

# Introdução aos Algoritmos Genéticos

© Artur Marques, 2005



## Índice

<b>Introdução</b> .....	3
<b>Relevância</b> .....	3
<b>Descrição</b> .....	4
Eis a estrutura do algoritmo genético básico: .....	4
Como codificar cromossomas? .....	5
Como fazer cruzamento e mutação? .....	5
Parâmetros de um AG .....	5
Seleção dos ascendentes, para reprodução sexual. ....	6
Codificações alternativas para os cromossomas. ....	7
<b>Conclusão</b> .....	8
<b>Referências</b> .....	8
<b>Projecto: G-Loto</b> .....	9
Resumo .....	9
Introdução .....	9
Observação .....	9
Flexibilidade e Genética .....	12
Crítica .....	13
Futuro .....	13

## Introdução

Os algoritmos genéticos (AGs) são uma técnica utilizada em problemas de procura/descoberta. A designação resulta de que os aspectos a descobrir são representados numa estrutura de dados, sobre a qual irão acontecer operações como cruzamento e mutação, de forma a que, iteração após iteração, a estrutura vá codificando valores «melhores», para os objectivos do problema.

Esta forma de atingir os objectivos (por evolução), mediante operações sobre um estrutura codificadora (cromossoma), tem elevado paralelo com o que se passa na natureza, conforme estudado em Genética.

Também a Teoria da Selecção Natural, de Charles Darwin, é importante para compreender a evolução, em AGs. De acordo com esta teoria, numa população de indivíduos, sobrevivem e reproduzem-se mais, os melhor adaptados ao meio ambiente; isto é, os mais aptos, para o problema. A prazo, os desadaptados extinguir-se-ão.

No arquipélago das Galápagos, Darwin encontrou, em ilhas próximas, espécies diferentes, com características exclusivas, para as particularidades do seu habitat, como pássaros cujos bicos estão especializados em certas sementes. Estes bicos foram o resultado de muitas gerações, que foram reforçando a expressão da característica vantajosa.

De notar que a teoria de Darwin foi atacada por não conseguir responder a (1) como surgem as variações nos indivíduos e a (2) como é que os indivíduos passam as suas características à descendência.

Só no século XX, com a Genética, é que estas questões seriam respondidas: os seres vivos têm, no núcleo das suas células, arranjos de DNA, chamados de genes, que determinam as suas características. Um cromossoma é uma sequência de genes. Durante a reprodução sexual, os cromossomas dos progenitores cruzam-se, gerando novos cromossomas.

Com baixa probabilidade, também podem acontecer mutações, que são alterações cromossómicas fortuitas, que podem conferir vantagens (caso das borboletas pretas de Manchester), desvantagens (síndrome de Down), ou não ter expressão.

Em AGs, todos estes conceitos estão presentes: cromossoma, características codificadas, reprodução (sexual), cruzamento (*crossover*), mutação (*mutation*), indivíduo, função de avaliação da aptidão de um indivíduo (*fitness function*), população e geração.

## Relevância

Em termos históricos, a paternidade dos AGs é atribuída a John Holland, principalmente depois do seu livro *Adaption in Natural and Artificial Systems* (1975). Em termos de posicionamento, os AGs podem considerar-se Computação Evolutiva, que é um ramo da Inteligência Artificial.

Holland pretendeu melhorar o entendimento do processo de adaptação natural e conceber sistemas artificiais, que o manifestassem. A aplicação principal são problemas de procura e de optimização, embora não se possam garantir soluções óptimas, devido à natureza heurística da função objectivo, que é explicada de seguida.

O problema dos métodos estritamente analíticos, muito eficazes para problemas de optimização, é que a «realidade» pode não ser fácil de descrever, com funções matemáticas e determinísticas – pense-se na classe de problemas *NP-hard* (*non deterministic polynomial*, com resposta não necessariamente binária).

## Descrição

Um conceito chave em AGs é o de **função objectivo**, ou *fitness function*. Esta função recebe como argumento um indivíduo da população e devolve a sua aptidão, relativamente aos restantes indivíduos. Quanto mais elevado o *fitness* de «alguém», maior é a sua probabilidade de reprodução.

