

Evolução Estrutural e Paramétrica de Redes Neurais Dinâmicas em Vida Artificial

Defesa de Mestrado

Candidato: **Cesar Gomes Miguel**

Orientador: **Prof. Dr. Marcio Lobo Netto**

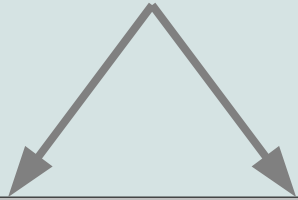
23 de Março de 2009



USP

Universidade de São Paulo
Escola Politécnica
Engenharia de Sistemas Eletrônicos

Comportamento

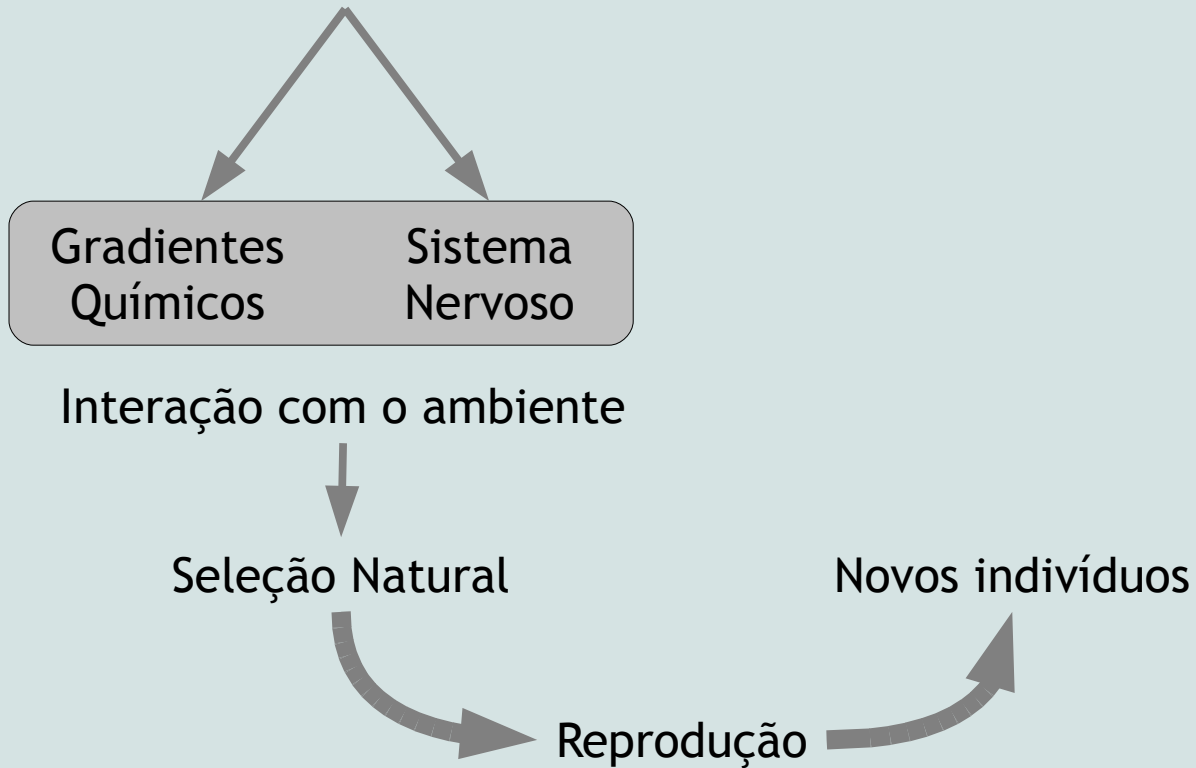


Gradientes
Químicos

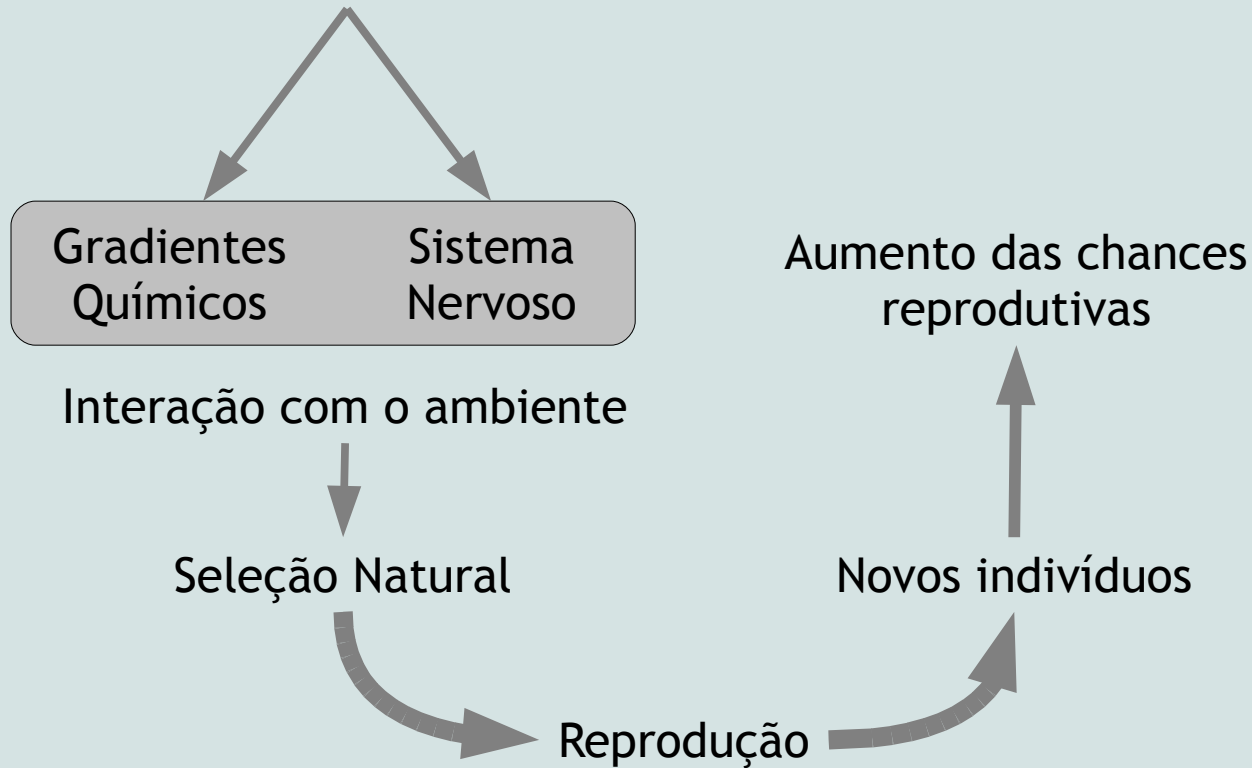
Sistema
Nervoso

Interação com o ambiente

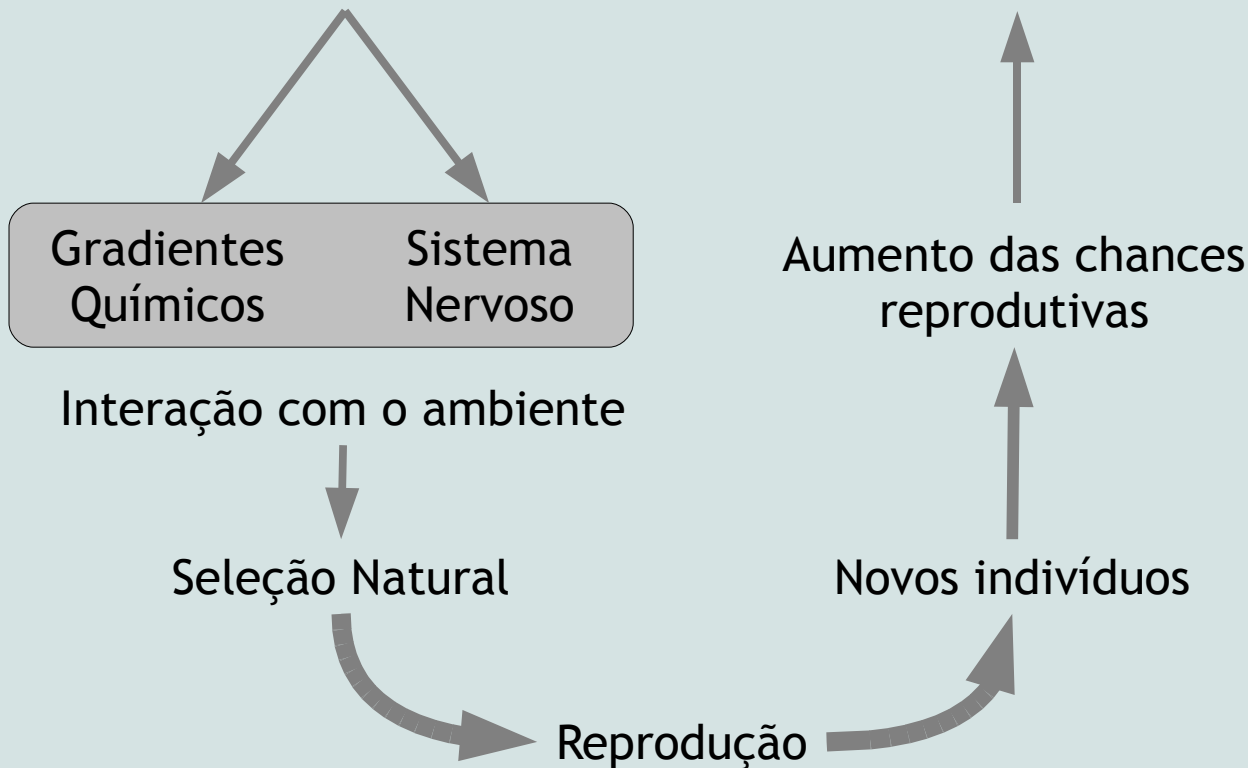
Comportamento



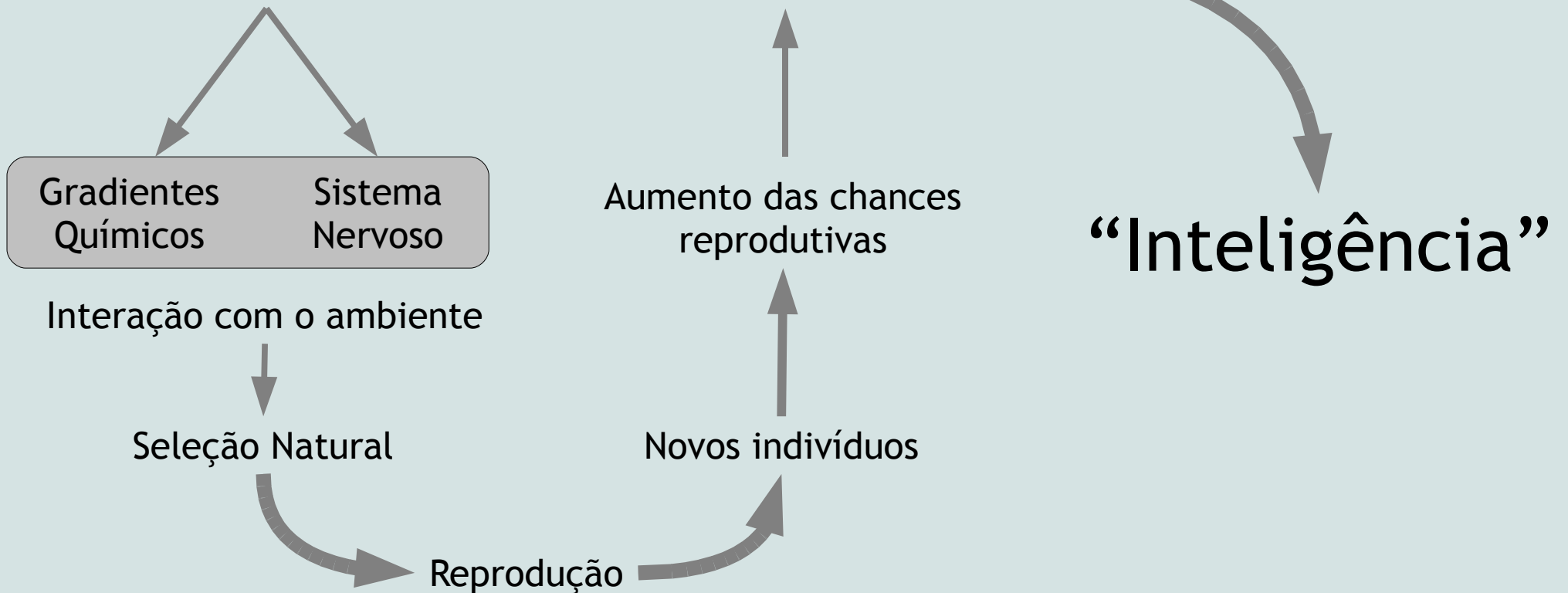
Comportamento



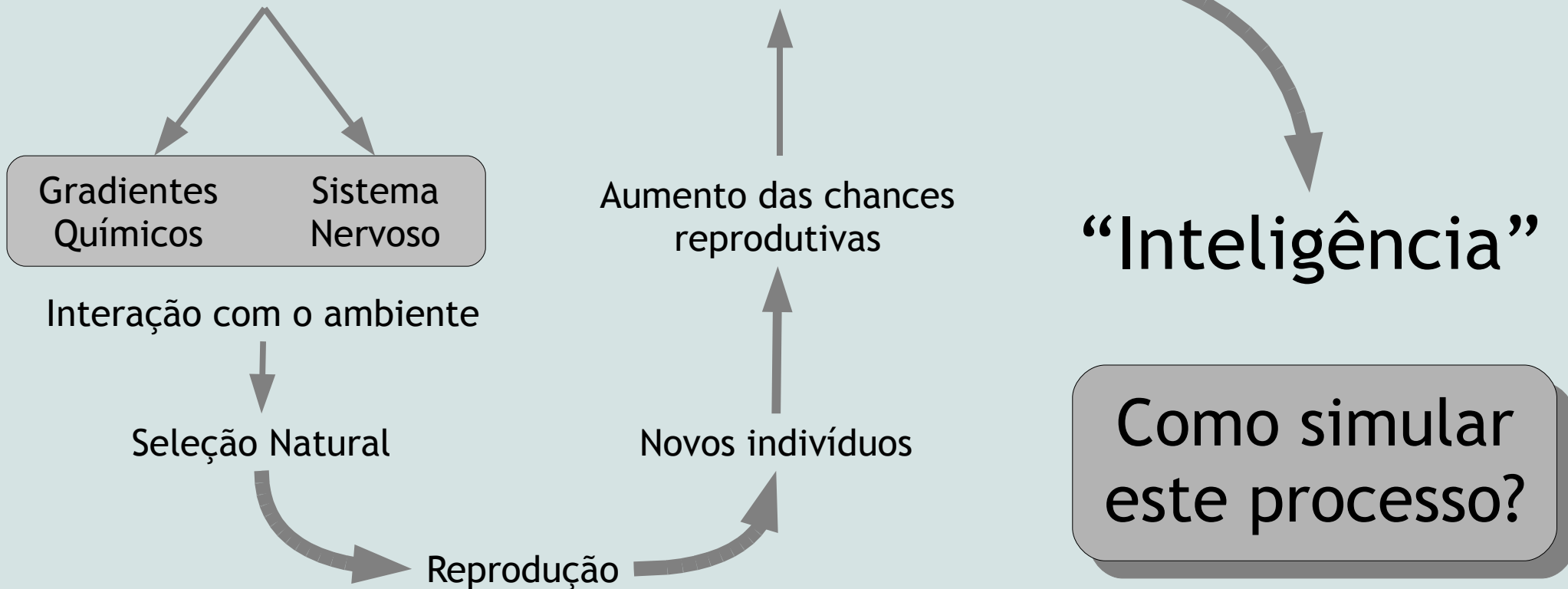
Comportamento Adaptativo



Comportamento Adaptativo



Comportamento Adaptativo



Uma forma possível é inspirada na própria biologia, usando como modelo:

- **Algoritmos Genéticos (AGs)**

Simulação do processo evolutivo.

Uma forma possível é inspirada na própria biologia, usando como modelo:

- **Algoritmos Genéticos (AGs)**

Simulação do processo evolutivo.

- **Redes Neurais Artificiais (RNAs)**

Modelo para o sistema nervoso, reproduzindo suas principais características.

Uma forma possível é inspirada na própria biologia, usando como modelo:

- **Algoritmos Genéticos (AGs)**
Simulação do processo evolutivo.
 - **Redes Neurais Artificiais (RNAs)**
Modelo para o sistema nervoso, reproduzindo suas principais características.
- } **Neuroevolução**

Objetivo Geral

Investigar métodos para combinar RNAs e AGs, de forma que possam ser usados no controle e adaptação de indivíduos (*organismos artificiais*) situados num ambiente virtual.

Objetivos Específicos

- Histórico e teoria sobre RNAs e AGs;
- Neuroevolução (RNAs e AGs);
- Análise do NEAT;
- Desenvolvimento de uma biblioteca para o NEAT;

Objetivos Específicos

- Análise de simuladores para ambientes virtuais;
