

Programação Evolutiva

Airton Bordin Junior
[airtonbjunior@gmail.com]

Mestrado em Ciência da Computação – Inteligência Computacional
Profº Dr Celso Gonçalves Camilo Junior

Universidade Federal de Goiania (UFG) - Instituto de Informática - Abril/2017

Programação

- Introdução
- Heurísticas e Metaheurísticas
- Algoritmos evolucionários
- Programação Evolutiva
- Referências





Introdução

- Podemos classificar os problemas computacionais em 2 grupos [SALIBA, 2010]
 - Tratáveis, também chamados de polinomiais
 - Podem ser resolvidos por algoritmos determinísticos.
 - Intratáveis, ou não polinomiais
 - Sem algoritmo determinístico para resolver o problema em tempo hábil.

Tratáveis

- Polinomiais
- Algoritmos determinísticos

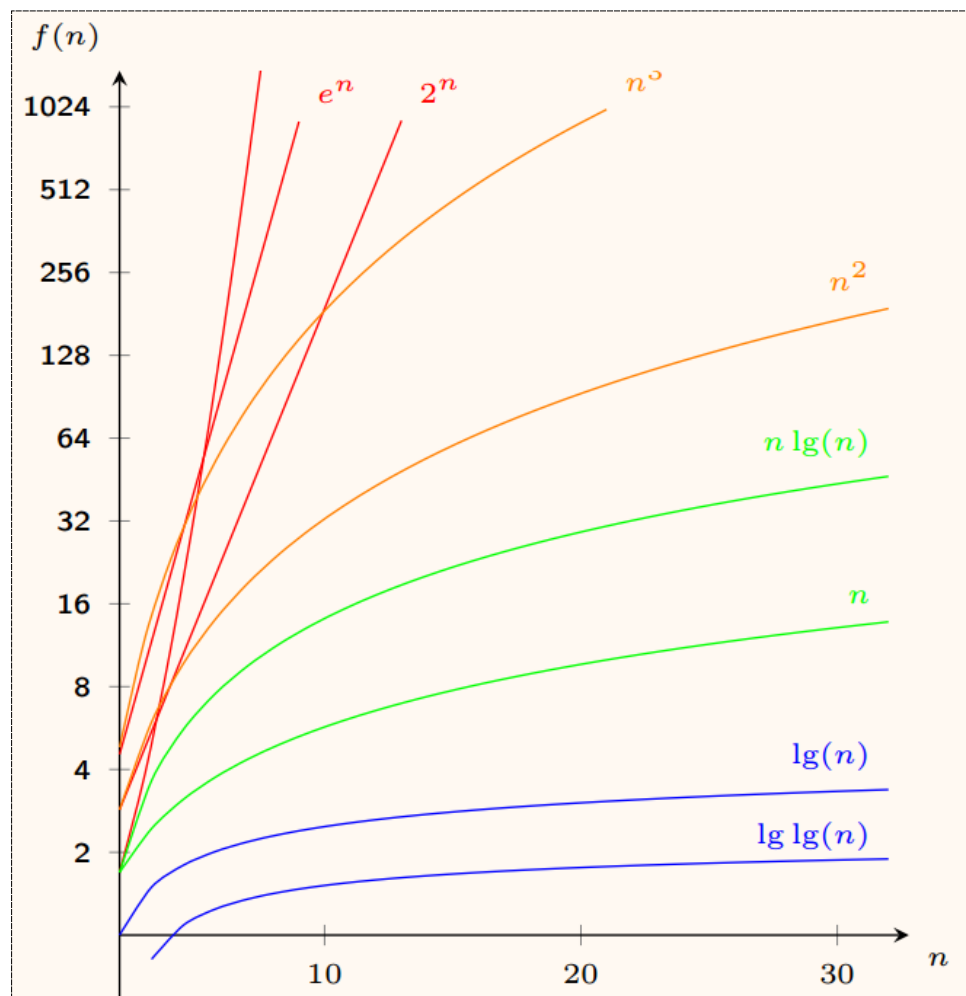
Intratáveis

- Não polinomiais
- Algoritmos não determinísticos
- Solução determinística inviável

