UNIFAL-MG
Universidade Federal de Alfenas

Projeto e Análise de Algoritmos

Projeto de Algoritmos

Heurísticas e Algoritmos Aproximados

Prof. Humberto Brandão
humberto@bcc.unifal-mg.edu.br

Universidade Federal de Alfenas
Departamento de Ciências Exatas
versão da aula: 0.4

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Problemas que podem ser resolvidos por:
  - algoritmos polinomiais são considerados "fáceis",
  - enquanto problemas que não podem ser resolvidos por algoritmos polinomiais são considerados "difíceis";

- Problemas considerados difíceis ou intratáveis são comuns;
  - Problema da Mochila;
  - Problema do Caixeiro Viajante;
  - Problema de Cobertura de Conjuntos

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Diante de um problema difícil, temos três possibilidades:
  - Tratar o mesmo com algoritmos que são ótimos;
    - São chamados de algoritmos exatos;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Diante de um problema difícil, temos três possibilidades:
  - Tratar o mesmo com algoritmos que são ótimos;
    - São chamados de algoritmos exatos;
  - Ou tratar com algoritmos que chegam próximo ao ótimo;
    - Neste caso a distância máxima é estabelecida;
    - São os algoritmos aproximados;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Diante de um problema difícil, temos três possibilidades:
  - Tratar o mesmo com **algoritmos que são ótimos**;
    - São chamados de **algoritmos exatos**;
  - Ou tratar com algoritmos que **chegam próximo ao ótimo**;
    - Neste caso a distância máxima é estabelecida;
    - São os **algoritmos aproximados**;
  - Ou tratar com algoritmos que tentam aproximar do ótimo;
    - Não existe garantia alguma de que a solução encontrada possui qualidade;
    - São chamados de algoritmos heurísticos;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- É importante para o projetista/analista conhecer uma grande quantidade de problemas clássicos e seus algoritmos eficientes;

Problemas

- É importante para o projetista/analista conhecer uma grande quantidade de problemas clássicos e seus algoritmos eficientes;
  - Isso ajudará a resolver problemas gerais encontrados em qualquer *software* a ser desenvolvidos;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- É **importante para o projetista/analista** **conhecer uma grande quantidade de problemas** clássicos e seus algoritmos eficientes;
  - Isso **ajudará a resolver problemas gerais** encontrados em qualquer *software* a ser desenvolvidos;
  - Para muitos destes problemas conhecemos soluções polinomiais;
    - Embora para alguns deles tal solução não é facilmente visualizada por um leigo no problema;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- É importante para o projetista/analista conhecer uma grande quantidade de problemas clássicos e seus algoritmos eficientes;
  - Isso ajudará a resolver problemas gerais encontrados em qualquer *software* a ser desenvolvidos;
  - Para muitos destes problemas conhecemos soluções polinomiais;
    - Embora para alguns deles tal solução não é facilmente visualizada por um leigo no problema;
- Identificando qual é o problema a ser tratado no *software*, o analista chega em duas opções;
  - Quais são?

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Duas opções possíveis:
  - Ou existe algoritmo polinomial conhecido;
  - Ou não existe;
- O que fazer em ambos os casos?

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- O que fazer em ambos os casos?
  - Se existe, pode ser que o polinômio é de grau baixo;
    - O analista pode implementar o algoritmo conhecido;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- O que fazer em ambos os casos?
  - Se existe, pode ser que o polinômio é de grau baixo;
    - O analista pode implementar o algoritmo conhecido;
  - Mas e se o polinômio é de grau elevado?
    - Em grafos, alguns problemas giram na casa de $O(n^{15})$;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Se o algoritmo mais eficiente conhecido é exponencial existe uma pergunta a ser feita:
  - Qual é a pergunta que o analista deve se fazer?

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Se o algoritmo mais eficiente conhecido é exponencial existe uma pergunta a ser feita:
  - **Qual é o tamanho do problema a ser resolvido?**

Problemas

- Se o algoritmo mais eficiente conhecido é exponencial existe uma pergunta a ser feita:
  - **Qual é o tamanho do problema a ser resolvido?**
  - Se sempre for pequeno, um algoritmo exponencial pode ser implementado;

Problemas

- Se o algoritmo mais eficiente conhecido é exponencial existe uma pergunta a ser feita:
  - **Qual é o tamanho do problema a ser resolvido?**
  - Se sempre for pequeno, um algoritmo exponencial pode ser implementado;
  - Se for de médio porte, alguns algoritmos exponenciais adaptados podem ser utilizados;
    - Como o *backtracking* com *forward-checking* com propagação de restrições;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Se o algoritmo mais eficiente conhecido é exponencial existe uma pergunta a ser feita:
  - **Qual é o tamanho do problema a ser resolvido?**
  - Se sempre for **pequeno**, um **algoritmo exponencial pode ser implementado**;
  - Se for de **médio porte**, alguns **algoritmos exponenciais adaptados** podem ser utilizados;
    - Como o *backtracking* com *forward-checking* com propagação de restrições;
  - Se for de grande porte... O que fazer???