- Realização de experimentos utilizando a biblioteca proposta para o NEAT;
- Avaliação dos resultados.

Neuroevolução



USP

Neuroevolução

Neuroevolução = 90% AG + 10% RNA



USP

Neuroevolução = 90% AG + 10% RNA

Idéia:

- Estabelecer uma codificação eficiente para representar arquiteturas de redes neurais (i.e., definir o genótipo);

Neuroevolução = 90% AG + 10% RNA

Idéia:

- Estabelecer uma codificação eficiente para representar arquiteturas de redes neurais (i.e., definir o genótipo);
- Para o Algoritmo Genético, a estrutura em si é irrelevante desde que:
 - (a) O fenótipo possa ser avaliado;
 - (b) Os operadores de recombinação e mutação “entendam” a estrutura utilizada;

Principais vertentes:

- Evolução dos pesos sinápticos (topologia fixa);
- Evolução dos pesos e da topologia;

Principais aplicações:

- Modelo bio-inspirado para a evolução do sistema nervoso.
- Método automático para determinar a topologia e os pesos sinápticos, assim como outros parâmetros.

**Vida
Artificial**

Otimização

Alguns problemas:

1. Como representar a rede neural num genótipo?
2. Como cruzar redes neurais de topologias/tamanhos arbitrários?
3. Como medir a similaridade entre duas redes?

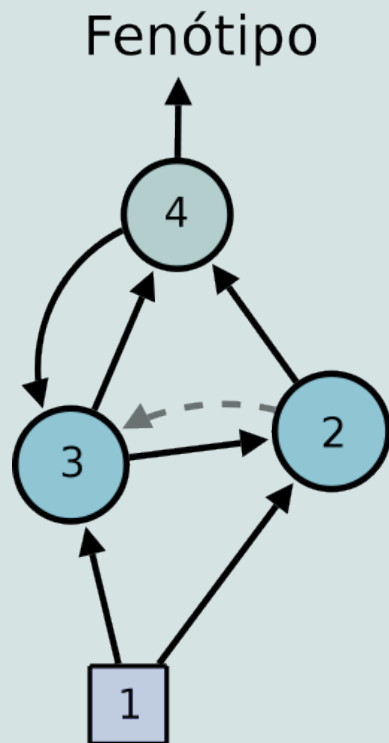
NEAT (*NeuroEvolution of Augmenting Topologies*)

- Início em 2002 (*Kenneth Stanley, UT em Austin*);
- Novas abordagens para os problemas anteriores;
- “Principal” método de neuroevolução;
- Largo campo de aplicação.

Principais características:

- Codificação genética flexível;
- Introdução de marcadores históricos (genes homólogos);
- Especiação;
- Minimiza o espaço de busca por complexificação,

Codificação genética do fenótipo:



Genótipo

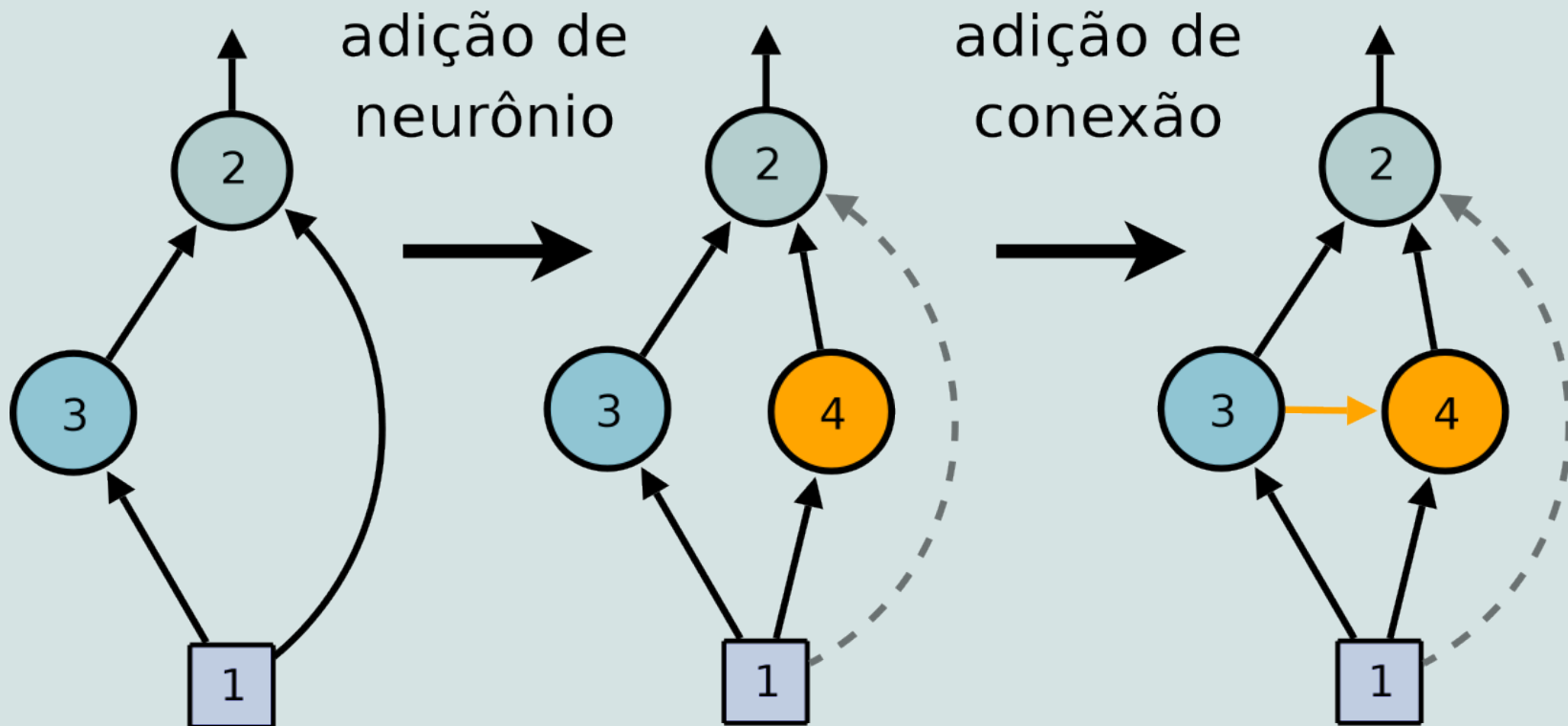
Genes de ganho sináptico

Peso:	0.25	2.31	1.55	0.98	5.11	1.17	0.07
De:	1	2	3	1	3	4	2
Para:	2	3	2	3	4	3	4
Habilitado:	Sim	Não	Sim	Sim	Sim	Sim	Sim
Registro:	7	6	8	4	5	9	11