A função objectivo serve assim para orientar o algoritmo no **espaço de procura**, pois o espaço de procura (ou espaço de estados) é a colecção de indivíduos possíveis, onde, iteração após iteração, se deverá assistir à desactivação das soluções pior avaliadas e à reprodução das de *fitness* superior, de forma a que se vão «activando» cromossomas melhores.

### Eis a estrutura do algoritmo genético básico:

#### 1) início

Gerar uma população aleatória inicial. Esta população pode ser vista como uma colecção de cromossomas, pois os indivíduos estão reduzidos à representação das suas características relevantes para o problema (cromossoma).

#### 2) *fitness*

Avaliar cada cromossoma, pela função de *fitness*.

#### 3) nova população

Produzir uma nova população (ou geração, ou descendência), por execução dos passos seguintes, tantas vezes, quantos os novos indivíduos pretendidos.

##### 3.1) selecção

Seleccionar dois cromossomas, para cruzamento, respeitando que quanto mais elevado o seu *fitness*, maior a sua probabilidade de selecção.

##### 3.2) cruzamento

Os cromossomas dois pais devem cruzar-se de alguma maneira.

##### 3.3) mutação

Considerar, com baixa probabilidade, a aplicação de mutação em alguma posição do cromossoma dos descendentes.

##### 3.4) aceitar

Aceitar o descendente e colocá-lo na nova população.

#### 4) substituição

Substituir a população (velha) pela nova população, gerada no passo 3.

#### 5) testar

Testar a condição de paragem do algoritmo; se satisfeita, terminar, produzindo como «melhor solução» (não confundir com solução óptima) a população corrente; caso contrário, continuar.

Uma condição típica de paragem, é um *fitness* médio superior a determinado valor, ou quando decorrido certo número de iterações.

#### 6) goto 2

O algoritmo genético básico (AGb) apresentado, deixou questões em aberto...

### Como codificar cromossomas?

Depende *completamente* do problema. Uma opção muito frequente é fazer uma codificação binária, em que cada bit corresponde à (não)activação de uma característica. Em alguns dos exemplos que se seguem, optou-se por essa representação, mas serão apresentadas alternativas.

### Como fazer cruzamento e mutação?

Cruzamento e mutação são os operadores dos AGs. O desempenho de qualquer AG é, acima de tudo, influenciado pela forma como estes operadores se comportam.

A forma mais usual de fazer *crossover*, consiste em escolher um ponto aleatório de cruzamento e depois juntar os genes à esquerda do «pai» com os genes à direita, da «mãe». Na representação abaixo, a barra vertical ( | ) representa o ponto de cruzamento.

Pai / cromossoma #1:	11011	00100110110
Mãe / cromossoma #2:	11011	11000011110
Filho #1 / cromossoma descendente #1:	11011	11000011110
Filho #2 / cromossoma descendente #2:	11011	00100110110

É possível fazer cruzamentos multi-ponto e com trocas de genes mais elaboradas, tudo dependendo da codificação para o problema.

A mutação só deve ocorrer depois do cruzamento, conforme descrito no algoritmo básico. O objectivo é diminuir a probabilidade de «óptimos locais», motivados pela falta de diversidade genética.

Para fazer mutação há que seleccionar quantos e quais os genes a mudar e depois mudá-los. No caso de uma codificação binária, isso corresponde a transformar alguns zeros em uns, e vice-versa. É importante compreender que a mutação tem uma violência variável; isto é, durante a produção da mesma nova população, pode afectar mais genes nuns indivíduos, do que noutros. Segue-se um exemplo, em que as mutações estão assinaladas com fundo amarelo.

Pai / cromossoma #1:	1101100100110110
Mãe / cromossoma #2:	1101111000011110
Filho #1 / cromossoma descendente #1:	1100111000011110
Filho #2 / cromossoma descendente #2:	1101101100110110

A técnica de mutação é dependente da estrutura de dados do cromossoma.