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Se for de grande porte... O que fazer???
  - Neste caso, os algoritmos exatos são inviáveis;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Se for de **grande porte**... O que fazer???
  - Neste caso, **os algoritmos exatos são inviáveis**;
  - A alternativa empregada são as heurísticas;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Se for de **grande porte**... O que fazer???
  - Neste caso, **os algoritmos exatos são inviáveis**;
  - A **alternativa** empregada **são as heurísticas**;
  - Existem meta-algoritmos, que são chamados de meta-heurísticas;

Problemas

UNIFAL-MG
Universidade Federal de Alfenas

- Meta-heurísticas não assumem características do problema;
- As heurísticas assumem;
- Exemplos de meta-heurísticas:
  - Algoritmos Evolucionários;
    - Algoritmo Genético;
    - Programação Genética;
    - Estratégia Evolucionária;
    - Programação Evolucionária;
  - *Simulated Annealing*; (recozimento simulado)
  - Busca Tabu;
  - Colônia de Formigas;
  - Otimização por Enxame de Partículas;

UNIFAL-MG
Universidade Federal de Alfenas

Algoritmos Genéticos

Antes de explicar AGs...

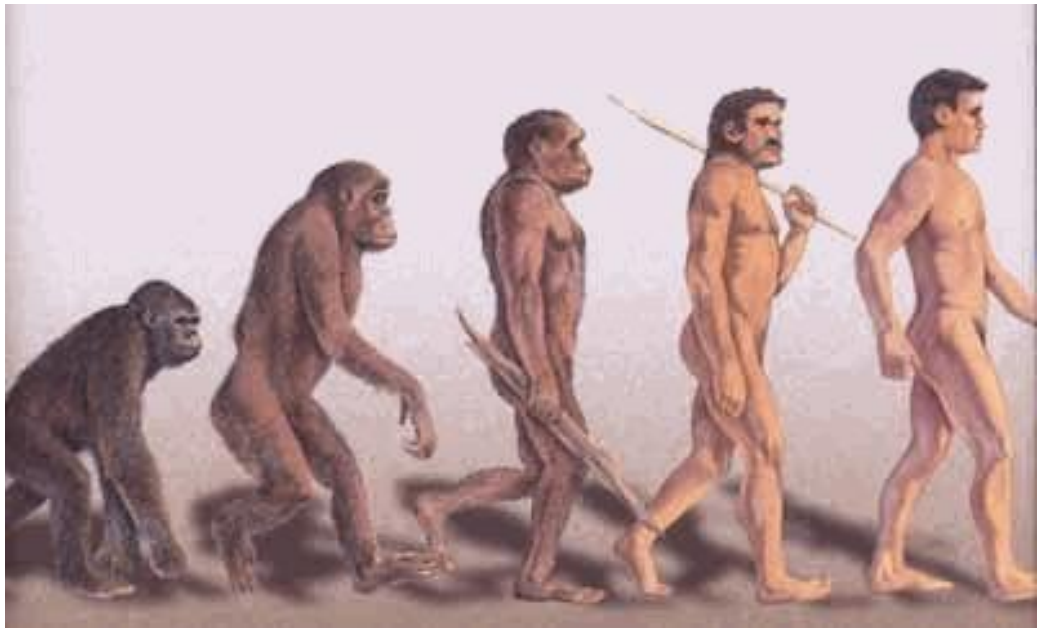
Técnica	Inspiração
Redes Neurais	Neurônios Biológicos
Sistemas Especialistas	Inferência Humana
Lógica Fuzzy	Proc. Lingüístico
Algoritmos Genéticos	Evolução Natural

Evolução natural

- Qual borboleta está mais adaptada ao ambiente?

