são determinadas pelo material genético de cada um e controladas por uma unidade básica chamada gene. Um conjunto de genes forma um cromossomo.

São as modificações no material genético das espécies que constituem a essência do processo de evolução. A evolução se dá através da ação conjunta da seleção natural e da recombinação de material genético durante a reprodução. Na natureza, a competição faz com que os indivíduos mais aptos tenham mais chances de sobreviver, enquanto os menos favorecidos tendam a desaparecer. O mecanismo de seleção natural leva à sobrevivência dos genes mais aptos.

O processo de evolução inicia-se na recombinação do material genético proveniente dos pais durante a reprodução. Novas combinações de genes são geradas a partir das já existentes. A troca de material genético entre cromossomos é chamada de crossover. Pedacos dos cromossomos dos pais são trocados durante o *crossover*, criando a possibilidade do surgimento de combinações mais “corretas” de genes e, conseqüentemente, de indivíduos mais capacitados. A repetição do processo de seleção e *crossover* ao longo das gerações permite a evolução contínua do material genético da espécie e a geração de indivíduos que sobrevivam em um ambiente competitivo.

5.3-MODELANDO A EVOLUÇÃO COM ALGORITMOS

Na década de 70, John Holland propôs os algoritmos genéticos como programas de computador que imitam o processo de evolução encontrado na natureza. Algoritmos genéticos são muito eficientes para busca de soluções ótimas, ou aproximadamente ótimas, em uma grande variedade de problemas, pois não impõem muitas das limitações encontradas nos métodos de busca tradicionais.

Os algoritmos genéticos manipulam uma população de soluções potenciais para um problema de busca ou otimização. Eles operam sobre a representação codificada de soluções, equivalente ao material genético dos indivíduos na natureza. O algoritmo genético codifica as soluções em cadeias de bits e cada solução é associada a uma medida de qualidade ou fitness, que reflete a capacitação de um indivíduo em relação aos demais.

Assim como na natureza, um mecanismo de seleção força a continua evolução da qualidade das gerações. Quanto maior a medida de *fitness* de um indivíduo, maiores as suas chances de se reproduzir e sobreviver à próxima geração. A recombinação de material genético é simulada através de um mecanismo de *crossover* que troca pedaços entre as cadeias de bits dos pais. Outra operação, chamada de mutação, provoca alterações esporádicas e randômicas nas cadeias de bits. A operação de mutação tem o papel de reintroduzir na população material genético perdido.

Inicialmente é gerada uma população formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema. Durante o processo evolutivo, esta população é avaliada: para cada indivíduo é dada uma nota, refletindo sua habilidade de adaptação a determinado ambiente. Uma porcentagem dos mais adaptados é mantida, enquanto os outros são descartados. Os membros mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamentos, gerando descendentes para a próxima geração. Este processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada.

Embora possam parecer simplistas do ponto de vista biológico, estes algoritmos são suficientemente complexos para fornecer mecanismos de busca adaptativos poderosos e robustos.

5.4-A ESTRUTURA DOS ALGORITMOS

Durante a geração t , o algoritmo mantém uma população de soluções potenciais (cromossomos) $P(t) = \{v_1, \dots, v_n\}$. Cada solução v_i é avaliada para fornecer a medida de sua qualidade ou aptidão. Uma nova população $P(t+1)$ é então formada, selecionando-se, probabilisticamente, os indivíduos mais aptos de $P(t)$. Alguns dos membros desta nova geração sofrem transformações por meio de operações de *crossover* e mutação, de acordo com uma probabilidade pré-definida.

Assim, o projeto de um algoritmo genético para uma dada aplicação deve conter os seguintes componentes:

- Uma representação na forma de cromossomos das soluções potenciais do problema.
- Uma forma de criar uma população inicial de soluções válidas.
- projeto de uma função de avaliação que classifique os indivíduos em função de sua aptidão.
- Operadores genéticos que alterem a composição dos filhos (*crossover* e *mutação*).
- Parâmetros de controle do algoritmo: tamanho da população, probabilidade da aplicação dos operadores genéticos, critério de parada, etc.