### Parâmetros de um AG.

Os parâmetros básicos de um AG são a **probabilidade de cruzamento**, a **probabilidade de mutação**, e o **tamanho da população**. Quando não há cruzamento nem mutação, os filhos são réplicas dos pais.

A probabilidade de cruzamento determina a probabilidade com que os descendentes terão genes de ambos os ascendentes; de notar que mesmo sem cruzamento, os descendentes não serão necessariamente cópias: tudo depende de serem alvo, ou não, de mutação.

Uma probabilidade de cruzamento de 100% ( $P=1$ ), faz com que não seja possível preservar indivíduos de gerações anteriores, na nova geração. Tipicamente, escolhe-se uma probabilidade de cruzamento elevada ( $P>0.8$ ), mas não certa, de forma a permitir indivíduos multi-geracionais.

A probabilidade de mutação deve ser baixa ( $P<0.1$ ), ou o algoritmo genético torna-se uma procura aleatória, pois o espaço de procura começa a ser povoado por cromossomas pouco relacionados com a respectiva ascendência.

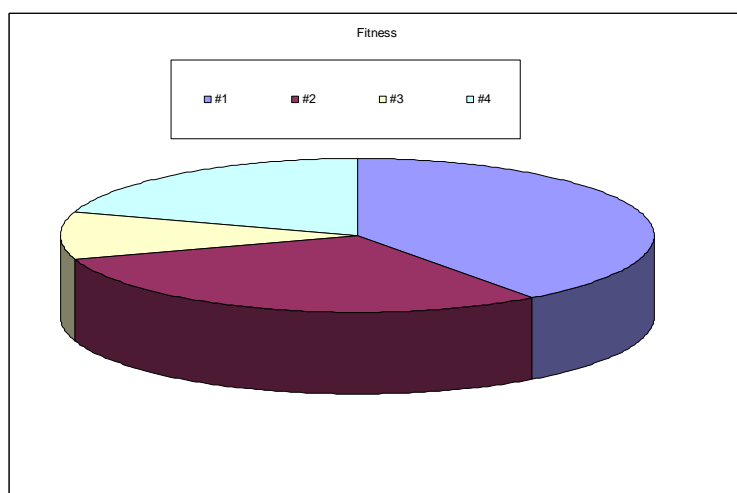
A mutação é um operador que, em certa perspectiva, colide com a função objectivo: quando nunca há mutação, todo AG é dirigido pela função de *fitness*; quando há muita mutação, o trabalho da função objectivo é desvirtuado, pelo acaso.

O tamanho da população é o número de cromossomas que são avaliados em cada iteração. Populações pequenas ajudam a iterações mais rápidas, mas não ajudam à exploração do espaço de procura, porque o património genético é reduzido; por outro lado, populações numerosas podem conduzir a execuções penosamente lentas.

## Seleção dos ascendentes, para reprodução sexual.

Normalmente, um cromossoma com o dobro do *fitness* de outro, deve ter o dobro da probabilidade de produzir descendência. Uma técnica para capturar as probabilidades de reprodução, em harmonia perfeita com os valores da função objectivo, é o sorteio por roleta directa, em que as maiores fatias estão proporcionalmente atribuídas aos progenitores melhor avaliados.

Para fazer um sorteio em roleta directa, é necessário (1) computar a soma do *fitness* de todos os elementos da população, para determinar o «tamanho da roda», (2) produzir um número aleatório em  $[0, \text{soma}]$ , e depois (3) decidir a fatia da roleta seleccionada.



Todavia, o sorteio por roleta directa é pouco utilizado quando há grandes diferenças entre as avaliações dos cromossomas, pois os indivíduos favorecidos podem monopolizar a descendência; por exemplo, imagine-se uma roleta em que 95% da área está atribuída a uma só fatia...

Uma alternativa é a **Rank Selection** ou **selecção por número de ordem**. Nesta técnica, as fatias da roleta não reflectem proporcionalmente os valores da função objectivo, mas antes o número de ordem (*rank*) dos cromossomas. Por exemplo, para uma população de cinco indivíduos, o pior cromossoma valerá 1; o segundo pior valerá 2... e o melhor valerá 5. Desta maneira, quanto melhor for o cromossoma, maior a sua probabilidade de eleição, sem comprometer enormemente os mal classificados.

