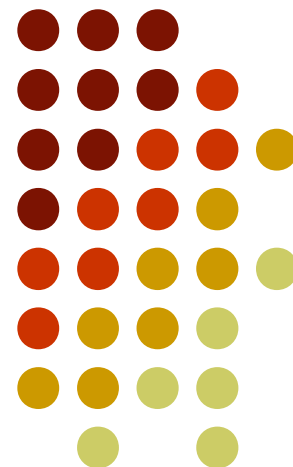


PGCA009 –Inteligência Computacional Aula 10 CE:Operadores

Prof. Angelo Loula
Mestrado em
Computação Aplicada (UEFS)



Computação Evolutiva

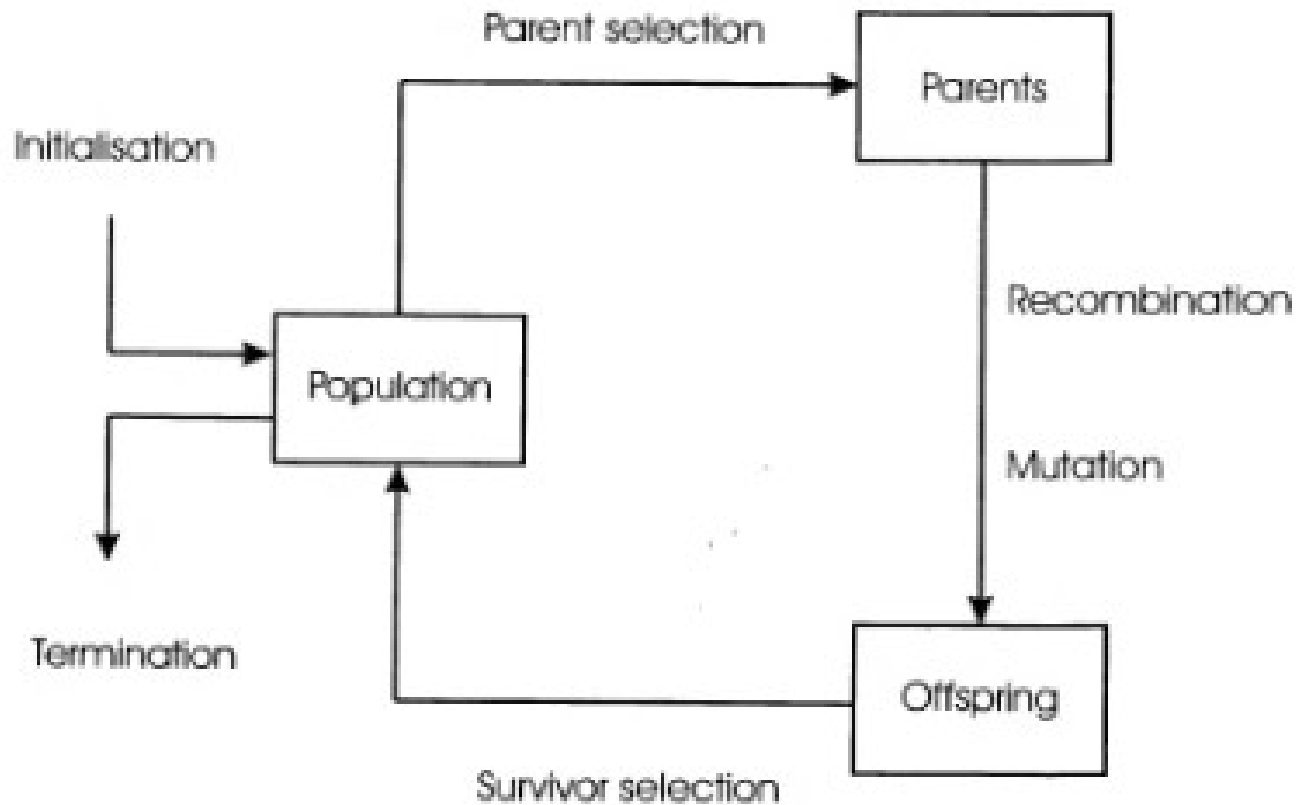


Fig. 2.2. The general scheme of an evolutionary algorithm as a flow-chart



Computação Evolutiva

- Componentes de um Algoritmo Evolutivo
 - Representação (definição dos indivíduos)
 - População
 - Função de Avaliação
 - Procedimento de Seleção
 - Progenitores (reprodução)
 - Sobreviventes (composição da nova geração)
 - Operadores de busca/variação (mutação e recombinação)

