Algoritmos Genéticos em Português Simples

Artigo original: http://www.ai-junkie.com/ga/intro/gat1.html
Autor: Mat Buckland
Tradução: Jerson Seling
Nota: Os capítulos 3, 4 e 5 do livro *AI Techniques For Game Programming* (Premier Press) do mesmo autor, apresentam este artigo de forma amplamente mais aprofundada. Recomendo que você adquira esse livro se ficou interessado pelo tema.

1. Introdução

O propósito deste capítulo é explicar os algoritmos genéticos o suficiente para habilitar você a usá-los em seus próprios projetos. Este é um capítulo com apenas o mais essencial do assunto. Eu não irei me aprofundar no assunto e não irei assustar você com uma matemática perturbadora ao mostrar equações do mau em todas as sentenças. De fato, eu não mostrarei nenhuma equação chata para você em todo este capítulo! Mas só neste capítulo...

Este capítulo foi feito para ser lido *duas vezes*... Então, não se preocupe se algo dele não fizer sentido na primeira vez que você estudá-lo.

2. Primeiro, uma aula de Biologia

Cada organismo possui um conjunto de regras, como uma planta de uma casa, descrevendo como o organismo é construído a partir dos pequenos tijolos de vida. Essas regras são codificadas nos *genes* de um organismo, os quais ficam conectados em longas sequências chamadas *cromossomos*. Cada gene representa uma característica específica do organismo, como a cor dos olhos ou dos cabelos, possuindo cada um várias configurações diferentes. Por exemplo, as configurações para o gene da cor do cabelo pode ser loiro, preto ou ruivo. Esses genes e suas configurações são usualmente referidos como o *genótipo* de um organismo. A expressão física do genótipo – no próprio organismo – é chamada de *fenótipo*.

Quando dois organismos acasalam, eles compartilham seus genes. A prole resultante acaba tendo metade dos genes de um pai e metade do outro. Esse processo é chamado de *recombinação*. Muito ocasionalmente um gene pode sofrer uma *mutação*. Normalmente esse gene mutante não afeta o desenvolvimento do fenótipo, mas muito ocasionalmente ele será expresso no organismo como uma característica completamente nova.

A vida na Terra se desenvolveu assim, pelo processo de seleção natural, recombinação e mutação. Para ilustrar como esse processo funcionou, produzindo a grande variedade de fauna e flora que convive no nosso planeta, deixe-me contar uma pequena história...

> *A muito tempo atrás viveu uma espécie de criatura chamada Gritador. Os gritadores se desenvolveram inteiramente dentro dos sombrios confins de um vasto sistema de profundas cavernas, oculto no interior de uma cordilheira. Eles tinham uma vida fácil, sentiam e farejavam uma alga nas paredes da caverna que eles adoravam comer, roncavam entre as rochas e na época do acasalamento, eles ouviam atentamente os gritos de outros gritadores. Não havia predadores na caverna, só os gritadores, a alga e ocasionalmente uma lesma amigável, assim os gritadores não tinham nada para temer (exceto talvez por um ocasional gritador mal-humorado). Um rio subterrâneo fluía pelo sistema de cavernas e a água continuamente gotejava pelas paredes trazendo nutrientes frescos que faziam as algas prosperarem, assim sempre havia o que comer e beber. Embora os gritadores pudessem sentir e farejar, eles nunca tiveram a necessidade de olhos no interior escuro das cavernas, assim o resultado era que eles eram totalmente cegos. Isso nunca pareceu interessar qualquer a um dos gritadores, assim eles só matavam o tempo gemendo e gritando na escuridão.*
>
> *Então um dia, um terremoto fez com que uma parte da caverna desmoronasse e pela primeira vez em muitos milênios, os gritadores sentiram o calor da luz do sol sobre sua pele e a suavidade do musgo sob seus pés. Alguns gritadores mais ousados provaram o musgo e viram que ele era melhor de comer que a alga da caverna. "Ooooooooooh!" gritaram eles entre montes de musgos e prontamente foram devorados por águias famintas que tinham voado até ali para ver o que era toda aquela comoção.*
>
> *Assim por um tempo pareceu que os gritadores poderiam ser caçados até a extinção, já que eles gostavam de comer o musgo e nunca podiam ver se uma águia estava voando acima de suas cabeças. Não apenas isso, eles também não podiam se esconder embaixo de uma rocha a menos que estivessem perto o bastante para tocá-la com seus sensores. Todo dia muitos*

gritadores que saíam da caverna com o doce cheiro do musgo em seus narizes eram rapidamente arrebatados para longe e comidos por uma águia. A situação deles parecia crítica.