Seguem-se duas outras alternativas, para selecção.

**Steady state selection**, em que, a cada geração, os melhores cromossomas são cruzados e os piores simplesmente removidos, de forma a que novos (potencialmente) melhores substituam velhos maus, facilitando indivíduos longevos.

**Elitism**, em que se garantem lugares nas gerações seguintes, aos melhores cromossomas de gerações anteriores. De notar que, devido aos operadores de cruzamento e de mutação, um AG básico que use outras técnicas de selecção, não assegura que os melhores representantes de cada população sobrevivem, pois as suas características podem alterar-se. O elitismo pode aumentar o desempenho de um AG, pois evita a possibilidade de perda de uma boa solução relativa.

## Codificações alternativas para os cromossomas.

Para além da codificação binária, outras codificações são habituais, para certos tipos de problemas.

A **codificação por permutações** é usual para problemas de ordenação. Nesta codificação, cada cromossoma é uma frase de números, cuja ordem é relevante. O problema do «caixeiro viajante» - em que há que visitar um conjunto de cidades, percorrendo a menor distância possível - é um exemplo em que faz sentido a codificação por permutações.

Nesta codificação, os operadores de cruzamento e de mutação têm de ter o cuidado de deixar o cromossoma «com lógica», ou então desvirtua-se a procura.

A **codificação por valor** é utilizada em situações em que os cromossomas sejam sequências de valores, fracamente tipificados; isto é, umas vezes sequências de inteiros, outras vezes sequências de números reais, sequências de direcções, etc...

Uma aplicação é a descoberta automática de pesos, para redes neuronais.

Cromossoma #1	1.2 1.3 1.4 0.8 9.11
Cromossoma #2	ALASKDKLWOIEPW
Cromossoma #3	Cima Baixo Esquerda Cima

Em programação genética utiliza-se a **codificação em árvore**, útil para representar hierarquias, como construções sintácticas.

### Possibilidades de cruzamento e de mutação

Para a **codificação binária**...

O cruzamento mais simples é o cruzamento **mono ponto**, conforme visto atrás.

Por exemplo, simbolicamente: **11001011** + **11011111** = **1001111**

O **cruzamento de duplo ponto** implica seleccionar dois pontos de cruzamento. O descendente herda do «pai», a sua estrutura até ao ponto#1 e depois do ponto#2; a restante estrutura é herdada da «mãe».

Por exemplo, simbolicamente: **11001011** + **11011111** = **11011111**

No **cruzamento uniforme**, os ascendentes contribuem com porções aleatórias, para o descendente.

Por exemplo, simbolicamente: **11001011** + **11011101** = **11011111**

No **cruzamento matemático**, os cromossomas cruzam-se, consoante uma operação matemática; por exemplo, o AND binário: **11001011** + **11011111** = **11001001**

Quanto à **mutação**, uma técnica habitual é a mutação por inversão.

Por exemplo, simbolicamente: **11001001** è **10001001**

Para a **codificação por permutações**,

O cruzamento típico é **mono ponto**, da seguinte forma: selecciona-se toda a estrutura do «pai» até ao ponto de *crossover*; depois a «mãe» é considerada, da esquerda para a direita, mas só se aceitam genes que não provoquem repetições.

Por exemplo, simbolicamente:

**(1 2 3 4 5 6 7 8 9)** + **(4 5 3 6 8 9 7 2 1)** = **(1 2 3 4 5 6 8 9 7)**

Quanto à **mutação**, dois números são escolhidos e trocados.

Por exemplo, simbolicamente: **(1 2 3 4 5 6 8 9 7)** è **(1 8 3 4 5 6 2 9 7)**

Para a **codificação por valor**,

O cruzamento pode ser como para a codificação binária.

A mutação consiste em somar/subtrair pequenos valores aos genes, consoante o seu tipo de dados.

Para a codificação em árvore,

O cruzamento faz-se pela ligação entre (sub)árvores dos ascendentes.