Computação Evolutiva



- Vamos conhecer mais sobre :
 - Representações
 - Seleção
 - Operadores de Busca (Mutação e Recombinação)
 - Função de Avaliação (Fitness)

Computação Evolutiva



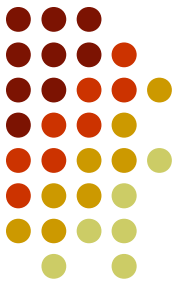
- Representações
 - Algoritmos evolutivos manipulam representações (codificações) de soluções para um dado problema
 - Esta representação depende do problema
 - Mas um mesmo problema pode ter várias representações
 - E influencia e depende fortemente dos operadores de busca
 - Uma boa escolha pode auxiliar muito
 - Uma má escolha pode dificultar muito

Computação Evolutiva



- Representações binárias
 - Em algoritmos genéticos, a representação usual é um vetor de números binários (bits)
 - Mesmo números inteiros ou reais podem ser representados por bit
 - Cuidado com a precisão
 - Representar $x \in [a, b]$ em um vetor de k bits terá precisão de $(b-a)/(2^k-1)$
 - Não tem bom desempenho para problemas numéricos, contínuos e de alta dimensionalidade

Computação Evolutiva



- Representações binárias
 - Representação binária usual: bits próximos não representam necessariamente locais próximos no espaço de soluções (genótipo x fenótipo)
 - Hamming Cliff Problem (distância de Hamming = número de bits diferentes)
 - 0111 (7) e 1111 (15), 0111 (7) e 1000 (8)
 - Codificação de Gray pode ser utilizada para posicionar cadeias adjacentes de bits de valores inteiros/reais adjacentes
 - Mas não garante resultados melhores

Computação Evolutiva



- Representações binárias

- Codificação de Gray

Dec	Gray	Binary
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

Computação Evolutiva



- Representações em ponto flutuante
 - O computador não representa adequadamente todos números reais
 - Números reais são representados pela notação de ponto flutuante
 - Problemas de otimização com variáveis reais contínuas: codificação por valores em ponto flutuante
 - Interpretação natural e direta

Computação Evolutiva



- Representações por permutações
 - Permutação de um conjunto finito: uma sequência ordenada de seus elementos, contendo cada elemento uma vez e somente uma vez
 - Dado n elementos, existem $n!$ permutações
 - Alguns problemas são problemas de busca de permutações e podem ser assim representados
 - Ex: caixeiro viajante, oito rainhas

Computação Evolutiva



- Representações por permutações
 - Novos desafios: em alguns casos (como aqueles que envolvem ciclos, dentre outros) rotações e reversão de uma solução também são soluções
 - Necessita de operadores específicos de mutação e recombinação