Felizmente, com os anos, a população de gritadores cresceu muito na segurança das cavernas e muitos deles sobreviveram para acasalar – afinal, uma águia não podia comer muito. Um dia, uma ninhada de gritadores nasceu compartilhando um gene mutante de célula de pele. Esse gene em particular era responsável pelo desenvolvimento das células de pele de suas testas. Durante o desenvolvimento dos bebês gritadores, quando suas células de pele se formaram a partir das instruções do gene mutante, elas ficaram ligeiramente sensíveis a luz. Cada novo bebê gritador podia sentir se algo estava bloqueando a luz em suas testas ou não. Quando esses pequenos bebês gritadores cresceram e se tornaram gritadores adultos e se aventuraram na luz para comer o musgo, eles podiam saber se algo estava sobrevoando sobre suas cabeças ou não. Assim esses gritadores cresceram tendo uma chance ligeiramente melhor de sobrevivência que a de seus primos totalmente cegos. E por causa de terem uma chance melhor de sobrevivência, eles de reproduziram muito mais, transmitindo o novo gene da célula de pele sensível a luz para sua prole. Depois de um curto período, a população se tornou dominada pelos gritadores com essa ligeira vantagem.

Agora vamos pular uma umas mil gerações no futuro. Se você extrapolar esse processo ao decorrer de muitos e muitos anos e contar com várias pequenas mutações que ocorressem nos genes das células da pele, é fácil imaginar um processo onde uma célula sensível a luz pode vir a se tornar um grupo de células sensíveis a luz e então as células do interior do grupo poderiam mutar para virar uma área em forma de lente, a qual ajudaria a colher a luz e focá-la em um só lugar. Também não é difícil imaginar uma mutação que levasse a produzir duas dessas áreas coletoras de luz, dando assim aos gritadores uma visão binocular. Isso poderia ser uma grande vantagem sobre seus primos Ciclopsianos, já que assim eles agora poderiam julgar distâncias perfeitamente e ter um grande campo de visão.

Como você pode ver, o processo de seleção natural – a sobrevivência do mais apto – e a mutação genética possuem regras muito poderosas para provocar a evolução de um organismo. Mas como a recombinação entra nesse esquema? Bom, para mostrar isso a você eu preciso contar algo sobre outros gritadores…

Ao mesmo tempo em que os gritadores com as células sensíveis a luz faziam algazarra em volta ao musgo e incomodavam as águias, outra ninhada de gritadores nasceu compartilhando um gene que afetava seu grito. Essa mutação dava a eles um grito mais forte que o de seus primos e este, portanto, podia atingir maiores distâncias. Isso se tornou útil na população cada vez menor, pois os gritadores com gritos maiores podiam chamar parceiros para o acasalamento que estivessem mais longe. Não só isso, mas as fêmeas gritadoras começaram a mostrar uma ligeira preferência pelos machos com gritos maiores. A conseqüência disso é claro, é que os gritadores mais bem dotados tiveram uma chance muito melhor de acasalar do que os outros gritadores. Com o tempo, os gritadores com gritos mais fortes prevaleceram na população.

Então em um belo dia, uma fêmea gritadora com o gene das células de pele sensíveis a luz encontrou um gritador com o gene para produzir gritos enormes. Eles caíram de amor um pelo outro e logo depois produziram uma ninhada de adoráveis bebês gritadores. Agora, por causa dos bebês terem cromossomos com a recombinação dos cromossomos de ambos os pais, alguns

dos bebês compartilharam ambos os genes especiais e cresceram não somente com as células de pele sensíveis a luz, mas com gritos enormes também! Essa nova ninhada foi extremamente boa para evitar águias e se reproduzir já que o processo da evolução os tinha favorecido, dessa forma, novamente o tipo melhorado de gritador se tornou dominante na população.

E assim continuou e continuou...

Os *algoritmos genéticos* são uma maneira de se resolver problemas ao se imitar o mesmo processo que a mãe natureza usa. Eles usam a mesma combinação de seleção, recombinação e mutação para desenvolver uma solução para um problema. Bom não é? Vire a página e encontre como exatamente isso é feito.

3. O Algoritmo Genético – uma visão geral

Antes que você possa usar um algoritmo genético para resolver um problema, deve-se encontrar uma maneira de *codificar* qualquer solução em potencial para o problema. Essa pode ser uma sequência de números reais ou, como é o caso mais típico, uma sequência de números binários. Eu me referirei a esta sequência de agora em diante como cromossomo. Um típico cromossomo pode parecer com isto:

1001010111010100101001110110111011111111101

(Não se apavore se isto não faz sentido no momento, isto começará a se tornar mais claro em breve. Por enquanto, apenas relaxe e deixe rolar).