Introdução

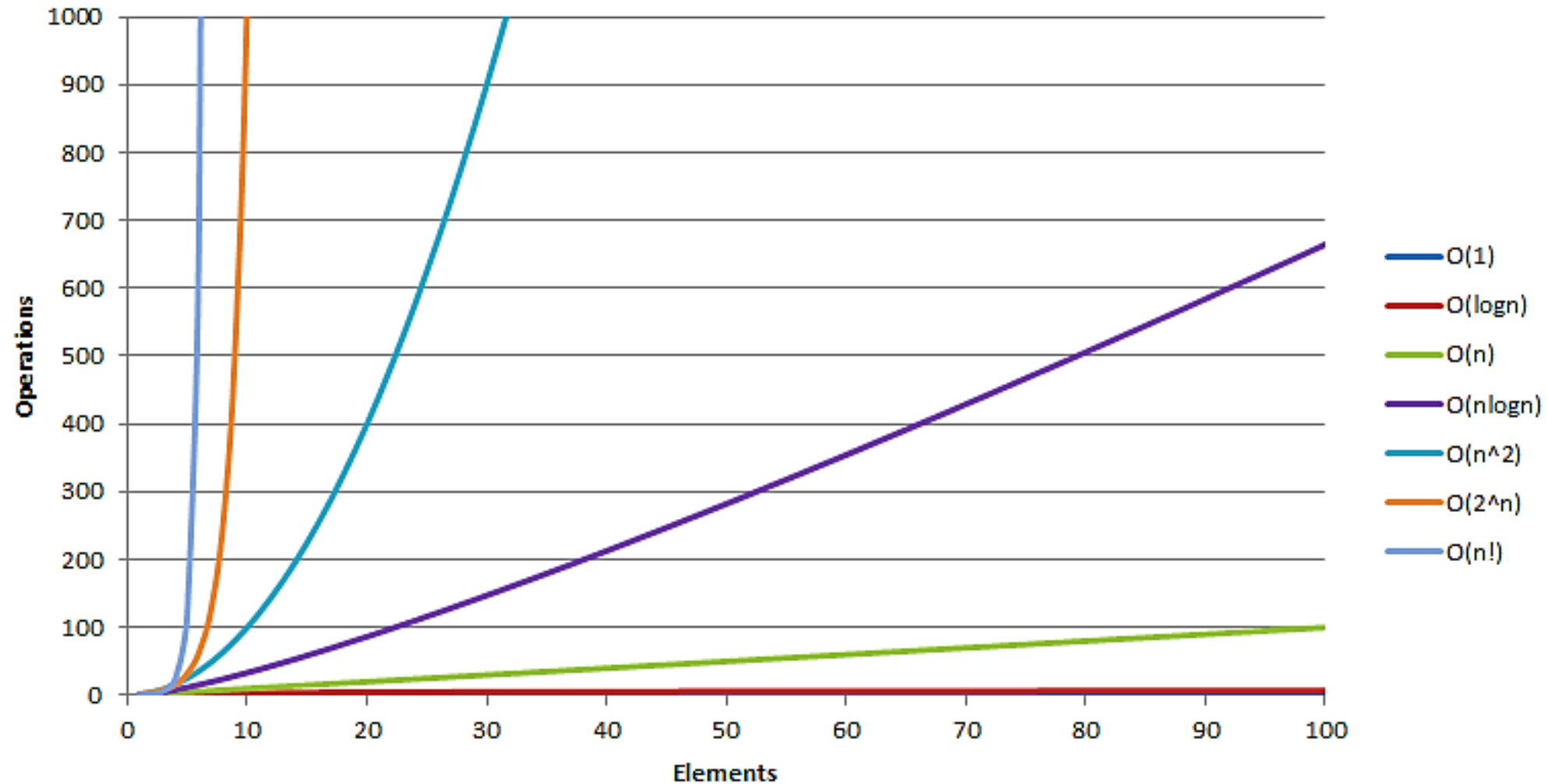


[CAPPELLE, 2017]





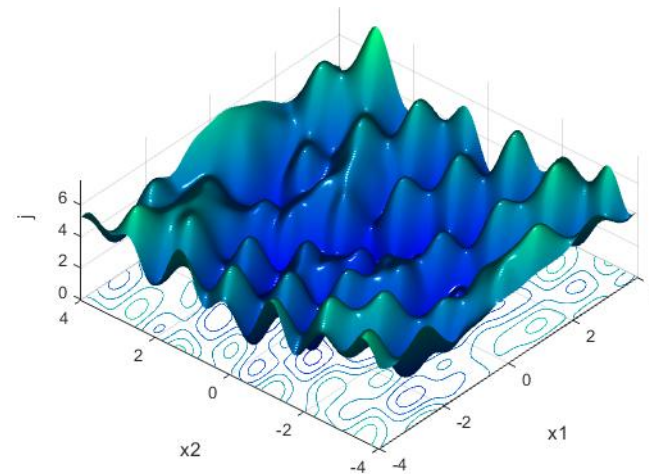
Introdução





Heurística

- Impraticabilidade de encontrar/calcular a melhor resposta para problemas não polinomiais;
- Desafio: produzir, em tempo reduzido, soluções tão próximas quanto possíveis da solução ótima.

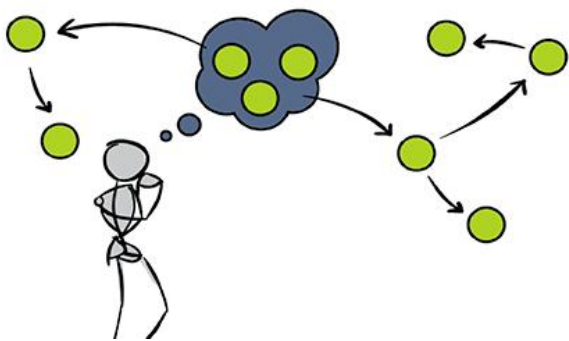




Metaheurística

Propriedades e características das metaheurísticas

[SALIBA, 2010]



Estratégias que guiam o processo de busca;

Exploração eficiente do espaço de busca - soluções ótimas ou quase ótimas;

De simples procedimentos de busca local a complexos processos de aprendizado;

Aproximados e usualmente não determinísticos;

