

Tópicos em Computação Evolucionária

Computação Natural

Gisele L. Pappa

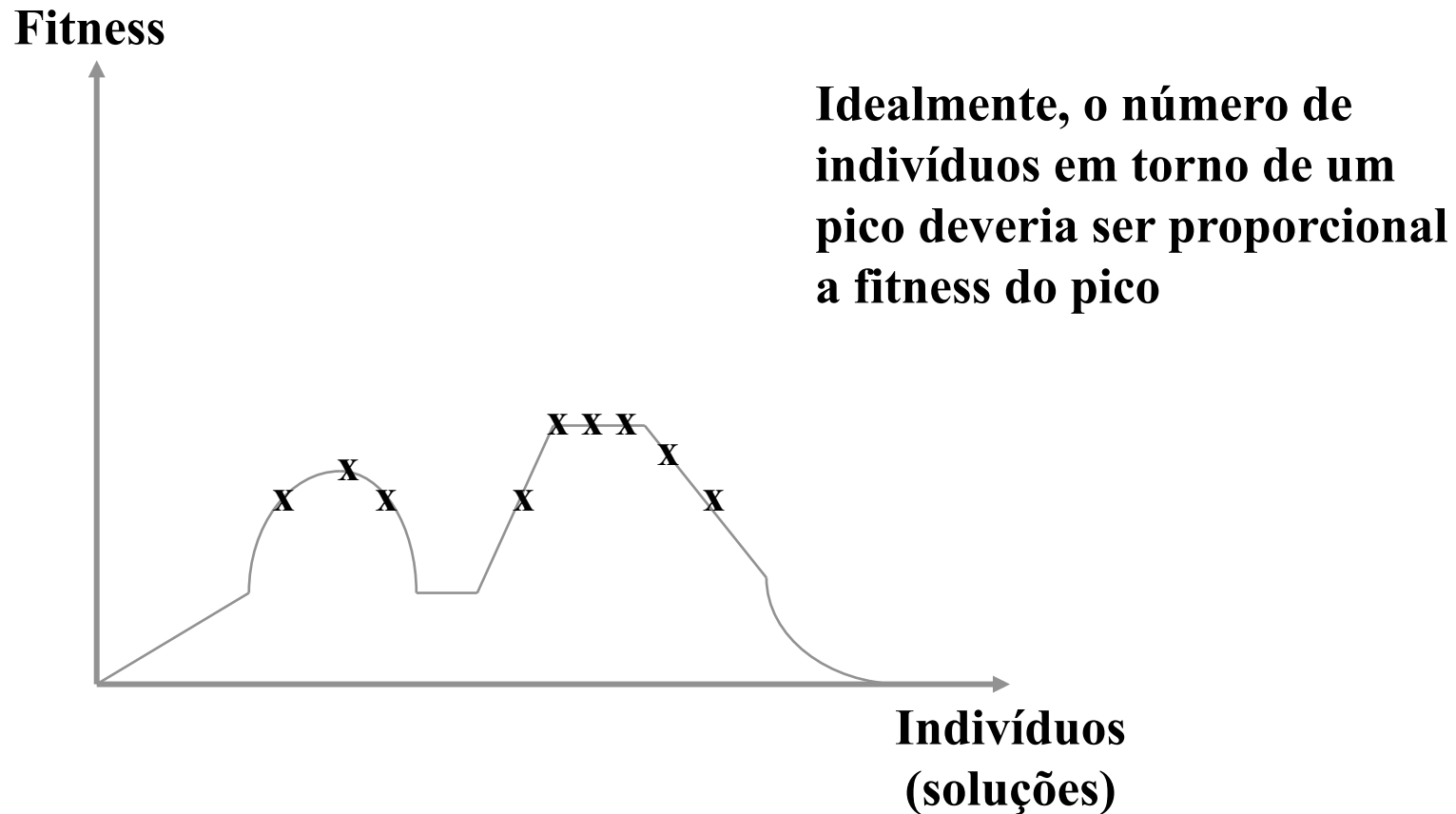
Tópicos

- Nichos (*niching*)
 - *Fitness Sharing* (método de *niching* mais popular)
 - *Crowding*
- Conceito de espécies (complementar a *niching*)
- Co-evolução
 - Competitiva
 - Cooperativa
- EAs paralelos