No início da execução de um algoritmo genético, uma grande população e cromossomos *aleatórios* é criada. Cada um, quando decodificado representará uma diferente solução para o problema em mão. Vamos dizer que são *N* os cromossomos na população inicial. Então, as seguintes etapas serão repetidas até que a solução seja encontrada:

1. Testar cada cromossomo para ver o quão bom ele é para resolver o problema em mão e designar uma *classificação de aptidão (fitness score)* de acordo. A classificação de aptidão é a medida de o quão bom esse cromossomo é para resolver o problema em mão;

2. Selecionar dois membros da população atual. A chance deles serem selecionados é proporcional a aptidão de seus cromossomos. O método de seleção com *roleta* é o mais usado;

3. Dependendo da *taxa de cruzamento* (*crossover rate*), serão feitos cruzamentos nos genes de cada cromossomo escolhido em pontos aleatoriamente escolhidos;

4. Percorrer os genes do cromossomo e alterá-los dependendo da taxa de mutação (*mutation rate*);

5. Repetir as etapas 2, 3 e 4 até que uma nova população de N membros seja criada.

3.1 Fale-me sobre a seleção com roleta (*roulette wheel*)

Esta é uma maneira de escolher membros da população de cromossomos de uma maneira proporcional a sua aptidão. Ela não garante que o membro mais apto seja escolhido para a próxima geração, ela meramente dá uma chance maior de isso acontecer a ele. Ela funciona assim:

Imagine que a classificação de aptidão total da população é representada por um gráfico em pizza, ou uma roleta. Agora designe um pedaço da roleta a cada membro da população. O tamanho do pedaço é proporcional a classificação de aptidão do cromossomo. Por exemplo, quanto mais apto é um membro, maior é seu pedaço de pizza. Agora, para escolher um cromossomo é só esperar a bolinha parar e pegar o cromossomo onde ela ficou.

3.2 O que é a taxa de cruzamento (*crossover rate*)?

Isto é simplesmente a chance de dois cromossomos trocarem seus genes. Um bom valor para isso é em torno de 0,7. O cruzamento é desempenhado pela seleção de um gene aleatório ao longo do cromossomo e pela troca de todos os genes a partir desse ponto.

Por exemplo, temos dois cromossomos:

1000100111001001 0
0101000100100001 1

Ao se escolher um gene aleatóriamente ao longo destes, digamos a posição 9, todos os genes que vem depois serão trocados, assim temos:

1000100110100001 1
0101000101001001 0

3.3 O que é a taxa de mutação (*mutation rate*)?

Esta é a chance de um gene dentro de um cromossomo ser alterado (0 vira 1 e 1 vira 0). Isso em geral é um valor muito baixo em genes codificados binariamente, digamos 0,001.

Assim, quando os cromossomos são escolhidos a partir da população, o algoritmo primeira testa para ver se o cruzamento pode ser aplicado, depois então o algoritmo volta e testa de novo o cromossomo para ver se é possível se mutar os genes dele.

4. Da teoria a prática

Para praticar a teoria que você recém aprendeu, vamos ver um problema simples:

Usando os dígitos de 0 a 9 e os operadores +, -, * e /, encontre uma sequência que resultará em um número alvo fornecido. Os operadores serão aplicados sequencialmente, da esquerda para direita como você lê.

Assim, se fornecido o número alvo 23, a sequência 6+5*4/2+1 pode ser uma solução possível.

Se 75,5 é o número escolhido, então 5/2+9*7-5 poderia ser uma solução possível.

Por favor. Garanta que você compreenda bem este problema antes de seguir. Eu sei que ele é um pouco inútil na vida real, mas o escolhi por ser muito simples.

4.1 Estágio 1: Codificação

Primeiro, nós precisamos codificar uma possível solução como uma sequência de genes… um cromossomo. Então, como faremos isso? Bom, primeiro precisamos representar todos os diferentes caracteres disponíveis para a solução… que são os de 0 a 9 e os sinais +, -, * e /. Estes representarão os genes. Cada cromossomo será feito de vários genes.

Quatros *bits* são necessários para representar todos esses caracteres usados:

0: **0000**
1: **0001**
2: **0010**
3: **0011**
4: **0100**
5: **0101**
6: **0110**
7: **0111**
8: **1000**
9: **1001**
+: **1010**
-: **1011**
*: **1100**
/: **1101**

Acima são mostrados todos os diferentes genes necessários para se codificar o problema descrito. Os possíveis genes **1110** e **1111** permanecerão sem uso e serão ignorados pelo algoritmo se encontrados.