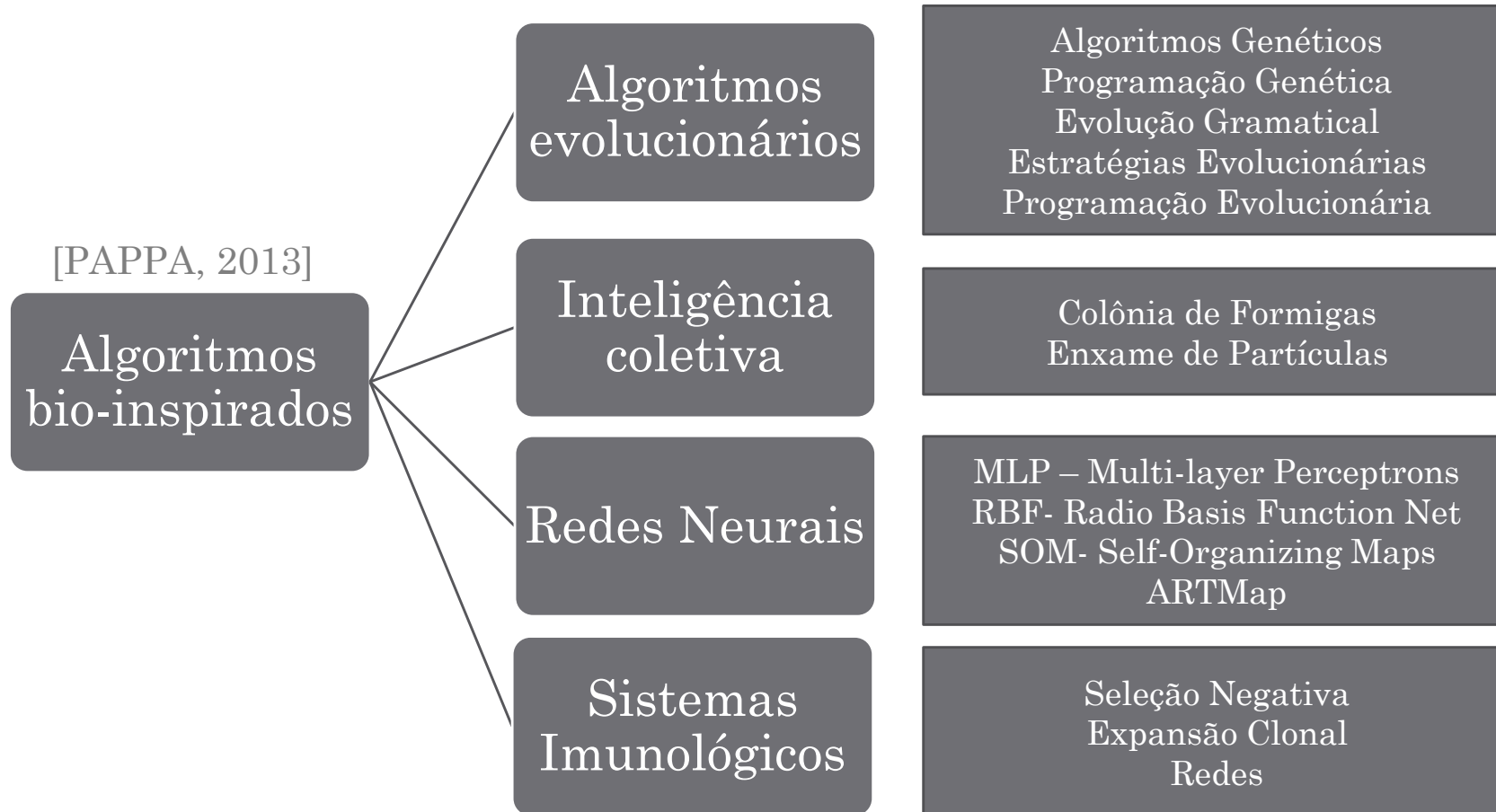
Podem incorporar mecanismos para evitar ficar presos em áreas confinadas do espaço de busca;

Não são específicas para um determinado problema;

Podem usar um conhecimento específico do problema na forma de heurísticas que são controladas por uma estratégia de nível superior.