Introdução a métodos de Niching

- Evolução natural mantém a diversidade das espécies
 - Cada espécie ocupa um nicho ecológico
- Em algoritmos evolucionários, a população normalmente converge para uma população uniforme, com muitas cópias do mesmo indivíduo
- Solução: modificar os algoritmos para utilizar um método de *niching*, simulando uma competição por recursos limitados

Introdução a métodos de Niching



Motivação para Niching

- Reduz a velocidade de convergência da população para um único indivíduo (evitando convergência prematura)
- Encontra um conjunto de soluções ótimas ou quase-ótimas, ao invés de uma solução ótima
- Onde utilizar niching
 - Otimização de funções multi-modais (muita soluções ótimas)
 - Otimização de funções multi-objetivas (soluções avaliadas de acordo com vários critérios)

Métodos de Niching

- 2 tipos principais:
 - *Fitness sharing*
 - *Crowding*

Fitness Sharing (1)

- Modifica apenas a maneira como a fitness de um indivíduo é determinada, o resto da evolução não é alterada
- Seja F_i a fitness original do indivíduo i
 F_i' a fitness compartilhada de um indivíduo i
 NC_i a contagem de nicho do indivíduo i , onde
 NC mede a saturação de um nicho

$$F_i' = F_i / NC_i$$

Fitness Sharing (2)

Seja SH uma função de compartilhamento, medindo a **similaridade** entre dois indivíduos

A contagem de nicho (NC) de um indivíduo é a soma das funções compartilhadas (SH) (das similaridades) entre o indivíduo i e todos os indivíduos da população (incluindo ele mesmo), ou seja

$$F_i' = \frac{F_i}{\sum_{j=1}^N SH(i,j)}$$

onde N é o número de indivíduos da população

Fitness Sharing (3)

- SH retorna um número entre $[0..1]$
 - Similaridade é inversamente proporcional a distância
- Seja d_{ij} a distância entre dois indivíduos i, j
 - Se $d_{ij} = 0$ (os indivíduos são idênticos), o valor da SH é 1
 - Se d_{ij} é maior ou igual a um limiar de distância, θ_{share} , o valor da SH é 0 (i e j estão em nichos diferentes)
 - Se $0 < d_{ij} < \theta_{share}$, então a função retorna um valor intermediário, entre 0 e 1

$$SH(d_{ij}) = \begin{cases} 1 - (d_{ij} / \theta_{share})^\alpha, & \text{se } d_{ij} < \theta_{share} \\ 0, & \text{nos outros casos} \end{cases}$$

- α é um parâmetro, que normalmente recebe valor 1

Fitness Sharing (4)

- 2 maneiras de medir a distância entre dois indivíduos
 - Compartilhamento de fenótipo ou de genótipo
 - Compartilhamento de **genótipo**: considera o material genéticos dos indivíduos
 - Exemplo: distância de hamming (se codificação binária é utilizada)
 - Número de bits diferentes
- | | | | | | | | | |
|----|---|---|---|---|---|----------------|----------------|---------------|
| 1) | 1 | 0 | 1 | 1 | 0 | $d_{1,1} = 0;$ | $d_{1,2} = 2;$ | $d_{1,3} = 1$ |
| 2) | 1 | 0 | 1 | 0 | 1 | | | |
| 3) | 0 | 0 | 1 | 1 | 0 | | | |

Fitness Sharing (5)