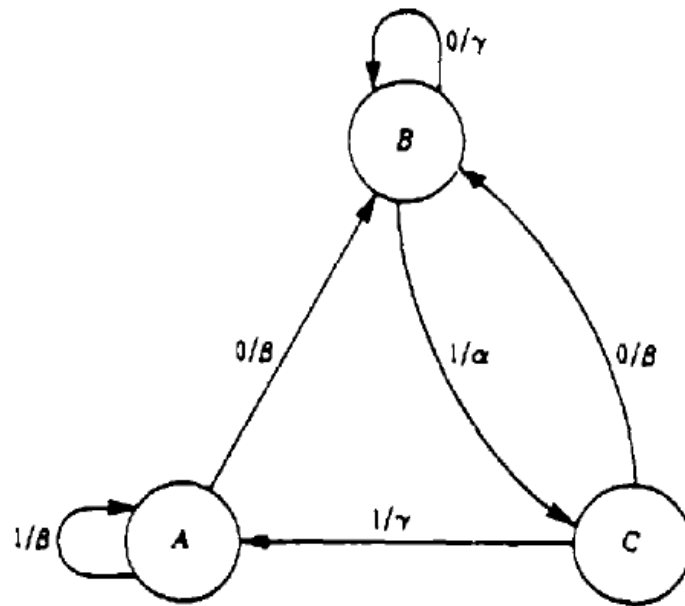
Computação Evolutiva

- Máquinas de estado finito
 - Uma máquina de estados finitos (finite state machines, FSM) é um sistema dinâmico a eventos discretos
 - Tomadas de decisão
 - Predição de séries
 - Uma FSM é uma quintupla $M = \langle Q, t, p, s, o \rangle$
 - Q : conjunto finito de estados
 - t : conjunto finito de símbolos de entradas
 - p : conjunto finito de símbolos de saída
 - s : função de próximo estado, $Q \times t \rightarrow Q$
 - o : função de saída, $Q \times t \rightarrow p$



Computação Evolutiva

- Máquinas de estado finito
- Dada uma entrada x , estando no estado q , terá como saída $o(q,x)$ e como próximo estado $s(q,x)$



Present state	<i>C</i>	<i>B</i>	<i>C</i>	<i>A</i>	<i>A</i>	<i>B</i>
Input symbol	0	1	1	1	0	1
Next state	<i>B</i>	<i>C</i>	<i>A</i>	<i>A</i>	<i>B</i>	<i>C</i>
Output symbol	β	α	γ	β	β	α

Computação Evolutiva



- Máquinas de estado finito
 - Saídas associadas às transições:
Máquina de Mealy, ocorre uma saída quando uma transição (estado, entrada) ocorre
 - Saídas associadas aos estados:
Máquina de Moore, ocorre uma saída quando um estado é alcançado
 - equivale a uma máquina de Mealy com cada estado tendo transições chegando a ele com a mesma saída
 - Possível representação: tabela de transições de estados, para cada estado e entrada indica a saída e o próximo estado



Computação Evolutiva

- Combinações de codificações
 - Dependendo do problema, podem ser utilizadas representações mistas, combinando os parâmetros do problema:
 - Binária
 - Ponto flutuante
 - Permutações
 - Matrizes
 - Árvores
 - Tabelas
 - Ou qualquer outra estrutura apropriada ao problema



Computação Evolutiva

- Função de avaliação (fitness)
 - Primeiro passo, decodificar indivíduos para soluções candidatas
 - Depois podemos avaliar a solução
 - Em problemas complexos é difícil calcular com objetividade um fitness para cada indivíduo
 - Fitness competitivo pode ser uma abordagem
 - Fitness padrão determina fitness independente da população (absoluto)
 - Fitness competitivo avalia indivíduo em relação a população (relativo)



Computação Evolutiva

- Função de avaliação (fitness)
 - Fitness pode ser obtido por cálculo de uma função matemática
 - Ou avaliando a solução no ambiente/ tarefa/ problema proposto
 - Classificador
 - Controlador industrial
 - Robô
 - Otimização multi-objetivo
 - Difícil definir uma função
 - Incorporar alguns objetivos como restrições

Computação Evolutiva



- Função de avaliação (fitness)
 - Otimização multi-objetivo
 - Ponderação dos objetivos em uma função única
 - Cada objetivo afeta seleção de uma parte da população
 - Minimax: minimização da diferença máxima entre o obtido e a meta desejada
 - Outras
 - De fitness de minimização para maximização
 - Diferença para o maior valor absoluto da função ou maior valor na população atual

Computação Evolutiva



- Função de avaliação (fitness)
 - Normalização
 - Dividir fitness pela soma dos fitness
 - Fitness Escalonado
 - Mapear fitness para outro intervalo
 - Salientar pequenos valores no fitness escalonado
 - Possibilidades:
 - Subtrair o menor fitness de todos os fitness
 - Subtrair de todos a média do fitness menos c vezes o desvio padrão

