Resolução do puzzle Pentóminos utilizando Algoritmos Genéticos

PAULO SILVA
Instituto Superior Técnico,
Universidade Técnica de Lisboa
pauloms@mail.telepac.pt

RUI TAVARES
Departamento de Matemática / Informática
Universidade da Beira Interior,
R. Marquês d'Ávila e Bolama
6200 Covilhã - PORTUGAL
rtavares@alpha2.ubi.pt

ANTÓNIO TEÓFILO
Instituto Politécnico de Setúbal
ateofilo@bocage.ips.pt

***Resumo*: Apresentamos a implementação de um Algoritmo Genético para a resolução do puzzle Pentóminos. A selecção é estacionária, probabilística e ponderada, com normalização da função de mérito, o cruzamento é multiponto uniforme, e apenas a mutação é não convencional, actuando sobre a representação com algum conhecimento de contexto. É também acrescentado um esquema de pontuações, para favorecer os indivíduos da população aparentemente com configuração melhor. No final, são apresentados resultados da implementação e propostos melhoramentos futuros.**

***Palavras chave*: Algoritmos Genéticos, Pentóminos.**

1. INTRODUÇÃO

O puzzle Pentóminos é bem conhecido tanto pela sua aparente simplicidade, como pela dificuldade extrema em conseguir encontrar soluções. Um cálculo simples (ver 3.1) mostra a dimensão vasta do espaço de procura. Propomos uma solução para este problema utilizando Algoritmos Genéticos, técnica que tem vindo a ser usada com suceso em problemas de procura em grande escala, e em particular naqueles em que não se conhecem ou é extremamente difícil formular estratégias de busca [1].

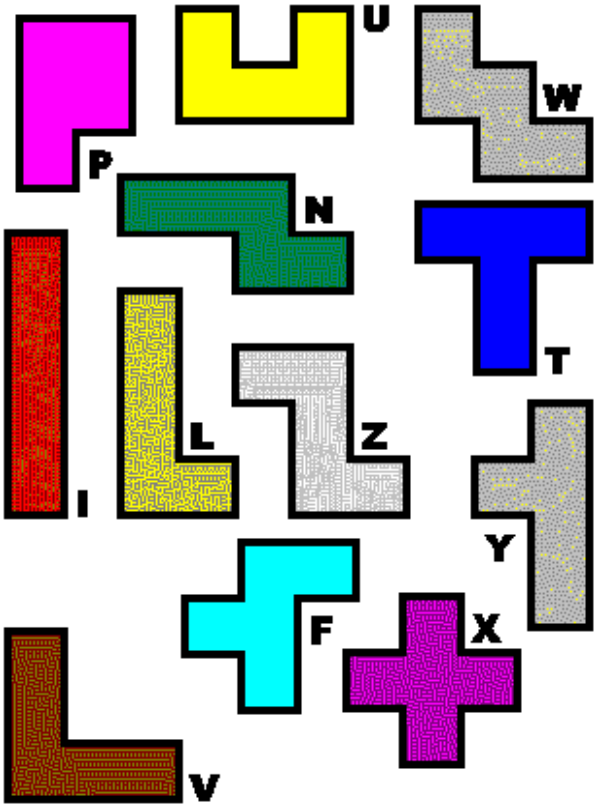

Figura 1: as doze peças

A secção 2 apresenta o puzzle Pentóminos. A secção 3 explica detalhadamente os passos envolvendo a concretização do Algoritmo Genético, incluíndo a representação utilizada, a correcção aplicada à função de mérito, e as estratégias não convencionais utilizadas. Na secção 4 apresentam-se resultados da execução do programa, para determinado conjunto de parâmetros. Por fim, na secção 5 são tiradas algumas conclusões e traçam-se direcções futuras.

2. O PUZZLE PENTÓMINOS

O puzzle pentóminos é baseado numa colecção das doze peças diferentes que resultam do agrupamento de cinco quadrados unidos pelos menos por uma aresta. A Figura 1 [1] mostra as doze peças e os nomes que lhes estão normalmente associados. É de notar que algumas peças são invariantes, umas vezes para as rotações múltiplo de 90º, e outras para o espelhamento sobre os eixos dos X ou Y.

Com estas doze peças, é possível formar, entre outras formas, rectângulos de 3x20, 4x15, 5x12 e 6x10, sem "buracos" no meio. Se se acrescentar uma décima-terceira "peça" – um quadrado de largura 2 – ou, em alternativa, quatro quadrados "soltos", pode-se resolver o puzzle 8x8. Uma possível solução para 8x8 com um quadrado 2x2 é dada na Figura 2 [1].

Neste trabalho, propomos uma implementação, utilizando Algoritmos Genéticos (AG), para o caso 8x8 em que os quatro quadrados são determinados e encontrados pelo algoritmo, e não fixos à partida.

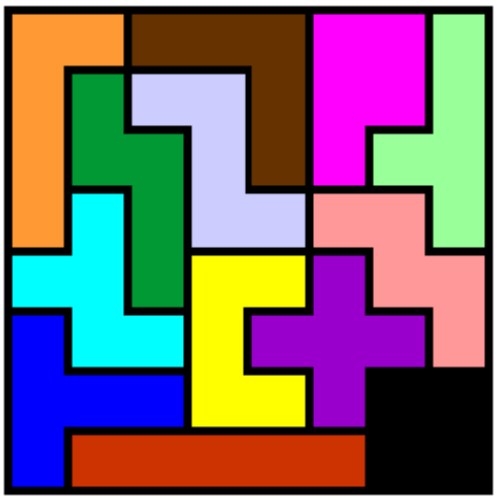

Figura 2: uma solução 8x8

3. IMPLEMENTAÇÃO DO AG

O AG opera sobre uma população de indivíduos – a que chamamos hipóteses – que são colecções de peças dispostas sobre um quadro, e eventualmente sobrepostas.

O objectivo a atingir é então uma hipótese em que todas as peças estão dispostas num quadro de 8x8, sem qualquer sobreposição entre elas.

3.1 Representação

Para cada hipótese, a representação utilizada é de posição fixa, sendo o cromossoma constituído por uma sequência de doze "genes" - cada um correspondendo a uma peça – de 9 bits de comprimento. Nestes 9 bits está codificada toda a informação sobre uma dada peça (Figura 3).

X	Y	R	E

Figura 3: um gene. Cada gene representa a posição de uma peça no quadro. R – rotação; E - espelhamento

O AG opera sobre uma população de hipóteses, dispostas sobre um quadro de 16x16, para contemplar a situção pior em que uma peça **I** existe apenas na fronteira do quadro interior e se dirige para o exterior.

Com esta representação, o número de hipóteses possíveis é então 3.25 x 10^{32} (aproximadamente 2^{9*12}). Note-se que não é feita a distinção para o caso das peças invariantes para a rotação e/ou espelhamento.

3.2 A Função de Mérito

A função de mérito aplicada a cada hipótese conta o número de casas ocupadas no quadro objectivo 8x8. Assim, o objectivo do AG é encontrar uma hipótese com mérito 60. No entanto, para efeitos de selecção, o mérito de cada hipótese é corrigido dinamicamente, de modo a aumentar a selectividade (3.3).

3.3 As etapas do AG

Depois da inicialização, e enquanto não for encontrada uma solução, ou não tiver sido atingido o número limite de gerações, são efectuadas as seguintes operações sobre a população:

- Selecção [1]: estacionária, probabilística e ponderada com o mérito corrigido relativo de cada hipótese, e elitismo de grau 1 (o melhor da população é sempre seleccionado). É feita com base, não no mérito das hipóteses, mas numa função de correcção daquele mérito, denominada NOLINEAR3, e dada pela fórmula:

$$f' = \frac{1}{5*(1-\frac{f-f_{\min}+1}{f_{\max}-f_{\min}+1})+1} - \frac{1}{6}$$

- Cruzamento: multiponto uniforme com base numa máscara aleatória criada para cada par de hipóteses cruzadas. Note-se que "ponto" neste operador significa "peça" ou gene, isto é, são trocadas peças por inteiro e não apenas bits ao acaso.
- Mutação: efectuada com uma probabilidade de mutação por hipótese, e com base numa máscara aleatória sobre os genes escolhidos. O número máximo de genes escolhidos para mutação é controlado por um outro parâmetro. Em cada gene sujeito a mutação, é escolhida aleatóriamente uma máscara de bits para inversão. No entanto, por si só, a mutação convencional quase sempre produz hipóteses de mérito baixo, e que rapidamente são elimindas na selecção seguinte. Por isso, efectuámos duas abordagens alternativas na construção deste operador.

3.4 O operador específico MutNear

Constatámos que o operador de mutação nas suas variantes tradicionais, produzia quase invariavelmente hipóteses com mérito baixo comparativamente à média das várias populações. Visto que a selecção que utilizamos é bastante restritiva (3.3), isto conduzia quase invariavelmente a que as hipóteses alternativas geradas por mutação se perdessem rapidamente nas selecções seguintes, a menos que a probabilidade de mutação fosse muito elevada, o que tendia para uma busca aleatória.

Assim, construímos uma variante do operador de mutação – denominado MutNear, ou Mutação Próxima - específica para este problema. Este operador, quando aplicado a uma hipótese, "observa" a codificação interna da representação, não a alterando arbitrariamente. Se o operador determina que um dado bit deve ser invertido, e esse bit pertence à zona da representação reservado à coordenada X (ou Y), o que acontece é que o valor de X (ou Y) é apenas incrementado de 1. Isto corresponde na prática a desviar apenas ligeiramente a peça da sua posição, sem a atirar ao acaso de novo para o quadro.

Se o bit escolhido não pertencer à zona reservada a X e a Y, é simplesmente invertido.

Apesar de interessante, esta estratégia mostrou ser pior em termos de resultados do que a mutação simples, desde que complementada com a estratégia descrita no ponto seguinte.

3.5 Esquema de Pontuações e implicações nos operadores.

Para favorecer as peças "boas", criámos um esquema de pontuações para as peças. Cada peça tem então uma pontuação associada, que vai de 0 a um certo máximo. Quando uma hipótese é avaliada, cada peça é observada e a sua pontuação altera-se da seguinte forma:

- Se a peça não estiver sobreposta com qualquer outra, e não estiver fora do quadro objectivo, a sua pontuação é incrementada de 1 (até ao máximo).
- Caso contrário, a sua pontuação é decrementada de 1 (até ao mínimo, 0).

Esta pontuação tem efeitos práticos na aplicação dos operadores Cruzamento e Mutação. No caso do Cruzamento – e para um dado par de hipóteses a cruzar – este só se opera sobre aquelas peças (indicadas pela máscara aleatória) cuja pontuação na hipótese de maior mérito seja zero. No caso da mutação, esta só se aplica às peças de pontuação nula, isto é, às peças menos "boas".

Em qualquer dos casos, na prática, isto corresponde a proteger as peças com pontuação não nula.

4. Resultados Obtidos

Os primeiros resultados apresentados são para uma população de 100 hipóteses, probabilidade de mutação (operador Mutação Próxima) de 0.8, número de mutações máximo por hipótese 2, e 100000 como número máximo

de gerações para encontrar uma solução. A probabilidade de mutação é alta visto que a correcção da função de mérito impõe uma selectividade elevada, e doutra forma o algoritmo facilmente ficaria encurralado em máximos locais [2]. Foram efectuados 126 ensaios. Na Figura 4 são apresentados, em percentagem, os resultados obtidos pelo AG.

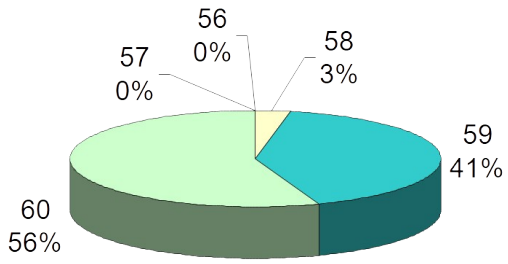

Figura 4: pertentagem de vezes que o AG parou num determinado mérito, com o máximo de 100000 gerações

A Figura 5 mostra o número médio de gerações que o AG demora até obter um dado mérito (são mostrados os valores 57, 58, 59 e 60). As várias médias dizem respeito, respectivamente, à média total, à média sempre que se encontrou uma solução, e às médias sempre que o AG parou com méritos 59 e 58. Observa-se que não existe variação significativa entre estas médias.

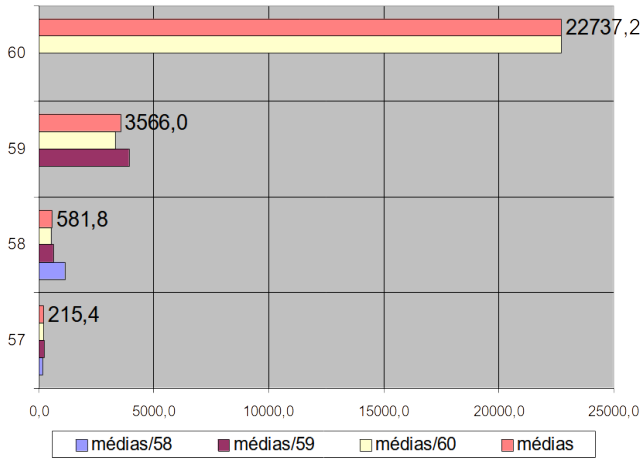

Figura 5: número médio de gerações até atingir um certo mérito

Se se alterar o operador mutação para a mutação convencional, com o esquema de pontuações, e aumentar a probabilidade de mutação para 1.0, o resultado é um surpreendente 100% de soluções em 100 ensaios, sendo 15743 o número médio de gerações para encontrar uma solução.

5. Conclusão e Melhoramentos Futuros

Propusemos uma implementação de um Algoritmo Genético para resolver o puzzle Pentóminos. Utilizando um operador para a mutação que não é indiferente à informação contida na representação utilizada, e juntamente com um esquema de pontuações para favorecer os indivíduos da população aparentemente mais próximos de uma solução, obtivemos resultados que indicam que, apesar da natureza do problema, esta metodologia também a este se adequa.

No futuro, pensamos utilizar uma estratégia, simplesmente denominada "Cristalização", para esgotar ainda mais depressa uma hipótese encurralada num máximo local. Criámos também um outro operador, já implementado, mas ainda não testado, que complementa o cruzamento, mas que é mais sensível ao problema, e que, numa só hipótese, troca entre si a posição origem de duas ou mais peças.

6. Agradecimentos

Agradecemos ao Professor Agostinho da Rosa, do Instituto de Sistemas e Robótica, pelo apoio prestado e pela paciência com que leu as inúmeras perguntas enviadas por correio electrónico.

Ainda aos fisioterapeutas Vera Farinha (Serviços Médico-Administrativos e Sociais) e Filipe Capitão (Centro de Medicina Física e Reabilitação de Santo Amaro de Oeiras Dr. José A. Capitão), os conselhos lúcidos e isentos de pensamento informático.

7. Bibliografia

1. Rosa, A. *"Algoritmos Genéticos e Vida Artificial"*. Material de suporte à cadeira com o mesmo nome no curso de Mestrado em Engenharia Electrotécnica e de Computadores, 1997/98, Instituto Superior Técnico.
2. Wolfrum, S. *"Stefan Wolfrum's Mind Puzzles Page"*. Em http://titan.cs.uni-bonn.de/~wolfrum/puzzle.html.
3. Miramontes, P. *"Genetic Algorithms vs In Vitro Selection?"*. Departamento de Bioquímica, Universidade de Montreal, Canada.
4. Allen Lima J., Gracias N., Pereira H., Rosa A. *"Fitness Function Design for Genetic Algorithms in Cost Evaluation Based Problems"*. Instituto de Sistemas e Robótica, Lisboa, Portugal.
5. Petridis V. and Kazarlis S. *"Varying Quality Function in Genetic Algorithms and the cutting Problem"*.Dpto de Eng. Electrotécnica, Faculdade de Engenharia, Universidade de Thessaloniki, Grécia.
6. Carreira C. *"Algoritmos Genéticos em «Game Playing»"*. Proceedings do 1º workshop de Algoritmos Genéticos e Vida Artificial, Instituto Superior Técnico, 1996.