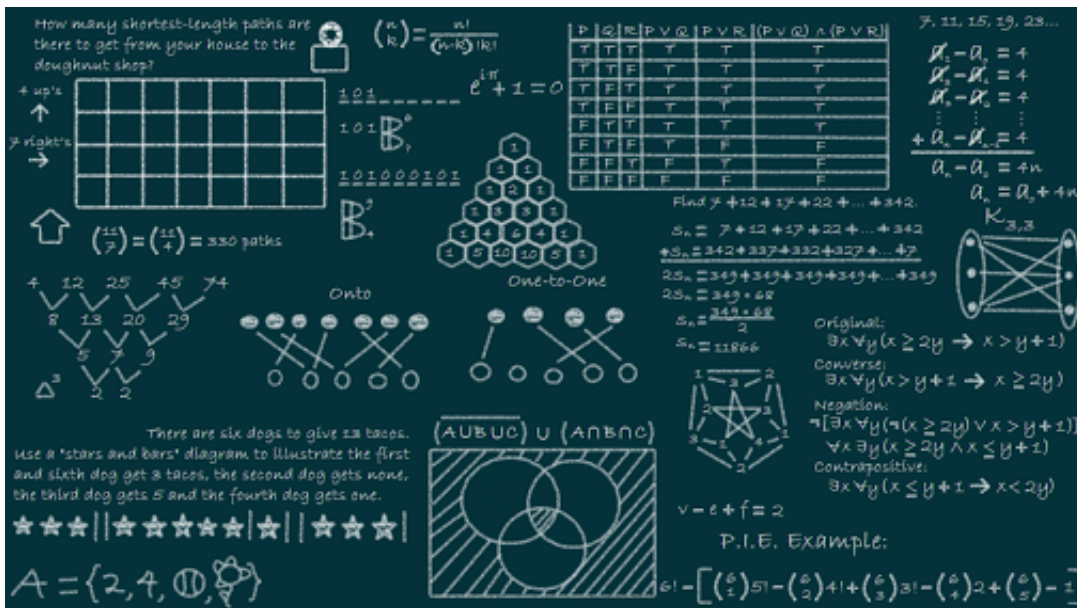


A Complexidade Irredutível nos Algoritmos Genéticos

Dedicado a [Diogenes Mota](#)

Este pequeno texto expõe algumas ideias claras para programadores sobre algoritmos genéticos, foi feito a partir de dois e-mails que enviei em 2015.



Uma definição geral do que é um algoritmo genético pode ser encontrada na Wikipédia:

Um algoritmo genético (AG) é uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca, fundamentado principalmente pelo americano John Henry Holland. Algoritmos genéticos são uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação (ou crossing over).

O sucesso do algoritmo genético é enganoso. Até mesmo intuitivamente a noção que o programador tem de economia de complexidade de tempo e complexidade de espaço aponta para um esquivo de tais soluções. Como diz Abel¹:

"Métodos computacionais muitas vezes empregam algoritmos genéticos (GA).

O apelo a eles é porque são modelados conforme a evolução biológica. Esta última é a principal motivação para se tolerar tal processo desagradável e ineficiente."

Ou seja, a principal motivação é realmente utilizar-se do algoritmo específico, o que depois é utilizado como exemplificação do sucesso da ideia darwiniana. Na verdade o sucesso dos algoritmos é proporcional a utilização do que Ewert et al. chamaram de **Informação Ativa**². Normalmente sempre uma entrada anterior de informação determinante no sistema, independente a natureza do experimento, principalmente quando existe esta busca por adequação:

*Isto porque, a fim de desenvolver um algoritmo genético bem sucedido, o desenvolvedor faz muitas decisões baseadas no conhecimento do problema que ele está tentando resolver. Como tal, a informação é derivada a partir deste conhecimento prévio do problema de pesquisa. **Ajustar** um algoritmo genético é outra fonte de informação ativa: é comum simplesmente porque funciona. A razão pela qual ele funciona é porque explora próprio conhecimento do programador no problema a ser resolvido.*

Como podemos ver acima a palavra "ajustar" está em negrito. Por que? Agora vem a *Complexidade Irredutível* presente nos algoritmos genéticos. Um sistema de variáveis aleatórias provavelmente nunca alcançaria qualquer estabilidade dentro dos recursos computacionais, dependendo do tamanho da entrada de dados. Um sistema com variáveis baseadas nas tendências da natureza seria decepcionante demais. A solução é o que chamam de **tuning** do algoritmo.

O **tuning** é o "ajuste". Ele faz "busca" ou "escolha" por parâmetros necessários e suficientes para o funcionamento do Algoritmo Genético. Encontrar os valores de parâmetros adequados para algoritmos evolutivos é um dos grandes desafios que persistem do campo de Computação Evolucionária.

Mas qual a natureza desse **tuning**?

*Encontrar um bom conjunto de valores de parâmetros é uma tarefa de otimização complexa, com uma função não-linear objetiva, variáveis que interagem, múltiplos ótimos locais, ruído (pela natureza estocástica dos Algoritmos Evolutivos serem ajustados), e uma falta de solucionadores de análise. **Ironicamente, é exatamente esse tipo de problemas, onde os Algoritmos Ev***

olutivos *são* *s*
olucionadores *heurísticos* *mu*
ito *competitivos.*

É, portanto, uma idéia natural para usar uma abordagem evolutiva para otimizar os parâmetros de um algoritmo evolutivo.

Interessante que ele nem percebe que está gerando um algoritmo da mesma natureza para resolver o problema do algoritmo inicial, sendo assim um verdadeiro meta-algoritmo.

Conclusão

Em suma, os algoritmos genéticos necessitam de uma solução qual eles mesmos estão propondo resolver. Isso caracteriza, ironicamente, uma necessidade circular (*complexidade irreduzível*).

Referências

- [1] **Abel, David L.** "The capabilities of chaos and complexity." International journal of molecular sciences 10.1 (2009): 247-291.
- [2] **Ewert, Winston, William A. Dembski, and Robert J. Marks II.** "Climbing the Steiner Tree--Sources of Active Information in a Genetic Algorithm for Solving the Euclidean Steiner Tree Problem." BIO-Complexity 2012 (2012).
- [3] **Smit, Selmar K., and Agoston E. Eiben.** "Comparing parameter tuning methods for evolutionary algorithms." Evolutionary Computation, 2009. CEC'09. IEEE Congress on. IEEE, 2009.