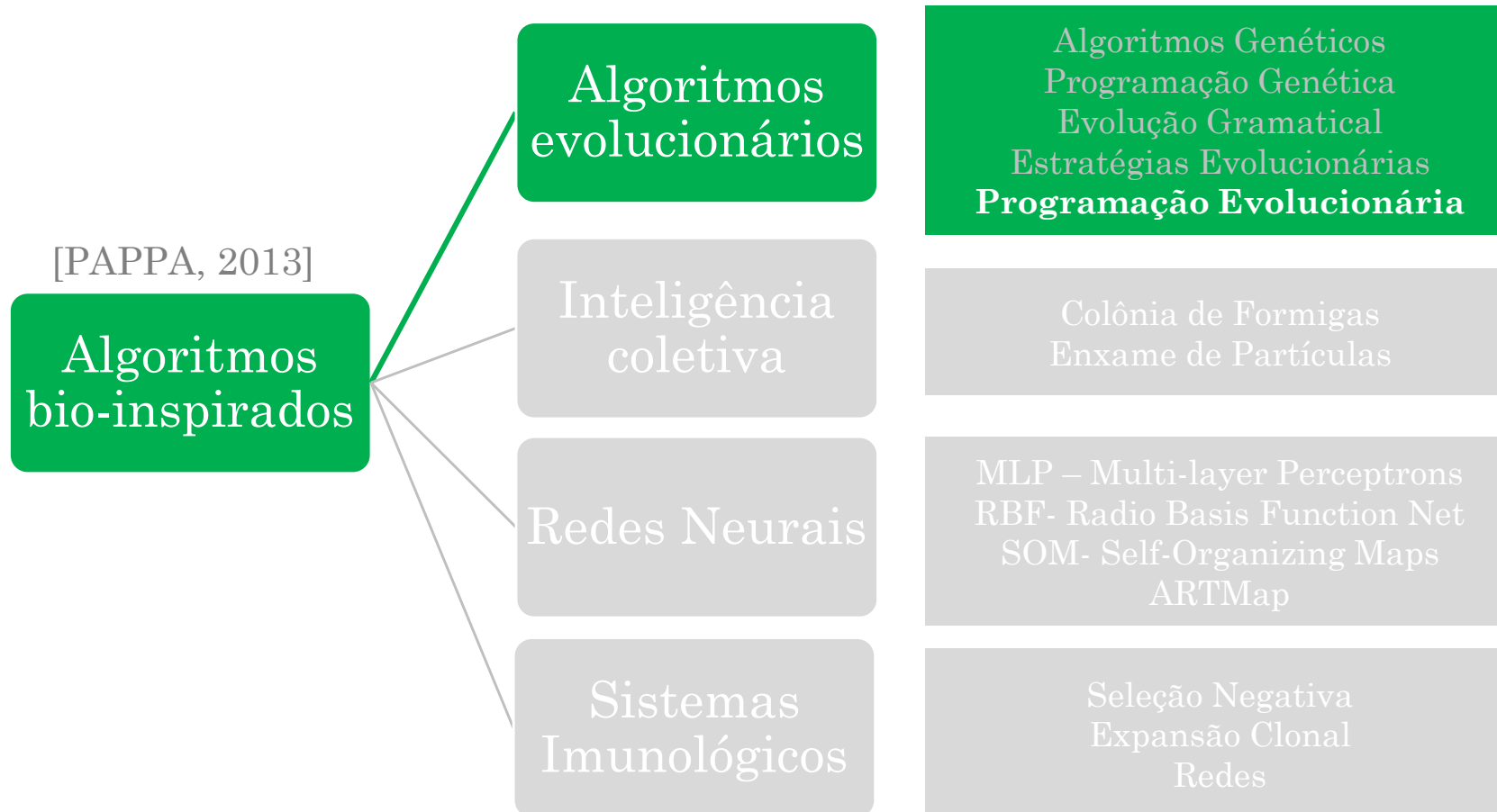


Algoritmos bio-inspirados





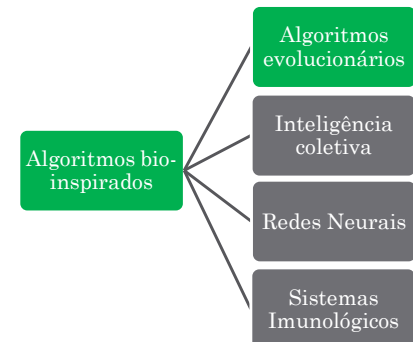
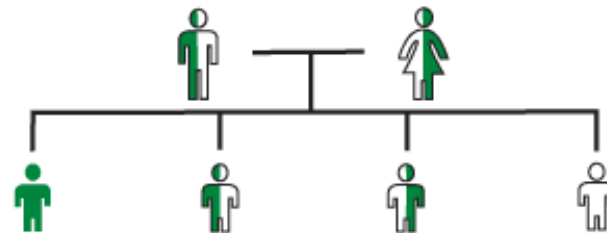
Algoritmos bio-inspirados





Algoritmos evolucionários

- Inspirados na teoria de evolução de Darwin;
- Evolução: mudança das características (genéticas) de uma população de uma geração para a próxima
 - Mutação dos genes;
 - Recombinação dos genes dos pais.

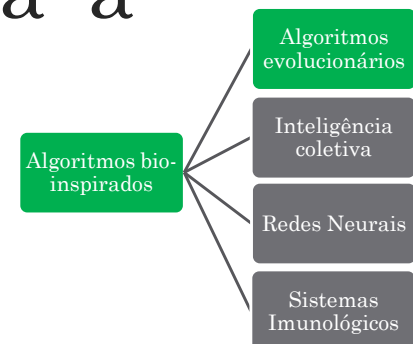




Algoritmos evolucionários

- Evolução é caracterizada basicamente por um processo constituído de 3 passos [VON ZUBEN, 2005]

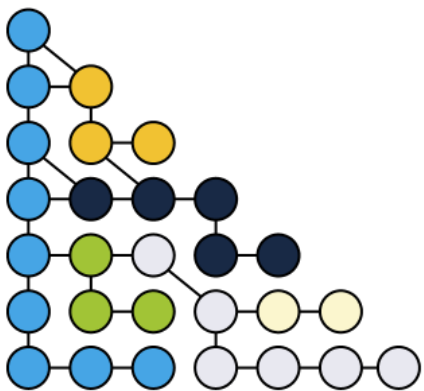
1. Reprodução com herança genética;
2. Introdução de variação aleatória em uma população de indivíduos;
3. Aplicação da “seleção natural” para a produção da próxima geração.





Algoritmos genéticos - Revisão

Principais
diferenças com
métodos
tradicionais



Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;

Trabalham com uma população e não com um único ponto;

Utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar;

Utilizam regras de transição probabilísticas e não determinísticas.



