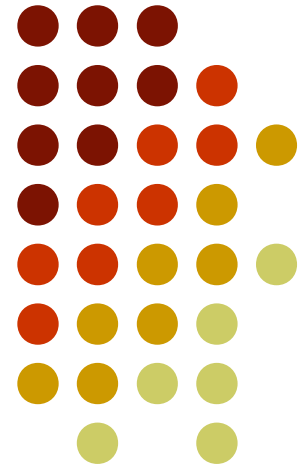


PGCA009 – Inteligência Computacional Aula 10 Tópicos Avançados em CE

Prof. Angelo Loula
Mestrado em
Computação Aplicada (UEFS)



Computação Evolutiva



- Niching
- Co-evolução
- Evolução de Redes Neurais

Computação Evolutiva



- Niching
 - Formação de grupos de indivíduos em uma população
 - Indivíduos no mesmo grupo são similares e fora do grupo são diferentes
 - Auxilia a manter diversidade e explorar o espaço de busca
 - Otimizar múltiplas funções objetivos simultaneamente

Computação Evolutiva



- Niching
 - Permite construir uma assembleia de máquinas (machine assemble) que cooperam
 - Otimizar múltiplas funções objetivos simultaneamente
 - Simula sistemas complexos e adaptativos
- Tipos:
 - Sharing (fitness sharing)
 - Crowding

Computação Evolutiva



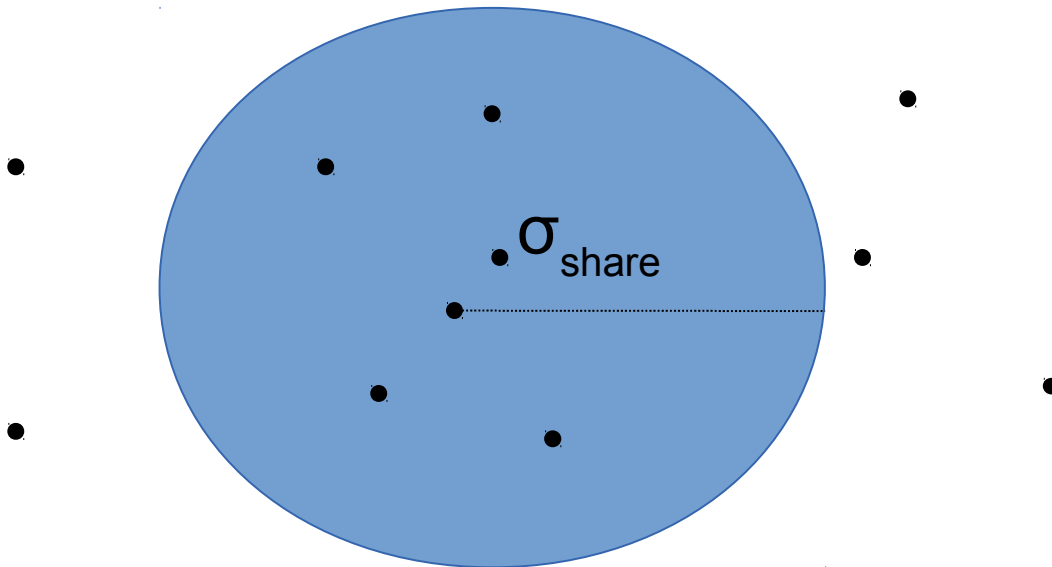
- Fitness Sharing
 - Fitness bruto transforma-se em fitness compartilhado
 - Há somente 'recursos' limitados em cada nicho, e ele deve ser compartilhado entre indivíduos no mesmo nicho
 - Permite explorar múltiplos picos em funções multimodais

Computação Evolutiva



- Fitness Sharing

- Raio de compartilhamento σ_{share} : define tamanho do nicho, indivíduos dentro do círculo deste raio devem compartilhar fitness