Genes de neurônios

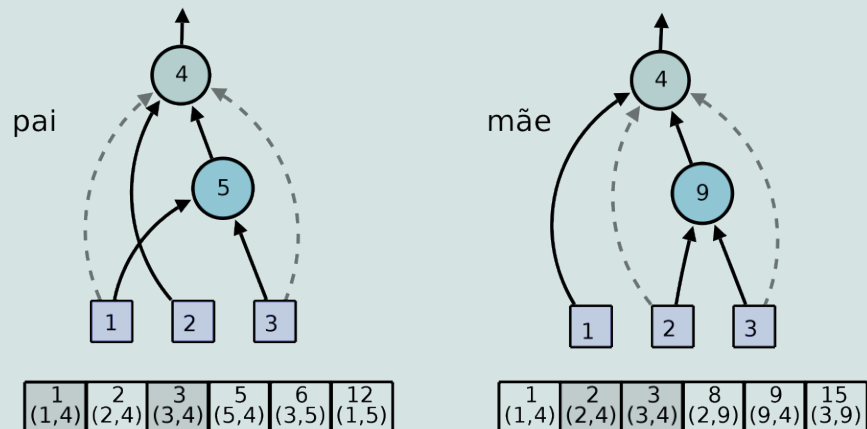
ID:	1	2	3	4
Tipo:	Entrada	Oculto	Oculto	Saída

Mutação estrutural:

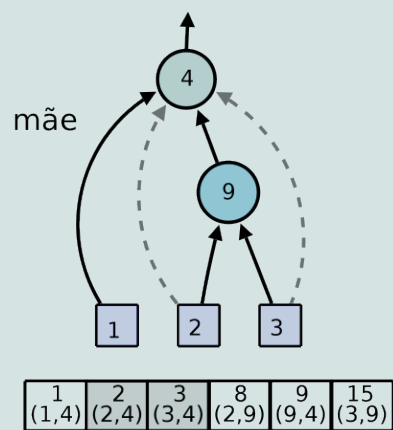
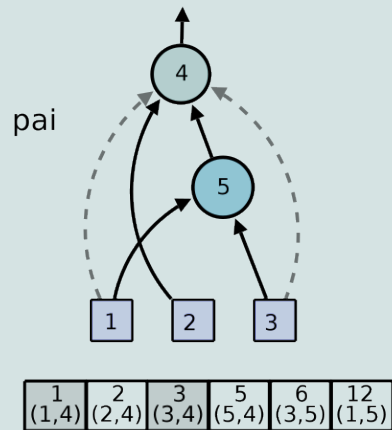


Neuroevolução: NEAT

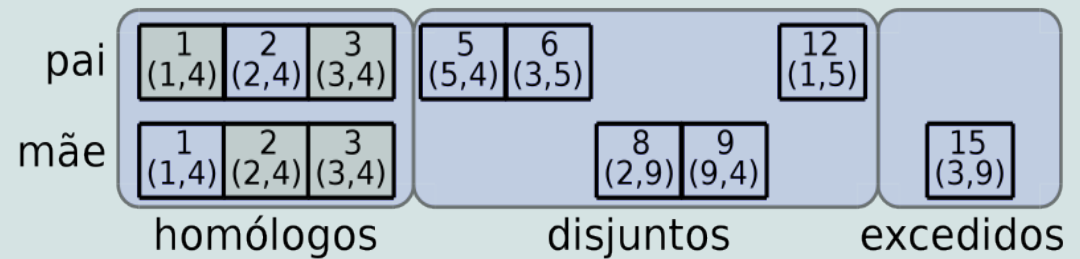
Operador de recombinação (*crossover*):



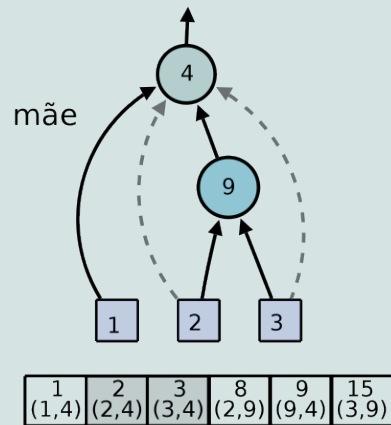
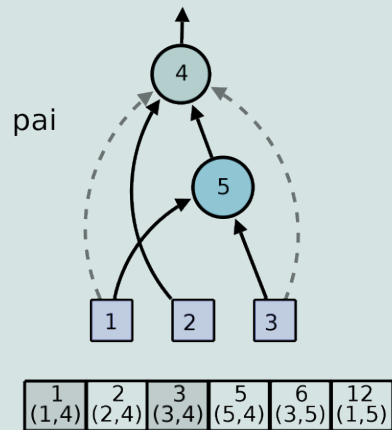
Operador de recombinação (*crossover*):



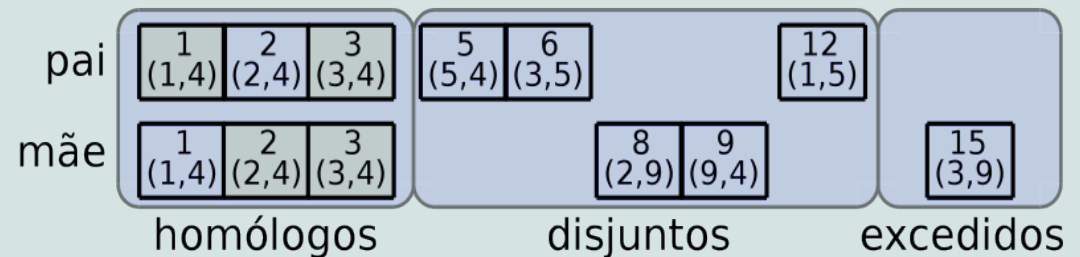
Alinhamento dos genes



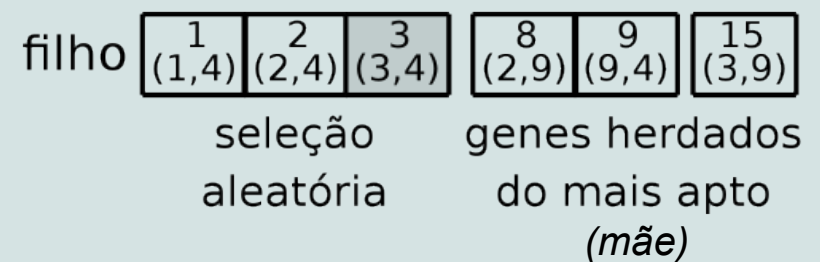
Operador de recombinação (*crossover*):



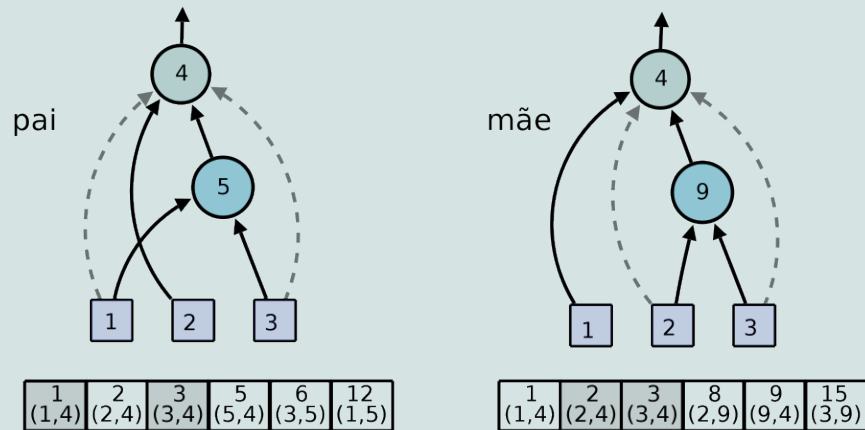
Alinhamento dos genes



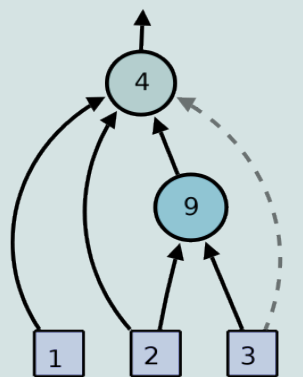
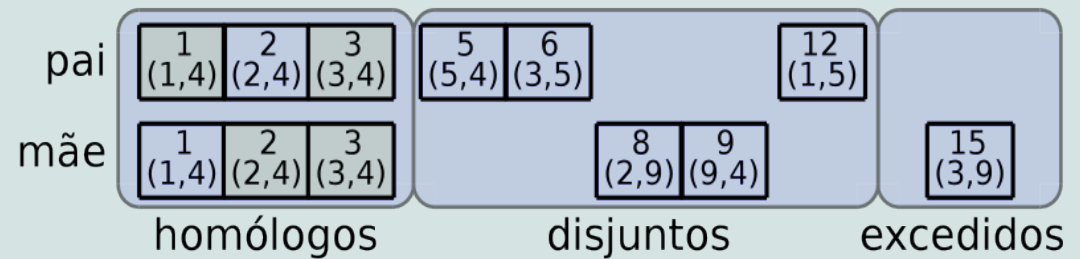
Crossover



Operador de recombinação (*crossover*):

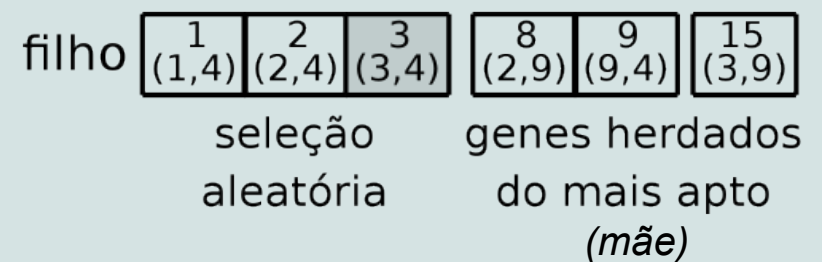


Alinhamento dos genes



fenótipo do filho

Crossover



Distância genética (*especiação*):

- Protege novas estruturas;
- Não cruza redes “incompatíveis”.

$$\delta = \frac{c_1}{N}E + \frac{c_2}{N}D + c_3\bar{W}$$

E: Genes em excesso

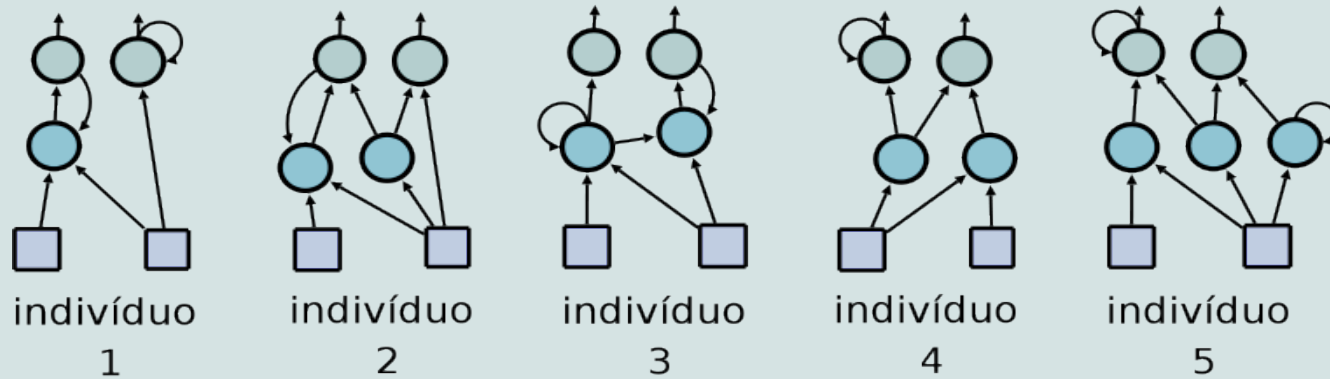
D: Genes disjuntos

W: Média dos genes homólogos

- Evita mínimos locais;