Algoritmos genéticos - Revisão

Algumas características

Paralelo: mantém uma população de soluções que podem ser avaliadas simultaneamente;

Global: AGs não usam somente informações locais, logo não necessariamente ficam presos em máximos locais;

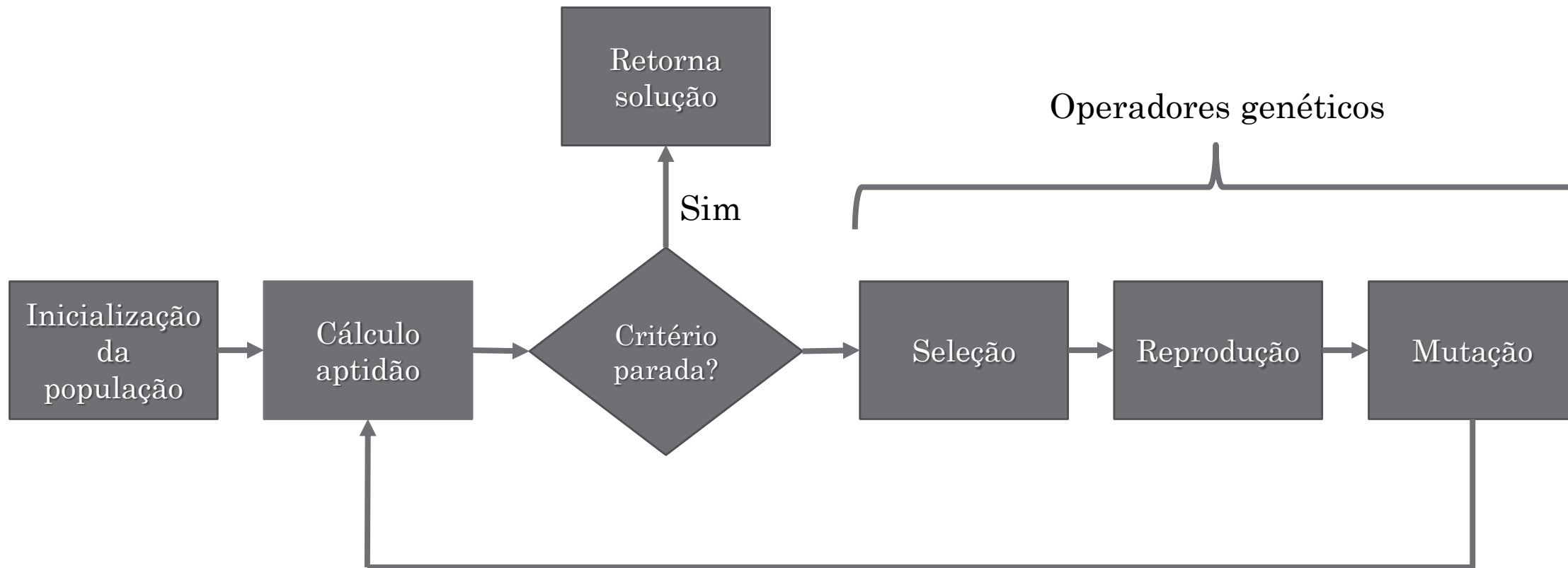
Não totalmente aleatório: usam informações da população atual para determinar o próximo estado de busca;

Não afetados por descontinuidades: não usam informações de derivadas nem necessitam informações de seu entorno;

Funções: Lidam com funções discretas e contínuas.



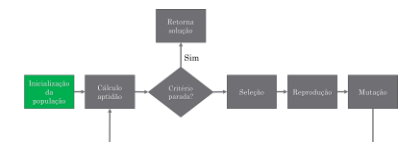
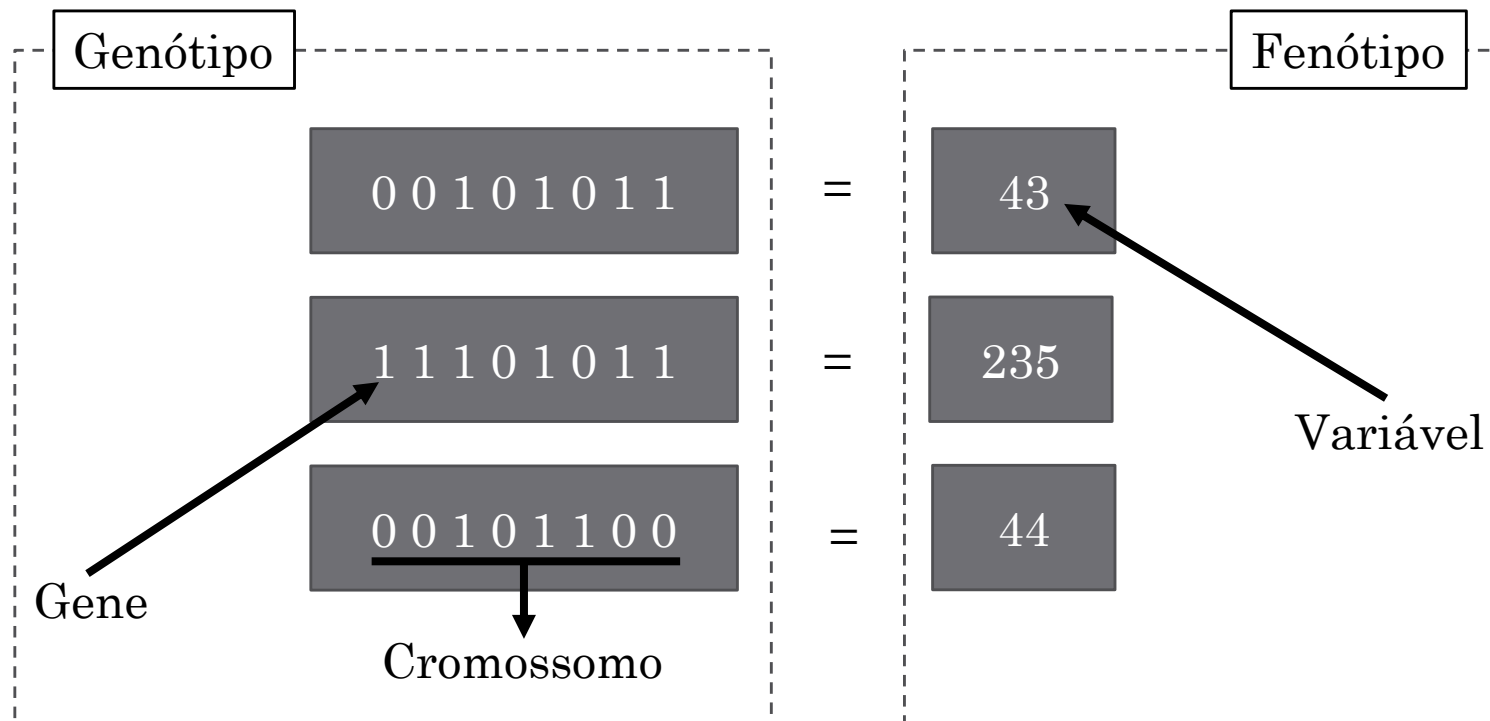
Algoritmos genéticos - Revisão