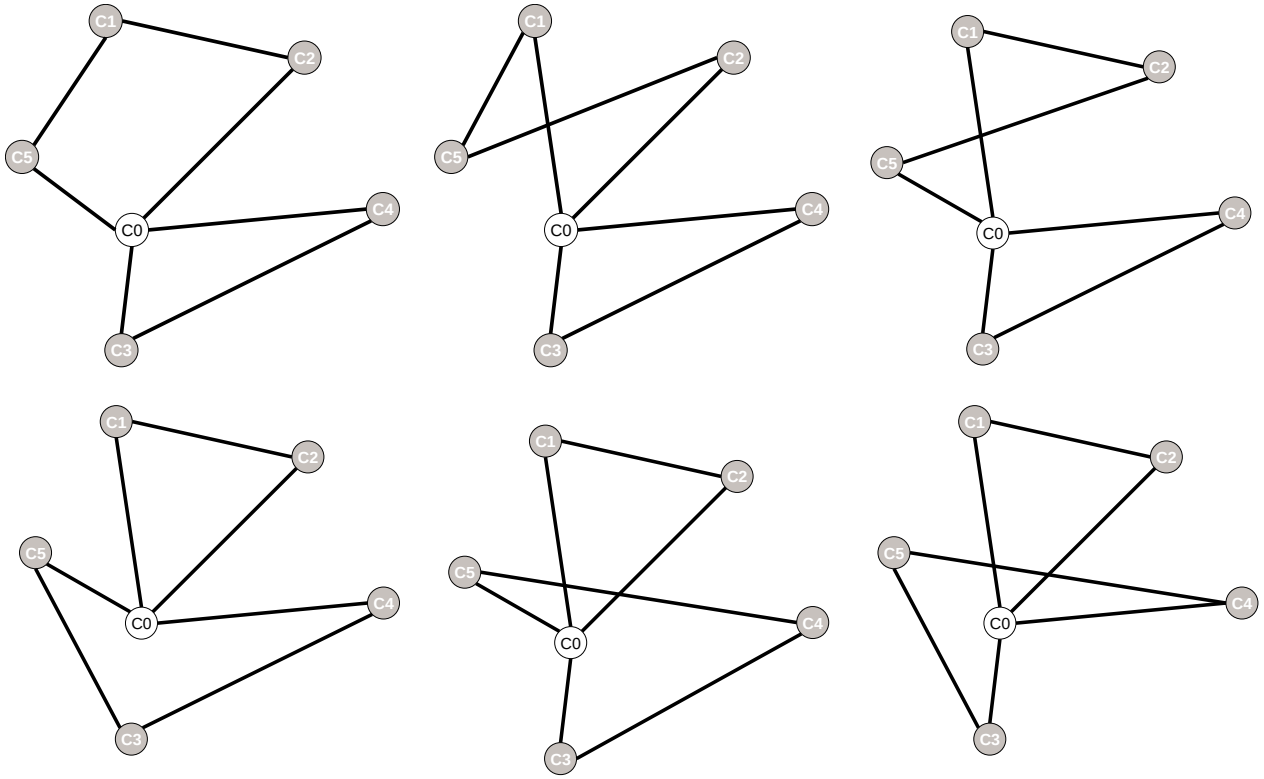
Evolução natural

- Qual animal está mais adaptado ao ambiente?



Evolução natural

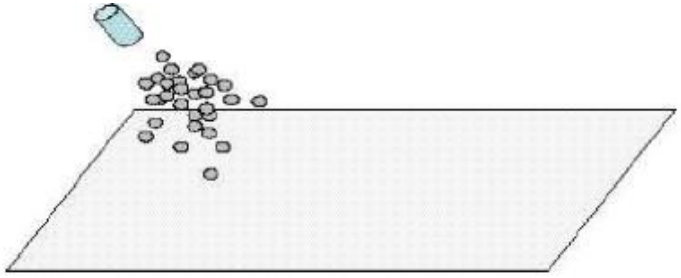
- Qual das 6 soluções está mais adaptada ao problema de roteamento de veículos?



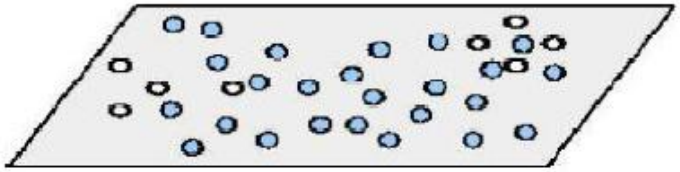
Um problema de exemplo...

Sink móvel em uma Rede de Sensores sem Fio (RSSF)