O algoritmo pode ser resumido em:

- I. Criar aleatoriamente uma população inicial de *strings*.
- II. Enquanto o critério de parada não for satisfeito:
 - a. Associar um grau de adaptação a cada indivíduo da população, usando a medida de adaptação escolhida.
 - b. Criar uma nova população, através da aplicação das três operações genéticas nos indivíduos escolhidos, com probabilidade baseada no grau de adaptação:
 - i. Reprodução de um indivíduo através da inserção de outra cópia sua na população.
 - ii. Recombinação de duas *strings*, usando a operação de *crossover*, para formar dois novos indivíduos.
 - iii. Mutação aleatória de um caractere, localizado em uma posição escolhida também aleatoriamente, criando um novo indivíduo.
- III. A *string* identificada pelo método de indicação de resultado é escolhida como o resultado do AG. Esse resultado pode ser a solução ideal ou aproximada do problema.

Critério de parada: pode-se utilizar o critério do número máximo de gerações, do tempo limite de processamento ou o critério de estagnação (parar quando há melhoria da população após um certo número de gerações consecutivas).

5.5-DETALHANDO AS OPERAÇÕES GENÉTICAS

As operações genéticas utilizadas pelo algoritmo são:

- Seleção: Baseada no princípio da sobrevivência do mais adaptado. Nessa operação, um indivíduo é selecionado probabilisticamente, baseando-se no seu grau de adaptação. Os que forem selecionados serão copiados, sem alterações, na próxima geração da população, de forma tal que, quanto mais adaptado, mais chance ele tem de ser selecionado. O interessante é que mesmo os pouco ou nada adaptados possuem alguma chance de se reproduzirem.
- Crossover: permite que novos indivíduos sejam criados e testados. A operação começa com dois pais selecionados probabilisticamente, baseando-se nos seus graus de adaptação. Como na operação anterior, os mais adaptados têm mais chance de serem selecionados porém mesmo os pouco adaptados possuem alguma chance. A operação produz dois filhos, cada um contendo algum material genético dos pais.
- Mutação: Permite a criação de novos indivíduos, através da seleção por adaptação e a escolha aleatória de um caractere da string para ser invertido. O novo indivíduo é inserido na próxima geração. Essa operação deve ser utilizada de forma econômica durante a execução do algoritmo genético.

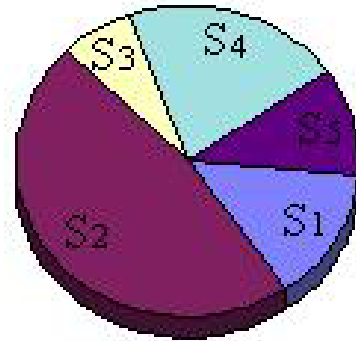
5.5.1 Seleção

O propósito desta operação é fazer a seleção dos pais para aumentar a probabilidade de reproduzir membros da população que tenham bons valores da função objetivo. Uma forma de implementação é a roleta, uma representação em que cada indivíduo tem uma probabilidade de ser selecionado que é proporcional ao seu grau de *fitness*. Um estágio de seleção então seria análogo a um giro dessa roleta, selecionando o indivíduo correspondente segundo o arco do círculo de onde está.

Para visualizar este método, considere um círculo (ou roleta) dividido em n regiões (onde n é o tamanho da população), sendo

a área de cada região proporcional a aptidão do indivíduo. Após cada giro da roleta escolhe-se um indivíduo. Este método é denominado amostragem universal estocástica. Cada indivíduo ocupa uma fatia proporcional ao valor da função objetivo. Assim, os indivíduos cujas regiões possuem maior área terão maior probabilidade de serem selecionados, inclusive, mais de uma vez. Veja o exemplo na figura abaixo:

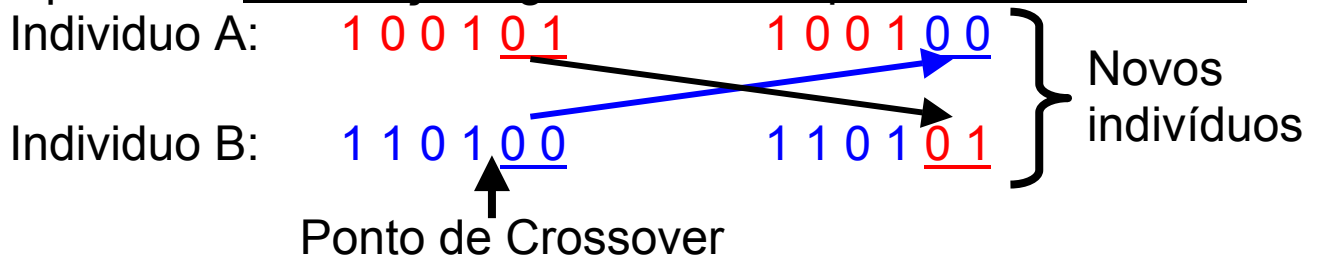
Indivíduo	Aptidão	Aptidão
S_i	$f(S_i)$	Relativa
S_1 10110	2.23	0.14
S_2 11000	7.27	0.47
S_3 11110	1.05	0.07
S_4 01001	3.35	0.21
S_5 00110	1.69	0.11



5.5.2 Crossover

Uma vez eleitos os pais, se utiliza um operador genético de cruzamento. O cruzamento responsável pela recombinação de características dos pais durante a reprodução, permitindo que as próximas gerações herdem suas características. Ele é considerado o operador genético predominante e é aplicado com probabilidade dada pela taxa de crossover P_c , que deve ser maior que a taxa de mutação. As formas de utilização desse operador são:

Um-ponto: um ponto de cruzamento é escolhido e a partir deste ponto as informações genéticas dos pais serão trocadas. Ex:



Multi-pontos: é uma generalização da idéia de troca de material genético através de pontos, onde muitos pontos de cruzamento podem ser utilizados.

Uniforme: não utiliza pontos de cruzamento, mas determina, através de um parâmetro global, qual a probabilidade de cada variável ser trocada entre os pais.

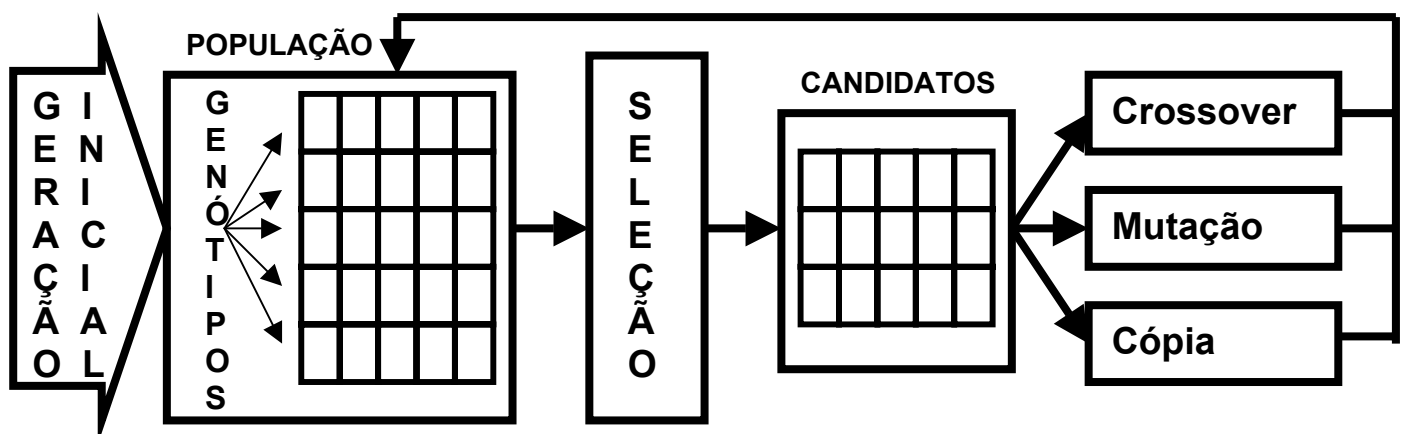
5.5.3 Mutação

Os operadores de mutação são necessários para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais componentes de uma estrutura escolhida, fornecendo assim, meios para a introdução de novos elementos na população. A mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero. O operador de mutação é aplicado aos indivíduos com uma probabilidade dada pela taxa de mutação P_m , geralmente pequena, pois a mutação é uma operação genética secundária.