Computação Evolutiva



- Seleção
 - O principal papel da seleção é dar ênfase às melhores soluções
 - Pressão seletiva está relacionado ao tempo de dominância, tempo para a melhor solução inicial dominar a população quando há somente seleção
 - Pressão maior, tempo menor, diversidade menor
 - Pressão menor, tempo maior, diversidade maior
 - Mas operadores de variação também afetam diversidade da população

Computação Evolutiva



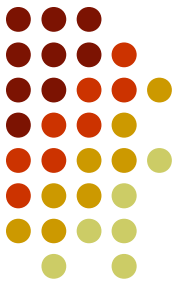
- Seleção
 - Seleção depende da avaliação obtida pelos indivíduos da população
 - Pode ser probabilística ou determinística
 - Envolve reprodução e sobrevivência
 - Pode haver seleção dos progenitores para reprodução
 - Ou seleção de sobrevivência dos descendentes (ou descendentes+progenitores)

Computação Evolutiva



- Seleção
 - Seleção proporcional
 - O número de cópias do indivíduo é proporcional ao seu fitness
 - Ex: seleção por Roleta (Roulette Wheel): o espaço na roleta é proporcional ao fitness e a roleta é então girada
 - Não permite fitness negativo, nem problemas de minimização

Computação Evolutiva



- Seleção
 - Seleção proporcional
 - Uma solução com fitness muito mais alto que os demais pode dominar a roleta
 - Ex: melhor 1.000.000 e demais menos que 100
 - Se o fitness da população é muito próximo, torna-se seleção aleatória
 - Ex: 100.010, 100.027, 100.053, 100.099
 - Normalização: mapear linearmente todos valores em um intervalo com valor superior e inferior

Computação Evolutiva



- Seleção
 - Seleção por torneio
 - q soluções sorteadas para torneio, selecionando os melhores de forma determinística ou probabilística conforme fitness
 - Podem ser selecionados outros ou podem participar em vários torneios
 - Torneio binário: 2 competem, melhor escolhido
 - Permite problemas de minimização e maximização
 - Não há limitação com fitness negativo
 - Pode ser paralelizado



Computação Evolutiva

- Seleção
 - Seleção por ranking
 - Indivíduos ordenados por fitness (crescente/maximização ou decrescente/minimização)
 - Atribuído novo fitness por posição no ranking
 - Seleção por ranking linear ou não linear
 - Seleção bi-classista
 - Seleciona $m\%$ melhores e $p\%$ piores, restante aleatório
 - Seleção elitista
 - Somente os melhores são selecionados

Computação Evolutiva



- Operadores de Busca
 - Dada uma população com certa representação, operadores de busca geram novos indivíduos para intensificar e diversificar população e assim buscar soluções
 - São transformações que recebem um ou mais indivíduos e podem gerar (pequenas) mudanças ou combinar indivíduos gerando novos
 - Projeto conjunto com representação: interdependência entre representação e operadores de busca

Computação Evolutiva



- Operadores de Mutação
 - Vetores de Bits
 - Mutação Uniforme: pequena probabilidade de cada posição inverter bit
 - Pode ser aplicado também vetores de símbolos ou inteiros, com probabilidade uniforme no sorteio de um novo valor diferente do valor atual
 - Estudos empíricos:
 - Probabilidade de mutação inicial alta decrescendo exponencialmente durante execução
 - Prob. Mínima: $1/\text{quant. posições}$

Computação Evolutiva



- Operadores de Mutação
 - Vetores de Núm. Ponto Flutuante
 - Adiciona-se um valor M a cada posição do vetor
 - M é obtido de distribuição de probabilidade
 - Uniforme: limitada a um intervalo $[-b, b]$, maior passo fica limitado
 - Gaussiana: média zero e desvio padrão ajustável
 - Auto-adaptação do desvio padrão, como vimos em EP e EE
 - Outras distribuições, incluindo ajustável durante execução, com grande variância inicial e pequena final