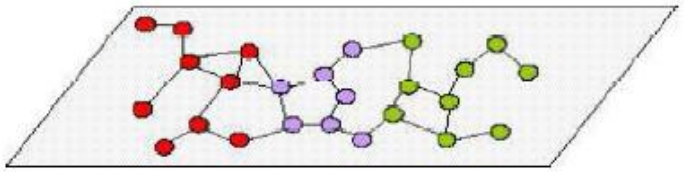
(a) Região de interesse



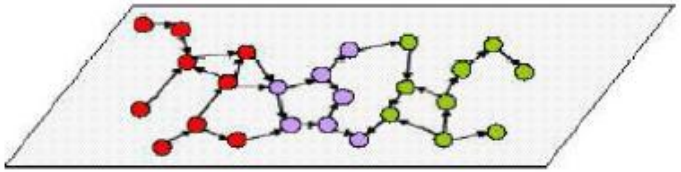
(b) Lançamento dos sensores



(c) Despertar dos sensores



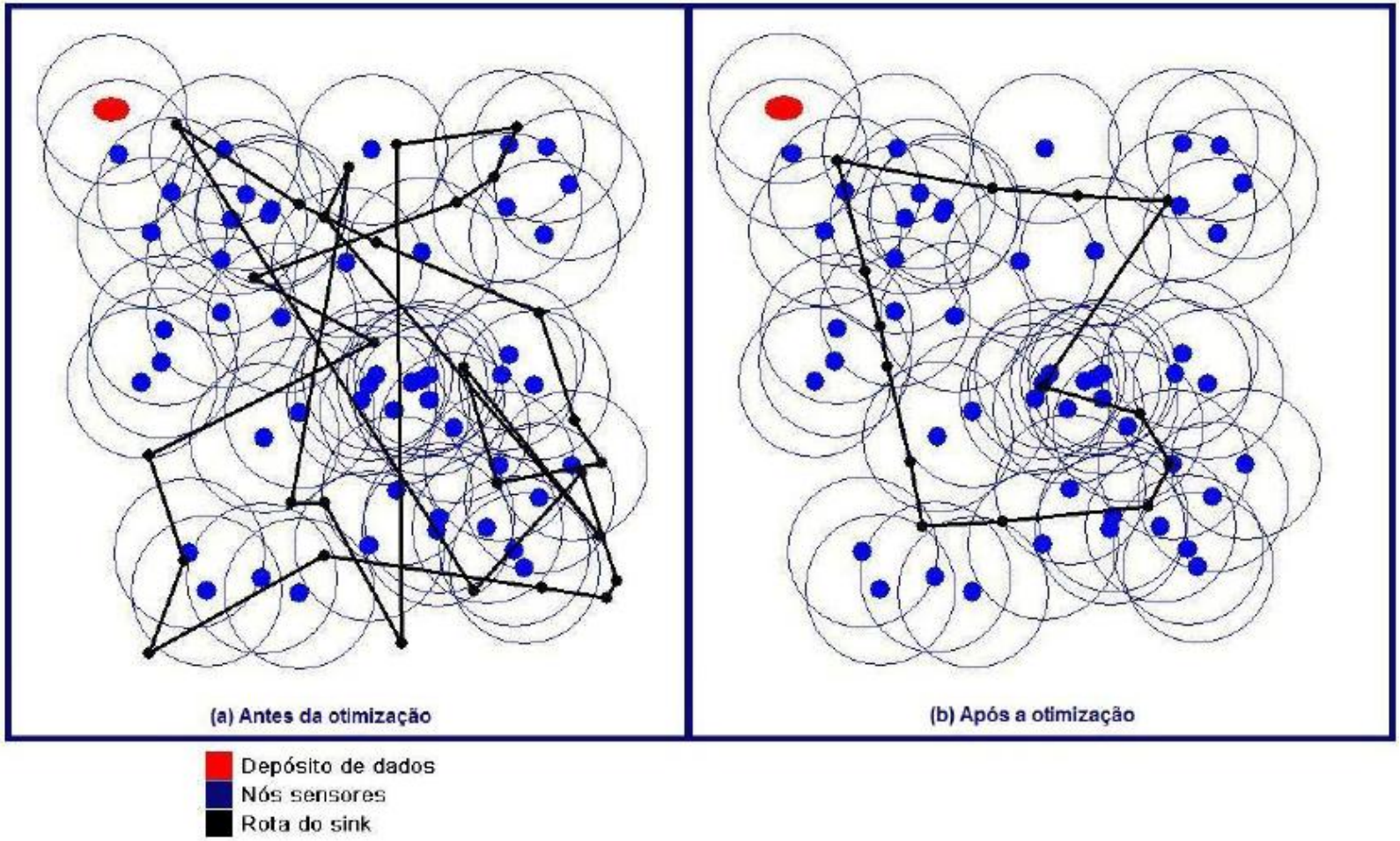
(d) Organização dos sensores



(e) Troca de dados entre os sensores

Evolução natural... de soluções

Solução inicia vs Solução Final



Relacionando Evolução Natural com AG

Evolução		**Resolvendo um problema**
Ambiente	⟷	Problema
Indivíduo	⟷	Solução Candidata
Fitness - Adaptabilidade	⟷	Qualidade da solução

Fitness → chance do indivíduo sobreviver e reproduzir

Algoritmos Genéticos

UNIFAL-MG
Universidade Federal de Alfenas

- Desenvolvido: EUA na década de 70.
- Primeiros pesquisadores:
  - J. Holland, K. DeJong, D. Goldberg
- Aplicado tipicamente na área de:
  - Otimização discreta
- Características:
  - Não muito rápido
  - Bom comportamento em problemas combinatórios
- Características especiais:
  - Ênfase tradicional na combinação de informações de "bons pais" (*crossover*)
  - Muitas variantes, por exemplo, modelos populacionais, operadores, ...

Algoritmos Genéticos

UNIFAL-MG
Universidade Federal de Alfenas

- O AG original de Holland é atualmente conhecido como Algoritmo Genético Simples - AGS);

Algoritmos Genéticos

UNIFAL-MG
Universidade Federal de Alfenas

- O **AG original** de Holland é atualmente **conhecido como Algoritmo Genético Simples** - AGS);

- Outros algoritmos genéticos utilizam diferentes:
  - Representações
  - Mutações
  - Cruzamentos
  - Mecanismos de seleção