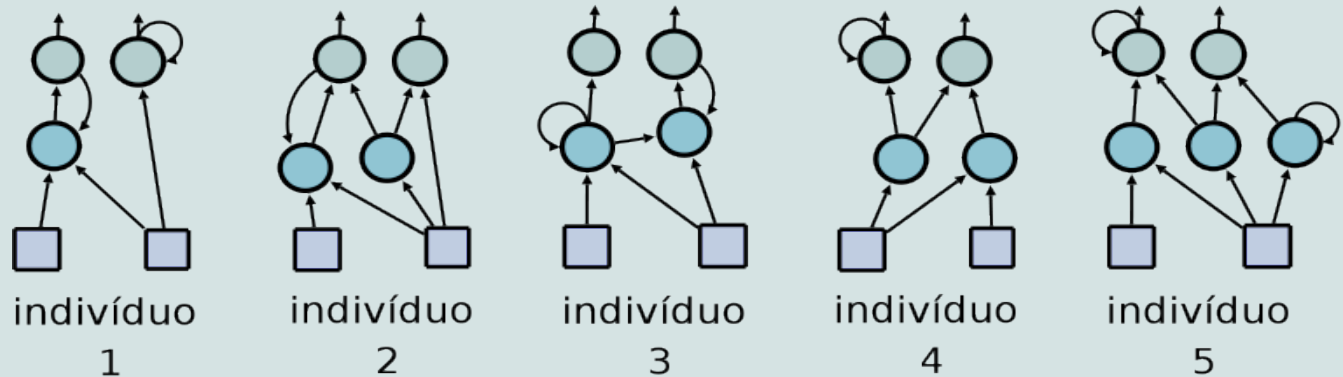
Neuroevolução: NEAT

Inicia com a menor topologia:

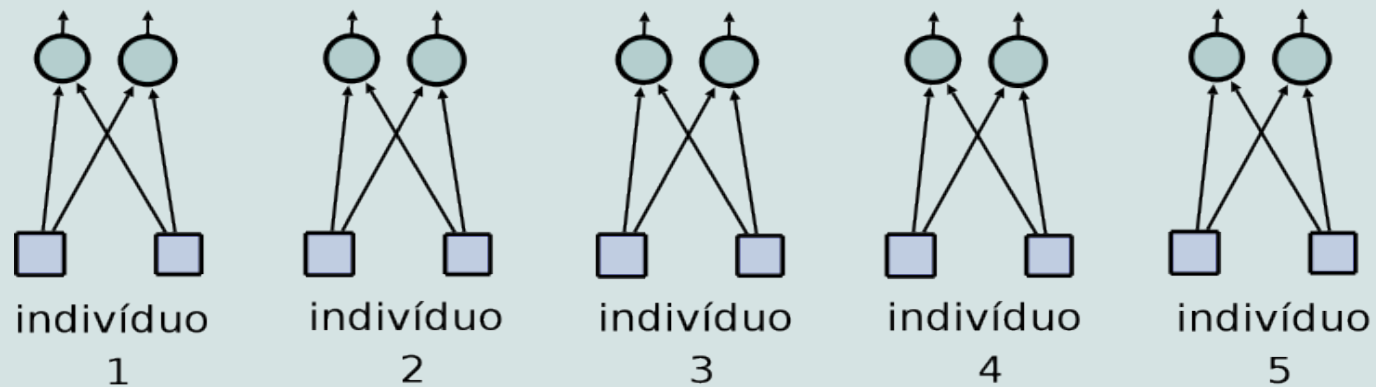


Neuroevolução: NEAT

Inicia com a menor topologia:



VS



Trabalhos Relacionados



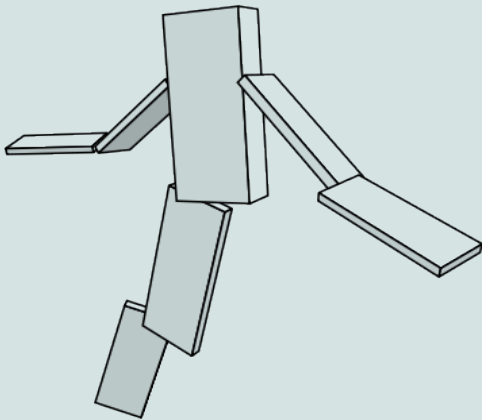
USP

Principais características:

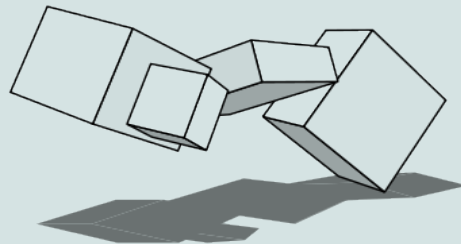
- Mapeamento genótipo-fenótipo indireto (usando recursividade)
- Desenvolvimento da morfologia;
- Evolução aberta (com poucas restrições).

Virtual Creatures (Karl Sims)

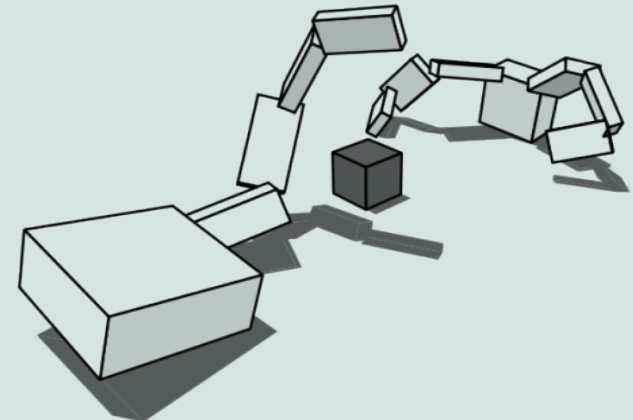
Exemplo de criaturas obtidas:



Nadar



Caminhar



Competir

Polyworld (Larry Yaeger)

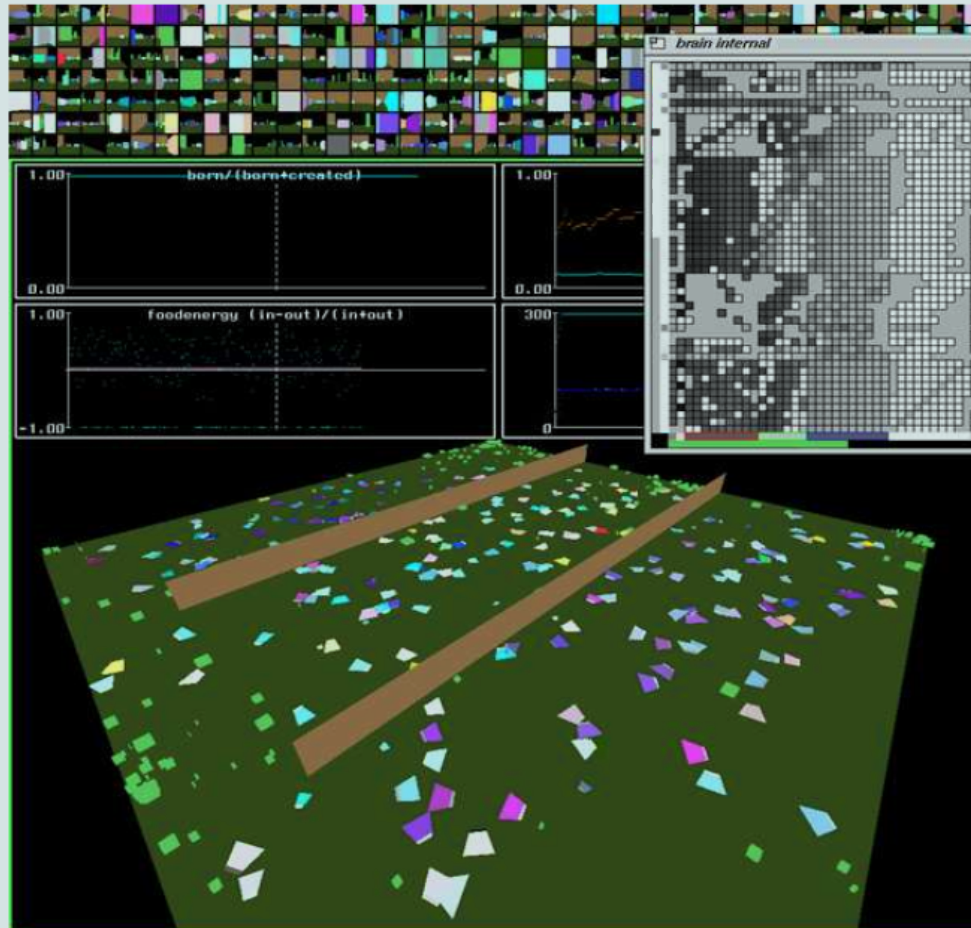
Principais características:

- Simulador para ecologia artificial;
- Evolução contínua (*não baseada em gerações fixas*);
- Evolução da topologia e aprendizado *hebbiano*;

Polyworld (Larry Yaeger)

Visão de cada
indivíduo da
população

Estatísticas
gerais da
simulação



Topologia da
rede neural de
um particular
indivíduo

Organismos
interagindo
no ambiente

Características comuns:

- Empregam métodos particulares de neuroevolução;
- Usam simuladores de ambiente.

Características comuns:

- Empregam métodos particulares de neuroevolução;
- Usam simuladores de ambiente.

Problemas identificados:

- Os métodos de neuroevolução são pouco conhecidos e detalhados (dificulta replicação e análises comparativas);
- Os simuladores compartilham muitas características que poderiam ser re-aproveitadas.

Características desejadas:

- Neuroevolução** {
- Flexibilidade para novas arquiteturas;
 - Detalhes de sua estrutura;
 - Implementações disponíveis livremente.

Características desejadas:

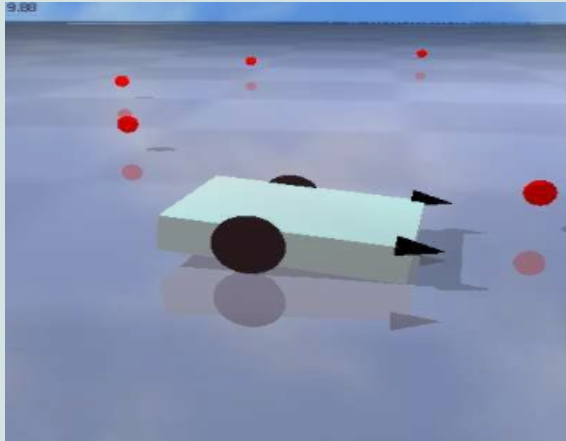
- | | | |
|----------------------|---|--|
| Neuroevolução | { | <ul style="list-style-type: none">• Flexibilidade para novas arquiteturas;• Detalhes de sua estrutura;• Implementações disponíveis livremente. |
| Simuladores | { | <ul style="list-style-type: none">• Simulação de física e visualização 3D;• Facilidade em estender (modular);• Multiplataforma. |

Desenvolvido para simulações de propósito geral em vida artificial e multi-agentes.