- Compartilhamento de fenótipo: considera os indivíduos decodificados (soluções candidatas)
 - Exemplo: decodificar 5 bits em uma variável x
- $d_{i,j} = |x_i - x_j|$, onde $|x|$ é o valor absoluto de s

material genético	decodificando x	
1) 1 0 1 1 0	22	$d_{1,1} = 0; d_{1,2} = 1; d_{1,3} = 16$
2) 1 0 1 0 1	21	
3) 0 0 1 1 0	6	

- O indivíduo 1 é mais similar ao indivíduo 2 que ao indivíduo 3 de acordo com a distância de fenótipos. Porém, indivíduo 1 é mais similar ao indivíduo 3 que ao 2 de acordo com a distância de genótipos.

Fitness Sharing (6)

- Exemplo do efeito da fitness compartilhada
(quanto maior a fitness, melhor o indivíduo)

$indiv_i$	F_i	NC_i	F_i'
1	10	2.5	4
2	12	4	3
3	6	1	6
4	8	2	4

O valor de F_i' é utilizado para seleção.

- Indivíduos 1 e 2 tem uma fitness original alta, mas são penalizados pelo valor alto de NC
- Indivíduo 3 tem a melhor função compartilhada

Fitness Sharing (7)

- Vantagens de fitness sharing
 - Conceitualmente simples, boa metáfora do niching natural
 - Tenta distribuir indivíduos proporcionalmente ao fitness do pico onde se encontram
- Desvantagens
 - Dificuldade de ajustar o valor do parâmetro θ_{share} : idealmente, requer conhecimento sobre o número e o tamanho dos picos, o que não é realista em problemas difíceis
 - Solução possível: ajustar dinamicamente o valor de θ_{share} durante a busca
 - Computacionalmente caro: precisa calcular a distância entre pares de indivíduos
 - Solução possível : computar o valor da função de compartilhamento baseado em uma amostra da população

Crowding

- Idéia básica: os novos indivíduos são inseridos na próxima geração substituindo pais similares
- Ao contrário de *fitness sharing*, *crowding* não aloca indivíduos proporcionalmente a fitness do pico
- Utiliza uma função de distância para determinar quando dois indivíduos são similares

Crowding Determinístico

- Agrupa os indivíduos da população em pares
- Cruza todos os pares
 - probabilidade de cruzamento = 1
- Muta os indivíduos gerados pelos pares
- Cada indivíduo compete com um pai
 - Indivíduos são forçados a competir com o pai menos distante
- O vencedor é inserido na nova população

Espécies

- Niching foca em distribuir indivíduos em diferentes picos, mas não na busca “dentro” do pico
- Problema: crossover pode trocar genes de indivíduos em picos diferentes, produzindo soluções ruins que não estão associadas a nenhum pico
- Solução: utilizar um método de “especialização” para restringir crossover a indivíduos similares, ou seja, indivíduos no mesmo pico/nicho
- O conceito de espécies complementa o niching, ajudando-o a encontrar e manter múltiplos picos

Co-evolução

- Em alguns problemas, o conceito de espécies é utilizado juntamente com o conceito de co-evolução
- Co-evolução na natureza
 - Influência evolucionária mútua entre duas espécies.
 - Alguns pesquisadores acreditam que todo tipo de evolução é, na verdade, alguma forma de co-evolução

Co-evolução

- Exemplos de co-evolução
- 2 tipos:
 - Cooperativa
 - Flores e insetos
 - Mitocôndria e a célula
 - Competitiva
 - Relações presa/predador ou parasita/hospedeiro

Co-evolução em Algs. Evolucionários

- Tipos de co-evolução:
 - por forma de interação: competitiva · cooperativa
 - por forma de organização: inter-populacional · intra-populacional
- Problemas:
 - das criaturas virtuais de Sims: competitivo e intra-populacional (vídeo)
 - evolução de estratégias de jogo para o dilema do prisioneiro: competitivo e inter-populacional.

