

ALGORITMOS GENÉTICOS

Evolução

Iremos agora analisar uma alternativa à técnicas usuais de busca baseada no conceito de *evolução*.

- Algoritmos baseados no conceito de evolução darwiniana foram introduzidos por Holland, em 1975.
- Idéia básica: simulação de processos evolutivos em sistemas artificiais poderiam ser usados para otimizar uma dada estrutura.
- Holland formalizou matematicamente o conceito de evolução, e explicou porque estes algoritmos genéticos poderiam operar melhor do que, por exemplo, busca aleatória.
- Algoritmos genéticos são a realização de mecanismos evolutivos em sistemas artificiais.

Genes e Alelos na Natureza

Na natureza, cada gene pode ter várias formas alternativas (*alelos*) que produzem diferenças no conjunto de características associadas ao gene. Por exemplo, ervilhas têm um único gene que determina a cor da florescência, um alelo pode determinar cor branca, o outro a cor rosa.

Vertebrados tipicamente têm milhares de genes / cromossomo, cada um com vários alelos. O conjunto de cromossomos, considerando-se todas as possíveis alelos para cada gene, tipicamente tem 10^{3000} indivíduos ... Mesmo uma população de 10 bilhões de indivíduos representa portanto uma minúscula fração do total de possibilidades.

Fica difícil acreditar em evolução nestes termos, mas existe um pequeno detalhe adicional ...

Coadaptação de Alelos e Evolução

Alelos interagem entre si: adaptação não é uma busca de indivíduos, mas uma busca de conjuntos de alelos coadaptados.

Cada indivíduo exemplifica um grande número de possíveis *padrões* de alelos coadaptados, ou *esquemas*.

“Testando” um indivíduo, estarei testando também todos os esquemas dos quais este indivíduo é uma instância.

Este paralelismo intrínseco é o “pulo do gato” para que algoritmos genéticos e processos evolutivos em geral) possam produzir resultados em tempo não-astronômico.

Componentes de um Algoritmo Genético

Objetivo: otimização de estruturas, cada estrutura correspondendo uma cadeia de genes (cromossomo).

Formalmente, seja \mathcal{A} o conjunto de possíveis estruturas, representadas por *strings* de comprimento fixo, e seja $v : \mathcal{A} \rightarrow \mathcal{R}^+$ uma medidade de desempenho (*fitness*). O objetivo é achar uma estrutura $a \in \mathcal{A}$ que maximize v .

Uma estrutura particular é chamada de *fenótipo*, e sua representação como uma *string* é seu *genótipo*.

Cada componente do genótipo é um gene, que pode assumir um valor correspondente a cada um de seus alelos. Na prática, usamos apenas dois alelos por gene (Por que?)

ILUSTRAÇÃO

Operação do Algoritmo Genético

Inicialmente, gere uma população aleatória de M indivíduos $S(0) = \{s(1,0), \dots, s(M,0)\}$. A seguir,

1. Para cada *string* $s(i,t) \in S(t)$, calcule e armazene sua medida de *fitness* $v(s(i,t))$.
2. Para cada $s(i,t) \in S(t)$, calcule a probabilidade de seleção definida por

$$p(i,t) = \frac{v(s(i,t))}{\sum_i v(s(i,t))} \quad (1)$$

3. Gere uma nova população $S(t+1)$ selecionando estruturas de $S(t)$ através da probabilidade de seleção e aplicando os operadores genéticos.
4. Volte ao passo 1.

Operadores Genéticos Básicos

Vamos representar uma string (genótipo) como $a_1a_2a_3 \dots a_l$, com $a_i \in \{0, 1\}$

Crossover Um ou dois pontos são selecionados aleatoriamente em dois indivíduos selecionados, e a operação de crossover é executada para a geração de dois filhos. ILUSTRAÇÃO.

- Variação: seleção de vários pontos de corte (caso extremo é o crossover uniforme).
- Projeto do operador de crossover é influenciado pela representação: não há garantias de que o fruto de um crossover seja um indivíduo válido. Exemplo: TSP. Soluções: mudança da representação ou do operador, conserto genético.