A mutação é a mutação de valores (no interior) dos nós da árvore (e não das ligações entre ramos).

## Conclusão

Os AGs são sistemas de inspiração natural, especialmente adequados para problemas de procura, *NP-hard*.

Não podem garantir soluções óptimas, mas podem ser utilizados em problemas de optimização.

## Referências

"Genetic Algorithms + Data Structures = Evolution Programs"

Zbigniew Michalewicz

1996, 3ª edição, Artificial Intelligence series, Springer

"Machine Learning"

Tom M. Mitchell

1997, Computer Science series, McGraw-Hill

[ENCORE](ftp://alife.santafe.edu/pub/USER-AREA/EC/), the EvolutioNary COmputation REpository network

<ftp://alife.santafe.edu/pub/USER-AREA/EC/>

The Hitch-Hiker's Guide to Evolutionary Computation

<ftp://alife.santafe.edu/pub/USER-AREA/EC/FAQ/www/index.html>

Genetic programming FAQ

<http://www-dept.cs.ucl.ac.uk/research/genprog/gp2faq/gp2faq.html>

The Genetic Algorithms Archive

<http://www.aic.nrl.navy.mil:80/galist/>

# Projecto: G-Loto

## Resumo

Esta parte do documento é sobre o projecto «G-Loto» (*Genetic Loto*), em que se procura gerar apostas de *Totoloto*, que aumentem a possibilidade de prémio de um concorrente, recorrendo a um algoritmo genético. Entende-se por *Totoloto*, um jogo em que são sorteadas sequências de números inteiros aleatórios, sem repetições. O projecto «G-Loto» é válido para qualquer concurso de *Totoloto*, independentemente do tamanho ou composição da sequência aleatória.

## Introdução

Uma vez que qualquer *Totoloto* é, por natureza, *absolutamente aleatório*, as soluções informáticas clássicas para apostar, são meros geradores de números inteiros, quase sempre limitadas a cenários concretos, como o *Totoloto* inglês ou o *Totoloto* português, por exemplo. Esta aproximação ao problema rejeita qualquer possibilidade de previsão.

O projecto «G-Loto» tem uma nova atitude perante o conceito de *absolutamente aleatório*...

Em Informática digital não existem geradores genuínos de números aleatórios; toma-se sempre uma raiz de geração, que é normalmente o estado de um relógio suportado em hardware. Ao fim de intervalos de tempo muito precisos, os resultados tendem a repetir-se.

A nova atitude do projecto «G-Loto», consiste em assumir que também na vida quotidiana, os processos aleatórios têm uma origem ou relação com a dimensão Tempo, tal que, em certas circunstâncias, tendem a repetir a sua história passada. Exactamente quais essas origens e/ou relações, é resposta que está para lá do objectivo do «G-Loto».

A Natureza está recheada de fenómenos periódicos, desde as estações meteorológicas, às ondas sinusoidais que neste momento sobrepovoam o espaço hertziano... A percepção que o ser humano faz desse meio condiciona grandemente as suas respostas, para os problemas. Por exemplo, só muito recentemente, com Sir Isaac Newton, factos que temos como triviais, como o efeito da força da gravidade, foram efectivamente *observados*, por omnipresentes que tenham estado, desde sempre.

O projecto «G-Loto», assume que o conceito de *absolutamente aleatório* não rejeita «a possibilidade de repetição, total ou parcial», que é, aliás, qualquer coisa de implícita à definição clássica do termo «aleatório», por contraditório que pareça e por desconsiderado que seja noutras abordagens. Esta assumpção acontece sem demonstração.

## Observação

Num esforço para diferenciar e valorizar distintamente sequências de números inteiros aleatórios, o projecto «G-Loto» utiliza uma função matemática, totalmente baseada num historial de «chaves», entendendo-se por «chaves», sequências de números inteiros aleatórios já sorteados, no âmbito de um concurso de *Totoloto*.

Esta função matemática será a função objectivo de um algoritmo genético simples, que terá como ponto de partida, ou população inicial, uma família de sequências ou «apostas» (cromossomas), que serão avaliadas e seleccionadas, de forma a envolverem-se num processo de evolução, do qual se espera que resulte uma população mais provável.

