

DIGITAL HOUSE

Curso de Ciência de Dados

Computação Natural – Introdução

Uma visão sobre a computação inspirada em fenômenos naturais

Mauricio Noris Freire
mauricio.noris@reachr.com.br

Ementa

- Apresentar o campo de estudo
- Elaboração e prática de algoritmos bio-inspirados

Objetivo

- Abordagens de computação inspiradas na natureza
- Solução de problemas complexos
- Projetar dispositivos computacionais que simulam, emulam, modelam e descrevem sistemas naturais
- Novos paradigmas de computação (DNA, dispositivos quânticos) que podem substituir o silício

A computação natural pode ser vista como uma versão computacional do processo de extração de ideias da natureza para o desenvolvimento de sistemas “artificiais”, ou então a utilização de materiais e mecanismos naturais para realizar computação.

fonte [1] – *Fundamentos da Computação Natural*

Copyrightc Leandro Nunes de Castro

O que compõe a Computação Natural

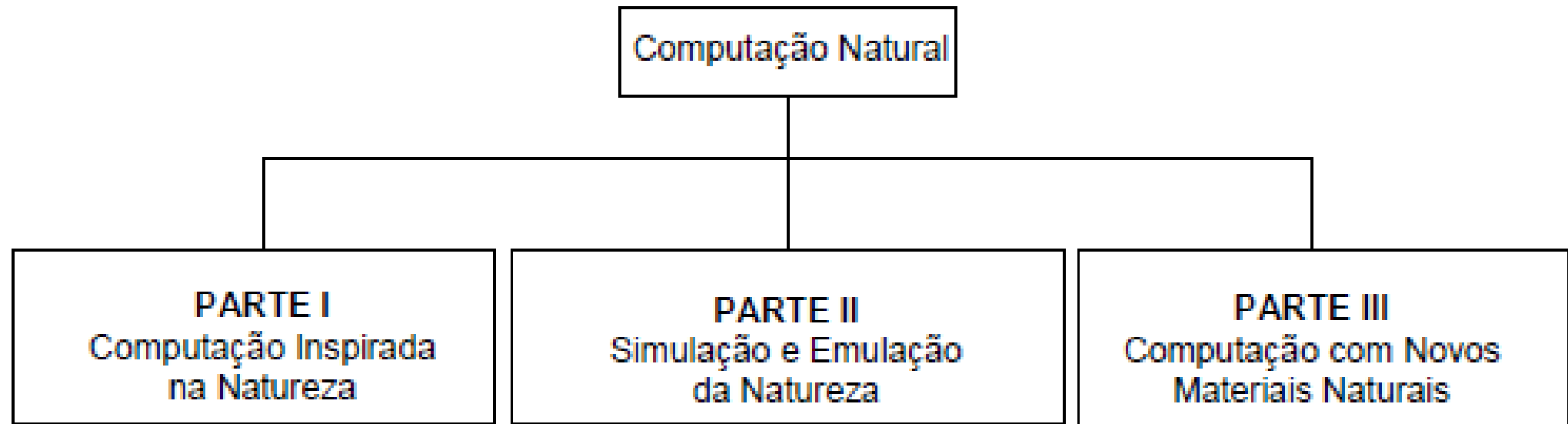


Figura 1: Três principais vertentes da computação natural.

Computação Inspirada na Natureza

- Computação Evolutiva
- Inteligência de Enxame
- Neurocomputação
- Sistemas Imunológicos Artificiais

fonte [1] – *Fundamentos da Computação Natural*

Quando usar

- Problema complexo, muitas soluções, dinâmicos
- Não determinístico, mas com amplo aspecto de soluções e passíveis de comparação
- Problema que não pode ser perfeitamente modelado
 - reconhecimento e classificação de padrões
- Uma única solução não é suficiente
- Sistemas físicos, químicos e biológicos que precisam ser modelados ou emulados com realismo
- Comportamentos e padrões naturais precisam ser reproduzidos com realismo
- Além dos limites da tecnologia computacional atual.

Indivíduos, Entidades e Agentes

- Coletividade:
 - populações de indivíduos,
 - colônias de insetos,
 - revoadas de pássaros,
 - genomas,
- repertórios de células, redes de neurônios
- Tipos de agentes:
 - biológicos (formigas, neurônios, etc.)
 - físicos (robôs, etc.)
 - virtuais (Tamagotchi, etc.)

Paralelismo e Distributividade

- processamento autônomo
- interdependente (mas sem coordenação)
- distribuído
- paralelo

Interatividade

- Os componentes interagem com outros componentes e com o ambiente
 - reprodutiva
 - simbiótica
 - competitiva
 - cooperativa
 - parasítica
- Conectividade (caminhões de interações entre unidades)
 - redes neurais
 - redes imunológicas
- Estigmergia
 - Comportamentos individuais modificam o ambiente resultando em novos comportamentos individuais
 - O ambiente media a comunicação entre os indivíduos (interação indireta)

Adaptação

“Habilidade de um sistema ajustar sua resposta a estímulos ambientais”

- Aprendizagem
 - ato ou processo de adquirir conhecimento, capacidade ou habilidade
 - ocorre pela experiência, o estudo ou interações
 - adaptam/mudam o comportamento baseado em exemplos
 - gradativa, altera padrão comportamental ou alguma característica
- Evolução
 - a vida muda com o tempo, formas mais jovens descendendo de formas mais antigas
 - requer uma população capaz de se reproduzir, sofrer variações genéticas e seleção natural

Realimentação (feedback)

“... a resposta a um estímulo possui algum tipo de efeito sobre o próprio estímulo”

- Positiva
 - quanto mais um evento ocorre, mais ele tende a ocorrer
- Negativa
 - regulador da anterior para manter um equilíbrio (dinâmico) do meio

Auto-organização

- *“De onde vem a ordem que observamos na natureza?”*
- trata da capacidade de formação de padrões em sistemas físicos e biológicos
 - interações internas dos sistemas
 - sem intervenção externa
- coletividade e interatividade, dinâmica, emergência, não linearidades, complexidade, baseada em regras, retroalimentação
- Não possui esta característica quando
 - existe uma liderança; um modelo; uma receita, fôrma ou molde !

Complexidade

- *“Um sistema complexo pode ser definido como aquele cujas propriedades não podem ser completamente compreendidas e nem explicadas estudando-se as partes”*
- componentes que interagem entre si e com o ambiente.
- comportamento global não linear
- sistemas complexos podem ser adaptativos (muda conforme o ambiente)
- emergência

Seleção Natural como um Algoritmo

- Reprodução com herança genética
 - Variação genética
 - Seleção natural
-
- *“De acordo com a teoria de Darwin, quando a seleção natural é imposta sobre as variações ocorre uma **movimentação mensurável no espaço de buscas**. Esta movimentação pode ser vista como um processo de pesquisa e desenvolvimento (P&D) que pode resultar em uma busca ou acúmulo de projeto **(otimização)**.”*

Biologia é Engenharia

- *“... o trabalho feito pela seleção natural é P&D, então biologia e fundamentalmente engenharia, uma conclusão que esclarece alguns de nossos mais profundos enigmas a distinção entre inteligência artificial e natural desaparece.”*
- O funcionamento de toda estrutura carrega implicitamente informações sobre o ambiente.
 - o A posição dos órgãos no corpo de uma serpente (distribuição e locomoção)
 - As asas de uma ave (aerodinâmica)

Reprodução

Assexuada

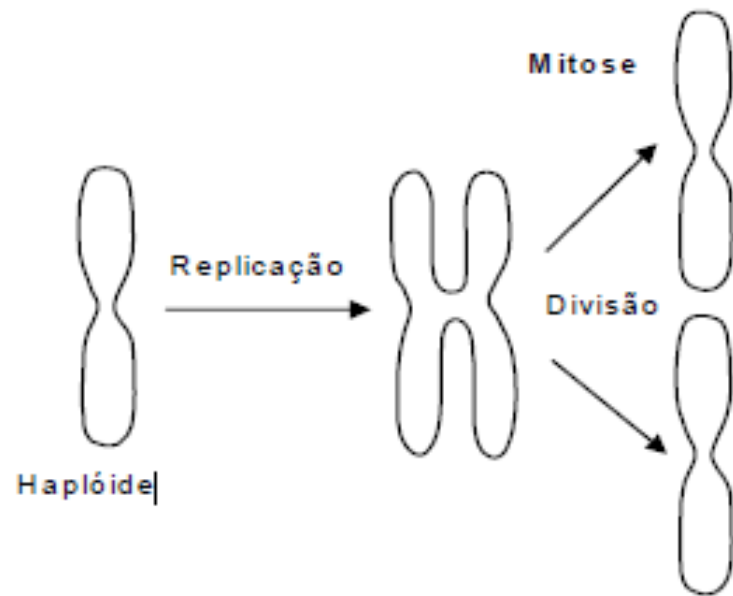


Figura 11: Reprodução assexuada em cromossomos haploides.

Sexuada

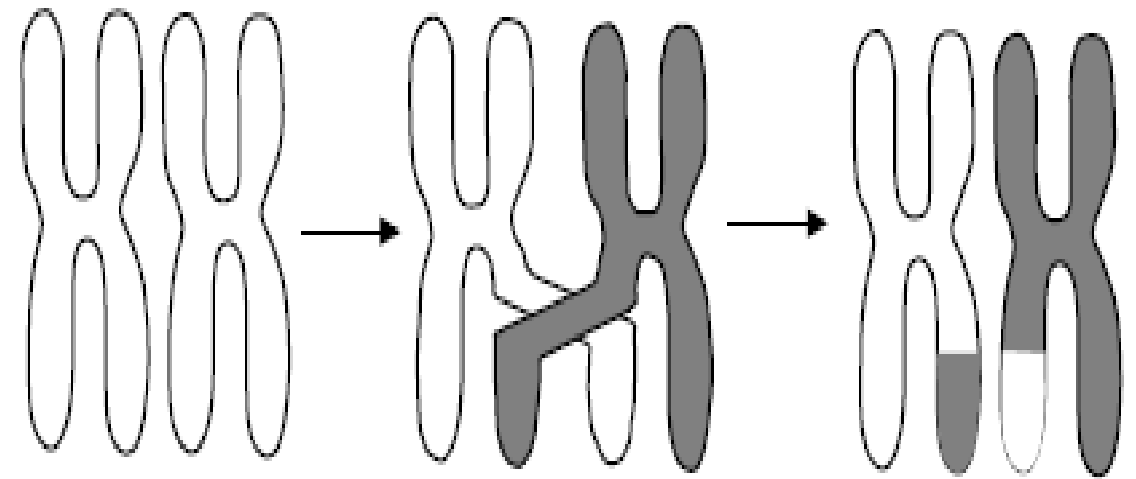


Figura 12: Recombinação genética (crossover) entre dois cromossomos.

Mutação

Tanto na reprodução assexuada como na sexuada, o processo de copia do material genético dos pais está sujeito a erros, denominados mutações.

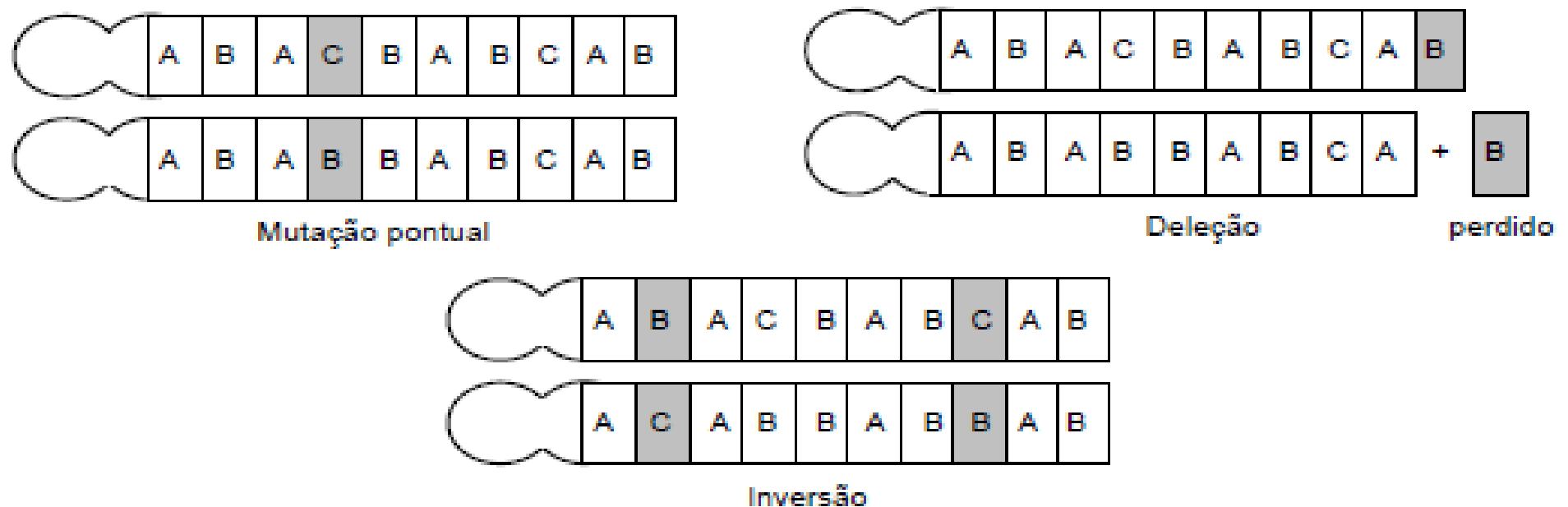


Figura 13: Ilustração de diferentes tipos de mutação.

Formulação do problema

- Representação dos dados
- Definição do objetivo
- Função de avaliação

Algoritmos Genéticos

- abstrair e rigorosamente explicar processos adaptativos
- dev. simulações que retenham os mecanismos originais (natureza)
- características
 - 2 espaços de trabalho: genótipo (a informação) e fenótipo (comportamento)
 - busca sobre uma população e não um único indivíduo
 - funções de fitness (avaliação)
 - regras de transição probabilísticas

Representação (Estrutura de dados)

- codificação do indivíduo
- O problema a ser resolvido e capturado em uma função objetivo que indica o fitness (adaptabilidade) de cada candidato a solução.

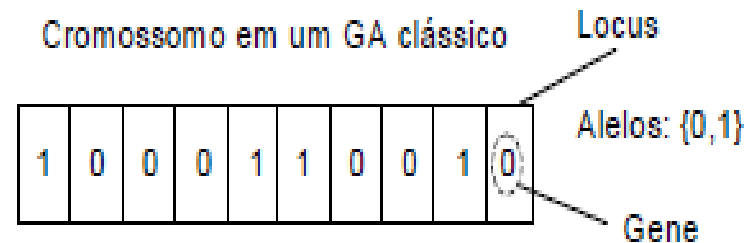
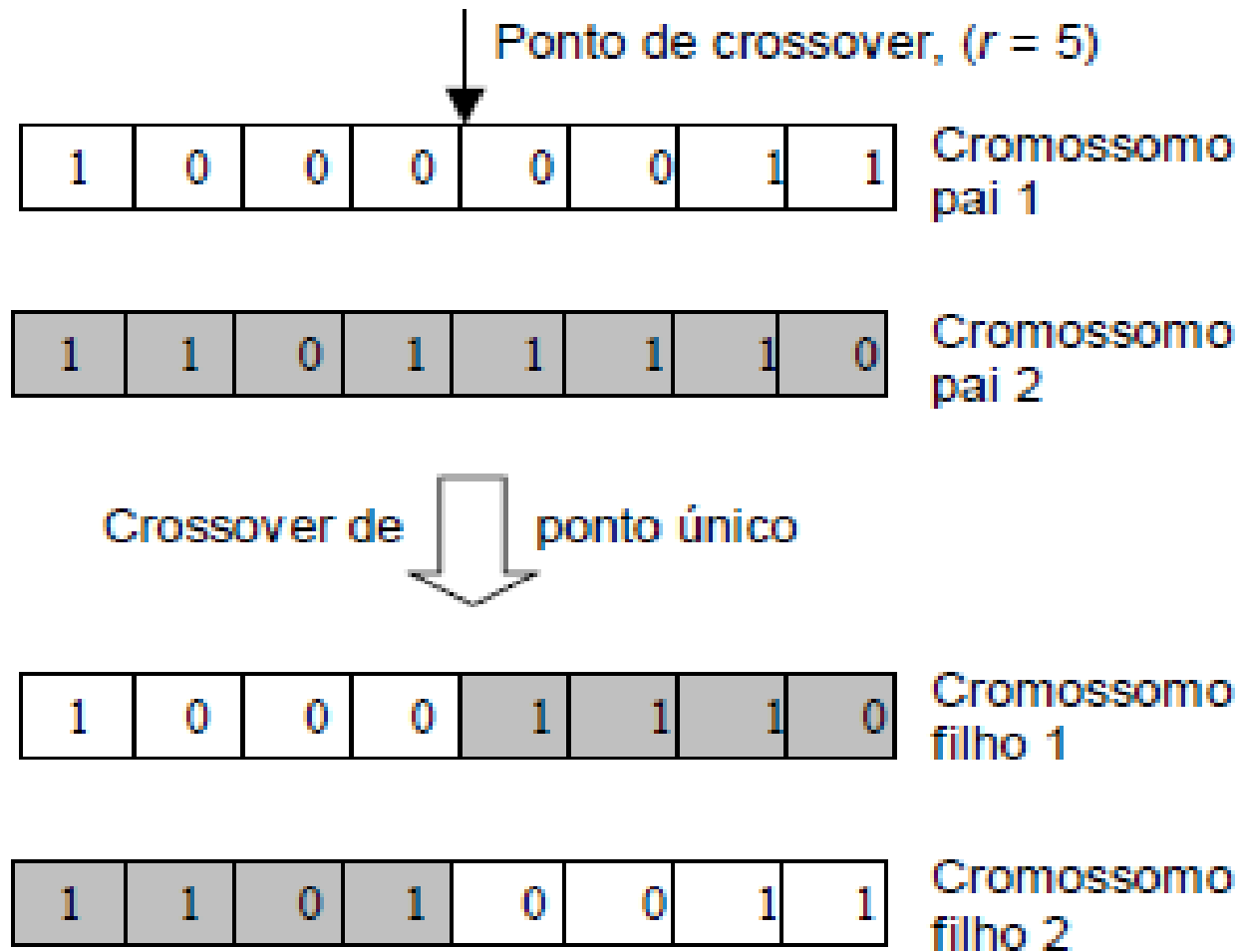


Figura 14: Cadeia binária de comprimento $l = 10$ correspondente à estrutura de dados de um GA clássico.

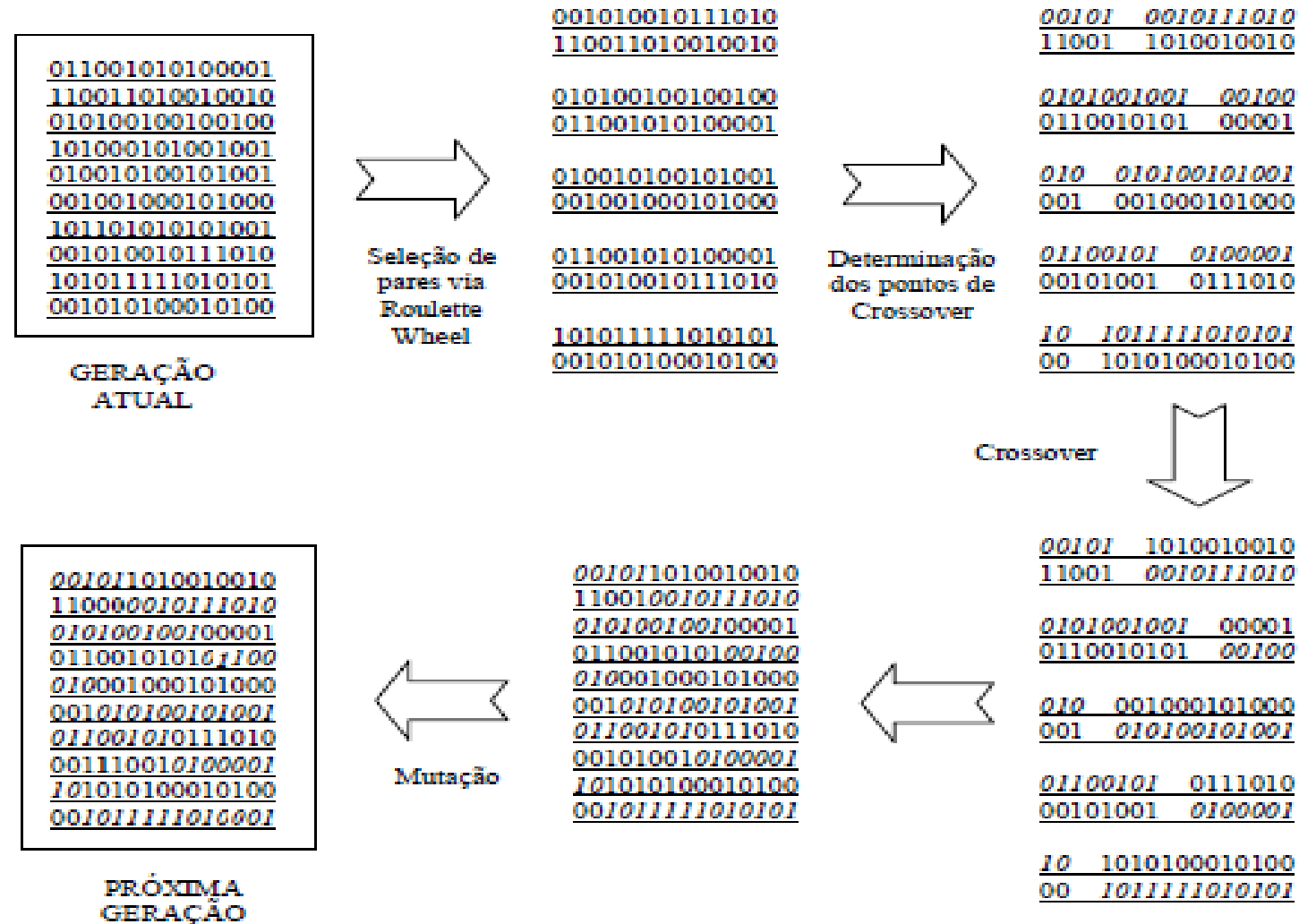
Mecanismo de Seleção

- gerador de números pseudo-aleatórios
- permite a “morte” do melhor indivíduo (melhor fitness)
- a probabilidade de reprodução de cada indivíduo determina a população seguinte
- esta probabilidade se dá por operadores genéticos específicos ao problema:
 - crossover
 - mutação

Crossover



Mutação



Fitness

- Verificar quanto determinado indivíduo está em relação ao objetivo
- Classificar os indivíduos
- Aplica-se após a reprodução e reordena os indivíduos para que os melhores (mais adaptados) se reproduzam e os outros não !

Aplicações

- Planejamento
 - Roteamento (Caixeiro viajante, Roteamento de veículos, Problemas de transporte, Robótica)
 - Sequenciamento de tarefas (*job shop scheduling*, *timetabling*, processamento computacional, empacotamento)
- Projeto
 - Filtros, Processamento de sinais, Sistemas inteligentes, Engenharia
- Simulação e identificação
- Classificação
 - Sistemas de Controle
 - Jogos

Atividade 1 - Criando um algoritmo evolutivo

Etapas:

- 1) Escolher um caractere da base MINST
- 2) Gerar uma população aleatória um e com fenótipo similar ao caractere escolhido na etapa anterior
- 3) Determinar e classificar o *fitness* da geração
- 4) Escolher o conjunto de indivíduos aptos a se reproduzir
- 5) Executar o *crossover* de modo que se gere uma nova geração. O genótipo de cada novo indivíduo será gerado é composto pela combinação genética de dois indivíduos da geração anterior
- 6) Efetuar a mutação aleatória de alguns genes nos indivíduos da nova geração
- 7) Retornar a etapa 3 e executar este processo por # gerações, ou até que o fitness esteja próximo de 100%

Inteligência de Enxame

- Sociabilidade
- Estudo e criação de soluções inspiradas em coleções de agentes simples que interagem em um mesmo ambiente de acordo com regras locais
- qualquer tentativa de projetar algoritmos ou dispositivos distribuídos inspirados no comportamento de insetos ou outros conjuntos
- composto por agentes não inteligentes e com capacidade individual limitada, capazes de apresentar comportamentos coletivos inteligentes

Características

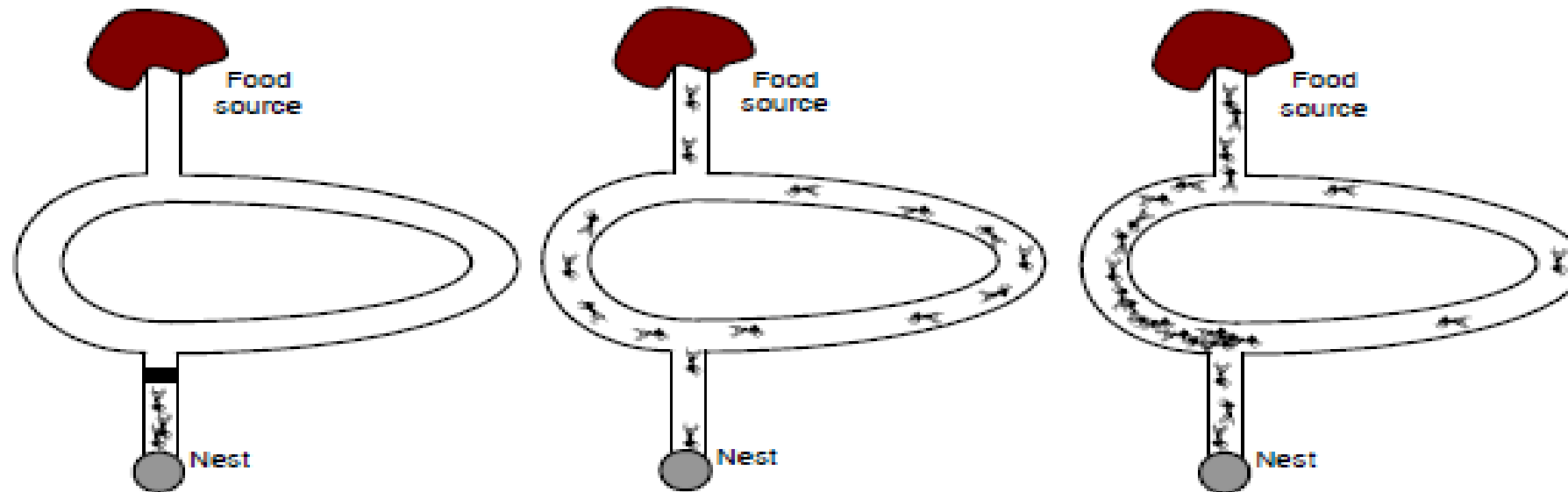
- **Proximidade:** os agentes devem ser capazes de interagir
- **Qualidade:** os agentes devem ser capazes de avaliar seus comportamentos
- **Diversidade:** permite ao sistema reagir a situações inesperadas
- **Estabilidade:** nem todas as variações ambientais devem afetar o comportamento de um agente
- **Adaptabilidade:** capacidade de se adequar a variações ambientais

Exemplos

- formigas
- revoada de pássaros
- engarrafamento de automóveis
- multidão
- mercado de ações visto como um enxame de investidores
- economia como um enxame e agentes econômicos

Coleta de Alimento pelas Formigas

- Comportamentos observados:
 - construção de uma trilha de feromônio
 - seguir a trilha com maior feromônio



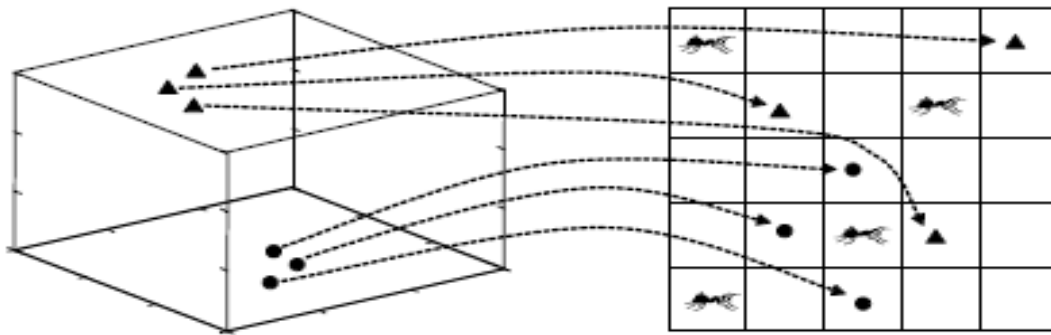
ACO - Algoritmo de Otimização por Colônias de Formigas

- procedimento paralelo para construção/modificação de trilhas
- regra de atualização de feromônio

```
procedure [best] = ACO(max_it)
    initialize  $\tau_{ij}$  //usually every edge is initialized with the same  $\tau_0$ 
    place each ant  $k$  on a randomly selected edge
     $t \leftarrow 1$ 
    while  $t < \text{max\_it}$  do,
        for  $i = 1$  to  $N$  do, //for each ant
            build a solution by applying a probabilistic transition
            rule  $(e-1)$  times. The rule is a function of  $\tau$  and  $\eta$ 
            //e is the number of edges on the graph  $G$ 
        end for
        eval the cost of every solution built
        if an improved solution is found,
            then update the best solution found
        end if
        update pheromone trails
         $t \leftarrow t + 1$ 
    end while
end procedure
```


ACA – Algoritmo Simples de Clusterização

- “agentes formigas” movem-se aleatoriamente em um grid bidimensional
- captura e transporta para outras posições
- função f de visibilidade das formigas



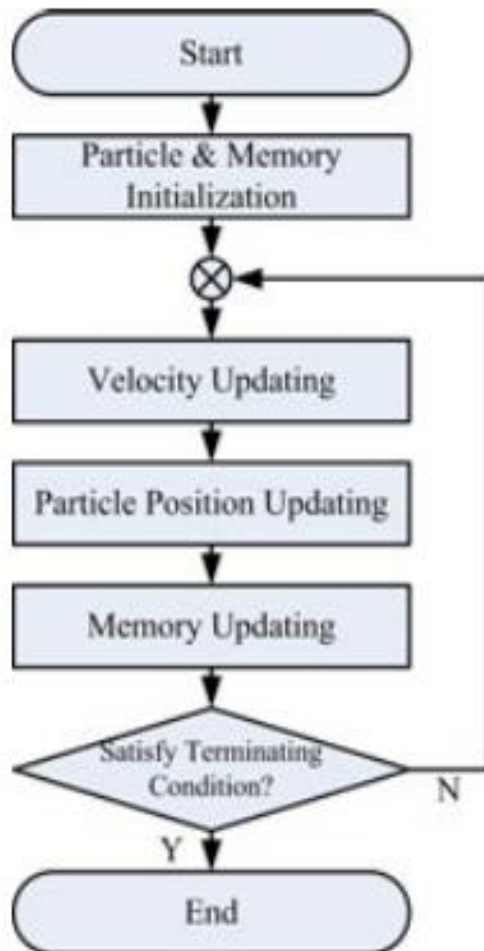
fonte [1] – *Fundamentos da Computação Natural*

```
procedure [] = ACA(max_it, N, k1, k2)
  place every item  $i$  on a random cell of the grid
  place every ant  $k$  on a random cell of the grid
  unoccupied by ants
   $t \leftarrow 1$ 
  while  $t < \text{max\_it}$  do,
    for  $i = 1$  to  $N$  do, //for every ant
      if unladen ant AND cell occupied by item  $\mathbf{x}_i$ , then
        compute  $f(\mathbf{x}_i)$  and  $p_p(\mathbf{x}_i)$ 
        pick up item  $\mathbf{x}_i$  with probability  $p_p(\mathbf{x}_i)$ 
      else if ant carrying item  $\mathbf{x}_i$  AND cell empty, then
        compute  $f(\mathbf{x}_i)$  and  $p_d(\mathbf{x}_i)$ 
        deposit (drop) item  $\mathbf{x}_i$  with probability  $p_d(\mathbf{x}_i)$ 
      end if
      move to a randomly selected neighboring and unoccupied cell
    end for
     $t \leftarrow t + 1$ 
  end while
  print location of items
end procedure
```

PSO – Particle Swarm Optimization

- Apresentado em 1995 por R. Eberhart and J. Kennedy [2]
- Diferentemente da maioria dos otimizadores, não utiliza gradiente descendente [2]
- Uma das principais críticas é que o algoritmo pode ficar preso em ótimos locais [7]
- Diversas variantes do algoritmo original têm sido apresentados desde então [6] e [7]

Conceitos



1. Inicialização:

- # partículas
- Velocidade
- Melhor solução local

2. Atualização da velocidade

$$v_i(t+1) = v_i(t) + c_1 \cdot r_1 \cdot (b_i(t) - x_i(t)) + c_2 \cdot r_2 \cdot (G(t) - x_i(t)).$$

3. Atualização da posição

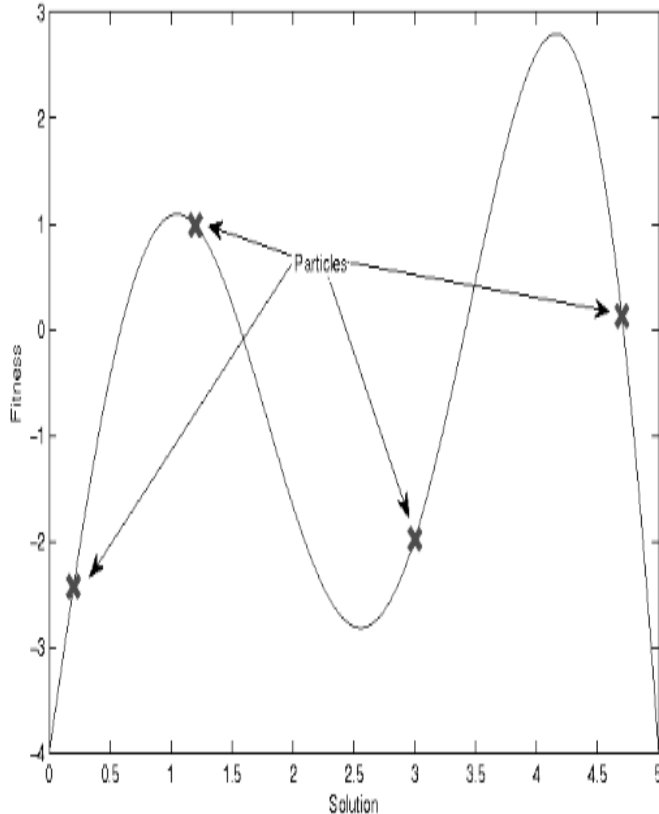
$$x_i(t+1) = x_i(t) + v_i(t), \quad i = 0, 1, \dots, M-1.$$

4. Atualização da Memória

- Atualiza o melhor global, senão ignora a interação

fonte: [4] – Procedimentos para a otimização usando partículas

Conceitos



fonte: [3] – PSO em estado inicial

1. Atualização de velocidade é a parte da otimização

$$v_i(t+1) = \underbrace{w}_{\text{Componente de Inércia}} \underbrace{v_i(t)}_{\text{Componente cognitivo}} + \underbrace{c_1 r_1 [\hat{x}_i(t) - x_i(t)]}_{\text{Componente cognitivo}} + \underbrace{c_2 r_2 [g(t) - x_i(t)]}_{\text{Componente social}}$$

Componente de Inércia

responsável por manter a partícula na mesma direção que originalmente se encontrava

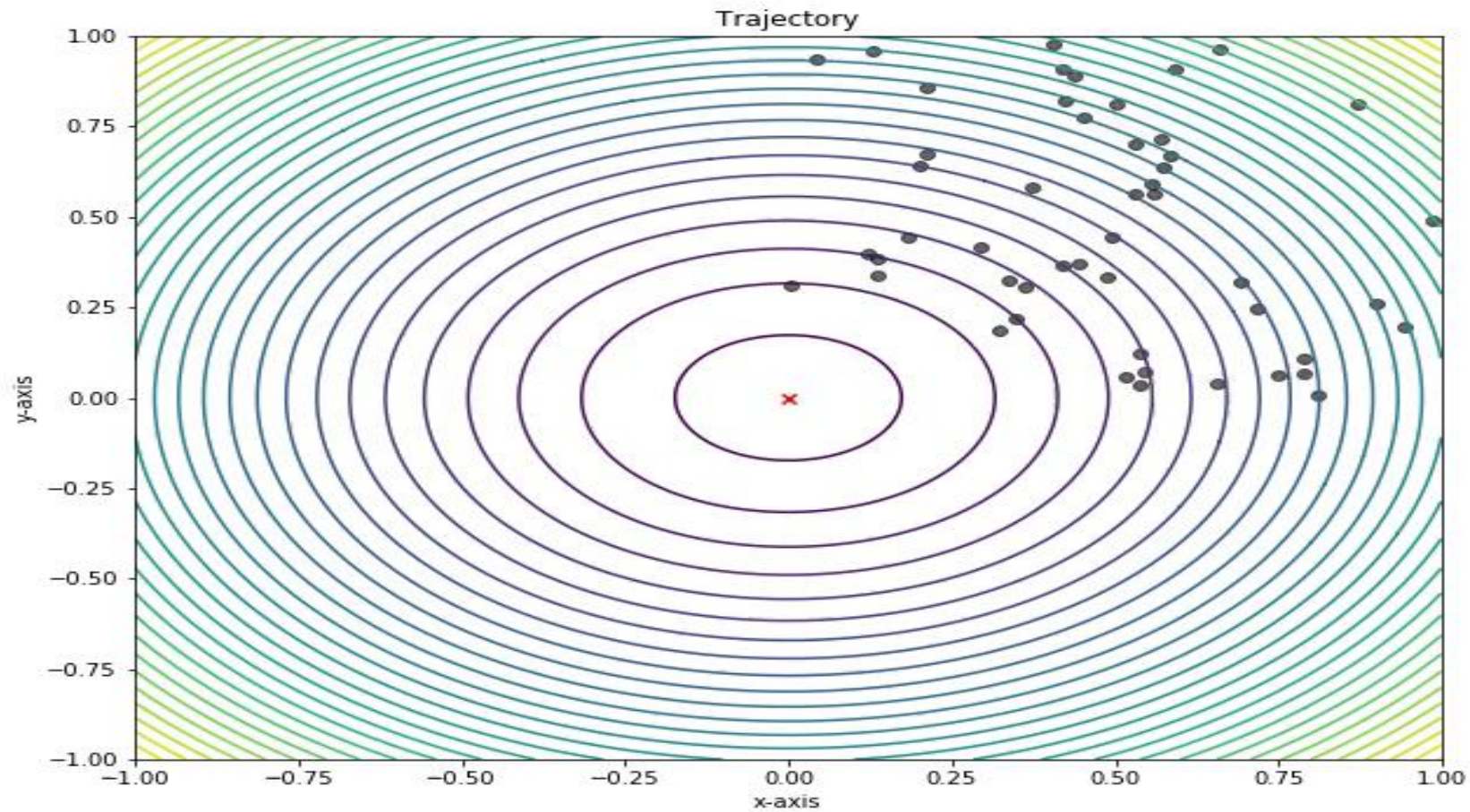
Componente cognitivo

atua como a memória da partícula, causando a tendência a retornar às regiões do espaço de busca em que experimentou altos padrões individuais

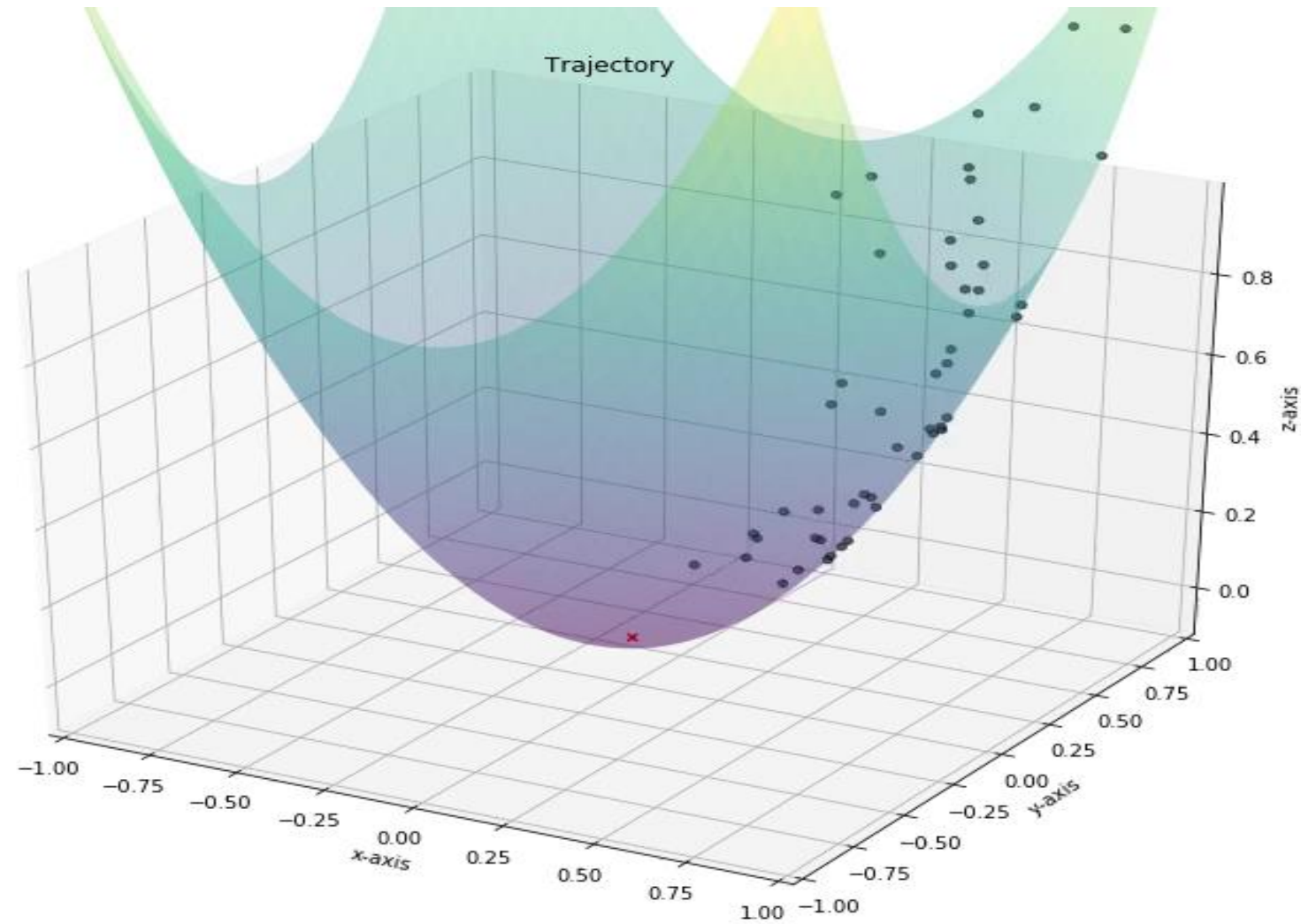
Componente social

faz com que a partícula se mova para o melhor região que o enxame encontrou até agora

Simulação 2D



Simulação 3D



PySwarms [5]

- PySwarms
 - Otimização de objetivo único em que o espaço de pesquisa é discreto. Útil para agendamento de trabalho, cacheiro viajante ou qualquer outro problema baseado em sequência
 - Otimização de objetivo único em que o espaço de pesquisa é contínuo. Perfeito para otimizar várias funções comuns

Atividade 2 - PSO

Imagine que você e seus amigos estejam procurando um tesouro juntos. O tesouro é mágico e recompensa não apenas aquele que o encontra, mas também aqueles que o procuram. Seu grupo sabe, aproximadamente, onde está o tesouro, mas não exatamente certo de sua localização definitiva.

Seu grupo então decidiu se separar de walkie-talkies e detectores de metal. Você usa seu walkie-talkie para informar a todos sobre sua posição atual e o detector de metais para verificar sua proximidade com o tesouro. Em troca, você ganha conhecimento das posições de seus amigos e também da distância deles ao tesouro.

Como membro do grupo, você tem duas opções:

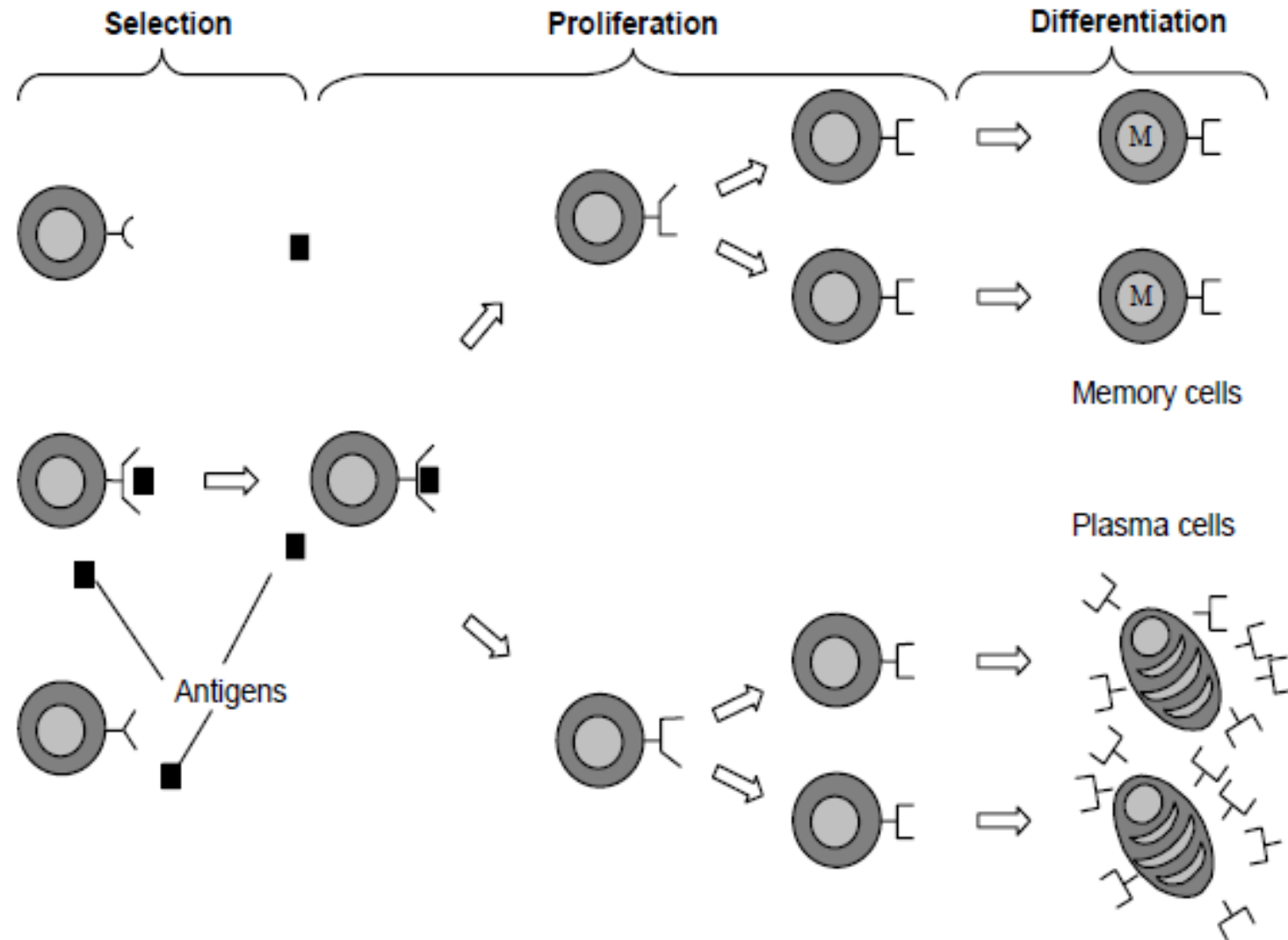
1. Ignore seus amigos e procure pelo tesouro do jeito que você quer. O problema é que, se você não encontrou, e você está longe disso, você recebe uma recompensa muito baixa.
2. Usando as informações coletadas do seu grupo, coordene e encontre o tesouro juntos. A melhor maneira é saber quem é o mais próximo do tesouro e ir em direção a essa pessoa.

fonte: [5] - Introdução

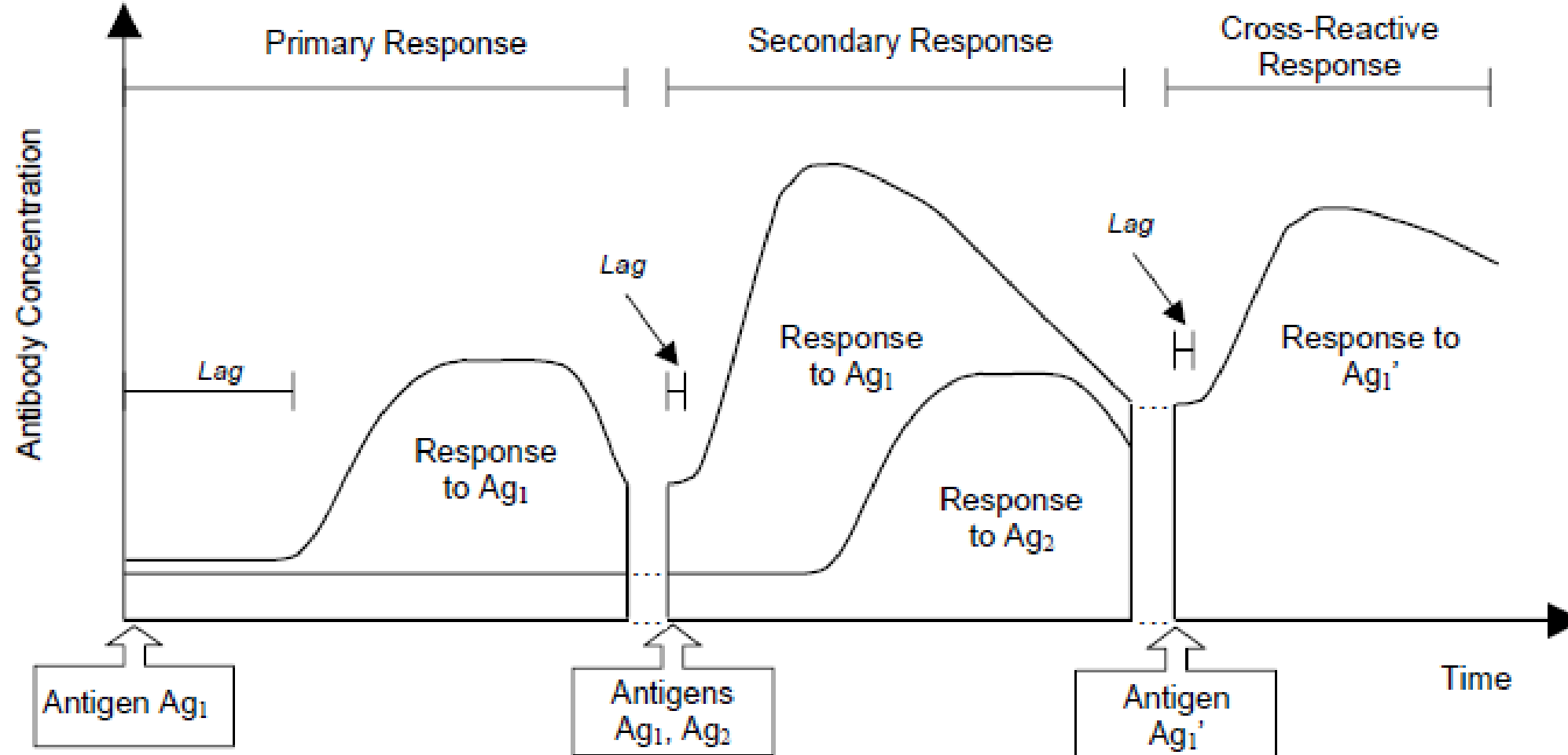
Sistemas Imunológicos Artificiais

- Imunologia é o estudo de mecanismo de defesa e resistência contra ataques externos
- responsável por nos proteger
- possui diferentes níveis, alguns até redundantes
- é adaptável (possui aprendizado e memória)

Seleção clonal e maturação da afinidade



Maturação e reatividade cruzada



fonte [8] – *Introduction to Artificial Immune Systems (AIS)*

Propriedades

- singularidade
- diversidade
- robustez
- autonomia
- divididos em camadas de resposta
- distribuída
- aprendizado guiado e memória
- “*predator-prey behavior*”
- tolerância a ruídos (o reconhecimento do antígeno não precisa ser perfeito)

Aplicações

- Reconhecimento de padrões
- Aproximação de funções
- Otimização
- Análise de dados e clustering
- Machine Learning
- Geração e manutenção de diversidade
- Geração de comportamentos emergentes
- Computação e programação evolucionária
- Detecção de falhas e anomalias

Algoritmo Clonal

```
Input:  $Population_{size}$ ,  $Selection_{size}$ ,  $Problem_{size}$ ,  $RandomCells_{num}$ ,  $Clone_{rate}$ ,  $Mutation_{rate}$   
Output: Population  
 $Population \leftarrow CreateRandomCells(Population_{size}, Problem_{size})$   
While ( $\neg StopCondition()$ )  
  For ( $p_i \in Population$ )  
    Affinity( $p_i$ )  
  End  
   $Population_{select} \leftarrow Select(Population, Selection_{size})$   
   $Population_{clones} \leftarrow \emptyset$   
  For ( $p_i \in Population_{select}$ )  
     $Population_{clones} \leftarrow Clone(p_i, Clone_{rate})$   
  End  
  For ( $p_i \in Population_{clones}$ )  
    Hypermutate( $p_i, Mutation_{rate}$ )  
    Affinity( $p_i$ )  
  End  
   $Population \leftarrow Select(Population, Population_{clones}, Population_{size})$   
   $Population_{rand} \leftarrow CreateRandomCells(RandomCells_{num})$   
  Replace( $Population, Population_{rand}$ )  
End  
Return (Population)
```

Pseudocode for CLONALG.

Atividade [3] – SIA

Baseando-se na implementação KAPA para o algoritmo Clonal (<https://github.com/kittsville/KAPA/> [9]), observe e analise os seguintes cenários de uso:

- 1) Dado um conjunto de antígenos, o KAPA produzirá anticorpos capazes de identificar os antígenos. Isso pode ser analisado com o KAPA no conjunto original de oito dígitos. Isso foi feito recriando os caracteres binários 10x12 no artigo original Os caracteres foram convertidos em antígenos representando-os usando o espaço de forma Hamming, onde cada caractere é um bitstring de comprimento 120. Os pixels preto e branco foram convertidos para valores True / False, respectivamente.
- 2) O número de gerações tomadas para a convergência permanece constante conforme aumenta o número de antígenos, dado $Ag \leq Ab$. Isto pode ser medido usando um modelo linear com o número de antígenos como a variável independente e o número de gerações tomadas para convergência como a variável dependente. KAPA foi executado 30 vezes cada em um número crescente de antígenos gerados aleatoriamente $Ag = 1..10$.

Referências

- [1] Fundamentos da Computação Natural <http://www.slideshare.net/Indecastro>.
- [2] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” 1995.
- [3] J. Blondin, “Particle Swarm Optimization: A Tutorial,” 2009.
- [4] S.-C. Chu, H.-C. Huang, J. F. Roddick, and J.-S. Pan, “Overview of Algorithms for Swarm Intelligence,” 2011.
- [5] https://pyswarms.readthedocs.io/en/latest/examples/basic_optimization.html
- [6] F. Van Den Bergh, “An Analysis of Particle Swarm Optimizers,” 2001.
- [7] D. Sedighizadeh and E. Masehian, “Particle Swarm Optimization Methods, Taxonomy and Applications,” Int. J. Comput. Theory Eng., pp. 486–502, 2009.
- [8] Introduction to Artificial Immune Systems (AIS) – Leandro Nunes de Castro
- [9] - <https://github.com/kittsville/KAPA/>
- [10] - <http://www.cleveralgorithms.com/nature-inspired/index.html>