Algoritmos Genéticos

UNIFAL-MG
Universidade Federal de Alfenas

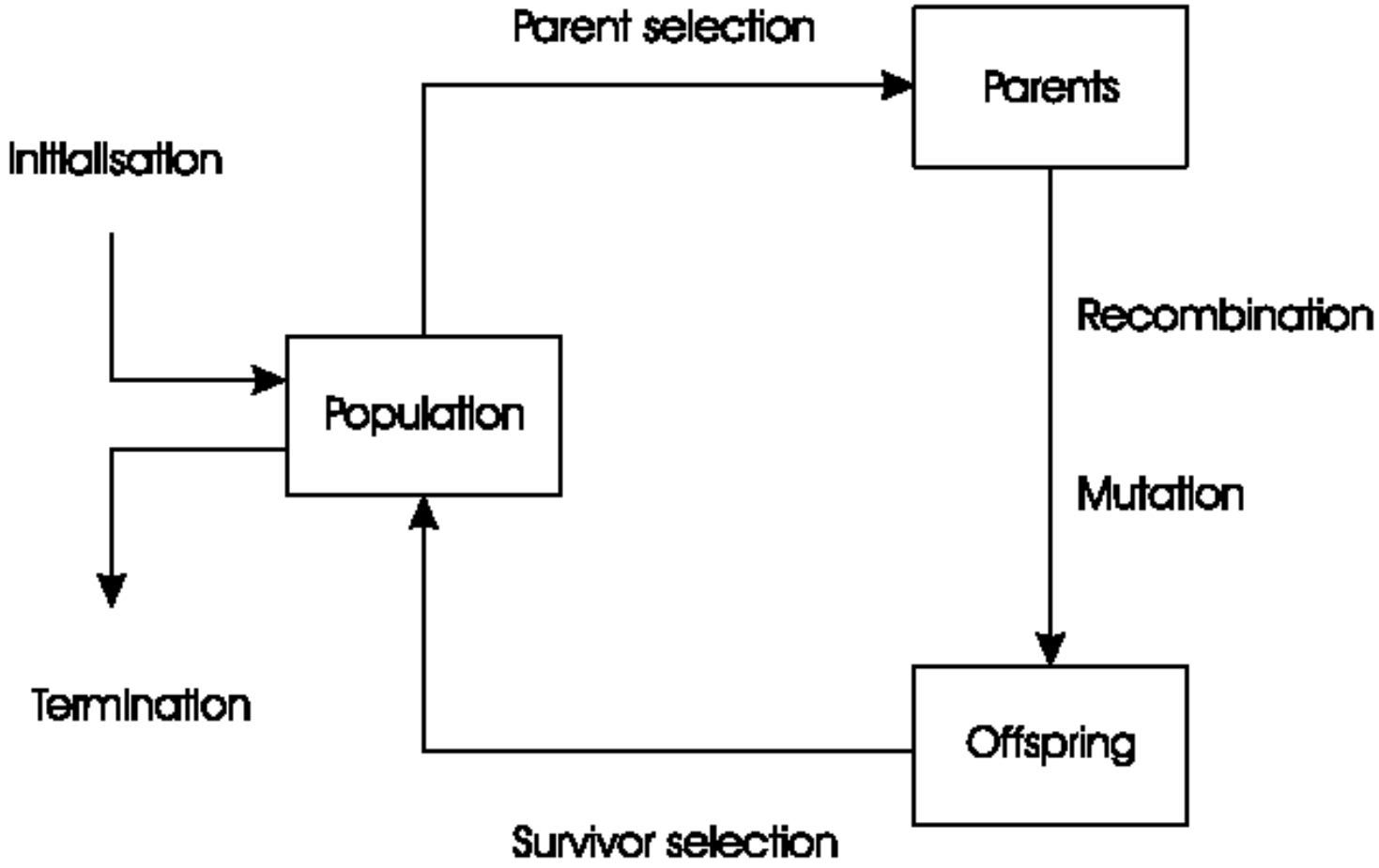
- O **AG original** de Holland é atualmente **conhecido como Algoritmo Genético Simples** - AGS);

- Outros algoritmos genéticos **utilizam diferentes**:
  - Representações
  - Mutações
  - Cruzamentos
  - Mecanismos de seleção

- Isso por que AG são gerais.

- Quando são instanciados as variações aparecem devido a natureza dos problemas.