No caso binário a mutação consiste em substituir com certa probabilidade (P_m) o valor de um bit. Escolhe-se aleatoriamente um bit do cromossomo selecionado. A seguir, um número aleatório entre 0 e 100 é gerado e, se ele for menor do que a taxa P_m , esse bit é invertido. Uma variante é, após o procedimento descrito, gerar aleatoriamente o valor (0 ou 1) do bit escolhido. Neste caso, a taxa de mutação de fato será a metade da taxa P_m . Ex.:

Indivíduo escolhido: 1 0 0 1 0 1 mutação 1 0 0 0 0 1
 Bit escolhido $\xrightarrow{\quad}$ $\xrightarrow{\quad}$

5.5.4 Esquema gráfico do algoritmo genético



5.6-O AJUSTE DOS PARÂMETROS GENÉTICOS

É também importante analisar de que maneira alguns parâmetros influem no comportamento dos algoritmos genéticos.

Tamanho da População: o tamanho da população afeta o desempenho global e a eficiência dos algoritmos genéticos. Uma população pequena oferece uma pequena cobertura do espaço de busca, causando uma queda no desempenho. Uma grande população fornece uma melhor cobertura do domínio do problema e previne a convergência prematura para soluções locais. Entretanto, com uma população muito grande há necessidade de recursos computacionais maiores, ou um tempo maior de processamento.

Taxa de Cruzamento: quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Entretanto, isto pode gerar um efeito indesejado pois a maior parte da população será substituída podendo ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento.

Taxa de Mutação: uma baixa taxa de mutação previne que uma dada posição fique estagnada em um valor, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória.

Intervalo de Geração: controla a porcentagem da população que será substituída durante a próxima geração.

Probabilidades: Em relação às probabilidades de recombinação e mutação, estudos têm mostrado que bons resultados geralmente são obtidos com altos valores de probabilidade de recombinação ($>0,7$) e baixos valores para probabilidade de mutação ($<0,01$).

5.7-O PROBLEMA DA REPRESENTAÇÃO

O ponto de partida para a utilização de algoritmos genéticos para a solução de problemas é a escolha da forma de representação destes problemas. Se um problema pode ser

representado por um conjunto de parâmetros (genes), estes podem ser unidos para formar uma cadeia de valores (cromossomo), no processo chamado codificação.

Tradicionalmente, os indivíduos são representados por vetores binários, onde cada elemento de um vetor denota a presença (1) ou ausência (0) de uma determinada característica: o seu genótipo.

Outra forma de representação muito comum é a de codificar valores inteiros em números binários. Os elementos podem ser combinados formando as características reais do indivíduo, ou o seu fenótipo.

Por exemplo, para um problema de maximizar uma função de três variáveis, $F(X, Y, Z)$, poderia-se representar cada variável por um número binário de 10 bits, obtendo-se assim um cromossomo de 30 bits e 3 genes.

Existem vários aspectos relacionados com a codificação de um problema a serem levados em conta:

- Deve-se utilizar um alfabeto o menor possível para representar os parâmetros. Normalmente se utiliza um dígito binário.
- As variáveis que representam os parâmetros do problema devem ser discretizadas, de forma a constituírem cadeias de bits.
- A maior parte dos problemas tratados com algoritmos genéticos são não lineares e muitas vezes existem relações "ocultas" entre as variáveis que formam a solução.
- O tratamento dos genótipos inválidos deve ser tomado em conta para o desenho da codificação.

5.8-A FUNÇÃO DE *FITNESS*

Geralmente a aptidão do indivíduo é determinada através do cálculo da função objetivo, ou função de avaliação, que depende das especificações de projeto. Dado um cromossomo, a função de avaliação consiste em associar um valor numérico de "adaptação" do indivíduo representado. Muito importante é que ela seja de cálculo rápido, já que vai ser aplicada para cada indivíduo de cada

população, nas sucessivas gerações. Grande parte do tempo gasto por um algoritmo genético se deve ao cálculo da função de avaliação.