Mutação Alelos são alterados com uma probabilidade fixa p . Em strings longas: número de genes alterados segue distribuição de Poisson. Este operador dispersa a população e executa um tipo de exploração do espaço de busca.

ESTRATÉGIAS DE MORTE E NASCIMENTO:

Filhos por geração: alguns estudos (Davidor) sugerem que um único filho é ótimo.

Estratégias de substituição: heurística que funciona bem na prática é sempre eliminar a string de mais baixa *fitness*.

Esquemas: Definição Formal e Discussão

Seja \mathcal{A} o conjunto de todas as strings. Podemos definir subconjuntos de \mathcal{A} com atributos em comum (esquemas) usando símbolos $*$ (*don't care*) em uma string. Por exemplo $a_1 * a_3 * * \dots *$ representa o esquema de todas as strings com primeiro elemento a_1 e terceiro elemento a_3 .

Exercício 1. Suponha k possíveis alelos para cada gene, e suponha que tenhamos apenas genótipos de tamanho l . Quantos possíveis esquemas existem?

Exercício 2. Quantos esquemas existem para uma dada string? Em outras palavras, quanto são os esquemas para os quais uma dada string é uma instância?

Supondo $k = 2$ e $l = 20$, temos então que:

- Uma string é uma instância de 2^l esquemas, cerca de 1.04×10^6
- Existem no total $(k + 1)^l$ esquemas, cerca de 3.48×10^9 .

Ou seja, o número de esquemas amostrados é pequena fração do total de esquemas. Para população de M strings, número de esquemas amostrados $\leq M2^l$. Pelo menos é exponencialmente maior do que o tamanho da população.

A razão entre o número máximo de esquemas instanciados pela população e o total de esquemas é

$$N\left(\frac{2}{k+1}\right)^l \tag{2}$$

que tende a zero quando $l \rightarrow \infty$. Note que esta razão é mais favorável para k pequeno ...

Teorema do Esquema: Definições Preliminares

- A *fitness* observável de um esquema ξ é a média

$$v(\xi, t) = \frac{1}{N(\xi, t)} \sum_{s \in I} v(s) \quad (3)$$

onde $I = \xi \cap Pop_t$ e $N(\xi, t)$ é o total de strings em I . Observe que $v(\xi, t)$ indica portanto a *fitness* média dos indivíduos da população atual que são instâncias do esquema ξ . Pergunta: por que chamo isso de *fitness* OBSERVÁVEL do esquema ξ ?

- Uma geração é o tempo esperado para que um indivíduo produza um descendente. A unidade de tempo t corresponderá a gerações. Assim, entre t e $t+1$ cada indivíduo terá produzido, em média, um filho. definimos $P(\xi, t)$ como a parcela da população na geração t que é uma instância do esquema ξ .
- Entre gerações sucessivas, a nova população é mantida separada da antiga população. Ao final de cada geração, a nova população substitui a antiga. Esta é uma condição necessária para enunciarmos o Teorema do Esquema, mas naturalmente não precisamos ser tão radicais em problemas reais ...

Teorema do Esquema

Suponha um mecanismo em que um dos pais é selecionado de acordo com uma prob. baseada em sua *fitness*. Para um crossover arbitrário, temos:

$$\langle P(\xi, t + 1) \rangle \geq H \frac{v(\xi, t)}{\langle v(t) \rangle} P(\xi, t) \quad (4)$$

onde $0 \leq H \leq 1$ é a *probabilidade de sobrevivência do esquema*, isto é, a probabilidade de que, sendo um pai uma instância de ξ , o filho também o será. Em geral, H depende da estrutura genotípica, do operador crossover utilizado e do mecanismo de seleção do segundo pai.

Observe que se a *fitness* observada de ξ for muito maior do que a *fitness* média da população, ou seja, se

$$H \frac{v(\xi, t)}{\langle v(t) \rangle} > 1 \quad (5)$$

então a proporção da população em ξ tende a crescer EXPONENCIALMENTE.