Esquema geral de um AE



Pseudo-código

UNIFAL-MG
Universidade Federal de Alfenas

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

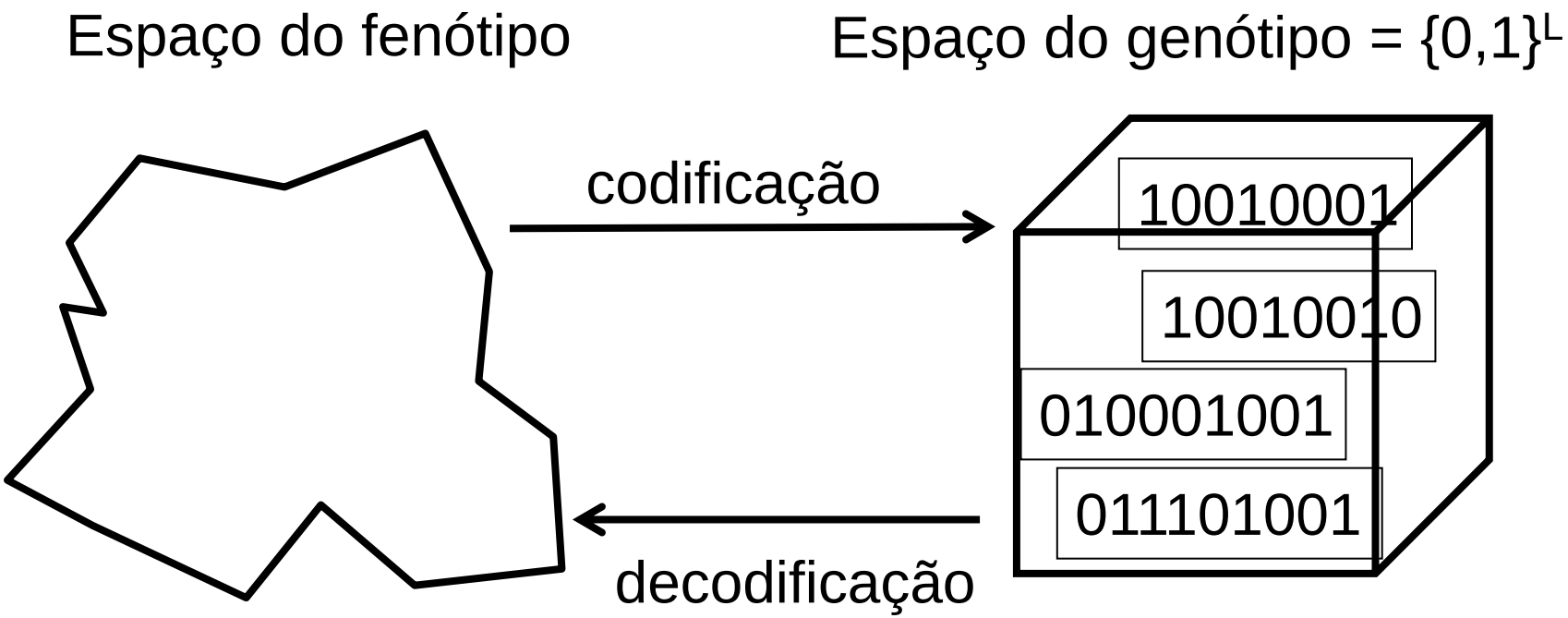
AGS

UNIFAL-MG
Universidade Federal de Alfenas

Representação	*String* binária
Recombinação	Corte de *n* pontos ou uniforme
Mutação	Mudança de *bits* com probabilidade fixa.
Seleção de pais	Proporcional ao *fitness*
Seleção de sobreviventes	Filhos substituem pais
Detalhe	Ênfase no cruzamento

Algoritmo Genético

Representação: String binária



Algoritmo Genético

Exemplo de execução do AG

UNIFAL-MG
Universidade Federal de Alfenas

- Maximizar x^2
  - Sujeito a
    - x>=0
    - x<=31

Algoritmo Genético

Exemplo de execução do AG

UNIFAL-MG
Universidade Federal de Alfenas

- Maximizar x^2
  - Sujeito a
    - x>=0
    - x<=31
- Codificação:
  - $\{0,1\}^5$
- Decodificação:
  - Sugira você mesmo...

Algoritmo Genético

Exemplo de execução do AG

UNIFAL-MG
Universidade Federal de Alfenas

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

Algoritmo Genético

Exemplo de execução do AG

Cruzamento

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Algoritmo Genético

Exemplo de execução do AG

Mutação

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

UNIFAL-MG
Universidade Federal de Alfenas

Representação de indivíduos e cruzamentos e mutações

String Binária

Representação Inteira

Representação de Ponto Flutuante

Representação de Permutação

Representação dos indivíduos

String Binária

UNIFAL-MG
Universidade Federal de Alfenas

- Muitos pesquisadores/programadores escolheram esta representação erroneamente ao longo das últimas décadas;
  - Como por exemplo, o exemplo anterior: $max\ x^2$.

Representação dos indivíduos

String Binária

UNIFAL-MG
Universidade Federal de Alfenas

- Muitos pesquisadores/programadores escolheram esta representação erroneamente ao longo das últimas décadas;
  - Como por exemplo, o exemplo anterior: *max* x^2.

- Para alguns problemas, é a representação mais natural:
  - Exemplo: Satisfabilidade Booleana (SAT)

Representação dos indivíduos

String Binária

UNIFAL-MG
Universidade Federal de Alfenas

- Exemplo de problema com esta codificação:
  - Inteiro 7 codificado em binário: 0111
  - Para chegarmos ao número 8 (que é próximo de 7 na escala dos números inteiros, precisamos aplicar com sucesso 4 alterações/mutações: 0111 → 1000.

Representação dos indivíduos

String Binária

- Exemplo de problema com esta codificação:
  - Inteiro 7 codificado em binário: 0111
  - Para chegarmos ao número 8 (que é próximo de 7 na escala dos números inteiros, precisamos aplicar com sucesso 4 alterações/mutações: 0111 → 1000.

  - Ou seja, a Distância de Hamming entre dois inteiros consecutivos não é igual a 1.
    - Isso prejudicaria uma busca (processo de otimização), por exemplo.

Representação dos indivíduos

String Binária - Mutação

- Mudança de bits:
  - Cada bit possui uma "pequena probabilidade" p_m de ser invertido.

UNIFAL-MG
Universidade Federal de Alfenas

Representação de indivíduos e cruzamentos e mutações

String Binária

Representação Inteira

Representação de Ponto Flutuante

Representação de Permutação

Representação dos indivíduos

Representação Inteira

- Pode ser uma representação mais direta para alguns problemas:
  - Por exemplo, *max* x^2.

Representação dos indivíduos

Representação Inteira

- Pode ser uma representação mais direta para alguns problemas:
  - Por exemplo, *max* x^2.