Inicialização da população

- Exemplo: representação em bits





Programação evolutiva

- Lawrence Jerome Fogel, década de 1960;
- Usando os postulados de Darwin, poderia criar algoritmos que simulassem a dinâmica da evolução e mecanismos da genética;
- *Técnica de simulação da evolução para desenvolver uma forma alternativa de inteligência artificial*



Lawrence Fogel



Programação evolutiva

- Comportamento inteligente – habilidades requeridas:
 - Predizer um determinado ambiente;
 - Responder apropriadamente a este ambiente baseado na predição feita.



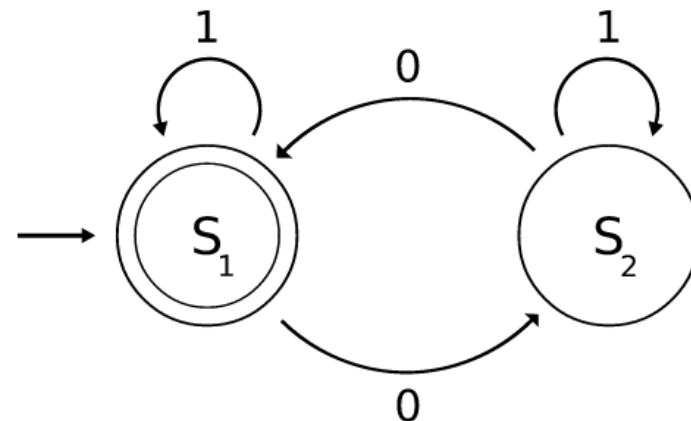
Programação evolutiva

- Ambiente: seqüência de símbolos a partir de um alfabeto finito;
- Operações no ambiente para produzir um símbolo de saída que maximize o desempenho do algoritmo em relação ao próximo símbolo (dada uma função de custo bem definida);
- Representação: máquinas de estado finito (MEF).



Programação evolutiva

- Máquinas de estado finito
 - Modelo matemático utilizado para representar programas;
 - Representa uma seqüência de instruções a serem executadas, cada qual dependendo de um estado atual da máquina e do estímulo atual





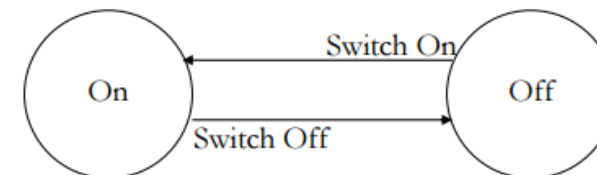
Programação evolutiva

- Máquinas de estado finito
 - Definição formal – 5-tupla $M = \langle Q, \tau, \rho, s, o \rangle$
 - Q : conjunto finito de estados;
 - τ : conjunto finito de símbolos de entrada;
 - ρ : conjunto finito de símbolos de saída;
 - $s: Q \times \tau \rightarrow Q$ é a função do próximo estado;
 - $o: Q \times \tau \rightarrow \rho$ é a função da próxima saída.



Programação evolutiva

- Máquinas de estado finito
 - Dado um símbolo de entrada x enquanto a máquina está no estado q , fornecerá como saída o símbolo $o(q,x)$ e mudará para o estado $s(q,x)$;
 - A informação contida no estado atual da máquina é suficiente para descrever seu comportamento em relação a uma dada entrada





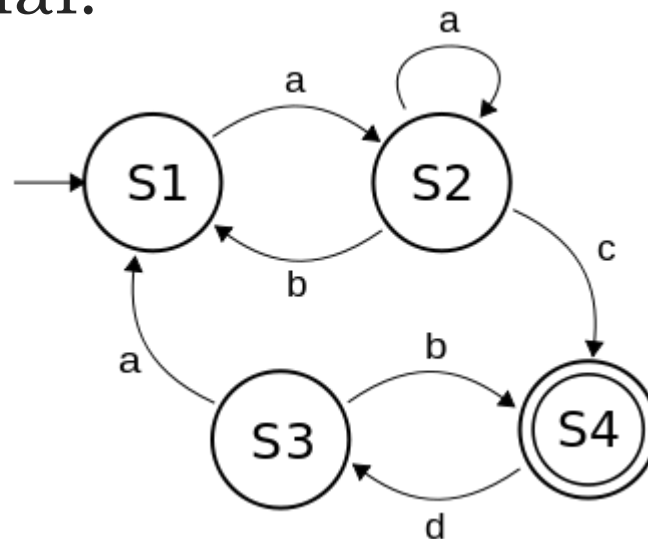
Programação evolutiva

- Máquinas de estado finito
 - O conjunto de estados serve como memória da máquina;
 - MEF é um transdutor que pode ser estimulado por um alfabeto finito de símbolos de entrada
 - Pode responder através de um alfabeto finito de saída, e que possui uma quantidade finita de estados internos