Assim, agora você pode ver que a solução mencionada anteriormente para 23, "6+5*4/2+1" pode ser representada por nove genes assim:

0110 1010 0101 1100 0100 1101 0010 1010 0001

6 + 5 * 4 / 2 + 1

Esses genes são unidos em sequência para formar o cromossomo:

01101010010111000100110100101010000 1

4.1.1 Uma palavrinha sobre a decodificação

Por causa do algoritmo tratar de arranjos aleatórios de *bits* é freqüente se encontrar uma sequência como esta:

00100010101011101011011100 10

Decodificados, esses *bits* representam:

0010 0010 1010 1110 1011 0111 0010

2 2 + n/a - 7 2

O que é inexpressivo no contexto deste problema! Assim, quando decodificar, o algoritmo terá de ignorar qualquer gene que não estiver conforme o padrão esperado de: número -> operador -> número -> operador... e assim em diante. Com isso em mente os cromossomos "sem sentido" acima são lidos (e testados) como:

2 + 7

4.2 Estágio 2: Escolha da Função de Aptidão

Esta pode ser a parte mais difícil do algoritmo de se entender. Ela na verdade depende de qual problema você está tentando resolver, mas a idéia geral é dar a classificação mais alta de aptidão para os cromossomos que estão mais perto de resolver o problema. Considerando o pequeno projeto que descrevi aqui, uma classificação de aptidão pode ser dada de uma forma inversamente proporcional a diferença entre a solução e o valor que um cromossomo decodificado representa.

Se considerarmos até o fim deste capítulo que o número alvo é 42, o cromossomo mencionado anteriormente:

01101010010111000100110100101010000 1

Tem uma classificação de aptidão de 1/(42-23) ou 1/19.

Assim se uma solução for encontrada, um erro de divisão por zero poderá ocorrer já que a aptidão será de 1/(42-42). Isso entre tanto não é um problema, pois já encontramos o que procurávamos... uma solução. Assim um teste pode ser feito nessa ocorrência e o algoritmo pode ser tratado de acordo.

4.3 Estágio 3: Mãos a Obra

Primeiro, por favor, leia este capítulo novamente.

Se você agora sente que entende o bastante para resolver este problema, eu recomendo que você tente codificar o algoritmo genético você mesmo. Não há melhor forma de aprender. Se, entretanto, você ainda está confuso, eu já preparei um código simples que você pode encontrar em http://www.ai-junkie.com/files/Basic_Genetic_Algorithm.zip. Preste atenção em coisas como a taxa de mutação, a taxa de cruzamento, tamanho do cromossomo, etc., para entender como cada parâmetro afeta o algoritmo. Também documente o código bem, para que você depois possa seguir o que está acontecendo!

Nota: O código fornecido irá converter a sequência de *bits* de um cromossomo nos valores já discutidos e a solução que ele encontrar (se encontrar) terá *todos* os símbolos válidos encontrados. Assim, se o alvo é 42, o resultado +6*7/2 não será uma solução válida, mesmo que seus quatro primeiros símbolos ("+6*7") formem uma solução válida.

Um código em *Delphi* feito por Asbjørn pode ser encontrado em:
http://www.ai-junkie.com/files/ga_tutorial_delphi.zip

Um código em *Java* feito por Tim Roberts pode ser encontrado em:
http://www.ai-junkie.com/files/GA.java