- O indivíduo pode ser visto como um *array* de *int's*.

Representação dos indivíduos

Representação Inteira

UNIFAL-MG
Universidade Federal de Alfenas

- Pode ser uma representação mais direta para alguns problemas:
  - Por exemplo, *max* x^2.

- O indivíduo pode ser visto como um *array* de *int's*.

- Facilita buscas locais. Exemplo.
  - 3 é próximo de 4, que é próximo de 5, que é próximo de 6...
    Isso não acontecia na representação de string binária.

Representação dos indivíduos

UNIFAL-MG
Universidade Federal de Alfenas

Representação Inteira

- Pode ser uma representação mais direta para alguns problemas:
  - Por exemplo, *max* x^2.
- O indivíduo pode ser visto como um *array* de *int's*.
- Facilita buscas locais. Exemplo.
  - 3 é próximo de 4, que é próximo de 5, que é próximo de 6...
    Isso não acontecia na representação de string binária.
- Pode ser utiliza na codificação de tipos enumerados:
  - Exemplo: Norte, Sul, Leste, Oeste...
  - Alguma relação com o PSR???

Representação dos indivíduos

Representação Inteira - Mutação

- Dois operadores clássicos de mutação para a representação inteira. Ambos mutacionam cada gene com uma probabilidade p_m.
  - Reinicio aleatório:
    - Se o gene *i* sofre uma mutação de reinicio aleatório, seu valor é "reiniciado", retirando-o de uma distribuição uniforme:
    - NovoValor ←U(LimiteInferior, LimiteSuperior);

Representação dos indivíduos

Representação Inteira - Mutação

UNIFAL-MG
Universidade Federal de Alfenas

- Incremento:
  - Se o gene *i* sofre uma mutação de incremento, seu valor é adicionado a um pequeno valor sorteado (positivo ou negativo).
    - É natural que este incremento saia de uma distribuição simétrica, com o seu centro em 0.
    - Pode ser uma distribuição uniforme: (int)U(-x, x)
    - Pode ser uma distribuição gaussiana (normal): (int)N(0, desvio);

UNIFAL-MG
Universidade Federal de Alfenas

Representação de indivíduos e cruzamentos e mutações

String Binária

Representação Inteira

Representação de Ponto Flutuante

Representação de Permutação

Representação dos indivíduos

Representação de Ponto Flutuante

UNIFAL-MG
Universidade Federal de Alfenas

- Utilizada quando queremos representar valores que possuem o domínio contínuo;
  - Aplicações também com a meta-heurística PSO para esta representação;

- Claro que em um computador, a continuidade é limitada pela implementação no *hardware*, mas podemos considerar que as arquiteturas atuais oferecem uma precisão adequada para a maioria dos problemas.

- O indivíduo pode ser visto como um *array* de *double*.

Representação dos indivíduos

Representação de Ponto Flutuante - Mutação

UNIFAL-MG
Universidade Federal de Alfenas

- Podem ser feitas as mesmas mutações da representação inteira.
  - Incremento;
  - Reinicio aleatório.
- Sem precisar truncar para inteiro, como no outro exemplo.

UNIFAL-MG
Universidade Federal de Alfenas

Representação de indivíduos e cruzamentos e mutações

String Binária

Representação Inteira

Representação de Ponto Flutuante

Representação de Permutação

Representação dos indivíduos

Representação de Permutação

UNIFAL-MG
Universidade Federal de Alfenas

- Muitos problemas possuem na solução um indivíduo que necessita informar a ordem de ocorrência de certos eventos;

- Exemplos:
  - Caminho mínimo;
  - Caixeiro Viajante;
  - Job-shop;
  - Roteamento;
  - Dentre inúmeros outros...

Representação dos indivíduos

Representação de Permutação - Mutação

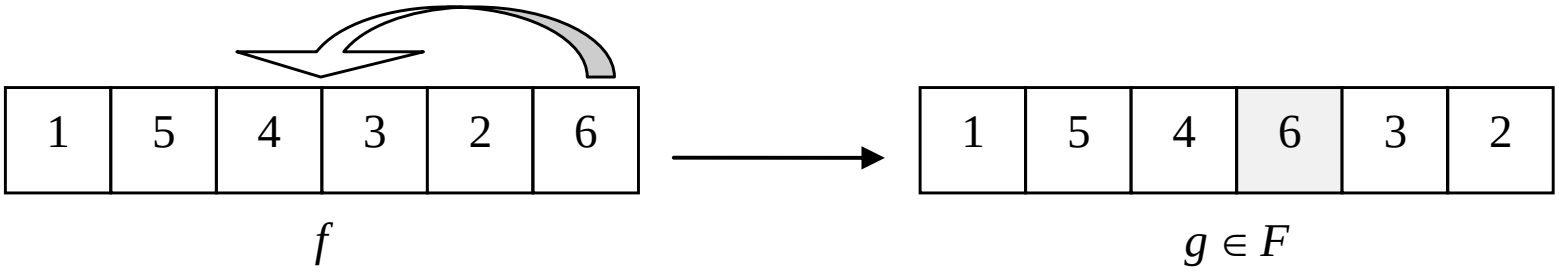
UNIFAL-MG
Universidade Federal de Alfenas

- E é claro que se a cidade X precisa ser visitada, e não faz sentido manter cromossomos que não possuem X na solução do caixeiro viajante.
  - Isso aconteceria se utilizássemos uma mutação de reinicio aleatório nos genes, por exemplo.
  - Portanto, precisamos descrever operadores que mantém tal estrutura (todos os elementos).