São consideradas 34 classes de informação, a partir do historial de chaves.

Por cada chave que conste do historial de chaves já sorteadas:

- A soma dos números da chave.
- A média dos números da chave.
- O número mínimo da chave.
- O número máximo da chave.
- A amplitude matemática da chave (MAX-MIN).
- O número de colunas diferentes que a chave exige, num boletim físico de *Totoloto*.
- O número de linhas diferentes que a chave exige, num boletim físico de *Totoloto*.
- O número de números pares, na chave.
- O número de números ímpares, na chave.

- O ano, o número de concurso e o dia, da chave.
- A lista de números propriamente ditos, que compõem a chave.
- A distância cronológica (em Semanas/Dias), da chave, em relação à chave mais recente do historial.

(12 medidas)

Para o conjunto de TODAS as chaves:

- A média da soma das chaves.
- A média da média das chaves.
- A média dos máximos das chaves.
- A média dos mínimos das chaves.
- A média da amplitude matemática das chaves.
- A média do número de colunas diferentes, que as chaves exigem, em boletins físicos de *Totoloto*.
- A média do número de linhas diferentes, que as chaves exigem, em boletins físicos de *Totoloto*.
- A média dos números pares, nas chaves.
- A média dos números ímpares, nas chaves.
- A lista de todos os números propriamente ditos, que alguma vez fizeram parte de alguma chave.
- A lista de quantas vezes saiu cada número que alguma vez foi sorteado, em alguma chave.
- A média de quantas vezes foram sorteados os números, que foram sorteados em chaves.
- A lista de todos os números sorteados mais vezes do que a média.
- A lista de todos os números sorteados menos vezes do que a média.
- A chave média, de todas as chaves sorteadas, ponderada por índice.
- A informação cronológica de todos os números que alguma vez foram sorteados.

(16 medidas)

Para cada número, alguma vez sorteado:

- Quantas vezes o número foi sorteado.
- A lista da evolução das ocasiões em que o número foi sorteado, na forma de uma recta temporal crescente, indexada por Semanas / Dias.
- A lista da evolução das distâncias das ocasiões em que o número foi sorteado, relativamente ao sorteio mais recente do historial. Esta lista está ordenada, na forma de uma recta temporal de idade decrescente; isto é, as menores distâncias cronológicas, para a chave mais recente, vêm no final.
- A soma das ocasiões em que o número foi sorteado, de acordo com a respectiva lista de sorteios. Um valor absoluto elevado desta soma *pode* significar que o número foi sorteado muitas vezes. Um valor relativo elevado desta soma é indicador de que o número tem um historial de sucessos.
- A soma das distâncias das ocasiões em que o número foi sorteado, relativamente à chave mais recente. Um valor modesto desta soma, *pode* significar que o número foi sorteado muito recentemente. Um valor relativo elevado desta soma, é indicador da frequência recente de sucessos do número.

(6 medidas)

Este total de (12+16+6) 34 informações de natureza distinta, é ponderado por uma fórmula de pesos variáveis. Por defeito, utiliza-se a seguinte função objectivo  $F$ , no caso da valoração de um sequência de  $n$  números inteiros aleatórios:

$$F=0.5 * F1 + 0.5 * F2$$

$$F1 = \left( \frac{f1 + f2 + \dots + fn}{n} \right)$$

$$F2 = 0.2 * Sum + 0.1 * (Max + Min + Amp + Evens + Odds + Average + Cols + Lines)$$

As parcelas  $f_1$  a  $f_n$ , consistem na avaliação de cada número individual da sequência, e são calculadas da seguinte forma:

$$f_n = 0.2 * A_1 + 0.2 * A_2 + 0.05 * A_3 + 0.05 * A_4 + 0.25 * A_5 + 0.25 * A_6$$

$$A_1 = \frac{\sum_{Semana} T_n}{\sum_{Semana} T} \quad A_2 = \frac{\sum_{Semana} D_n}{\sum_{Semana} D} \quad A_3 = \frac{\sum_{Dia} T_n}{\sum_{Dia} T} \quad A_4 = \frac{\sum_{Dia} D_n}{\sum_{Dia} D}$$

$$A_5 = \frac{Dn(\text{semanas})}{D \max(\text{semanas})} \quad A_6 = \frac{Dn(\text{dias})}{D \max(\text{dias})}$$

**A1** é uma medida da frequência de «sucesso», de um número  $n$ , versus um número hipotético que tivesse estado em todas chaves.