A pressão seletiva está implicitamente relacionada com a diversidade da população. Alta pressão seletiva tende a fazer a diversidade cair rapidamente, levando a população a convergir em poucas gerações, o que pode resultar em convergência prematura. Por outro lado, uma baixa pressão seletiva pode tornar o processo de busca extremamente lento. Métodos de escalamento evitam que alguns super-indivíduos dominem a população no início da busca e, geralmente, produzem melhores resultados. A função clássica de medida de *fitness* tem a forma:

$$F = (Cx - Cx_{\min}) / (Cx_{\max} - Cx_{\min})$$

5.9-A CONVERGÊNCIA DO ALGORITMO

Se o algoritmo genético for corretamente implementado, a população evoluirá ao longo de sucessivas gerações até que a adaptação do melhor indivíduo convirja para um ótimo global. A convergência é uma progressão uniforme. Um gene será convertido quando em 95% da população tiver o mesmo valor. A população converge quando todos os genes de cada indivíduo convergirem.

O algoritmo genético procura uma solução no espaço de busca definido pelas *strings* possíveis de serem construídas, de forma a encontrar a mais adaptada e utilizando para isso, apenas os valores numéricos de adaptação. Independentemente da complexidade e do domínio do problema em particular, os algoritmos genéticos executam sempre as mesmas operações simples de reprodução, recombinação e mutação. Além disso, os algoritmos genéticos são surpreendentemente rápidos e eficientes em espaços de buscas complexos, não lineares e multidimensionais.

5.10-EXEMPLOS DE APLICAÇÕES

5.10.1 Encontrar o valor máximo de uma função

Seja a função:

$f(x) = -x^2 + 33x$, onde $x \in [0, 31]$ e x é um número inteiro.

O primeiro passo consiste em codificar o parâmetro x como uma cadeia de bits de tamanho finito. Usando codificação binária, necessitamos de 5 bits. A seguir, devemos gerar aleatoriamente uma população de cadeias. Neste caso, usaremos uma população de tamanho $m = 4$. Por exemplo:

00101 11000 01000 10011

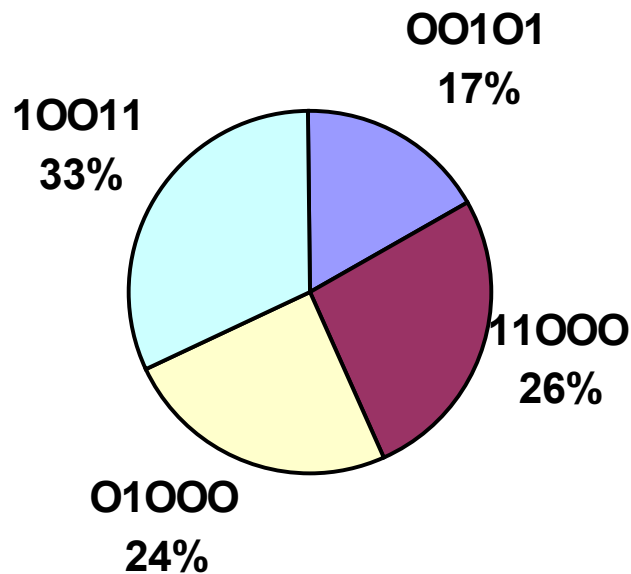
A seguir, calculamos os valores da função objetivo para cada cadeia ($f(x)$):

Cadeia	x	$f(x)$	% do total	% acumulado
00101	5	140	17,03163	17,03163017
11000	24	216	26,27737	43,30900243
01000	8	200	24,3309	67,63990268
10011	19	266	32,3601	100
Total		822	100	

Somando os valores da função objetivo individual de cada cadeia, chegamos ao valor 822, a partir do qual encontramos a participação da função objetivo individual de cada cadeia (% do total).

O processo de seleção ou rolagem da roleta inicia-se com a geração de um número aleatório $n \in [0,100)$. A seguir, consulta-se à qual elemento esse número n corresponde, na tabela de probabilidades acumuladas (veja a figura abaixo).