Programação evolutiva

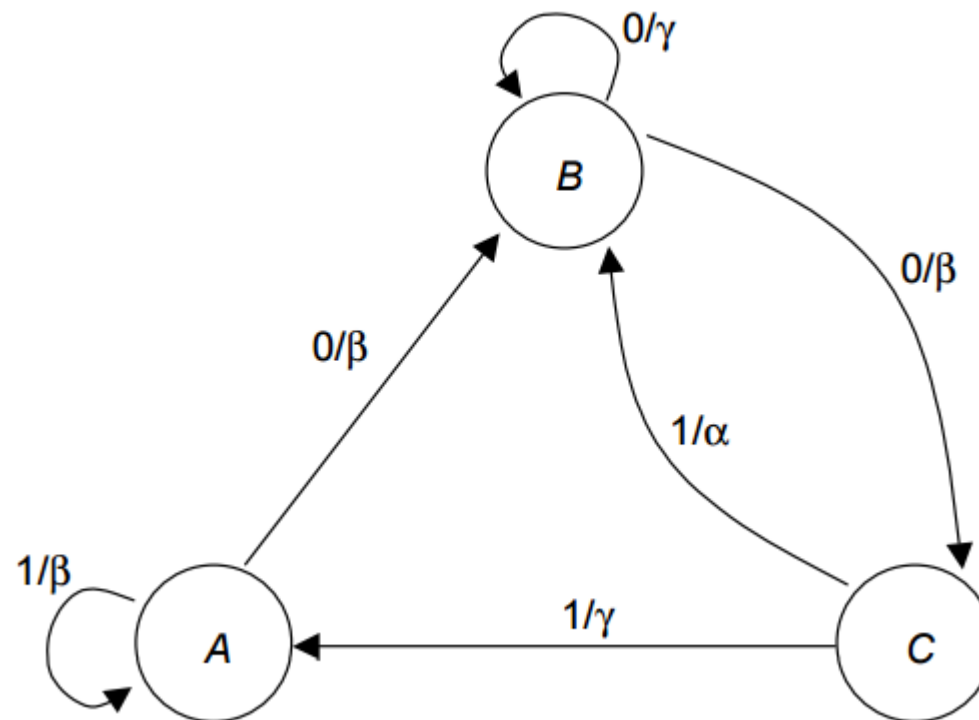
- Máquinas de estado finito
 - Os pares de entrada/saída e transições para o próximo estado de cada símbolo de entrada, tomados sobre cada estado, especificam o comportamento de cada MEF, dada uma condição inicial.





Programação evolutiva

- Exemplo: [VON ZUBBEN, 2002]

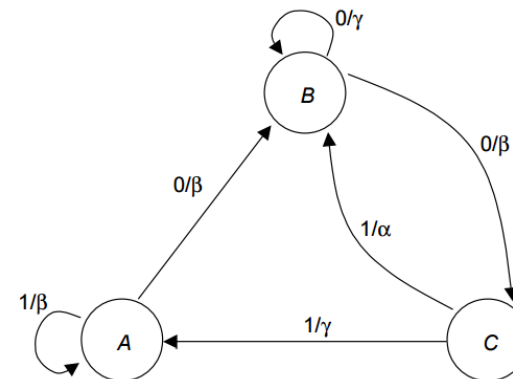
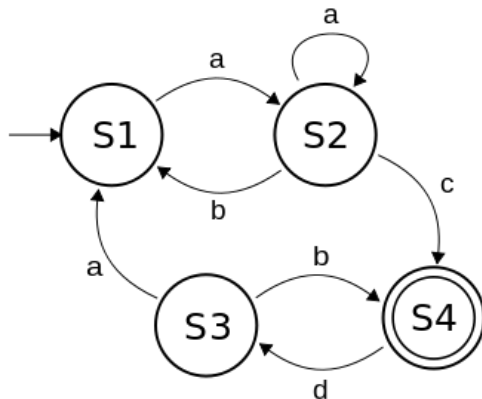