Um número que tenha saído sempre tem  $A_1=1$  (100%); outros números terão  $A_1 < 1$ .

T representa a recta temporal de chaves.

O denominador é a soma do número de todas as semanas, em historial. Por exemplo, se em historial estiverem (apenas) as semanas 20 e 21, o denominador será  $20+21=41$ .

O numerador é a soma do número das semanas, em que  $n$  esteve presente. Por exemplo, se  $n$  saiu apenas na semana 20,  $A_1 = 20/(20+21)$ .

**A2** é uma medida do «peso da idade» de um número  $n$ , versus um número hipotético que tivesse estado em TODAS as chaves.

A2 elevados indicam que o número está “velho”, ou é muito frequente.

A2 baixos indicam que o número é “novo” ou infrequente.

Para esclarecer entre idade e frequência, podem usar-se a recta temporal e o indicador A1.

D representa a recta temporal de chaves, numa perspectiva de distâncias, para a chave mais recente. Por exemplo, a semana anterior à «actual», tem  $D=1$ ; a anterior a essa tem  $D=2$ , e assim sucessivamente.

Em numerador tem-se a soma das distâncias, medidas em semanas, das chaves em que  $n$  foi sorteado, relativamente à chave mais recente.

Por exemplo, se  $n$  tiver saído, para o histórico de chaves disponíveis, há 10, 5 e 4 semanas atrás, o numerador será  $10+5+4$ .

O denominador é a soma das distâncias, para um número hipotético, que esteja em todas as chaves.

**A3** e **A4** são análogos de **A1** e **A2**, mas em dias.

Na medição em dias, considera-se que as semanas começam aos domingos (dia 1) e terminam aos sábados (dia 7), embora qualquer outra consideração fosse possível, desde que usada consistentemente.

**A5** e **A6** representam o valor relativo das distâncias médias, em semanas e dias, do número  $n$ , relativamente às distâncias máximas, na mesma unidade.

**A1**, **A2**, **A3** e **A4**, tendem a avaliar um número  $n$ , pela sua regularidade; isto é, números que constem frequentemente do historial de chaves disponível, serão melhor cotados (valores mais próximos de 1).

**A5** e **A6**, tendem a avaliar um número  $n$ , pelo seu insucesso (ou ausência); isto é, números invulgares, relativamente ao historial de chaves, tendem a ser melhor cotados (valores mais próximos de 1), porque têm estado, em média, mais distantes.

*Sum*, *Max*, *Min*, *Amp*, *Evens*, *Odds*, *Average*, *Cols* e *Lines*, são válidos, não para um número  $n$  singular, mas para o conjunto de números da aposta. Estes parâmetros medem a percentagem de

proximidade da aposta em causa, relativamente aos valores médios obtidos para o total do historial de chaves.

Qualquer um destes parâmetros, vê a sua percentagem de proximidade, medida pela seguinte expressão, indicada para um parâmetro genérico P:

$$P = 1 - \frac{|\bar{P} - V(\text{aposta})|}{\bar{P}}$$

V(aposta) representa a valoração da aposta, para o parâmetro.

Por exemplo, se o valor médio para um parâmetro, relativamente ao historial de todas as chaves disponíveis, é 150, e se uma aposta aleatória em particular, é medida também em 150 para esse parâmetro, então o valor percentual da sua proximidade à média é:

$$P = 1 - |(150-150)/150| = 1 - 0 = 1$$

Ou seja, essa aposta seria medida como absolutamente coincidente, relativamente à média do parâmetro.

*Sum* representa a proximidade, relativamente à soma média.

*Max* representa a proximidade, relativamente ao valor máximo médio.

*Min* representa a proximidade, relativamente ao valor mínimo médio.

*Amp* representa a proximidade, relativamente ao valor da amplitude matemática média.

*Evens* representa a proximidade, relativamente ao valor médio de números pares.

*Odds* representa a proximidade, relativamente ao valor médio de números ímpares.

*Average* representa a proximidade, relativamente ao valor médio das médias.

*Cols* representa a proximidade, relativamente ao valor médio do número de colunas exigido.

*Lines* representa a proximidade, relativamente ao valor médio do número de linhas exigido.

## Flexibilidade e Genética

O projecto «G-Loto» foi concebido de forma a ser o mais versátil possível. O programa é totalmente dinâmico, com alocação de memória em tempo real, sendo pois capaz de representar e trabalhar com um historial de chaves potencialmente ilimitado. Não há limites para o número de números por aposta, e a configuração gráfica das matrizes físicas do *Totoloto*, é definível, de forma a que parâmetros como *Cols* e *Lines* não sejam afectados por boletins futuros, ou de diferentes partes do mundo.

Todas as opções que afectem qualquer algoritmo do programa são parametrizáveis [também] por ficheiros externos à aplicação.

Todo os *outputs* do programa podem acontecer para ficheiros, e algumas operações geram automaticamente ficheiros.

A forma como o programa foi escrito, permite o *design* fácil de interfaces gráficas de alto nível, que apenas precisarão de conhecer as possíveis entradas e saídas do «G-Loto», na forma de ficheiros, de forma a permitirem toda a interacção tolerada pelo programa.

Para lá da geração de apostas aconselhadas, relativamente a um historial de chaves, o «G-Loto» permite ainda:

- A correcção automática de apostas, relativamente a chaves.
- A geração de estatísticas e avaliações cronológicas, para leitura, pelo apostador.
- Editores de alto nível para ficheiros que contêm apostas, chaves e configurações do utilizador.

Utiliza-se um algoritmo genético, para chegar ao objectivo máximo do programa, que é a geração de uma família de apostas, potencialmente mais prováveis, relativamente ao historial de chaves disponibilizado, assumindo o princípio da localidade.

Neste algoritmo genético, o cromossoma são os próprios números da aposta.

Em traços gerais, este AGb comporta-se assim:

- Parte-se de uma população inicial aleatória...
- ...cujos elementos são avaliados pela função objectivo já descrita.
- Os melhores 30% elementos estão garantidos na geração seguinte (Elitismo).
- Os restantes 70% elementos da próxima geração, são obtidos em sorteio por roleta. Neste processo são escolhidos os «pais» das apostas que seguem, e que serão cruzados em 50% da dimensão do respectivo cromossoma. Por exemplo, para sequências de tamanho 6 (seis números), serão cruzados 3 dos números que compõem a aposta. As posições que cruzam são aleatórias, o que significa que se utiliza uma máscara *random*.
- Daqui se depreende que também o cromossoma com que o algoritmo genético trabalha é flexível; isto é, não se está limitado a apostas de X números; pode trabalhar com apostas (e chaves), de qualquer dimensão.
- O algoritmo pára ao fim de  $100 * \text{Número\_de\_chaves\_no\_historial}$  gerações. Por exemplo, para 84 chaves conhecidas, a família de apostas aconselhadas, será gerada ao cabo de 8400 gerações / iterações. Todavia, este valor (100) é definível. Nada é estático no «G-Loto».

## Crítica

Só a recolha de longas séries temporais de resultados, de apostas feitas com e sem G-Loto, poderá sugerir se este algoritmo genético aumenta, ou não, a probabilidade de prémio.

Em todo o caso, assumindo um cenário de sequências numéricas sujeitas ao princípio da localidade, aqui apresenta-se uma algoritmo genético, com uma função objectivo que procura capturar essa localidade.

## Futuro

O futuro do «G-Loto» passa pela concepção de um meta-G-Loto; isto é, de um meta-Algoritmo Genético (meta-AG), que avalie populações de soluções, e gere automaticamente parametrizações aconselhadas, para a função objectivo.