% do total de cada elemento



Dessa maneira, cadeias com maior valor de função objetivo tem maior chance de serem selecionadas para sobreviver. O processo é repetido até que m cadeias sejam selecionadas.

Supondo que os quatro números gerados foram: 35; 52; 91 e 66. Assim, temos que:

- 35 está entre 17,03 e 43,31, correspondendo à cadeia 11000;
- 52 corresponde à cadeia 01000;
- 91 corresponde à cadeia 10011; e,
- 66 corresponde à cadeia 01000.

Assim, conseguimos as seguintes cadeias de candidatos à sobrevivência: 11000, 01000, 10011, 01000.

O operador de cruzamento é agora aplicado mediante a seleção aleatória de q pares de cadeias, as quais trocarão material entre si (o valor de q depende do valor da taxa de crossover, P_c , definida).

Para realizar o cruzamento, o processo mais elementar consiste na escolha do ponto de cruzamento, representado por um número inteiro aleatório k entre 1 e $(\text{tamanho_da_cadeia} - 1)$.

As cadeias podem manter a parte inicial e trocar entre si a parte posterior à posição k . Caso as cadeias selecionadas sejam 11000 e 10011 e $k = 3$, então 110 | 00 e 100 | 11 trocarão material, gerando 11011 e 10000.

O operador de mutação atua ao nível de bit, usando uma probabilidade de troca de P_m . No exemplo, se $P_m = 1\%$, quando uma cadeia for escolhida a mesma é varrida bit a bit. Para cada bit desta, gera-se um número aleatório entre $[1,100]$. Se, o número for igual à 1, o bit é invertido, se não, ele é mantido (se a probabilidade P_m fosse 3%, inverteríamos o bit se o número gerado aleatoriamente fosse ≤ 3).

A aplicação dos operadores de crossover e mutação está resumida na tabela abaixo:

Seleção	Cross point	Após o cruzamento	Após a mutação	x	$f(x)=-x^2+33x$
110 00	3	11011	11011	27	162
01000	-	01000	010(1)0	10	230
100 11	3	10000	10000	16	272
01000	-	01000	01000	8	200
				Total	864

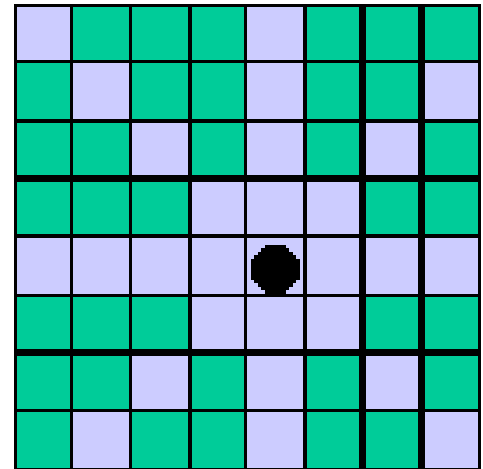
Podemos verificar que a função objetivo máxima cresceu de 266 para 272 e a soma das aptidões dos indivíduos da população cresceu de 822 para 864. Na população atual poderia até ocorrer uma queda no valor máximo da função objetivo, mas a tendência de crescimento é sempre esperada, o que pode ser observado pelo valor total. O mecanismo de seleção aumenta a probabilidade de reprodução de indivíduos com melhor aptidão, assim sendo é esperado o crescimento do valor total.

O algoritmo continua até que a condição de parada seja atihgida (haja pouca evolução, um valor absoluto seja atingido ou um número fixo de gerações tenham sido produzidas).

5.10.2 O jogo das oito Rainhas

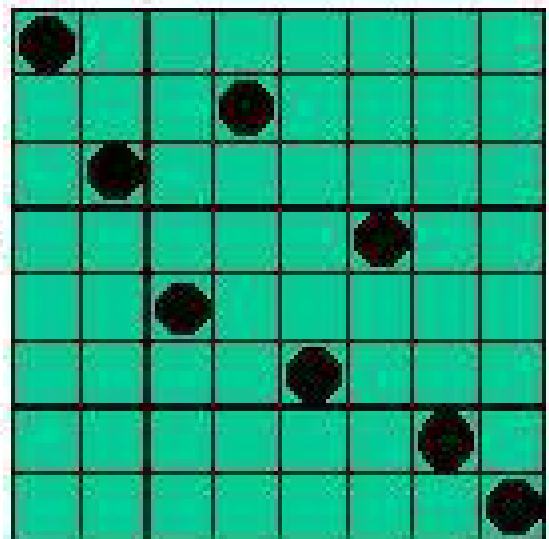
Problema: Colocar 8 rainhas num tabuleiro de 8x8 casas, de tal modo que as rainhas não se possam atacar mutuamente.

Uma idéia para representar o problema, pode ser criar para cada coluna um cromossomo com a posição em que aparece a rainha naquela coluna. Assim uma posição qualquer do tabuleiro (fenótipo) equivaleria a um genótipo de oito posições, como ilustrado abaixo:



Genótipo: uma permutação dos números de 1 a 8

1 3 5 2 6 4 7 8



Fenótipo: uma configuração →

A mutação pode ser feita, ao contrário do caso anterior, trocando-se elementos aleatoriamente dentro do genótipo:

Mutação: troca de dois números

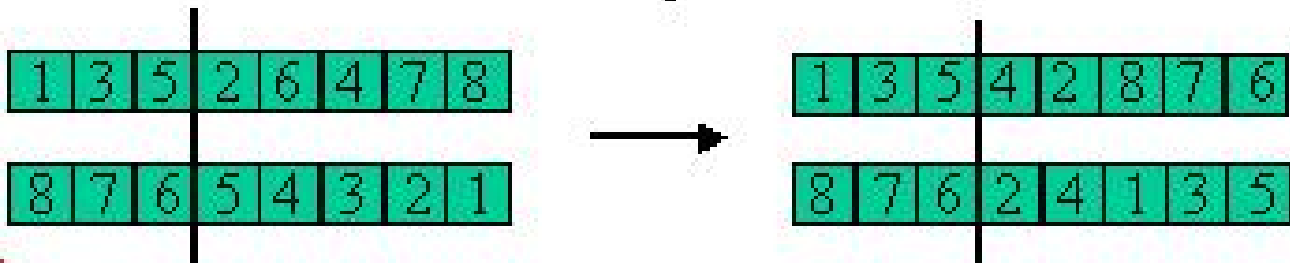
1 3 5 2 6 4 7 8



1 3 7 2 6 4 5 8

O crossover pode ser feito com ponto único:

Crossover: combinar dois pais



Função de avaliação (fitness): cada rainha é penalizada pelo número de outras rainhas que pode atacar. A qualidade da solução é dada pela soma da quantidade de rainhas que podem ser atacadas.

Naturalmente, como o ideal neste caso é que a quantidade de rainhas atacadas seja zero, a qualidade da avaliação será tanto melhor quanto menor for o número dado pela função de avaliação.

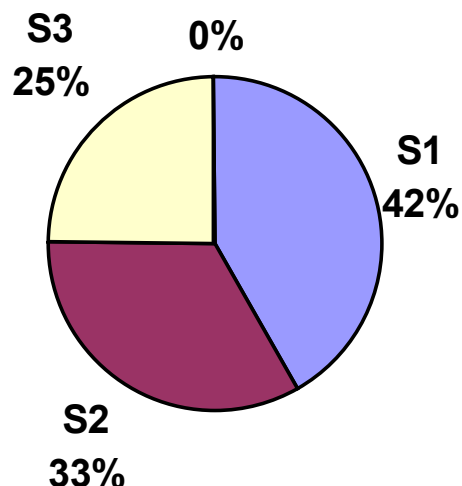
Assim, para fazermos com que a área ocupada na roleta seja proporcional à qualidade da solução, podemos fazer a área de uma configuração igual à diferença entre o valor da função de avaliação e soma das penalidades de todas as rainhas, sobre o valor desta soma.

Ex:

$$\text{Fitness}(S_1) = 2$$

$$\text{Fitness}(S_2) = 4$$

$$\text{Fitness}(S_3) = 6$$



Realizando as operações genéticas utilizando parâmetros adequados, o algoritmo tende a encontrar a solução que, neste caso, minimiza a função de custo (fitness).