Exercício: o que vai acontecer com a fração acima depois de algum tempo?

Assim, o Teorema do Esquema garante aumento de população *fit* apenas por um tempo limitado . . .

Para que o teorema do esquema se aplique fortemente a um dado problema, devemos satisfazer:

1. Ter um H próximo de 1. Em alguns casos, limites inferiores para H podem ser definidos em termos de parâmetros controláveis.
2. Ter crossover que garanta “mistura 50:50” da informação genética dos pais. Isto evita que o crossover se assemelhe a duplicação, cujas garantias de geração de populações *fit* são mais restritas.

Algoritmos Genéticos: Aspectos de Projeto

Como método de busca, um algoritmo genético tem as seguintes vantagens:

- Indivíduos são selecionados mais e mais segundo esquemas instanciadores de fenótipos de maior *fitness*, com garantias teóricas pelo menos durante um tempo limitado.
- Desde que a população seja suficientemente grande, a possibilidade do algoritmo ficar preso em mínimos locais é baixa, já que a base de conhecimento (população de estruturas) é distribuída por todo o espaço de busca.

Apesar disso, a teoria matemática de AGs é bem incompleta, a não ser em casos muito simples. Precisamos usar algumas heurísticas para guiar um projeto baseado em AGs.

Um projeto pode exigir uma das seguintes três características de um AG:

- Convergência para um ótimo global (melhor solução).
- Produção de várias soluções boas (mas subótimas).
- Adaptação em tempo real à uma função de *fitness* variante no tempo v_t .

Convergência para Solução Ótima

Muito difícil garantir . . .

Teoria de AGs exige grandes populações, nem sempre facilmente geradas ou avaliadas em problemas típicos de IA → populações pequenas tendem a produzir *convergência prematura* (para soluções fracas) e perda de diversidade. Fenômeno observado na natureza: variações alélicas de baixa frequência produzindo populações isoladas com fenótipos de grupo sanguíneo diferentes de população maior.

Compensar com aumento da taxa de mutação NÃO é boa idéia: mutação excessiva elimina contribuição do crossover em sua busca de esquemas *fit* e faz o algoritmo se assemelhar a uma *busca exaustiva*.

Compensar convergência prematura é uma instância do conflito entre *exploração* das soluções encontradas e *exploração* do espaço de busca, necessária para que soluções melhores sejam encontradas.

Produção de Soluções Boas

Várias propostas:

- Cavicchio, 1970: Mecanismos de pré-seleção, tais como substituição de pais pelos melhores filhos, *etc*
- De Jong, 1975: Mecanismo *crowding*. Filhos gerados substituem strings similares, obtidas de sub-populações de tamanho fixo. Introduce pressão evolutiva proporcional à similaridade entre membros da população.
- Mauldin, 1984: Mecanismo de censura. Um operador de *unicidade* ajuda a manter diversidade, evitando que filhos excessivamente similares a outros membros da população sejam gerados.
- Goldberg, 1987: Mecanismo de *sharing*. Nichos prematuros (produzidos por strings com utilidade alta concentradas em uma região pequena) são evitados por um mecanismo diversificador que aumenta a probabilidade de seleção de nichos de baixa densidade populacional.

Adaptação à $v = f(t)$

Melhor não se preocupar muito com isso . . .

- Algoritmos genéticos tipicamente necessitam de muitas gerações para convergir, o que demanda tempo (especialmente para o cálculo da *fitness* da população e seleção de indivíduos).
- Adaptação requer convergência relativamente rápida.
- Convergência rápida em AGs só se o problema for relativamente simples. Nesse caso, talvez uma técnica de busca mais simples ou teoricamente mais sólida resolva o problema . . .

Isto não quer dizer que pesquisa de métodos para aumentar a velocidade de convergência de AGs não seja útil. Pelo contrário, AGs se prestam bem à implementações paralelas, o que certamente permite aumento efetivo de populações.

Exemplo: algoritmo AG paralelo Georges-Schlutter resolve o TSP para 520 cidades em 10 minutos . . .