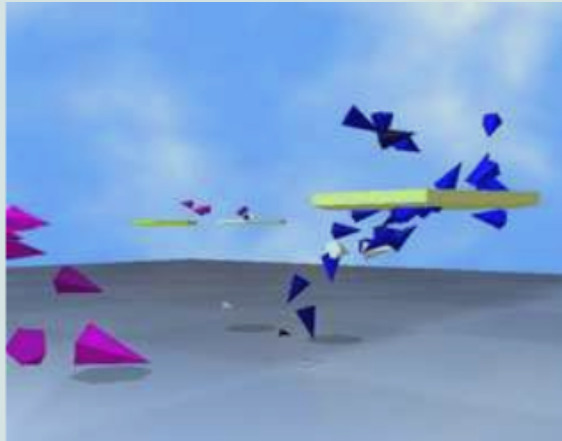


- Início em 2002 (Jon Klein);
- Simulação de física (baseado no *Open Dynamics Engine*);
- Visualização 3D (pode ser desabilitada para desempenho);
- Permite corpos articulados;
- Modular e escrito em C/C++ (facilmente extensível);
- Experimentos podem ser escritos em Python;

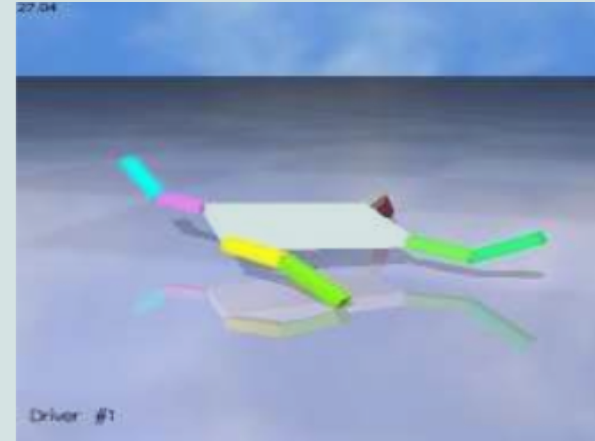
Simulador Breve



Veículo de
Braitenberg



Multi-agentes



Corpo articulado

NEAT

+

Breve

- Amplamente analisado/aplicado;
 - Dezenas de implementações;
 - Adequado para Vida Artificial e Engenharia
-
- Código aberto;
 - Desenvolvimento em atividade desde 2002;
 - Documentação completa;
 - Facilidade de integração.

Projeto NEAT-Python

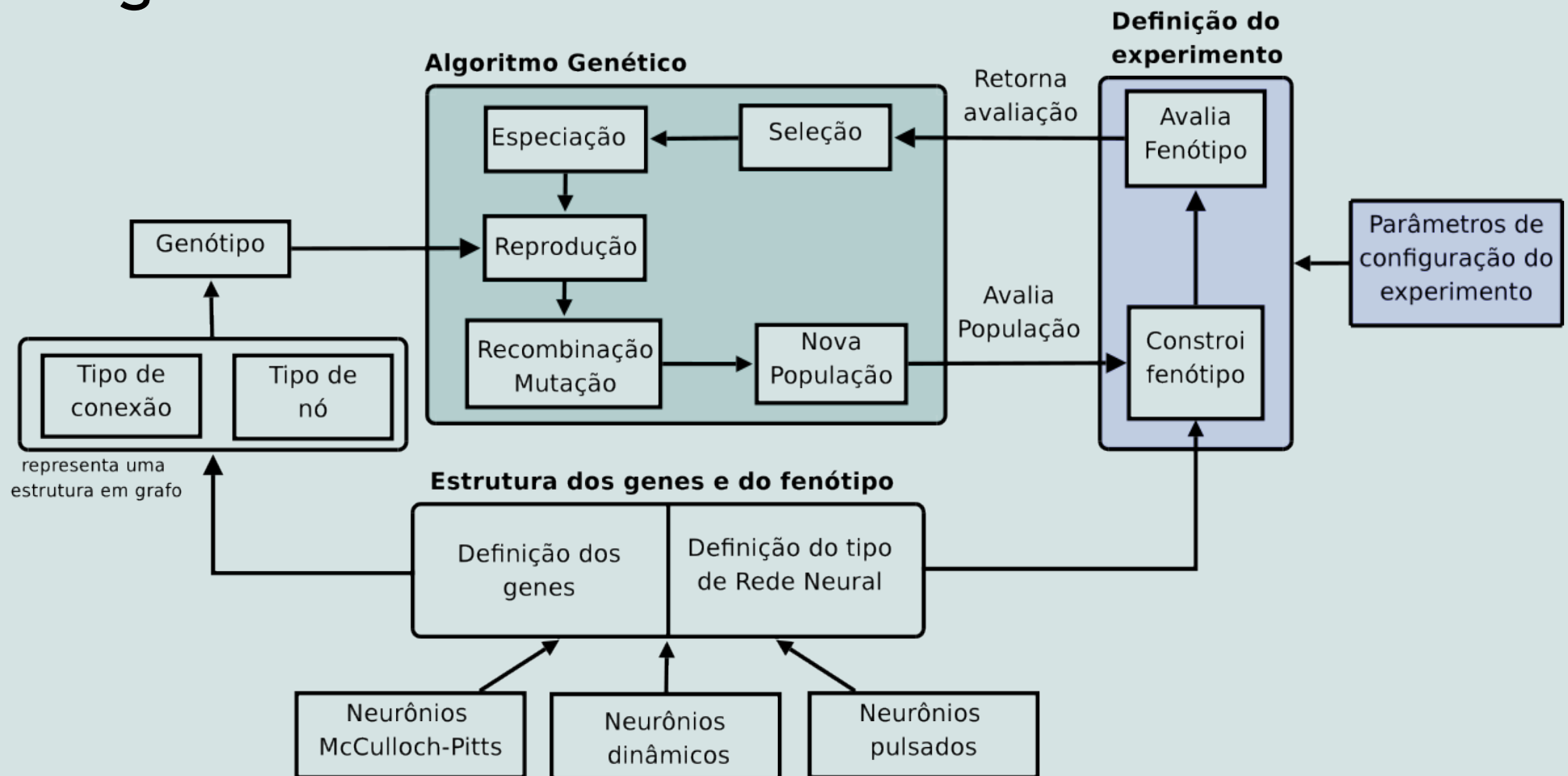


USP

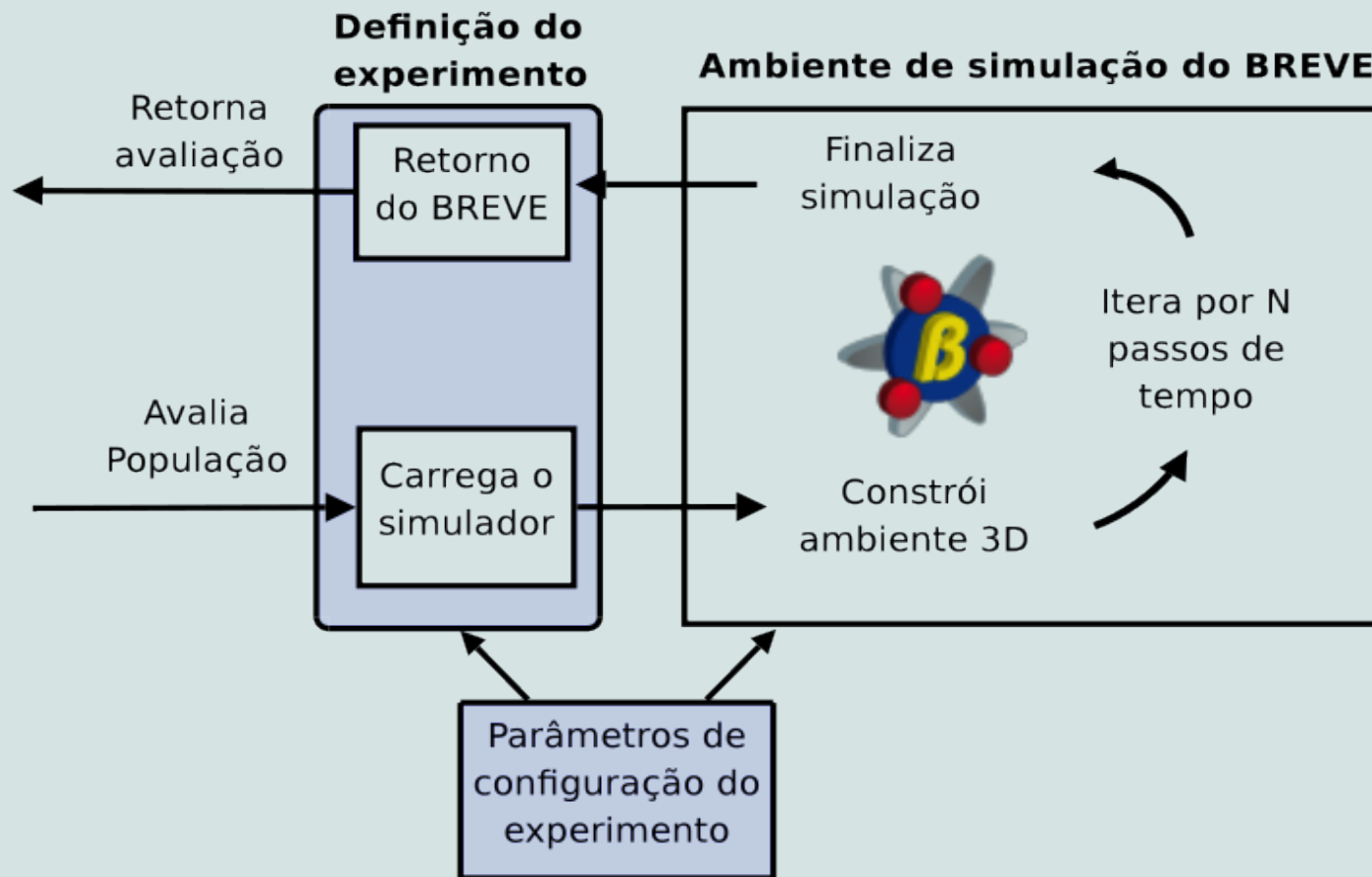
Propósito: desenvolvimento de uma biblioteca para o NEAT escrita em Python.

- Fácil integração com Breve;
- Estrutura modular (AG + RNA);
- Permite adicionar novos modelos neurais;
- Desenvolvido em colaboração com Carolina Feher (ICB-USP).

Diagrama em blocos:



Integração com o Breve:



Estende o NEAT incorporando Redes Neurais Dinâmicas do tipo CTRNNs (*Continuous-time Recurrent Neural Networks*)

$$\tau_i \frac{dy}{dt} = -y_i + \sum_{j=1}^N w_{ji} \sigma(y_j - \theta_j) + \sum_{k=1}^S S_{ki} I_k$$

Estende o NEAT incorporando Redes Neurais Dinâmicas do tipo CTRNNs (*Continuous-time Recurrent Neural Networks*)

$$\tau_i \frac{dy}{dt} = -y_i + \sum_{j=1}^N w_{ji} \sigma(y_j - \theta_j) + \sum_{k=1}^S S_{ki} I_k$$

Constante de decaimento

Ativação em $(t-1)$


Função de ativação

Entradas externas (sensores)

Em oposição ao modelo tradicional (sigmoidal), as CTRNNs apresentam:

- Maior realismo biológico;
- Dinâmica rica (aproximam qualquer sistema dinâmico);
- Meio termo entre neurônios sigmoidais e pulsados;
- Apropriadas para simulações de tempo contínuo (i.e., Vida Artificial, Sistemas de Controle e etc).

NEAT-Python

**neat-python**
A NEAT (NeuroEvolution of Augmenting Topologies) implementation in Python

Search Projects

Search the Web

Project Home

Downloads

Source

Administer

About

NEAT (NeuroEvolution? of Augmenting Topologies) is a method for evolving arbitrary neural networks developed by Kenneth O. Stanley. This project aims to implement NEAT in the Python programming language.

For further information regarding general concepts and theory, please visit: [Selected Publications](#) in Stanley's website.

Testing

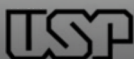
If you want to try neat-python, please check out from the subversion repository. This project is still in development and not totally bug free. However, you can start playing with the XOR example and then create your own experiment.