Co-evolução em Algs. Evolucionários

- O fitness de um indivíduo depende de outros indivíduos (da interação com outros indivíduos).
 - O fitness de um indivíduo pode variar mesmo que o indivíduo não varie.
 - Mudança em um indivíduo pode acarretar alteração no fitness de outros.

Co-evolução Competitiva

- Aplicação mais comum: jogos
 - Imagine um algoritmo onde cada indivíduo representa a estratégia de um jogador
- Para uma estratégia ser robusta, é mais interessante que ela seja testada em um conjunto dinâmico de jogadores
- Co-evolução competitiva
- Com o passar do tempo, melhor estratégia melhora, mas as outras estratégias também melhoram

Co-evolução Competitiva

- Seu sucesso depende da estratégia de competição:
 - Um conta todos
 - Um contra um número k de indivíduos aleatórios
 - Todos contra a melhor
- A organização populacional também têm papel importante no desempenho do algoritmo (i.e. uso de uma ou mais populações)

Vantagens da Co-evolução

- Ajuda a manter a diversidade na população
- Permite construir soluções complexas de forma incremental
- Pode acelerar o processo evolutivo;
- Tende a reduzir a necessidade de conhecimento sobre o problema

Desvantagens da Co-evolução

- Difícil de calibrar para se chegar aos resultados esperados
- Pode apresentar comportamentos de alta complexidade e não-intuitivos
- Não necessariamente admite um “propósito evolutivo” absoluto, apenas relativo (*arms race*)
 - Arms race é qualquer competição em que não existe objetivo absoluto, apenas o objetivo relativo de permanecer à frente de seus competidores.
- Requer a definição de métricas que atestem (ou não) progressos na busca por soluções de alta qualidade.

Onde aplicar co-evolução ?

- Problemas onde existe dificuldade no cálculo do fitness:
 - testar todos os casos: custoso ou intratável.
 - testar um subconjunto de casos: qual?
- Problemas sem nenhuma função de fitness conhecida.
- Problemas modularizáveis: dividir-para-conquistar

Algoritmos Paralelos

- 3 abordagens principais
 - Paralelização global
 - Paralelização em alto nível de abstração
 - Modelo da ilha (*island model*)
 - Paralelização em baixo nível de abstração
- Atualmente, paralelização de AEs em GPUs estão em alta.

Paralelização Global

- O algoritmo genético é executado em uma máquina e a fitness distribuída em diversas máquinas.
- Eficiente apenas quando o custo maior do algoritmo está no cálculo da fitness.

Paralelização em Alto Nível

- Mais utilizada
- População é dividida em múltiplas sub-populações
- Evolução ocorre de maneira isolada, com algumas fases de migração
- Qualidade da performance é influenciada pelo número e tamanho de *demes* (sub-populações) disponíveis, além do material genético trocado durante as fases de migração

Paralelização em Baixo Nível

- Utiliza muitas sub-populações com poucos elementos
- Appropriada para estruturas de supercomputadores
- Área de pesquisa sendo explorada é a de criação de modelos híbridos
 - Combinam paralelização em alto e baixo níveis

Migração

- Um dos fatores determinantes no sucesso de algoritmos evolucionários paralelos
- 3 problemas
 - Quando migrar
 - Quem migrar
 - Quantos migrar
- Migração normalmente ocorre em intervalos fixos de tempo
- Maioria dos artigos usa uma taxa de migração de 5% a 20% da população
 - Depende do problema

Leitura Recomendada

- Sareni, B.; Krahenbuhl, L., "Fitness sharing and niching methods revisited," *IEEE Transactions on Evolutionary Computation*, vol.2, no.3, pp.97-106, 1998
- Solution concepts in coevolutionary algorithms
2004, Sevan Gregory Ficici , Brandeis University, USA

Agradecimentos

- Alguns dos slides foram retirados das aulas de Computação Natural ministradas por Alex Freitas.