- Símbolos de entrada: $\{0,1\}$
- Símbolos de saída: $\{\alpha,\beta,\gamma\}$



Programação evolutiva

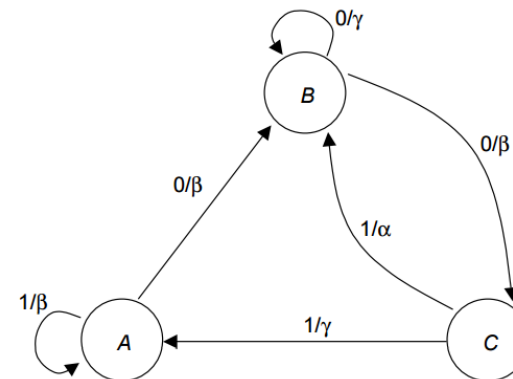
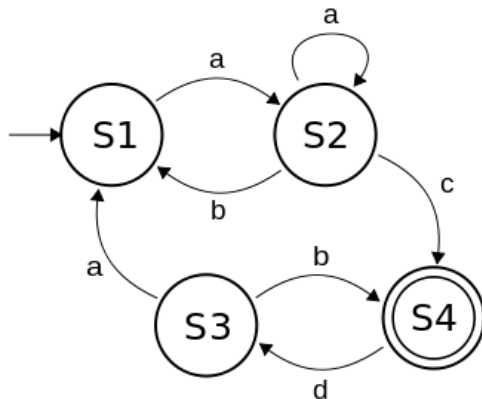
- Funcionamento básico usando MEF [FOGEL, 1966]
 - População de MEFs exposta ao ambiente
 - Seqüência de símbolos observados até o momento
 - Símbolo de saída é observado e comparado ao próximo símbolo de entrada.





Programação evolutiva

- Funcionamento básico usando MEF [FOGEL, 1966]
 - Qualidade da predição é medida usando uma função de avaliação;
 - Após a última predição - função de avaliação para cada símbolo indica o *fitness* de cada máquina.





Programação evolutiva

- Funcionamento básico usando MEF [FOGEL, 1966]
 - Máquinas são geradas aplicando mutação aleatória em cada máquina pai
 - Pai produz um único filho.
 - Tipos básicos de mutação
 - Mudança símbolo de saída;
 - Mudança transição de estado;
 - Adição e remoção estado;
 - Mudança estado inicial.



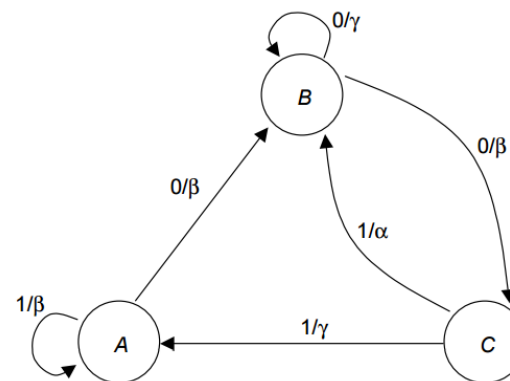
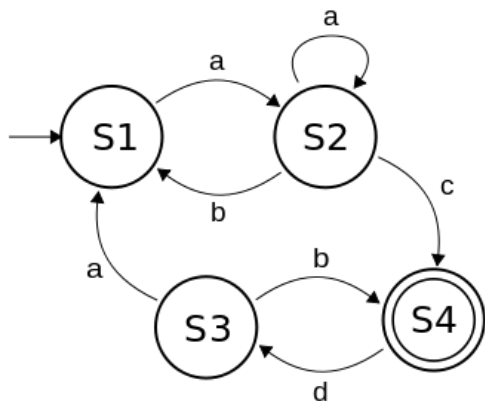
Programação evolutiva

- Filhos são avaliados como os pais;
- MEFs com maior fitness são escolhidas como pais da próxima geração;
- Em geral: metade da quantidade total de máquinas (pais e filhos) selecionada para manter população constante.



Programação evolutiva

- Processo é repetido até a requisição da predição de um novo símbolo;
- Exclusão de um estado e mutação do estado inicial só são permitidos quando existe mais de um estado inicial.





Programação evolutiva

- Abordagens Top-down e Bottom-up
 - Estratégias Evolutivas (EE), assim como a Programação Evolutiva (PE), possuem uma proposta filosófica diferente das outras abordagens de computação evolutiva;
 - EEs e PE são vistas como abordagens Top-down;
 - Algoritmos Genéticos (AG) e Programação Genética (PG) são considerados do tipo Bottom-up.



Programação evolutiva

	AG	EE	PE	PG
Gene	Cada bit ou segmento de bits	Cada x_i, σ_i, θ_i	Cada x_i, σ_i	Cada função e terminal
Cromossomo	Cadeia binária completa	Vetores completos $\mathbf{x}, \boldsymbol{\sigma}, \boldsymbol{\theta}$	Vetores completos $\mathbf{x}, \boldsymbol{\sigma}$	Cada programa
Genótipo	Coleção de cadeias binárias	Coleção de cromossomos: $(\mathbf{x}, \boldsymbol{\sigma}, \boldsymbol{\theta})$	Coleção de cromossomos: $(\mathbf{x}, \boldsymbol{\sigma})$	Coleção de programas
Fenótipo	Valor decodificado de \mathbf{x}	Componente \mathbf{x}	Componente \mathbf{x}	Valor retornado pelo programa



Referências

- FOGEL, L. J. Artificial Intelligence through Simulated Evolution
- ZUBEN, F. V. Representação e Operadores Evolutivos
- ZUBBEN, F. B. Programação Evolutiva
- LUCAS, D.C. Algoritmos Genéticos: Uma Introdução
- CARVALHO, A. P. L. Algoritmos Genéticos
- ROSA, T. O.; LUZ, H. S. Conceitos Básicos de Algoritmos Genéticos: Teoria e Prática
- LIMA, E. S. Algoritmos Genéticos
- MONÇÃO, A. C. B. L. Uma Abordagem Evolucionária para o Teste de Instruções SELECT SQL com o uso da Análise de Mutantes
- PARREIRAS, R. O. Algoritmos Evolucionários e Técnicas de Tomada de Decisão em Análise Multicritério