1. Computação Evolutiva

Prof. Renato Tinós

Depto. de Computação e Matemática (FFCLRP/USP)

1.7. Outras Metaheurísticas Populacionais

1.7.1. Metaheurísticas Populacionais

1.7.2. *Differential Evolution* (DE)

1.7.3. *Particle Swarm Optimization* (PSO)

1.7.4. *Ant Colony Optimization* (ACO)

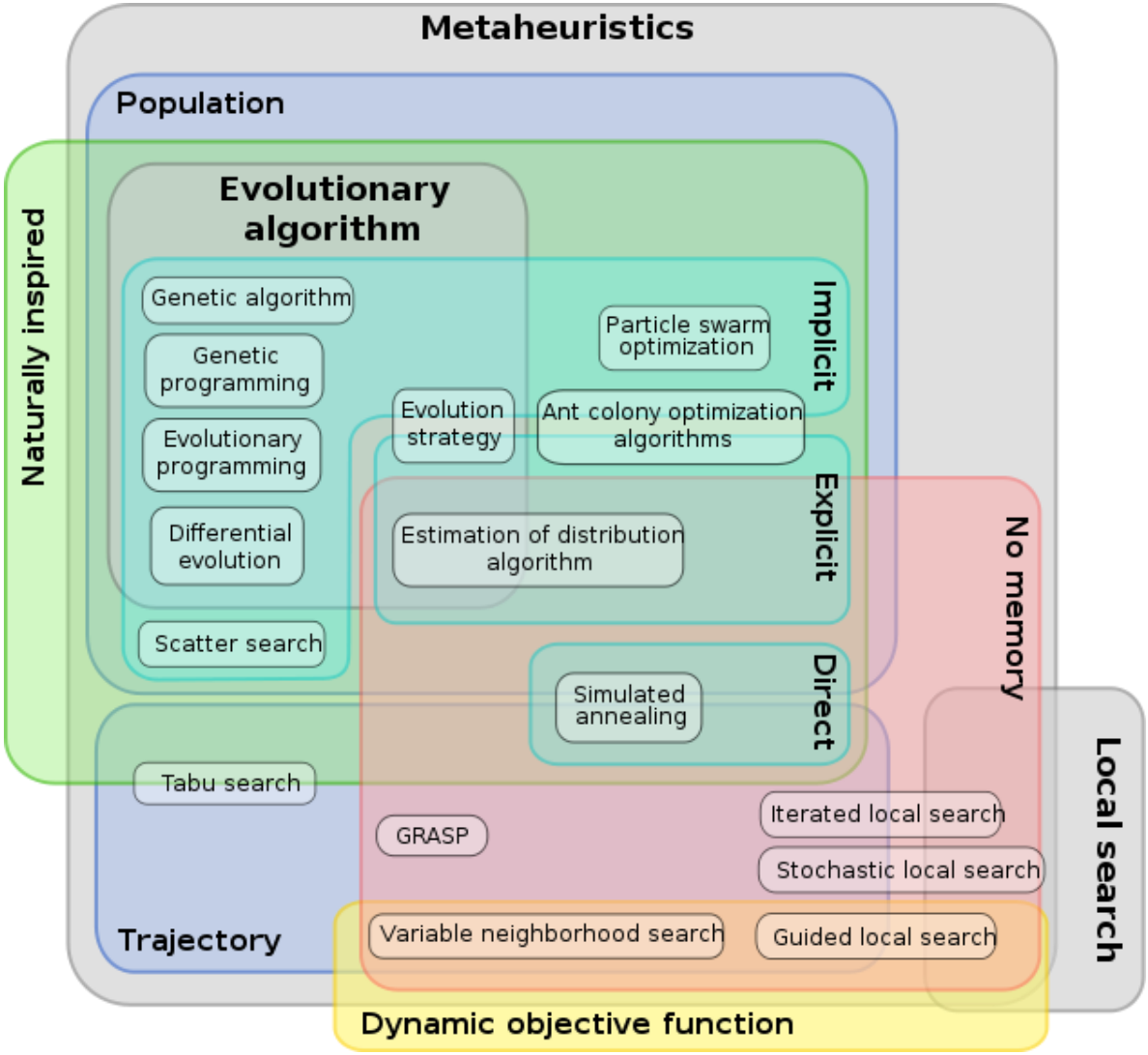
1.7.1. Metaheurísticas Populacionais

- **Metaheurísticas**
  - Procedimento de alto nível usado para encontrar, gerar ou selecionar uma heurística que pode prover uma solução suficientemente boa em problemas de otimização com informação incompleta ou imperfeita e/ou com **limitada capacidade computacional**
    - Podem, nestes problemas, encontrar boas soluções mais eficientemente que métodos tradicionais
  - Amostram soluções do espaço de soluções em problemas em que este é muito grande para ser completamente examinado
    - Assim, quando a capacidade de computação é um fator limitante, **não garantem necessariamente que o ótimo global será encontrado**

1.7.1. Metaheurísticas Populacionais

- **Metaheurísticas**
  - Podem utilizar pouca informação sobre o problema a ser resolvido
    - Podem, portanto, serem utilizadas em uma grande variedade de problemas
  - Muitas metaheurísticas utilizam métodos estocásticos para vasculhar o espaço de soluções
    - Exemplos:
      - *Simulated Annealing* (Recozimento Simulado)
      - Computação Evolutiva

1.7.1. Metaheurísticas Populacionais



Fonte: https://commons.wikimedia.org/wiki/File:Metaheuristics_classification.svg
Authors: Johann Dréo, Caner Candan

1.7.1. Metaheurísticas Populacionais

- **Metaheurísticas Populacionais**
  - Em cada iteração do algoritmo, utilizam uma população de soluções ao invés de uma solução
  - Ao utilizar em paralelo mais de uma solução, estas podem ser **recombinadas** para gerar novas soluções
  - Existem diversas metaheurísticas populacionais
  - Exemplos:
    - Algoritmos evolutivos
    - *Particle Swarm Optimization (PSO)*
    - *Differential Evolution (DE)*
    - *Ant Colony Optimization (ACO)*

1.7.1. Metaheurísticas Populacionais

- **Estratégias de recombinação: codificação real**
  - Recombinar 2 ou mais soluções da população atual selecionadas aleatoriamente
    - Formas: discreta, média aritmética
    - Ex.: EEs
  - Recombinar 2 soluções da população atual selecionadas de acordo com o fitness
    - Formas: discreta
    - Ex.: AGs
  - Recombinar 1 solução com a diferença de outras duas
    - Forma: média ponderada
    - Ex.: DE
  - Recombinar soluções da população atual utilizando as melhores soluções encontradas
    - Forma: média ponderada
    - Ex.: PSO

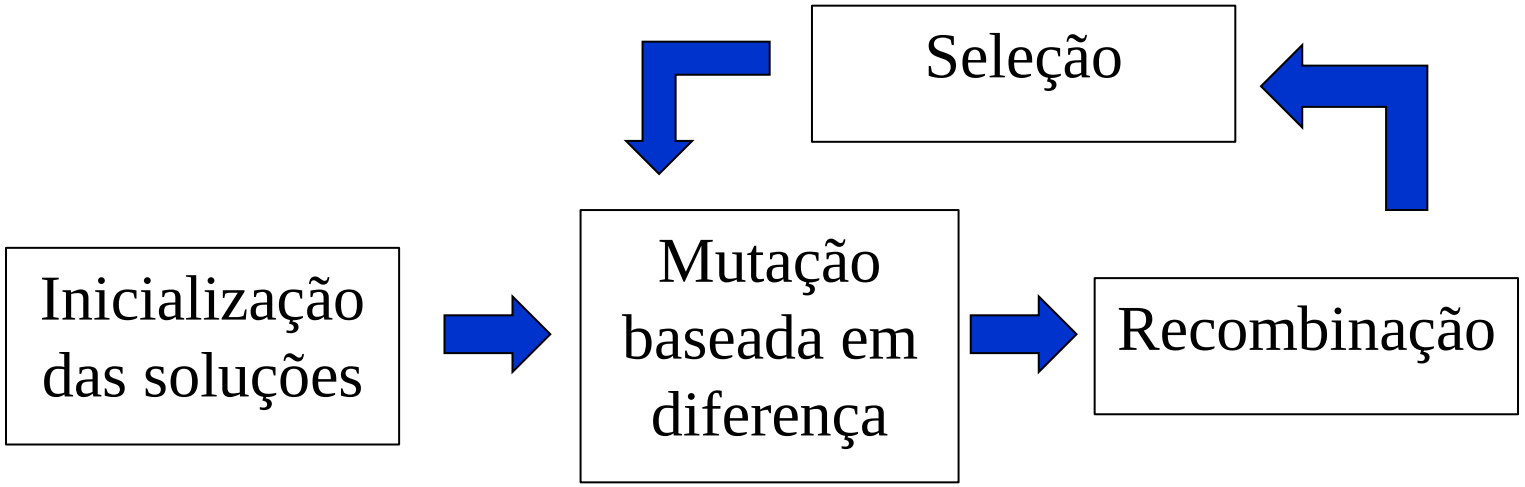
1.7.2. *Differential Evolution* (DE)

- ***Differential Evolution*** **(Evolução Diferencial)**
  - Algoritmo introduzido por Storn e Price (1995)

    http://www1.icsi.berkeley.edu/ftp/pub/techreports/1995/tr-95-012.pdf
  - Utilizado em otimização contínua
    - Tem apresentado ótimos resultados em diversos problemas
  - Característica principal: usar a diferença entre duas soluções para modificar uma terceira solução

1.7.2. *Differential Evolution* (DE)

- **Estágios principais de um algoritmo de DE**



  - Parâmetros
    - taxa de crossover
    - *F* : fator de escala (usado no vetor de diferenças)
    - tamanho da população

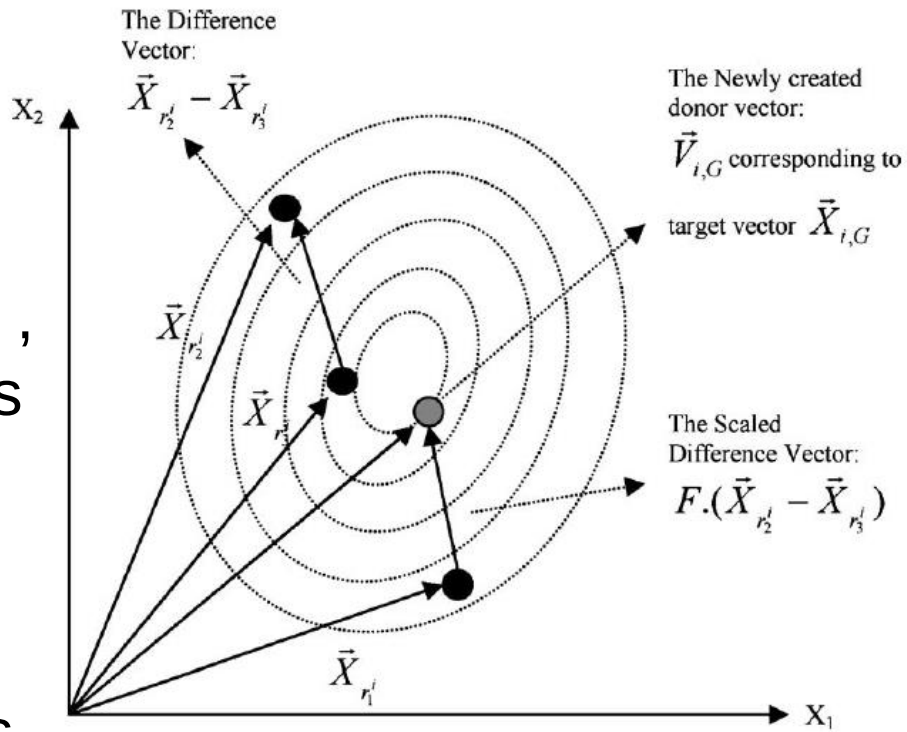
1.7.2. *Differential Evolution* (DE)

- **Mutação**

$$\mathbf{y} = \mathbf{x}_i + F(\mathbf{x}_j - \mathbf{x}_k)$$

sendo **y** o vetor resultante, $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ vetores pais escolhidos aleatoriamente e *F* o fator de escala

- Existem diversas variações desta equação, resultando em variantes da DE



Fonte: Das, S., & Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*(1), 4-31.

1.7.2. *Differential Evolution* (DE)

- **Atrativos**
  - Comparado a vários outras metaheurísticas populacionais, DE é simples de implementar
  - Apresenta bom desempenho em vários problemas com codificação contínua, superando diversos outros algoritmos
  - O número de parâmetros é baixo
    - Além disso, eles podem ser adaptados durante a otimização
  - A complexidade de memória é baixa, quando comparada com o CMA-ES
    - O CMA-ES tem se destacado como a mais competitiva das metaheurísticas para otimização contínua

1.7.3. *Particle Swarm Optimization* (PSO)

- ***Particle Swarm Optimization*** **(Otimização por Enxame de Partículas)**
  - Algoritmo introduzido por Eberhart e Kennedy (1995)

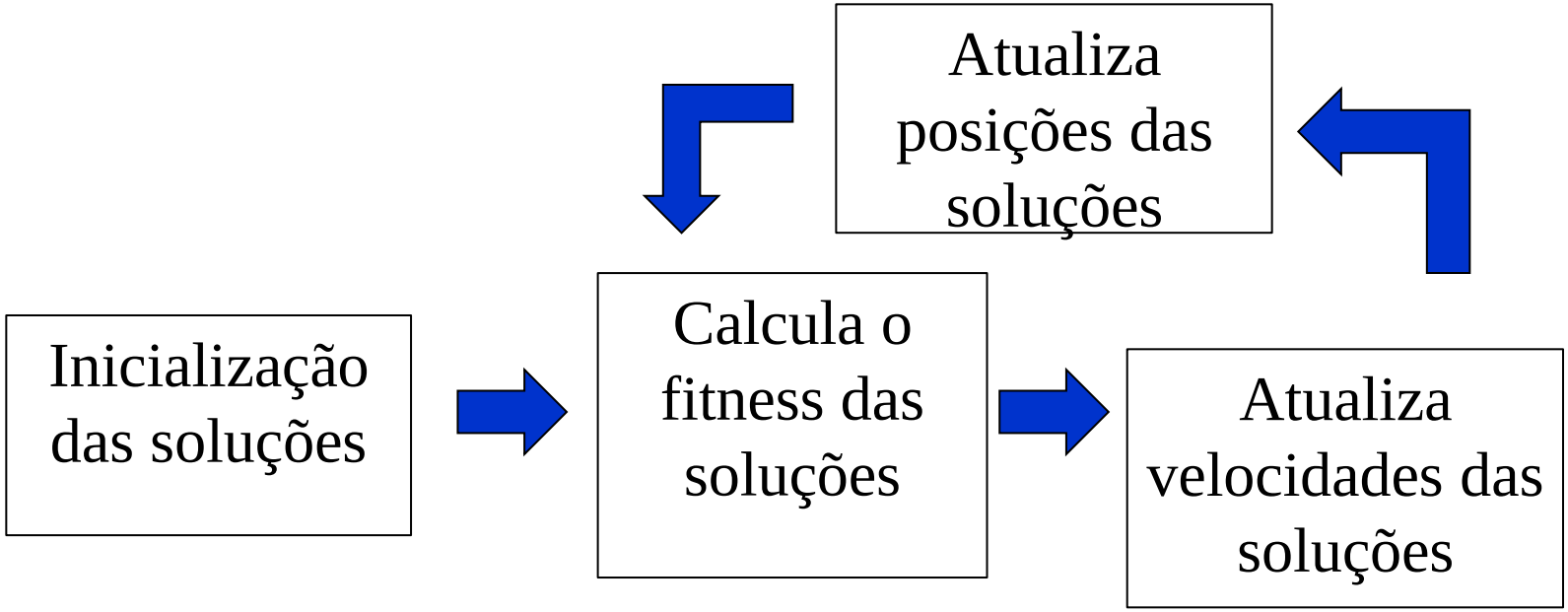
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.3577&rep=rep1&type=pdf

  - Inspirado inicialmente em princípios sócio-psicológicos
    - Mecanismo de reprodução baseado no movimento coletivo de indivíduos (exemplos: pássaros, peixes)
  - Característica principal: as posições e velocidades das soluções são alteradas de acordo com as melhores soluções encontradas

1.7.3. *Particle Swarm Optimization* (PSO)

- **Geralmente utilizada em otimização contínua**

- **Etapas principais**



1.7.3. *Particle Swarm Optimization* (PSO)

- **Atualização das velocidades e posições da *i*-ésima solução em cada dimensão *d***

$$v_{i,d} = v_{i,d} + \alpha\, r_1 (x_{gbest,d} - x_{i,d}) + \beta\, r_2 (x_{nibest,d} - x_{i,d})$$

$$x_{i,d} = x_{i,d} + v_{i,d}$$

sendo que:

- $x_{i,d}$ e $v_{i,d}$ são respectivamente a posição (solução) e velocidade da *i*-ésima solução na dimensão *d*,
- r_1 e r_2 são números aleatórios (distribuição uniforme) entre $[0,\beta]$
- α e β são parâmetros (respectivamente conhecidos como peso de inércia e constante de aceleração; usualmente $\alpha = \beta = 2$)
- $\mathbf{x}_{gbest}$ é a melhor solução encontrada até o momento
- $\mathbf{x}_{nibest}$ é a melhor solução vizinha à *i*-ésima solução
  - Diferentes topologias podem ser consideradas

https://en.wikipedia.org/wiki/File:ParticleSwarmArrowsAnimation.gif

1.7.3. *Particle Swarm Optimization* (PSO)

- **Atrativos**
  - As melhores soluções atuais são utilizadas para atrair as outras soluções
  - Operador de seleção não é explicitamente utilizado
  - Mecanismos para aliviar a convergência prematura podem ser facilmente aplicados
    - Exemplo: uso de repulsão entre as partículas
  - Pode ser adaptado para vários tipos de problemas, como problemas de otimização discreta e **problemas multiobjetivo**
  - Pode ser facilmente hibridizado

1.7.4. *Ant Colony Optimization* (ACO)

- ***Ant Colony Optimization*** **(Otimização por Colônia de Formigas)**
  - Algoritmo introduzido por Dorigo(1992)

    Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*.
  - Inspirado no comportamento de formigas
    - Ao procurar um caminho entre a colônia e uma fonte de comida
  - Técnica probabilística para resolver problemas computacionais que podem ser representados por grafos e no qual as soluções são representadas por caminhos nestes grafos

1.7.4. *Ant Colony Optimization* (ACO)

- **Inspiração**
  - Inicialmente, formigas andam aleatoriamente ao procurarem comida
    - Ao passarem, deixam um rastro de feromônios
      - Os rastros de feromônios evaporam com o tempo
    - Se uma formiga acha um rastro de feromônios, ela tem uma certa probabilidade de seguir este rastro
      - Quando maior a concentração de feromônios, maior é a probabilidade
    - Se formigas podem inicialmente escolher qualquer um dentre vários caminhos até uma fonte de comida, aqueles caminhos menores terão uma maior concentração de feromônios

1.7.4. *Ant Colony Optimization* (ACO)

- **Atualização ("típica") da taxa de feromônio na aresta entre vértices *i* e *j* :**

$$\tau_{i,j} = (1-\rho)\cdot\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{\ k}$$

Taxa de evaporação

Quantidade de feromônio depositada na aresta pela formiga *k*

$$\Delta\tau_{i,j}^{\ k} = \begin{cases} Q/L_k & \text{se a formiga passou pela aresta} \\ 0 & \text{caso contrário} \end{cases}$$

Constante

Qualidade da solução dada pela formiga *k*

1.7.4. *Ant Colony Optimization* (ACO)

- **Estando no vértices *i* , a probabilidade de a formiga *k* ir para o vértice *j* é:**

Constante

$$p_{i,j}^{\ k} = \begin{cases} \dfrac{\tau_{i,j}^{\ \alpha} \cdot \eta_{i,j}^{\ \beta}}{\sum\limits_{c_{i,l} \in \mathbf{N}(s^p)} \tau_{i,l}^{\ \alpha} \cdot \eta_{i,l}^{\ \beta}} & \text{se } c_{i,j} \in \mathbf{N}(s^p) \\ 0 & \text{caso contrário} \end{cases}$$

Conjunto de componentes factíveis

Vértice ainda não visitado

$$\eta_{i,j} = \frac{1}{d_{i,j}}$$

Peso da aresta

1.7.4. *Ant Colony Optimization* (ACO)

Resources on Ant Colony Optimization

- **Web pages:**
  - www.aco-metaheuristic.org: The official Web site of the ant colony metaheuristic (see Figure 4).
  - www.metaheuristics.org: Web site of the "Metaheuristics Network" project. This European Union funded project was dedicated to the theoretical analysis and experimental comparison of metaheuristics.
- **Books:**
  - M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
  - E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
- **Scientific Journals:** Scientific articles on ACO are published in many journals, including "IEEE Transactions on Systems, Man, and Cybernetics", "IEEE Transactions on Evolutionary Computation", "Artificial Life", "INFORMS Journal on Computing", "Journal of Heuristics", "Computers and Operations Research", "Computational Optimization and Applications", and "European Journal of Operational Research". The new journal "Swarm Intelligence," whose first issue is forecast for June 2007, will certainly become the main scientific periodical for disemination of ACO and related research.
- **Conferences:**
  - The biannual series of workshops "ANTS – The International Workshop on Ant Colony Optimization and Swarm Intelligence" (iridia.ulb.ac.be/~ants), held for the first time in 1998, is the oldest conference in the ACO and swarm intelligence fields.
  - The series of conferences "IEEE Swarm Intelligence" (www.computelligence.org/sis) focuses on swarm intelligence techniques and also ant colony optimization.
  - Articles on ACO are regularly presented at other conferences such as "IEEE Congress on Evolutionary Computation (CEC)", "Genetic and Evolutionary Computation Conference (GECCO)", "Parallel Problem Solving from Nature (PPSN)", "INFORMS" meetings, "European Chapter on Combinatorial Optimization (ECCO)" meetings, the "Metaheuristics International Conference (MIC)" and many others.
- **Software:** Software, distributed under the GNU license, is available at: www.aco-metaheuristic.org/aco-code/
- **Popular press:** ACO is often covered by the popular press. Pointers to popularization articles can be found at: www.aco-metaheuristic. org/aco-in-the-press.html
- **Mailing list:** A moderated mailing list dedicated to the exchange of information related to ACO is accessible at: www.aco-metaheuristic. org/mailing-list.html

Comentários

- **Referências**

  - Das, S., & Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on*, *15*(1), 4-31.
  - Verbete PSO em: Sammut, C., & Webb, G. I. (Eds.). (2011). *Encyclopedia of machine learning*. Springer Science & Business Media.
  - Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization. *Computational Intelligence Magazine, IEEE*, *1*(4), 28-39.