Computação Evolutiva



- Fitness Sharing

- função de compartilhamento $sh(d_{ij})$

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}} \right)^\alpha, & \text{if } d_{ij} < \sigma_{share}, \\ 0, & \text{otherwise,} \end{cases}$$

d_{ij} : distância entre indivíduos i e j

- fitness compartilhado

$$f_{share}(i) = \frac{f_{raw}(i)}{\sum_{j=1}^{\mu} sh(d_{ij})},$$

μ : tamanho da população

Computação Evolutiva



- Fitness Sharing
 - A distância pode ser genotípica ou fenotípica
 - Definir o raio pode ser difícil
 - Tamanho da população é importante
 - Pode não localizar todos picos e pode perder picos
 - Computar fitness compartilhado é mais custoso
 - Pode não funcionar adequadamente
 - Fitness pode precisar ser escalonado, ressaltando melhores fitness bruto

Computação Evolutiva



- Implicit Fitness Sharing
 - Para um problema com vários casos de teste
 - Escolha um caso de teste i e repita C vezes:
 - Selecione uma amostra de x indivíduos
 - Verifique o indivíduo com melhor desempenho no caso de teste i
 - Só melhor indivíduo recebe uma recompensa

Computação Evolutiva



- Niching
 - Crowding: indivíduos substituem indivíduos similares
 - Deterministic Crowding: se reproduzem com qualquer outro, mas substituem o pai mais similar se fitness for melhor
 - Competição localizada, intensificar busca em picos
 - Cuidado com diversidade na população inicial

Computação Evolutiva



- Niching
 - Speciation: complementa Niching
 - Somente indivíduos similares podem se juntar para reproduzir
 - Similaridade por tag ou por distância
 - Niching localiza picos, Speciation converge no pico

Computação Evolutiva



- Co-evolução
- Na biologia, co-evolução é o processo de duas ou mais espécies reciprocamente afetarem a evolução uma da outra.
- Exemplo: a mudança evolutiva da morfologia de uma planta pode afetar a evolução de herbívoros, que por sua vez afetam a evolução da planta, e assim por diante

Computação Evolutiva



- Co-evolução
- Acontece co-evolução entre espécies que interagem ecologicamente:
 - Presa/Predador
 - Espécies Competitivas
 - Espécies Mutualistas

Computação Evolutiva



- Co-evolução
- Ao elaborarmos um Algoritmo Evolutivo, normalmente temos um processo evolutivo isolado
 - Não há interação entre os indivíduos evoluindo com outros indivíduos também evoluindo
- Algoritmos Co-Evolutivos podem ser aplicados em Artificial Life, Jogos, Otimização, etc.

Computação Evolutiva



- Algoritmos Co-Evolutivos
- O espaço de busca não é fixo, mas acoplado
- O mesmo indivíduo pode ter fitness diferente em populações diferentes
- Passos evolutivos de um indivíduos geram mudanças no espaço de fitness de outro indivíduo

Computação Evolutiva



- Algoritmos Co-Evolutivos
- Quantidade de populações
 - Co-evolução inter-populacional: duas ou mais populações
 - Co-evolução intra-populacional: uma população
- Tipo de relação:
 - Co-evolução competitiva: competição entre indivíduos, afetando negativamente fitness
 - Co-evolução cooperativa: cooperação entre indivíduos, afetando positivamente fitness

Computação Evolutiva



- Algoritmos Co-Evolutivos
- Um exemplo:
 - Dilema do Prisioneiro
 - Dois prisioneiros A e B, um pode delatar o outro
 - Se X delata e Y cala-se, X está livre e Y tem 3 anos de prisão
 - Se X delata e Y delata, X e Y tem 2 anos de prisão
 - Se X e Y calam-se, X e Y tem 1 ano de prisão
 - Dilema do Prisioneiro Iterativo
 - Repete-se, podendo lembrar escolhas anteriores

Computação Evolutiva



- Algoritmos Co-Evolutivos
- Um exemplo:
 - Para o Dilema do Prisioneiro (sem iteração)
 - O equilíbrio de Nash é obtido quando ambos se denunciam, porque neste ponto não há benefício para mudar de estratégia individualmente

	B cala-se	B denuncia
A cala-se	A: 1 , B: 1	A: 3 , B: 0
A denuncia	A: 0 , B: 3	A: 2 , B: 2

Computação Evolutiva



- Algoritmos Co-Evolutivos
- Um exemplo:
 - O que esperar do Dilema do Prisioneiro Iterativo?
 - Projete um Algoritmo Evolutivo para co-evoluir jogadores do Dilema do Prisioneiro Iterativo com 50 rodadas
 - Jogador pode lembrar 3 últimas rodadas

Computação Evolutiva



- Algoritmos Co-Evolutivos
- Um exemplo:
 - Projete um Algoritmo Evolutivo para co-evoluir jogadores do Dilema do Prisioneiro Iterativo
 - Vetor de 70 bits
 - 64 possibilidades: 4 combinações por rodada, 3 rodadas
 - + 6 bits de memória pé-jogo para rodadas iniciais
 - 0 – calar-se; 1 – denunciar
 - Avaliação jogando com todos da população

Computação Evolutiva

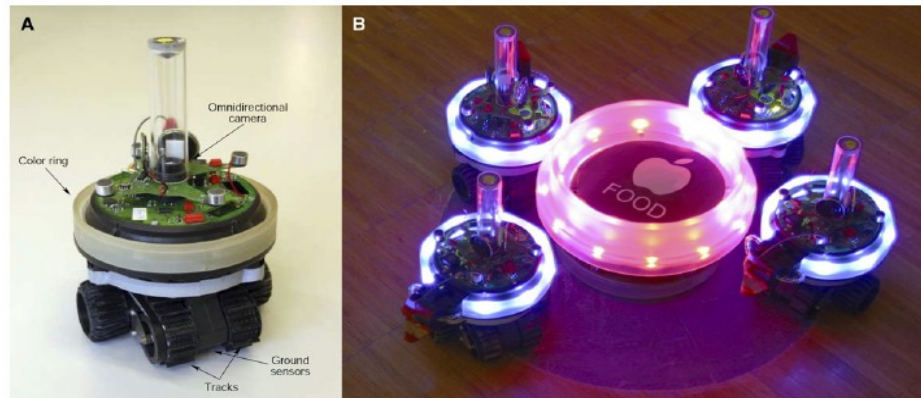


- Algoritmos Co-Evolutivos
- Um exemplo:
 - Projete um Algoritmo Evolutivo para co-evoluir jogadores do Dilema do Prisioneiro Iterativo
 - Vetor de 70 bits
 - Avaliação jogando com todos da população
 - Evolução converge para jogadores cooperativos,
Ver Darwen P. and Yao X. On Evolving Robust Strategies for Iterated Prisoners Dilemma. In Progress in Evolutionary Computation, LNAI 956: 276-292, 1995

Computação Evolutiva



- Algoritmos Co-Evolutivos
- Outro Exemplo:
 - Evolução de comunicação pela co-evolução de falantes e intérpretes



Floreano, D., Mitri, S., Magnenat, S., and Keller, L. (2007) Evolutionary conditions for the emergence of communication in robots. *Current Biology*, 17(6):514--519

Computação Evolutiva



- Algoritmos Co-Evolutivos
 - Evolução de comunicação pela co-evolução de falantes e intérprete
 - Divididos em grupos, tarefa de aproximar-se e permanecerem próximos a uma fonte de comida e não ficar próximo de uma fonte de veneno
 - Robôs poderiam usar um sinal visual, ligando ou desligando seu anel de luz azul
 - Fonte e Veneno emitem luz vermelha e robô só diferencia quando está do lado, pela cor no chão
 - Robô vê luzes e sua direção, tem sensor de chão, ativa motores das rodas e anel de luz

Computação Evolutiva



- Algoritmos Co-Evolutivos
 - Evolução de comunicação pela co-evolução de falantes e intérprete
 - Eles irão cooperar ou competir?

Computação Evolutiva



- Algoritmos Co-Evolutivos
 - Evolução de comunicação pela co-evolução de falantes e intérprete
 - Eles irão cooperar ou competir?
 - Se a seleção atual no grupo (e não indivíduo) ou se indivíduos do grupo são clones um do outro, há cooperação
 - Senão, há competição e um sistema de comunicação enganoso pode surgir

Computação Evolutiva



- Algoritmos Co-Evolutivos
 - Evolução de comunicação pela co-evolução de falantes e intérprete
 - Divididos em grupos, tarefa de aproximar-se e permanecerem próximos a uma fonte de comida e não se aproximar de uma fonte de veneno
 - Robôs poderiam usar um sinal visual, ligando ou desligando seu anel de luz azul
 - Fonte e Veneno emitem luz vermelha e robô só diferencia quando está do lado
 - Evoluir rede neurais controlar os robôs

Computação Evolutiva



- Algoritmos Co-Evolutivos
 - Quando usar co-evolução?
 - Não sabemos como calcular fitness
 - Há situações demais para testar, testa-se para um subconjunto, mas qual?
 - Situações de teste co-evoluem com soluções
 - Fitness muda ao longo do tempo
 - Aumentar e manter diversidade
 - Obter robustez e tolerância a falha
 - Mesmo quando não é preciso, pode melhorar resultados

Computação Evolutiva



- Evolução de Redes Neurais
 - Aprendizado e evolução são duas formas de adaptação
 - Pode-se combinar redes neurais com algoritmos evolutivos de muitas maneiras
 - Muitos problemas em redes neurais podem ser vistos como busca e otimização
 - Pesos das conexões
 - Funções de transferência
 - Arquitetura da rede

Computação Evolutiva



- Evolução de Redes Neurais
 - Aprendizado em rede neurais normalmente é um problema de minimização de erro
 - Métodos baseados em gradiente (como backpropagation) são muito usados
 - Porém são sensíveis aos pesos iniciais
 - São ruins para funções multi-modais e podem ficar presos em mínimos locais
 - Necessitam diferenciabilidade da função de erro
 - Podem não ajudar a generalizar

Computação Evolutiva



- Evolução de Redes Neurais
 - Algoritmos Evolutivos são métodos probabilísticos de busca
 - Adequado a problemas complexos,
 - a problemas multimodais
 - a problemas com não-diferenciabilidade
 - a problemas com não-continuidade
 - São robustos e menos sensíveis a condições iniciais
 - Podem ser aplicados sozinhos ou com métodos de treinamento de redes neurais

Computação Evolutiva



- Evolução de Redes Neurais
 - Representação
 - Pesos:
 - valores binários ou em ponto flutuante
 - Arquitetura(topologia):
 - todos neurônios e conexões,
 - ou certos parâmetros (ex: quant. neurônios na camada intermediária)
 - funções de transferência ou seus parâmetros
 - Regras de aprendizado

Computação Evolutiva



- Evolução de Redes Neurais
 - Decodifique cada indivíduo em uma rede neural
 - Treine a rede, se for o caso
 - Mas o treino não altera genótipo
 - Avalie a rede e atribua fitness
 - Baseado no erro de treino ou de validação, na complexidade da rede, outros critérios
 - Baseado na realização da tarefa proposta
 - Realize seleção e variação

Computação Evolutiva



- Evolução de Redes Neurais
 - Mutação deve ser adequada aos elementos representados para evoluir
 - Recombinação precisa ter cuidado para não ser destrutiva
 - Identificar building blocks nos elementos representados
 - ex: um crossover uniforme de cada peso não irá gerar descendentes melhores
 - Pode não utilizar recombinação

Computação Evolutiva



- Evolução de Redes Neurais
 - Existe uma variedade muito grande de métodos/algoritmos para evolução de redes neurais
 - Aplicações também muito diversificadas
 - Exemplo: Robótica Evolutiva
 - Evolução da rede neural que controla o robô
 - Co-evolução de robôs diferentes
 - Co-evolução do corpo e da rede neural de um robô
 - Evolução em simulação e transferência para realidade