Code License: [GNU General Public License v3](#)

Labels: [neuralnetworks](#), [geneticalgorithms](#), [evolution](#), [Python](#), [NEAT](#)

Project owners:
[mirrorballu2](#), [cesar.gomes](#)

<http://code.google.com/p/neat-python/>



Experimentos

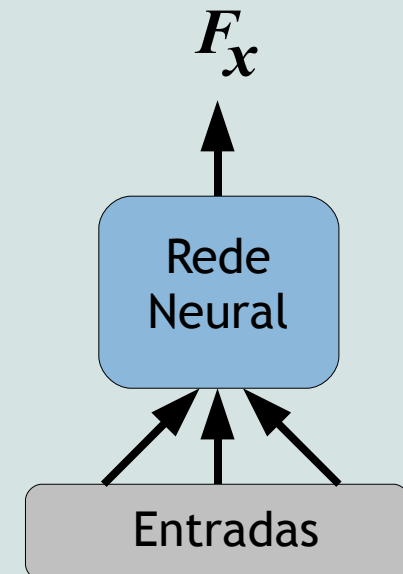
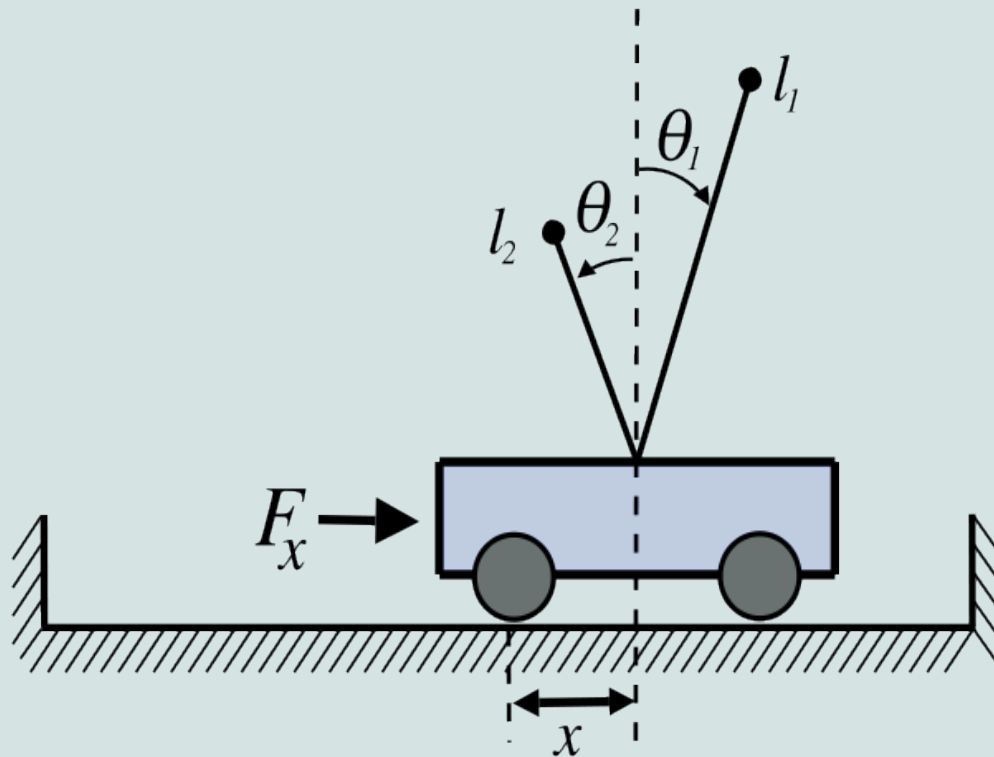


USP

Pêndulo Invertido

Objetivo:

Encontrar uma arquitetura de rede neural capaz de equilibrar, na posição vertical, um ou mais pêndulos acoplados a um carro.



Pêndulo Invertido

Modelo:

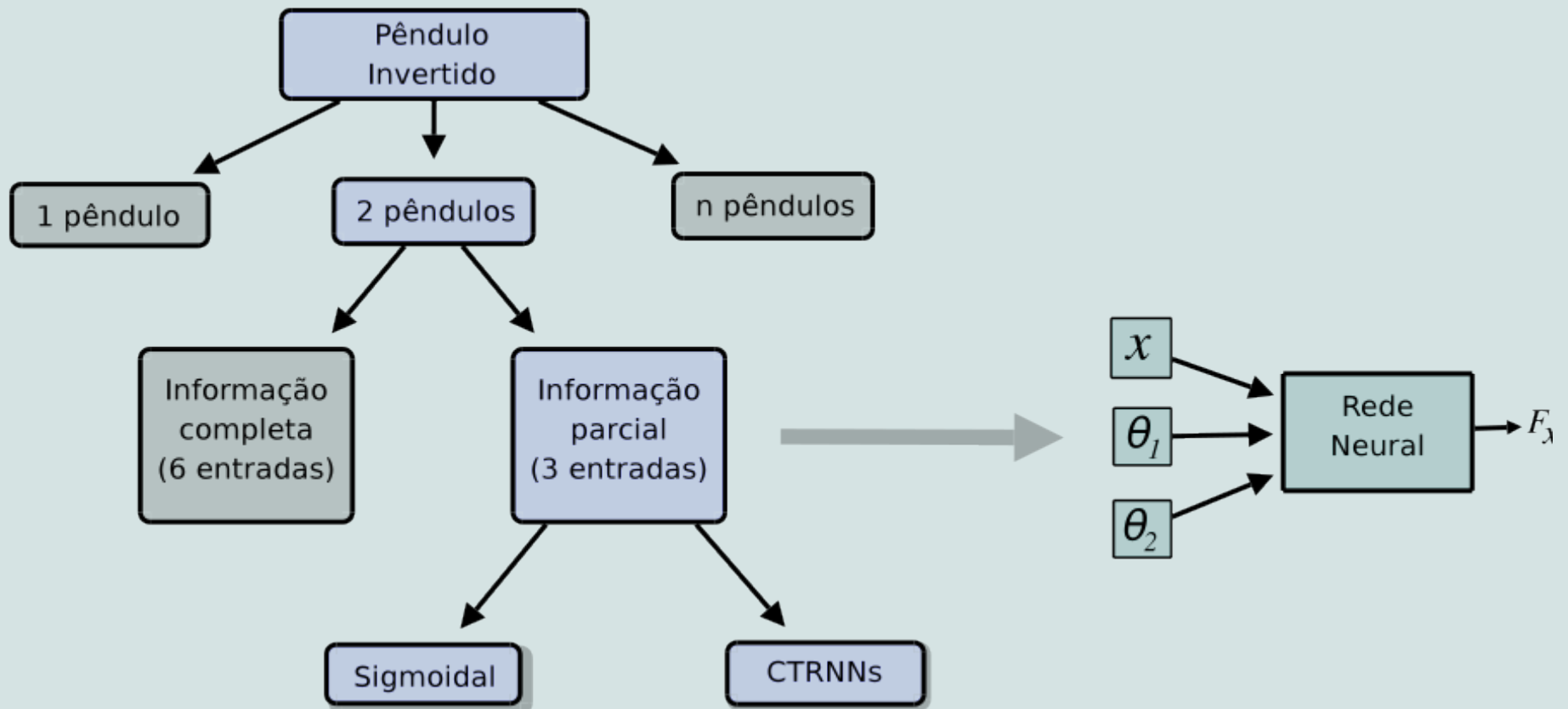
$$\begin{cases} \ddot{x} &= \frac{F - \mu_c \operatorname{sgn}(\dot{x}) + \sum_{i=1}^N \bar{F}_i}{M + \sum_{i=1}^N \bar{m}_i} \\ \ddot{\theta}_i &= -\frac{3}{4l_i} \left(\ddot{x} \cos \theta_i + g \sin \theta_i + \frac{\mu_{p_i} \dot{\theta}_i}{m_i l_i} \right) \end{cases}$$

$$\begin{cases} \bar{m}_i &= m_i \left(1 - \frac{3}{4} \cos^2 \theta_i \right) \\ \bar{F}_i &= m_i l_i \dot{\theta}_i^2 \sin \theta_i + \frac{3}{4} m_i \cos \theta_i \left(\frac{\mu_{p_i} \dot{\theta}_i}{m_i l_i} + g \sin \theta_i \right) \end{cases}$$



Pêndulo Invertido

Diversas variantes:



Pêndulo Invertido

Metodologia:

Fase I Equilibrar por 100k passos.

Fase II Ao passar pela Fase I, a mesma rede é avaliada por 1k passos com diferentes condições iniciais:

$$(x, \dot{x}, \theta_1, \dot{\theta}_1) \longrightarrow [0.05, 0.25, 0.5, 0.75, 0.95]$$

Totalizando $5^4 = 625$ estados iniciais. A solução é válida se a rede generalizar para pelo menos 200 estados e recebe um “coeficiente de generalização”.



Experimento I:

Comparar o desempenho do NEAT-Python com a versão oficial de Kenneth Stanley. **Ambos devem desempenhar de forma similar.**

Pêndulo Invertido: Resultados

Experimento I:

Comparar o desempenho do NEAT-Python com a versão oficial de Kenneth Stanley. **Ambos devem desempenhar de forma similar.**

Método	Avaliações	$\Delta\bar{x}$	CG	$\Delta\bar{x}$
NEAT C++	23777	718.80	257	1.95
NEAT Python	24091	519.32	272	1.98

Média para 544 rodadas.

Não há significância estatística: **ambos resolvem de maneira similar.**

Experimento II:

- Comparar o desempenho entre redes de neurônios sigmoidais e dinâmicos (CTRNNs) ;
- A natureza dinâmica das CTRNNs deve ser mais adequada na resolução de problemas em domínio contínuo.

Pêndulo Invertido: Resultados

Experimento II:

Método	Avaliações	$\Delta \bar{x}$	CG	$\Delta \bar{x}$
CE	840000	—	300	—
CNE	87623	—	—	—
ESP	26342	—	—	—
AGE	25065	4360	317	—

Pêndulo Invertido: Resultados

Experimento II:

Método	Avaliações	$\Delta\bar{x}$	CG	$\Delta\bar{x}$
CE	840000	—	300	—
CNE	87623	—	—	—
ESP	26342	—	—	—
AGE	25065	4360	317	—
NEAT	23777	718	257	1.95

Pêndulo Invertido: Resultados

Experimento II:

Método	Avaliações	$\Delta\bar{x}$	CG	$\Delta\bar{x}$
CE	840000	—	300	—
CNE	87623	—	—	—
ESP	26342	—	—	—
AGE	25065	4360	317	—
NEAT	23777	718	257	1.95
NEAT-CTRNN	4048	105	274	2.11

Pêndulo Invertido: Resultados

Experimento II:

Método	Avaliações	$\Delta\bar{x}$	CG	$\Delta\bar{x}$
CE	840000	—	300	—
CNE	87623	—	—	—
ESP	26342	—	—	—
AGE	25065	4360	317	—
NEAT	23777	718	257	1.95
NEAT-CTRNN	4048	105	274	2.11

A vantagem obtida não é mérito do NEAT-Python simplesmente, mas sim da união do NEAT com CTRNNs (*que até então não havia sido experimentado*).

Busca de Alimento



USP

- Problema fundamental em Vida Artificial;
- Exige um ambiente virtual com o qual o indivíduo possa interagir;
- Permite explorar diversas configurações e observar a emergência de comportamentos;

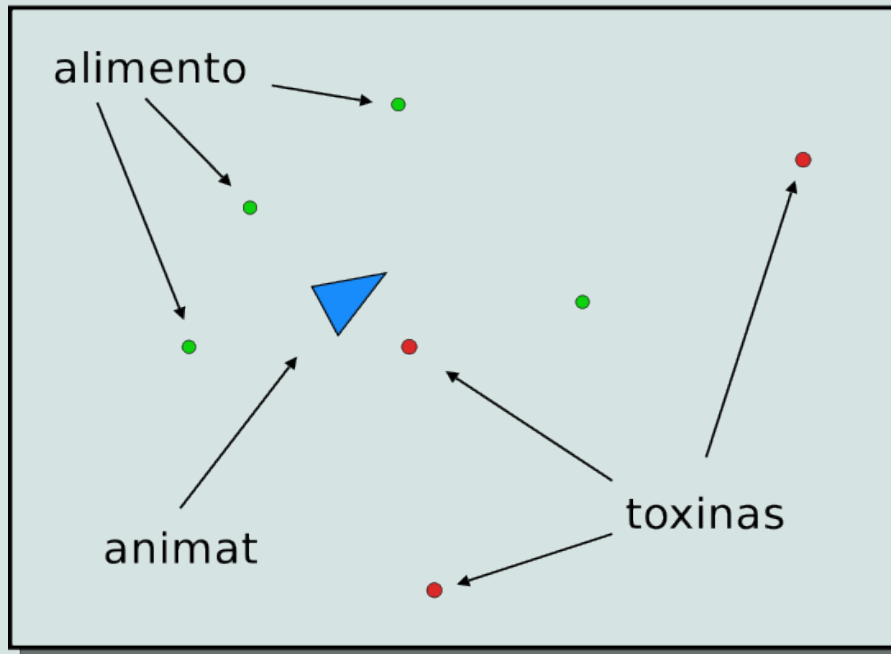
Metodologia:

- Definição do ambiente (estrutura);
- Morfologia do indivíduo (sensores e atuadores);
- Objetivo e critério de seleção.

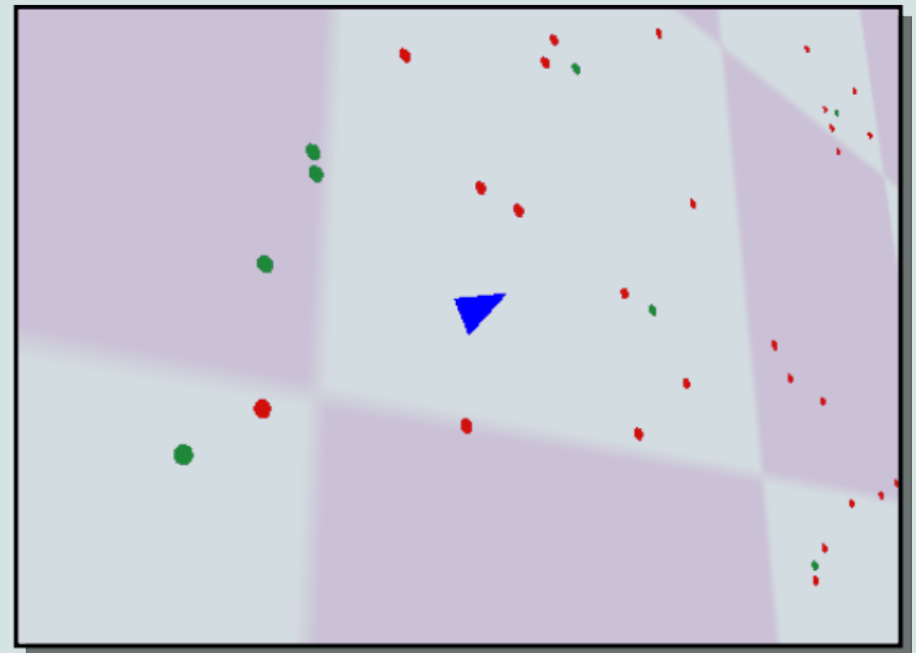
Busca de Alimento

Ambiente:

Representação 2D

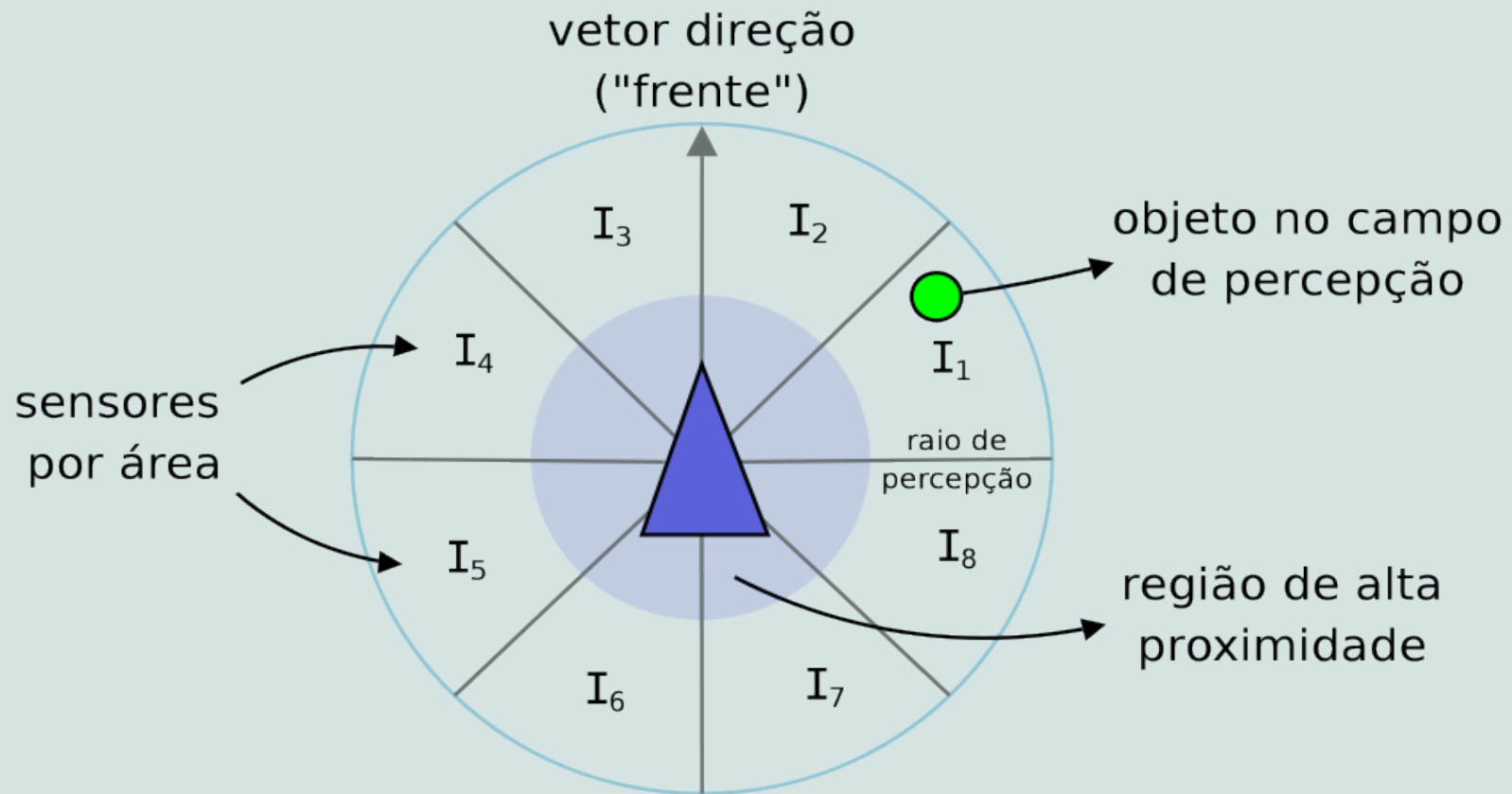


Visualização no Breve



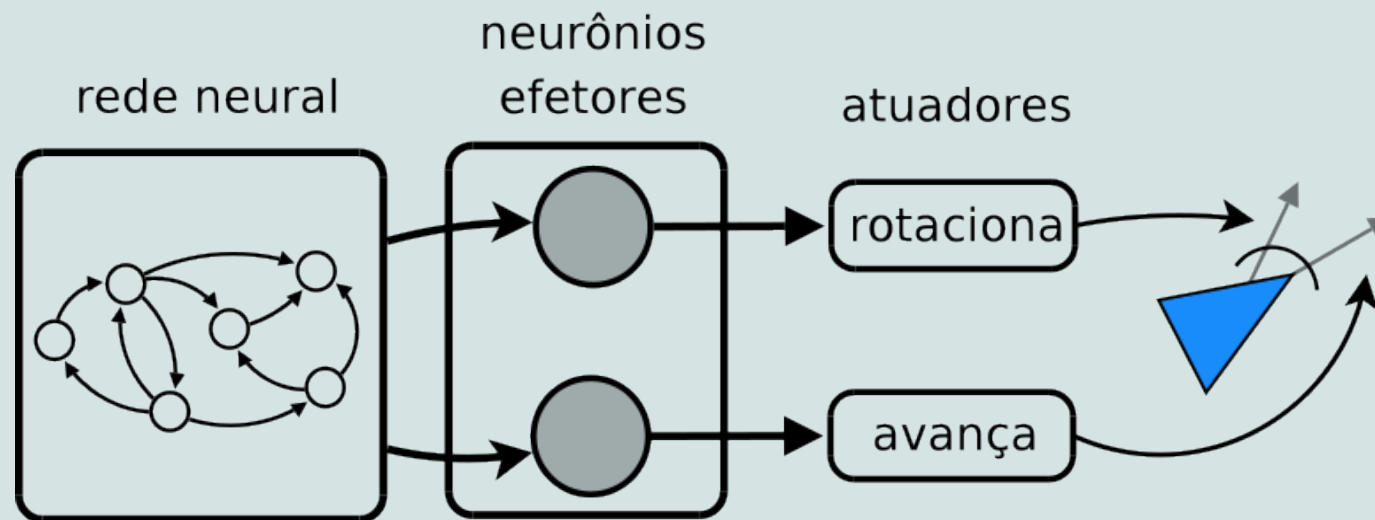
Busca de Alimento

Percepção do ambiente através de sensores:



Busca de Alimento

Interação com o ambiente através dos atuadores:



Busca de Alimento

Objetivo:

Desenvolver a habilidade de maximizar o consumo de alimentos, evitando toxinas.

Seleção:

Cada alimento vale 1 (um) ponto de “energia” e cada toxina equivale a perder 0.5 ponto.

Ao fim de cada rodada (5000 passos de tempo) cada *animat* recebe seu valor adaptativo em função do consumo de alimentos e toxinas:

$$F = A - \frac{T}{2}$$



Resultados:

Desempenho do melhor indivíduo para a média de 100 rodadas.

100 alimentos, 100 toxinas (distribuição normal)

Cenário I

28.28 ± 1.03 (alimento)

5.71 ± 0.32 (toxinas)

28%
(eficiência)

Resultados:

Desempenho do melhor indivíduo para a média de 100 rodadas.

Cenário I

100 alimentos, 100 toxinas (distribuição normal)

28.28 ± 1.03 (alimento)

5.71 ± 0.32 (toxinas)

28%
(eficiência)

Cenário II

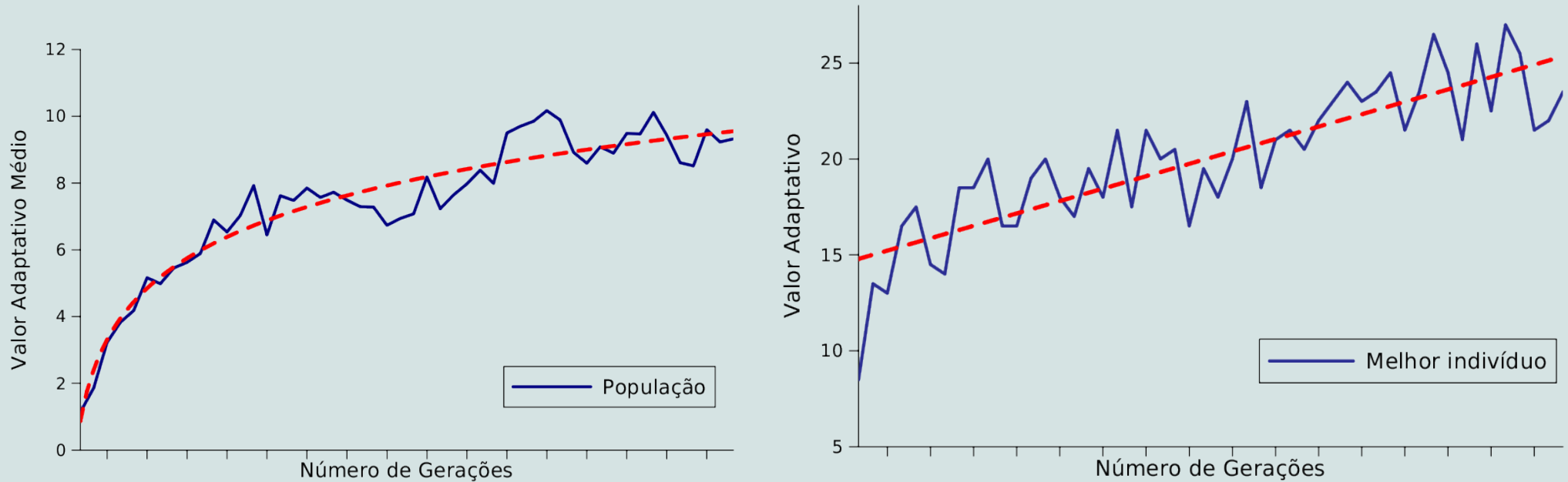
50 alimentos, 100 toxinas (distribuição normal)

16.64 ± 0.59 (alimento)

4.70 ± 0.30 (toxinas)

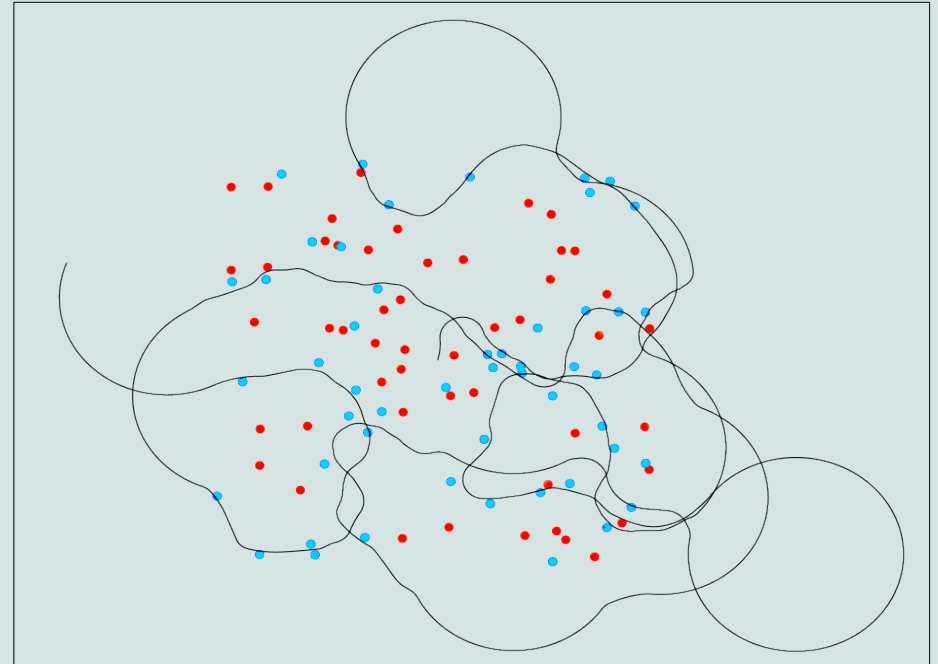
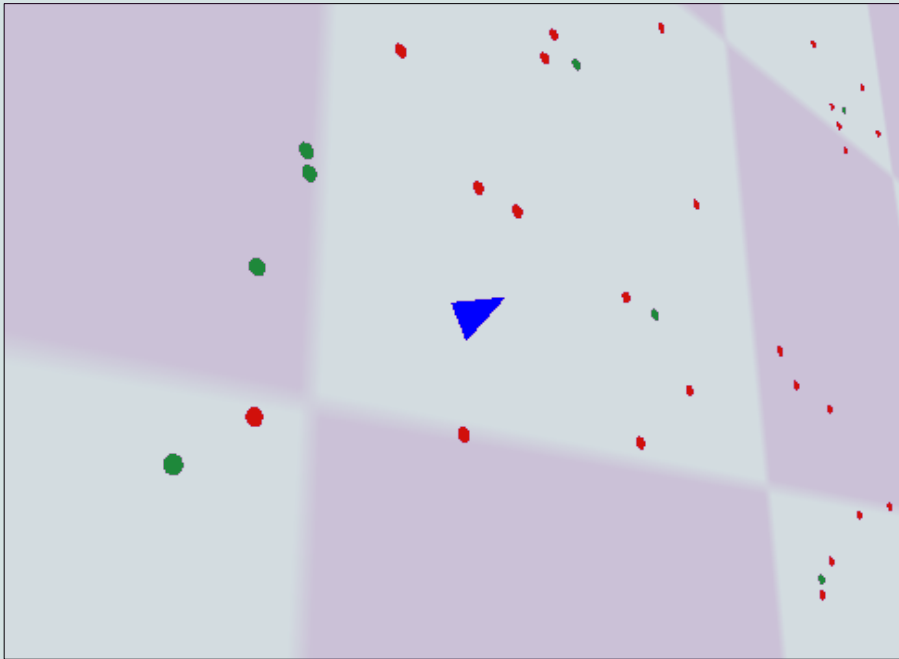
27%
(eficiência)

Resultados:



Após 50 gerações

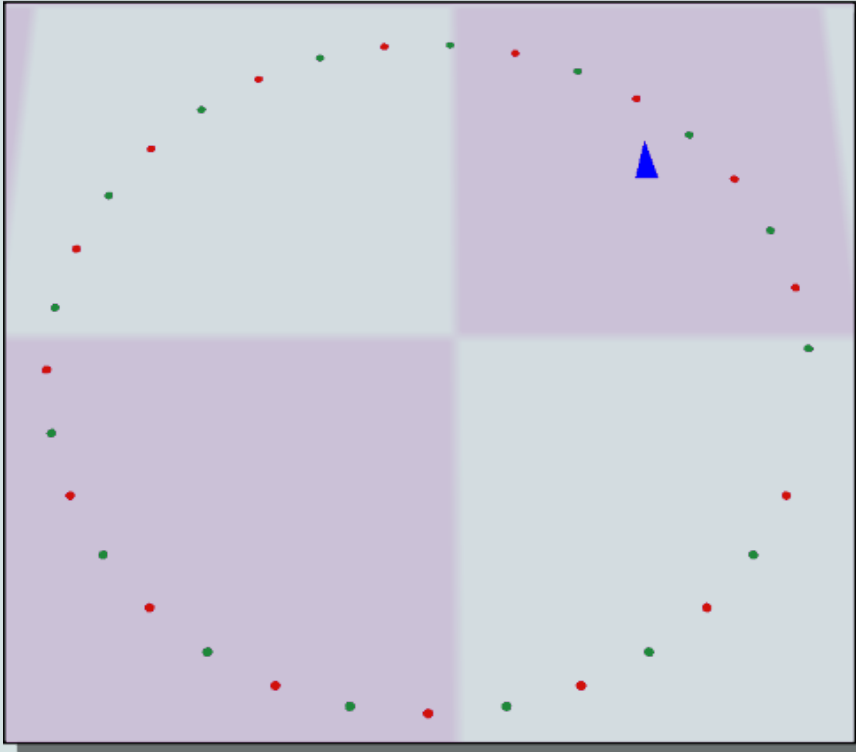
Exemplo de caminho percorrido



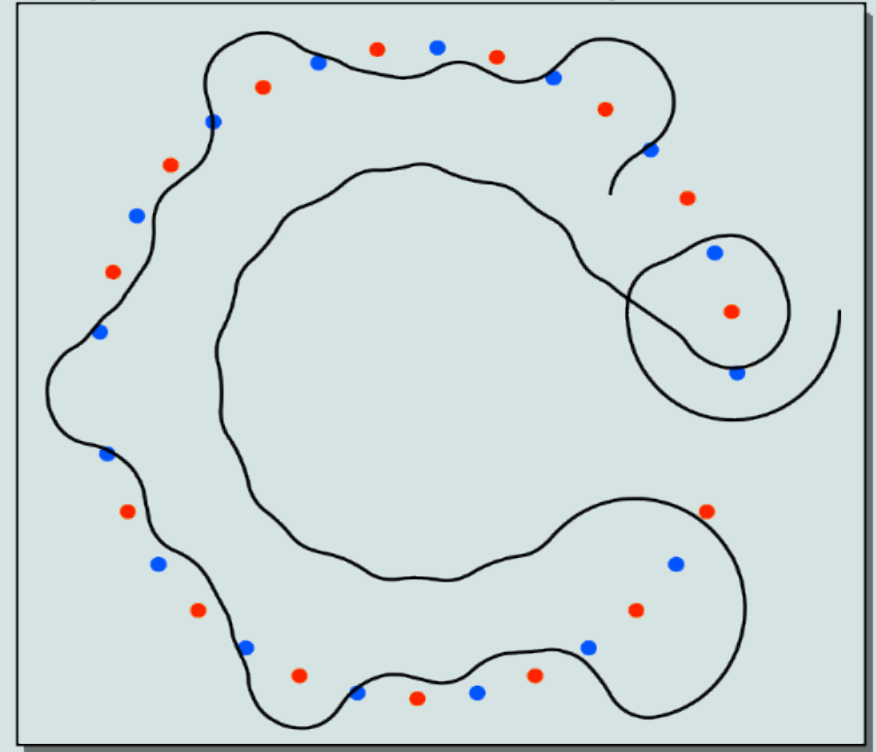
Distribuição normal

Exemplo de caminho percorrido

Visualização no Breve



Representação 2D do percurso



Ambiente estruturado (em círculo)

Observações:

- Bom grau de generalização para o tipo de ambiente;
- Mantem a eficiência em condições de maior dificuldade.

Principal problema:

- Comportamento restrito ao tipo de seleção, morfologia e configuração do ambiente.

Considerações Finais



USP

Principais dificuldades:

- Grande número de parâmetros na configuração do NEAT;
- Instabilidade numérica na solução das CTRNNs;
- Pouca informação a respeito do espaço de busca.

Principais dificuldades:

- Grande número de parâmetros na configuração do NEAT;
- Instabilidade numérica na solução das CTRNNs;
- Pouca informação a respeito do espaço de busca.

Contribuições:

- Análise do NEAT utilizando CTRNNs;
- Implementação em Python, modular e de fácil uso, disponibilizada como biblioteca de código aberto.

Trabalhos futuros:

- Codificação genética indireta (em oposição à direta);
- Redes neurais pulsadas (maior realismo biológico).

Considerações Finais

Trabalhos futuros:

- Codificação genética indireta (em oposição à direta);
- Redes neurais pulsadas (maior realismo biológico).

Publicações:

- MIGUEL, C. G., NETTO, M. L. (2008) **Using a General Purpose Virtual Environment for Artificial Life Simulations.** In: *X Symposium on Virtual Reality*, João Pessoa, PB. (Resumo estendido)
- MIGUEL, C. G., SILVA, C. F., NETTO, M. L. (2008) **Structural and Parametric Evolution of Continuous-time Recurrent Neural Networks.** In: *10th Brazilian Symposium on Neural Networks*, Salvador, BA: IEEE. (Artigo completo).



Disciplinas cursadas no período (2006-2007):

- Fundamentos de Ciências Cognitivas
- Modelos Computacionais em Ciências Cognitivas
- Vida Artificial em Ambientes Virtuais
- Princípios de Neurocomputação
- Jogos Eletrônicos Interativos
- Visualização Científica
- Robôs Móveis Autônomos