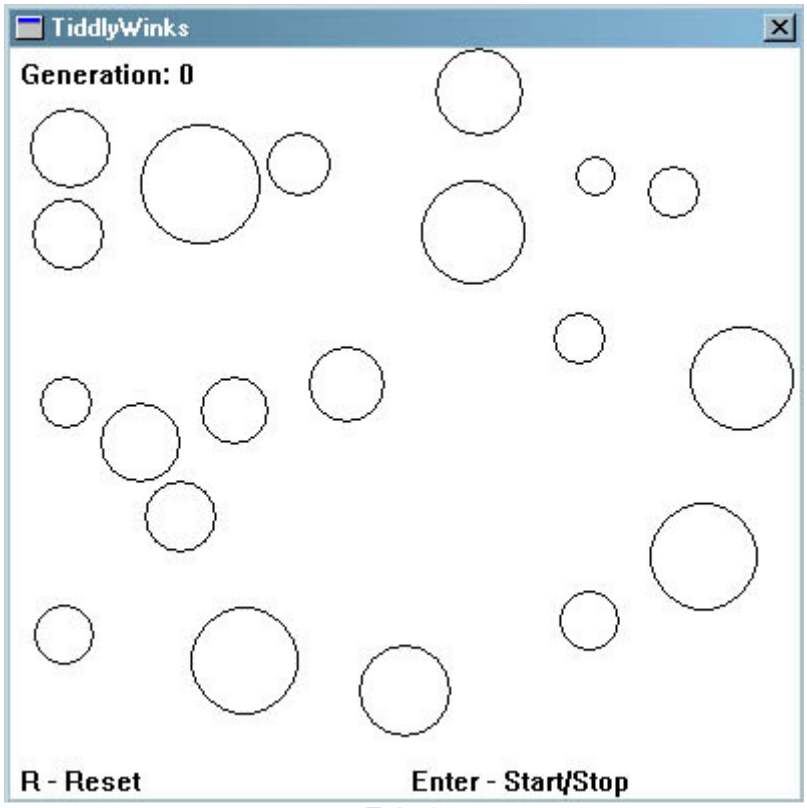
4.4 Últimas Palavras

Eu espero que este capítulo tenha ajudado você a entender o básico sobre algoritmos genéticos. Por favor, note que eu apenas cobri o mais básico aqui. Se você achou os algoritmos genéticos interessantes, então há muito mais para você aprender. Há diferentes técnicas para se usar, diferentes operadores de cruzamento e mutação para se tentar e coisas mais esotéricas como compartilhamento de aptidão (*fitness sharing*) e evolução de espécies (*speciation*) para se testar. Todas ou algumas dessas técnicas irão melhorar o desempenho de seus algoritmos genéticos consideravelmente.

4.5 Coisas para se Tentar

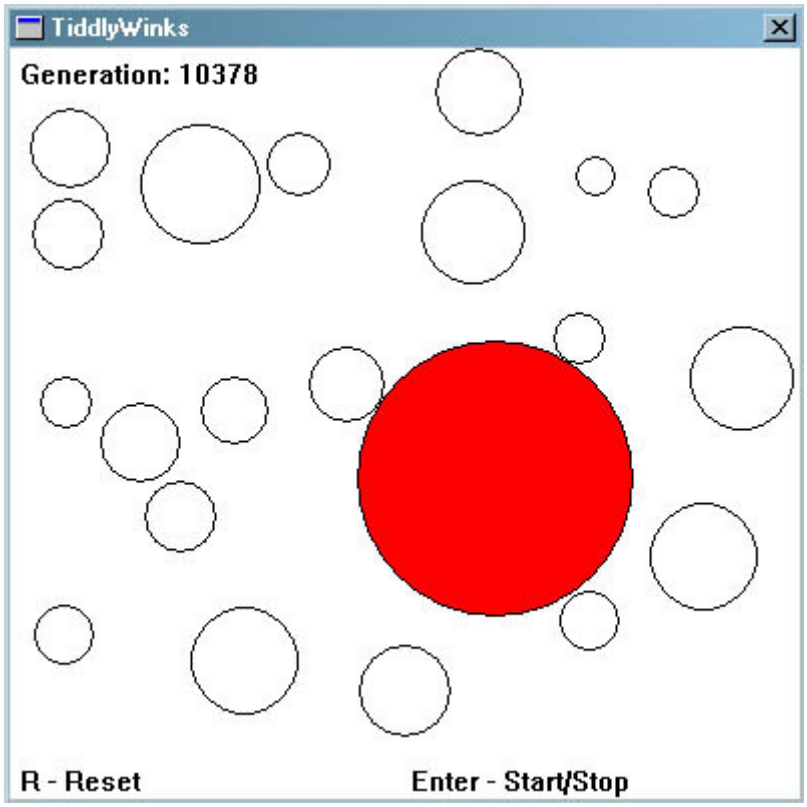
Se você teve sucesso em codificar um algoritmo genético para resolver o problema dado neste tutorial, tente seguir este problema mais difícil:

Dada uma área com discos não sobreponíveis espalhados em sua superfície como mostrado na Tela 1:



Tela 1

Use um algoritmo genético para encontrar o disco de maior raio que possa ser colocado entre esses discos sem sobrepor nenhum deles. Veja a Tela 2.



Tela 2

Como você já pode ter percebido, eu já escrevi um código que resolve esse problema, então se você quiser vê-lo, você pode encontrá-lo em http://www.ai-junkie.com/files/tiddlywinks.zip (mas você irá

tentar resolver sozinho antes não é?). Para os que estão sem compilador, peguem o executável em http://www.ai-junkie.com/files/tiddlywinks_exe.zip.