Computação Evolutiva



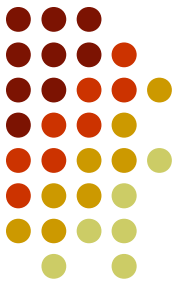
- Operadores de Mutação
 - Permutações
 - Mutações devem gerar permutações
 - 2-opt: seleciona dois pontos e reverte sequência entre eles
 - 3-opt: corte em 3 segmentos, pode trocar posições dos segmentos, e reverte-los ou não
 - Inserção: escolher um elemento e inseri-lo em outra posição
 - Troca: escolher dois e trocar posições

Computação Evolutiva



- Operadores de Mutação
 - FSM
 - Sorteio da quantidade de mutações (distribuição poisson)
 - Mudar símbolo de saída de uma transição
 - Mudar próximo estado de uma transição
 - Adicionar novo estado (até quant.máx)
 - Remover um estado
 - Mudar estado inicial

Computação Evolutiva



- Operadores de Recombinação
 - Vetores de bits (também símbolos ou inteiros)
 - Crossover de 1 ou n pontos: segmentos são trocados entre 2 progenitores
 - Há uma probabilidade de ocorrer ou não
 - 2 ou 1 (descartando o outro) vai para nova população
 - Pode ter mais de 2 progenitores
 - Crossover uniforme: cada posição tem uma probabilidade de haver trocar de alelos entre progenitores

Computação Evolutiva



- Operadores de Recombinação
 - Representação pode identificar posição do alelo na codificação
 - supera limitação de somente posições vizinhas serem blocos construtivos
 - Ex: (0 1) (1 5) (1 4) (0 2) (0 3)
 - mutação altera posição e/ou valor
 - blocos construtivos são dinâmicos

Computação Evolutiva



- Operadores de Recombinação
 - Vetores de núm. ponto flutuante
 - Pode ser crossover de n-pontos ou uniforme
 - Crossover intermediário: média dos valores dos progenitores
 - Combinação linear: $x1' = \alpha \cdot x1 + (1 - \alpha) \cdot x2$;
 $x2' = (1 - \alpha) \cdot x1 + \alpha \cdot x2$
 - Podem ser aplicados em todo vetor ou em partes dele (global vs local) e com +2 progenitores
 - Outros: heurístico, simplex, geométrico, etc

Computação Evolutiva



- Operadores de Recombinação
 - Permutações
 - Crossover anteriores não geram indivíduos válidos
 - Crossover de ordem (Davi's):
 - Um pai é o vetor cut, o outro é filler
 - Escolhe dois pontos de corte, copia segmento entre cortes do cut mantendo as mesmas posições
 - Começando no segundo ponto de corte, procura símbolos do filler que não aparecem no filho e copia para posição vazia do filho, girando para o começo do vetor após o final

Computação Evolutiva



- Operadores de Recombinação
 - Permutações
 - Crossover 2 de ordem
 - Escolha K posições aleatórias no pai 2
 - Localize no pai 1 os símbolos nestas posições do pai 2
 - Reordene símbolos do pai 1 para ficarem na mesma ordem relativa do pai 2
 - Demais símbolos são copiados nas mesmas posições

Computação Evolutiva



- Operadores de Recombinação
 - Permutações
 - Crossover de posição
 - Copie K símbolos do pai 1 para o filho, mesma posição
 - Varredura no pai 2 (esq>dir) buscando elementos que faltam e copiando na ordem
- Existem outros, alguns mais adequados para certos problemas devido a suas particularidades



Computação Evolutiva

- Componentes de um Algoritmo Evolutivo
 - Representação (definição dos indivíduos)
 - População
 - Função de Avaliação
 - Procedimento de Seleção
 - Progenitores (reprodução)
 - Sobreviventes (composição da nova geração)
 - Operadores de busca/variação (mutação e recombinação)

Computação Evolutiva



- Componentes de um Algoritmo Evolutivo
 - Não faça escolhas ingênuas
 - Mesmo arbitrárias precisam ser planejadas
 - Analise as consequências de suas escolhas
 - Procure entender o problema em mãos
 - Verifique se enquadra-se como algum meta-problema
 - Conheça as técnicas de busca e heurísticas utilizadas
 - A literatura pode ser lhe ajudar com o problema ou com os componentes do AE