Representação dos indivíduos

Representação de Permutação - Mutação

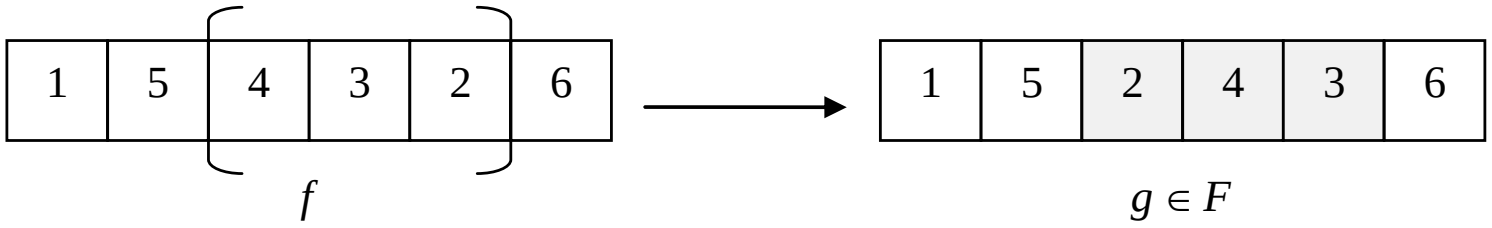
- Mutação de Inserção:
  - Escolhe aleatoriamente um gene
  - Escolhe aleatoriamente uma nova posição
  - Efetua a troca desde gene de ordem, causando uma perturbação na seqüência dos elementos.



Representação dos indivíduos

Representação de Permutação - Mutação

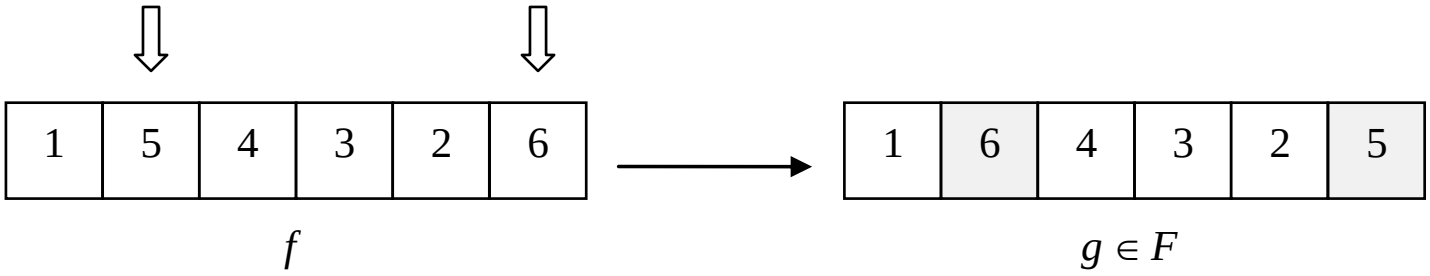
- Mutação de Mistura:
  - Escolhe aleatoriamente um intervalo e mistura aleatoriamente os genes daquele intervalo.



Representação dos indivíduos

Representação de Permutação - Mutação

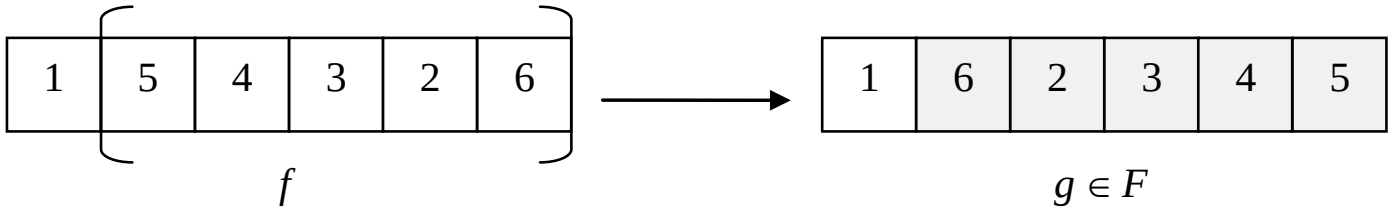
- Mutação de Troca:
  - Escolhe aleatoriamente dois genes e os troca de posição.



Representação dos indivíduos

Representação de Permutação - Mutação

- Mutação de Inversão:
  - Escolhe aleatoriamente um intervalo e inverte a ordem dos genes.



Próxima aula

UNIFAL-MG
Universidade Federal de Alfenas

- Visão Geral sobre:
  - Programação Genética;
  - *Simulated Annealing*;
  - Otimização por Enxame de Partículas;

Bibliografia

- EIBEN, A. E.; SMITH, J. E.; Introduction to Evolutionary Computaing; Natural Computing Series; Springer. 2003.

- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson;