

# Programação Genética Distribuída: uma extensão do algoritmo da programação genética para suportar a evolução do sistema de controle em uma população de robôs móveis

**Anderson Luiz Fernandes Perez**

Programa de Pós-Graduação em Engenharia Elétrica  
Universidade Federal de Santa Catarina - UFSC

**Guilherme Bittencourt**

Departamento de Automação e Sistemas  
Universidade Federal de Santa Catarina - UFSC

**Mauro Roisenberg**

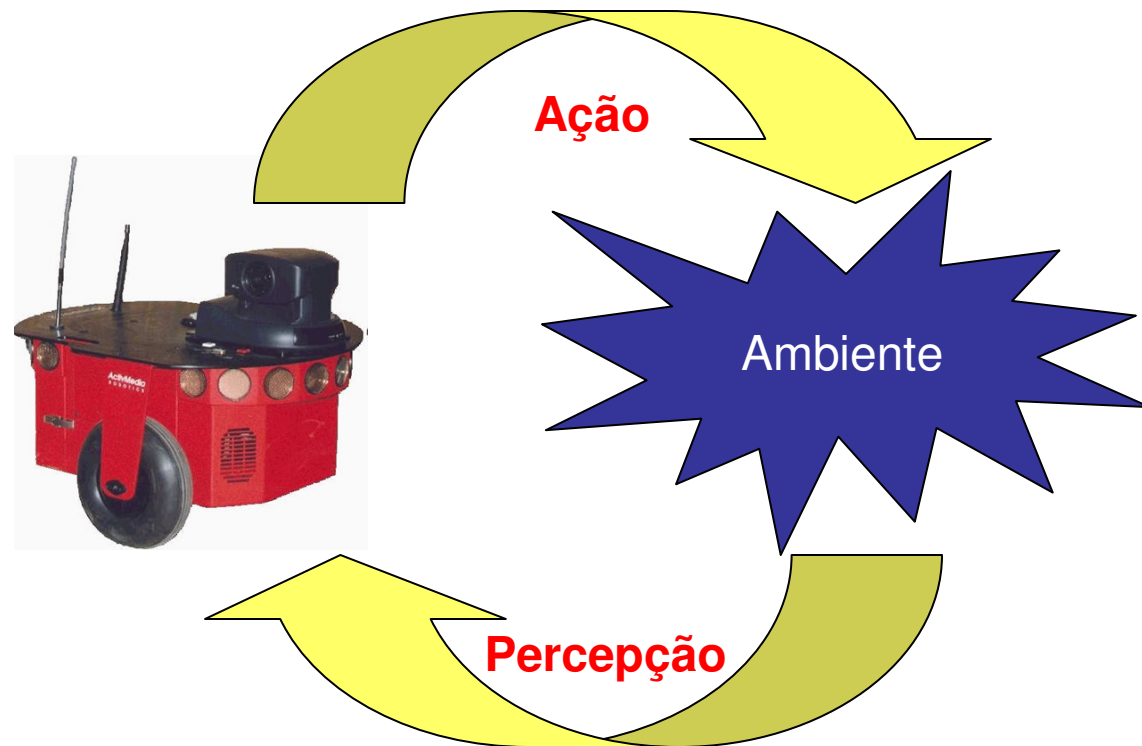
Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina - UFSC

# Programa

- Introdução
- Computação Evolutiva
- Robótica Evolutiva
- Programação Genética Distribuída
- Resultados

# Introdução

- Controle de Robôs Móveis



# Introdução

- Controle de Robôs Móveis
  - Sistema de Controle (ideal):
    1. **Recebe** informações dos sensores e determina alguma ação a ser executada.
    2. **Planeja** as ações para executar uma determinada tarefa.
    3. **Controla** a seqüência de comandos para executar uma tarefa.
    4. **Modifica** as ações do robô em resposta a mudanças no ambiente.

# Introdução

- Controle de Robôs Móveis
  - Como programar um robô móvel para explorar um ambiente ruidoso, desconhecido ou não estruturado?

# Computação Evolutiva

# Computação Evolutiva

- A **computação evolutiva** é um paradigma da **computação bioinspirada**, que:
  - investiga como computadores podem ser utilizados para modelar a natureza, e
  - como soluções encontradas pela natureza podem originar novos paradigmas de computação.

# Computação Evolutiva

- Em computação evolutiva existem diferentes tipos de algoritmos evolutivos:
  - Algoritmos Genéticos
  - **Programação Genética**
  - Programação Evolutiva
  - Estratégias Evolutivas
  - ...



# Computação Evolutiva

- Características comuns de um algoritmo evolutivo:
  - utilizam populações de indivíduos;
  - introduzem variações genéticas na população usando um ou mais operadores genéticos;
  - selecionam, de acordo com a aptidão, os indivíduos que depois se reproduzem para criar uma nova geração.

# Computação Evolutiva

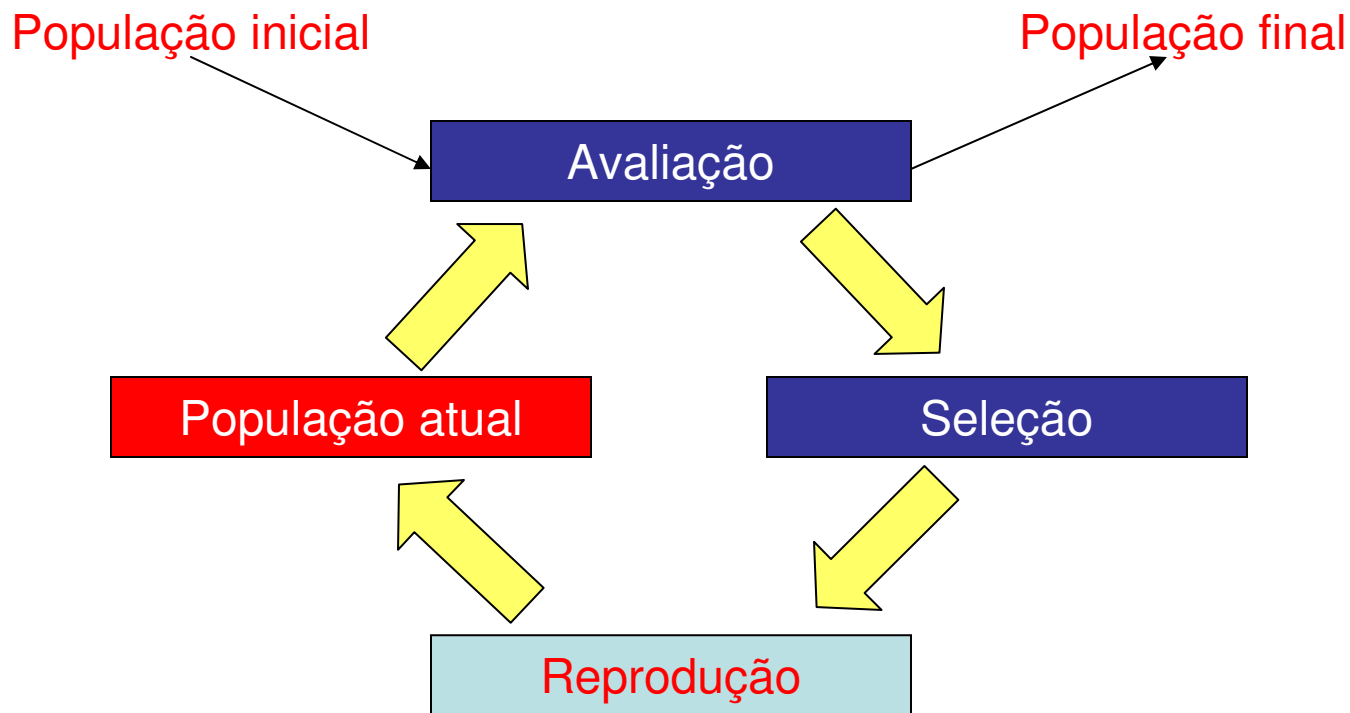
- Aspectos fundamentais para o desempenho de uma abordagem evolutiva:
  - codificação dos indivíduos numa representação genética, em que cada indivíduo representa um candidato à solução de um dado problema.
  - uma função de avaliação que associa a cada indivíduo um valor de adaptação (fitness) que determina o quão adaptado o indivíduo está.

# Computação Evolutiva

- Algoritmo evolutivo
  - *Cria aleatoriamente uma população inicial;*
  - *Executa interativamente os seguintes passos até que algum critério de parada seja satisfeito:*
    - *Avalia cada indivíduo da população através de uma função de avaliação que expressa a sua aptidão;*
    - *Seleciona os melhores indivíduos com base no valor de aptidão;*
    - *Aplica aos indivíduos selecionados operadores genéticos;*
  - *Retorna com o melhor indivíduo encontrado.*

# Computação Evolutiva

- Ciclo da CE:

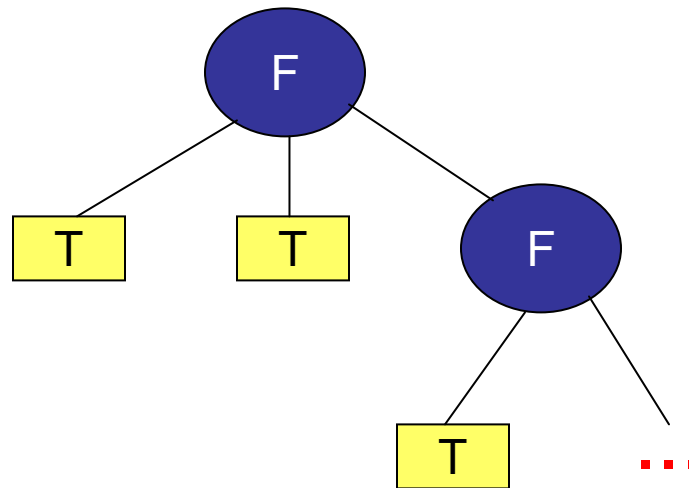


# Computação Evolutiva

- Programação Genética
  - Técnica para geração automática de programas de computador.
  - Criada por John Koza em 1992.
  - Objetivo:
    - **Ensinar computadores a se programar**, isto é, a partir de especificações de comportamento, o computador deve ser capaz de evoluir um programa que as satisfaça.

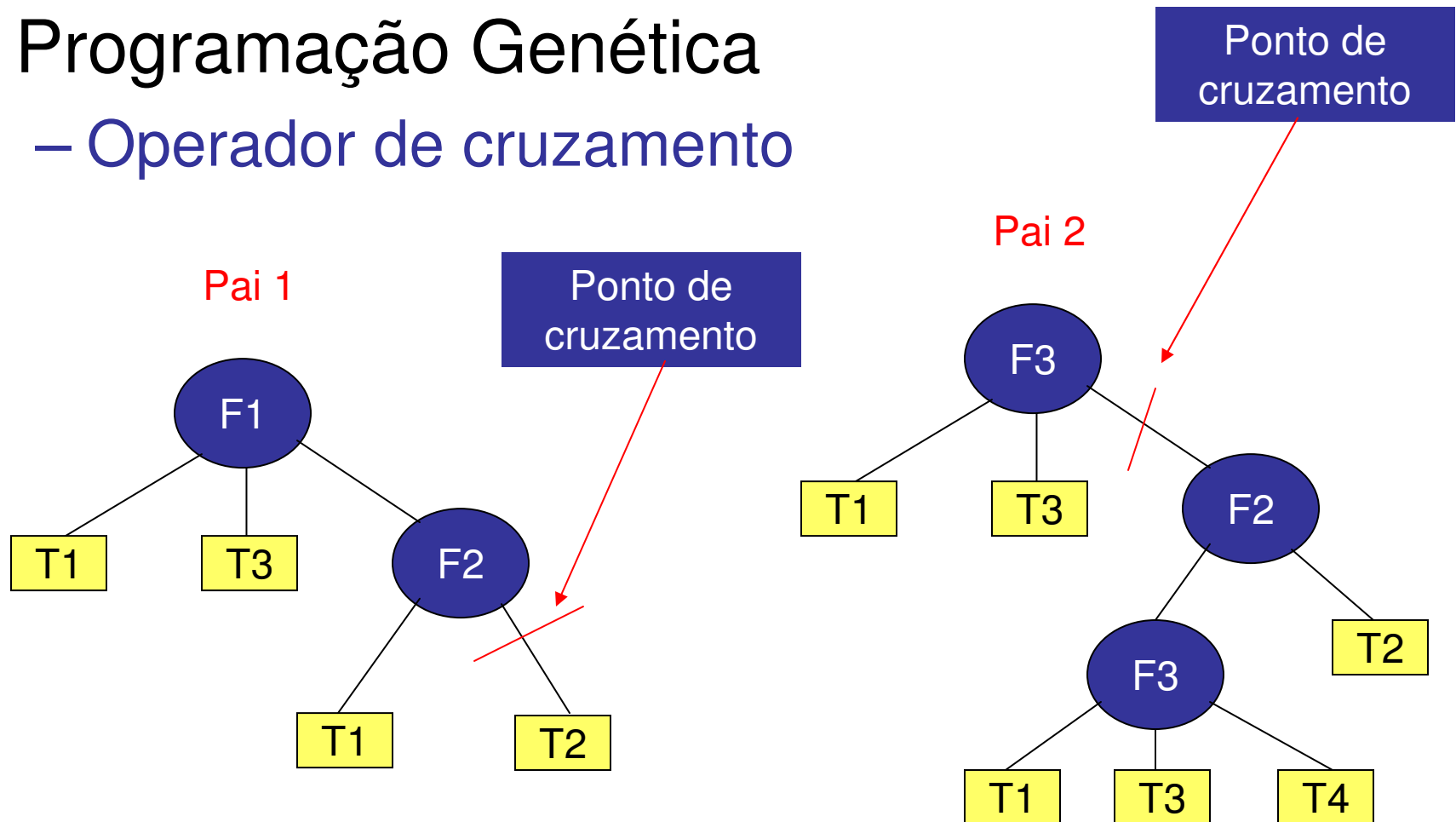
# Computação Evolutiva

- Programação Genética
  - Representação genotípica
    - Estrutura em árvore
      - Conjunto de funções {F}
      - Conjunto de terminais {T}



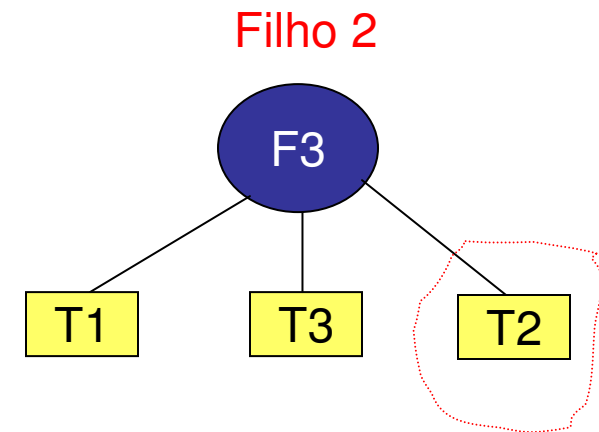
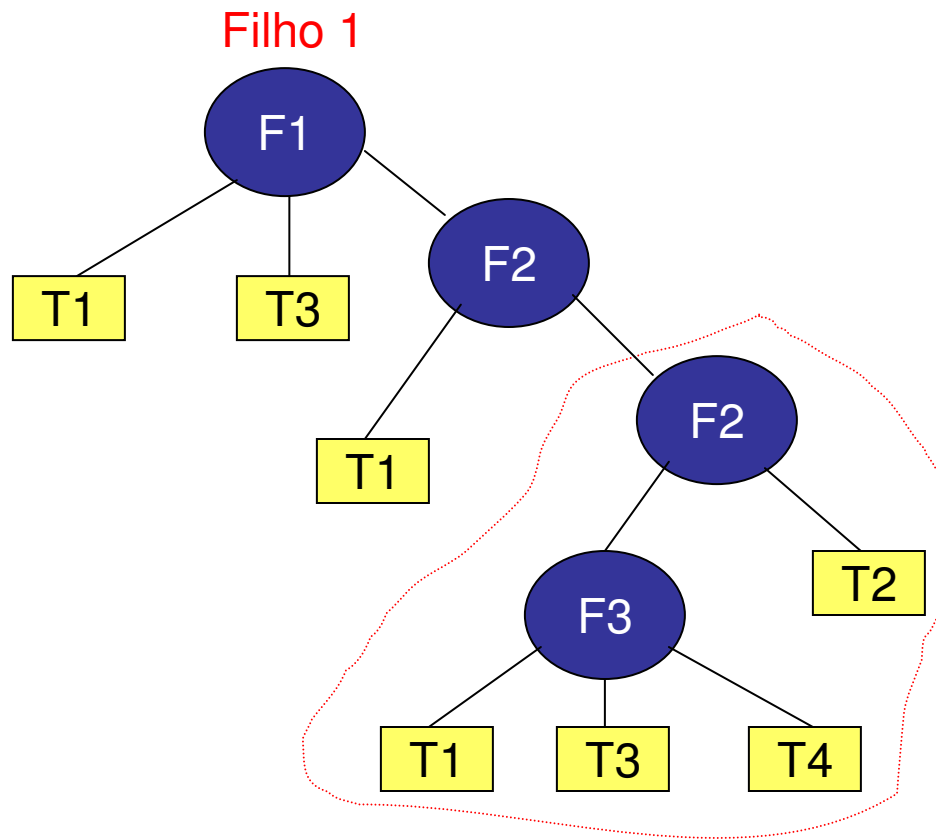
# Computação Evolutiva

- Programação Genética
  - Operador de cruzamento



# Computação Evolutiva

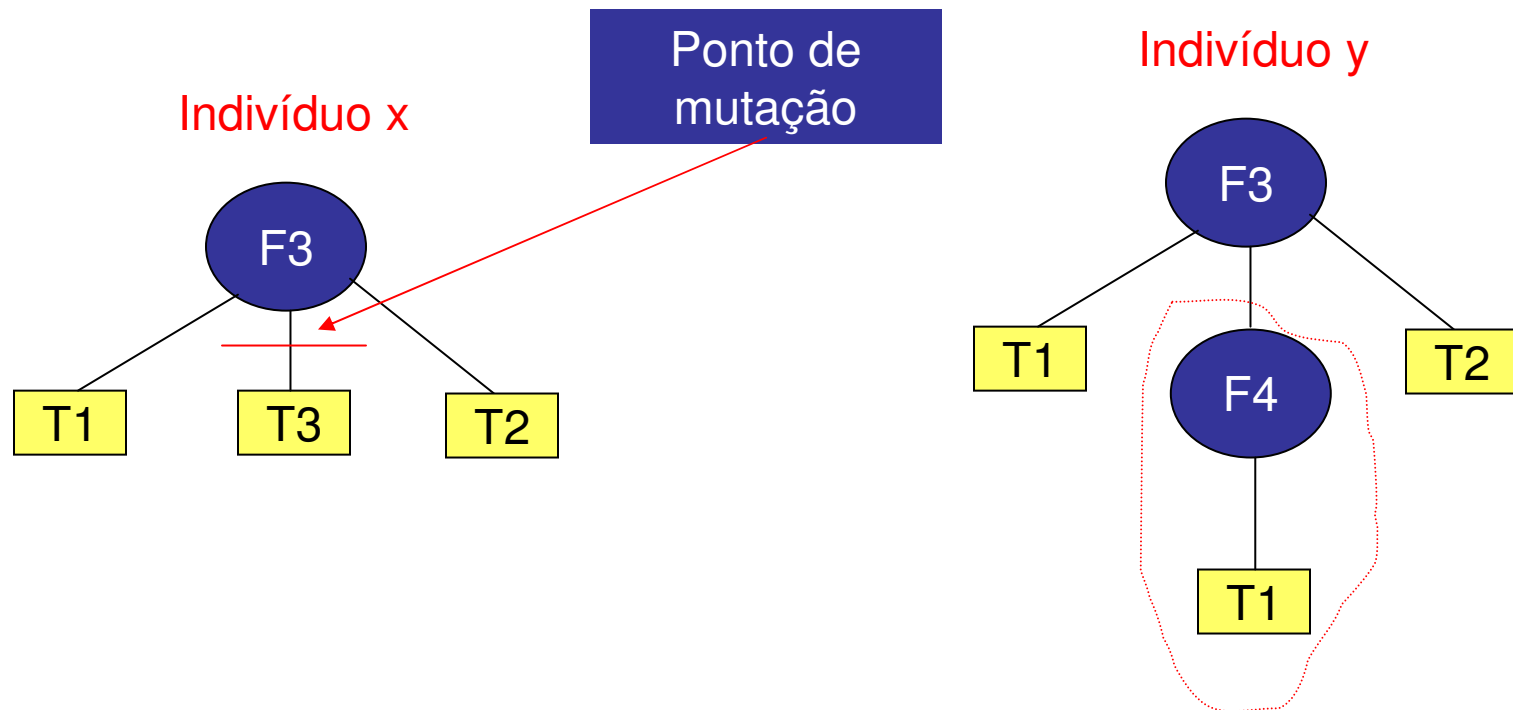
- Programação Genética
  - Operador de cruzamento



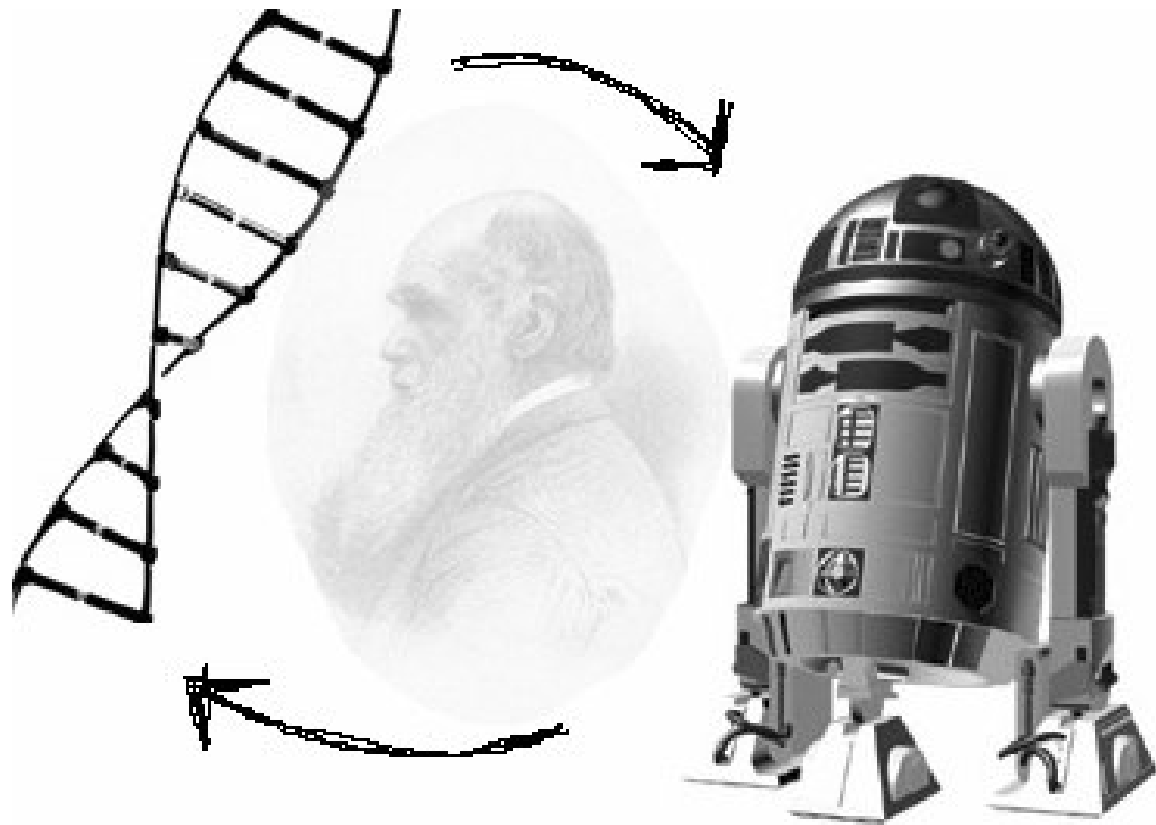


# Computação Evolutiva

- Programação Genética
  - Operador de mutação



# Robótica Evolutiva

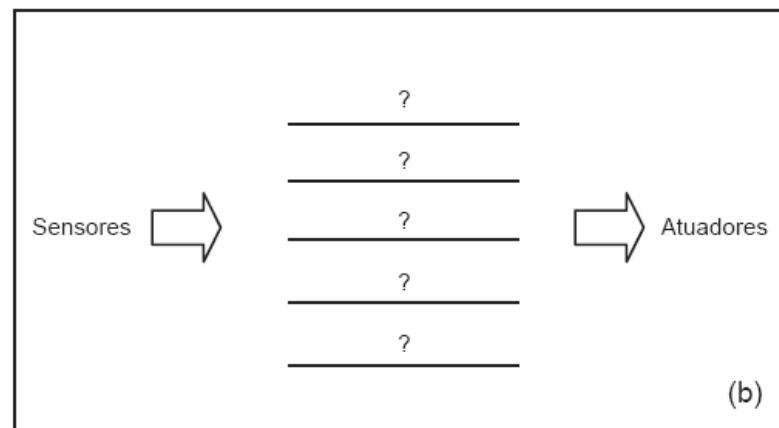
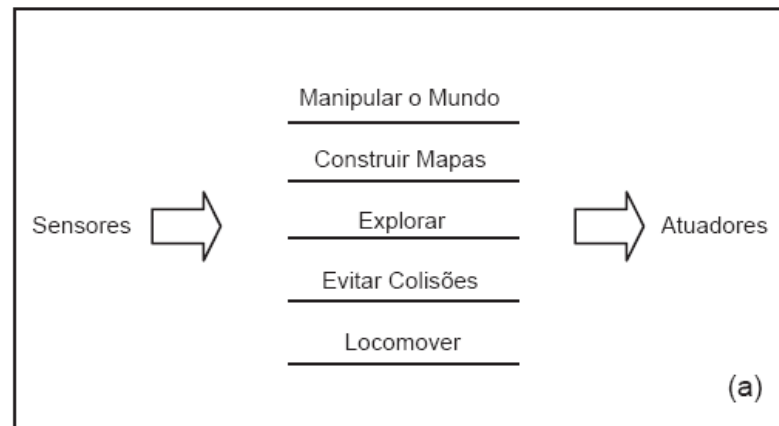


# Robótica Evolutiva

- Surgiu da utilização de técnicas de computação evolutiva para **synthesize automatically controllers** para robôs.
- Na RE o **comportamento global** é constantemente **avaliado** e **modificado** por um **processo de avaliação automático**.
- A RE pode ser descrita como um **processo** de **evolução e adaptação contínuas** de um robô situado em um determinado ambiente.

# Robótica Evolutiva

- Comparação RE versus Robótica convencional

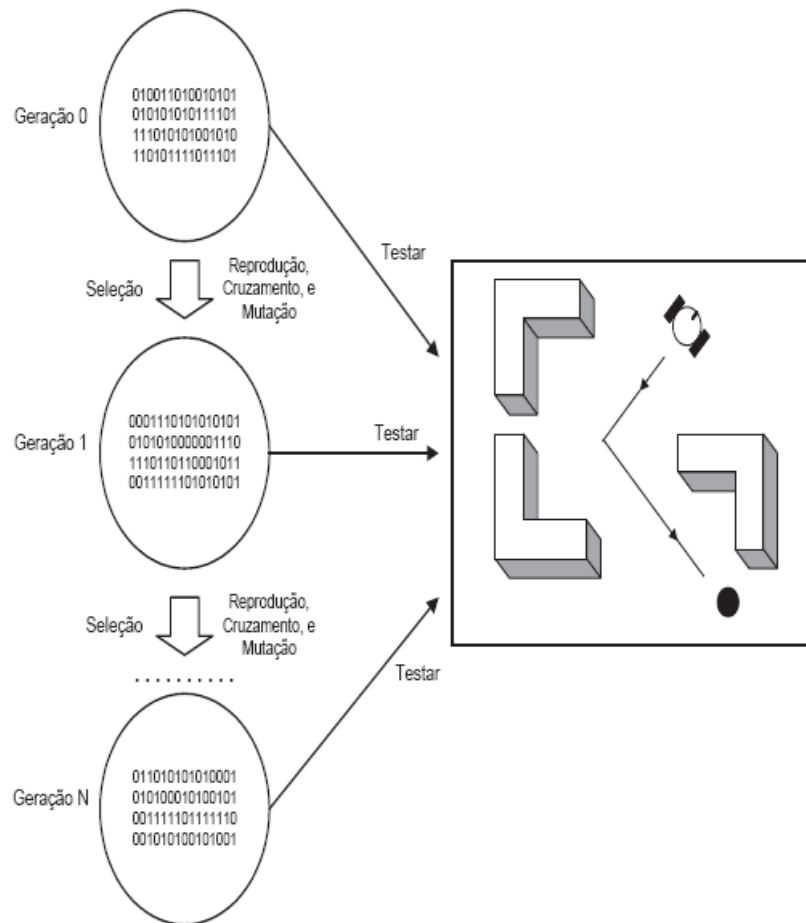


# Robótica Evolutiva

- Principais questões abordadas pela RE:
  - Sintetizar automaticamente comportamentos mais complexos do que aqueles que podem ser produzidos manualmente;
  - Explorar amplamente as características do ambiente e dos indivíduos, mesmo que algumas delas sejam obscuras ao projetista;
  - Produzir o comportamento esperado especificando-se o que o robô deve fazer e não como ele deve operar;
  - Mostrar que técnicas evolutivas podem reduzir o esforço humano necessário para produzir um sistema de controle em comparação com métodos manuais de projeto.

# Robótica Evolutiva

- Visão geral da RE:



# Robótica Evolutiva

- Abordagens
  - Evolução em simulação com testes em robôs reais.
  - Evolução em robôs reais.
  - Evolução embarcada

# Programação Genética Distribuída

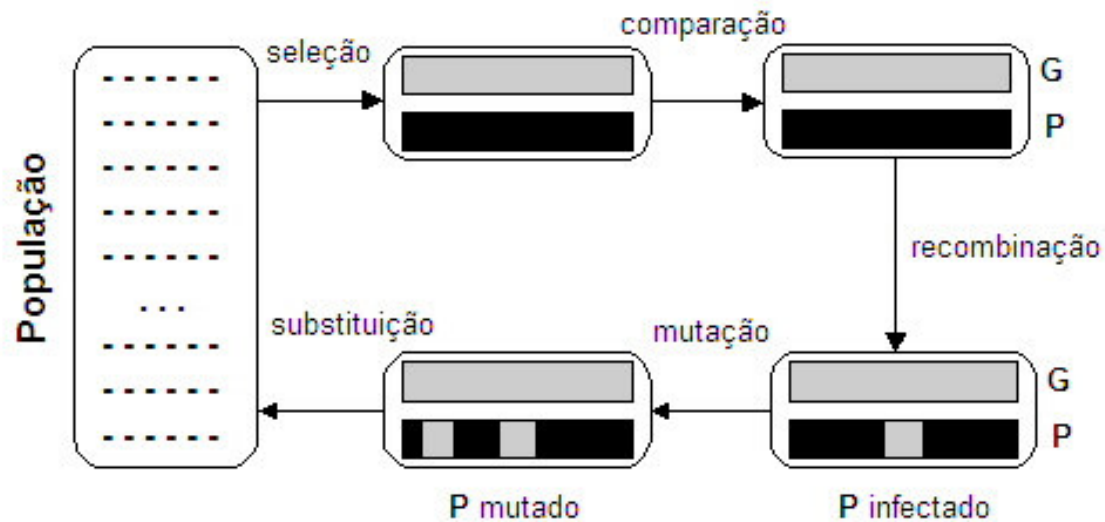


# Programação Genética Distribuída

- A PGD é uma extensão do algoritmo tradicional da PG.
- É baseada no Microbial GA, uma variação do algoritmo genético.
- O funcionamento do Microbial AG se assemelha à recombinação genética que acontece nas bactérias.

# Programação Genética Distribuída

- Microbial AG

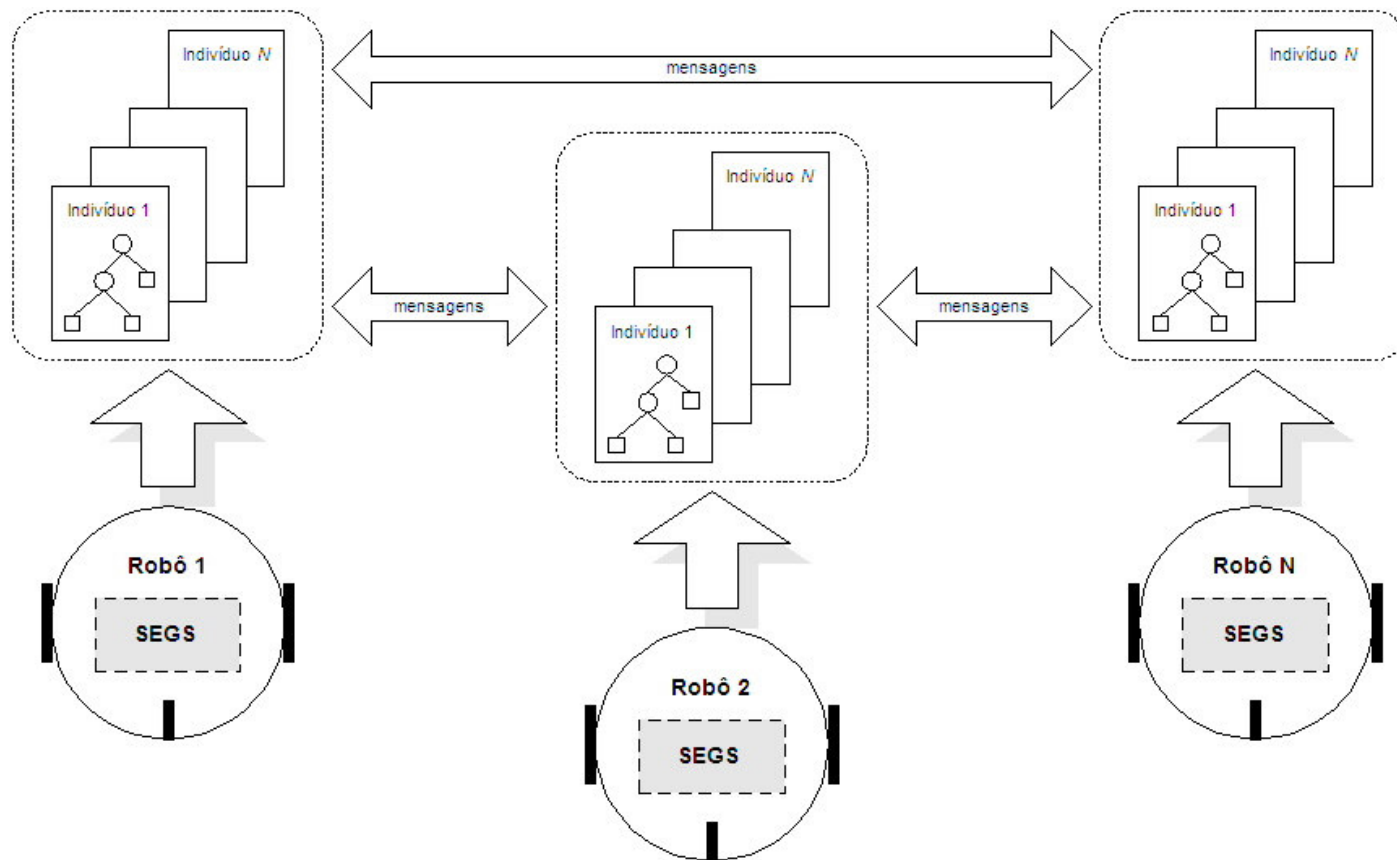


# Programação Genética Distribuída

- Na PGD são considerados dois conjuntos de populações:
  - Um conjunto local com todos os indivíduos que estão embarcados em cada robô.
  - O conjunto total que representa a união de todos os conjuntos locais (da população de robôs).

# Programação Genética Distribuída

- Visão geral da população de indivíduos



# Programação Genética Distribuída

- Algoritmo da PGD

1. Criar aleatoriamente uma população de programas;
2. Executar iterativamente os seguintes passos até que algum critério de parada seja satisfeito:
  - a) Avaliar cada programa da população através de uma função de avaliação, que expressa a sua aptidão;
  - b) Receber uma mensagem **M** de um indivíduo remoto enviadas por outro robô;
  - c) Selecionar os  $t$  melhores indivíduos da população local usando o método de seleção por torneio;
  - d) Selecionar aleatoriamente uma parte do melhor indivíduo local (mais adaptado) e enviar uma mensagem **M**, contendo a parte selecionada mais o valor do *fitness*, em *broadcast* (difusão) para os outros robôs;
  - e) Comparar se o *fitness* do pior indivíduo selecionado localmente é menor que o *fitness* do indivíduo remoto. Se sim, executa o operador de mutação remota substituindo uma parte, selecionada aleatoriamente, do indivíduo local pela parte recebida do indivíduo remoto;
  - f) Executar os operadores de cruzamento e mutação;
3. Retornar com o melhor programa encontrado.

# Programação Genética Distribuída

- Operador de Mutação Remota

$$P_{local}(I) = \left\{ \begin{array}{ll} OMR(I) & \text{if } Fitness(I_{remoto}) > Fitness(I_{local}) \\ I & \text{if } Fitness(I_{remoto}) \leq Fitness(I_{local}) \end{array} \right\}$$

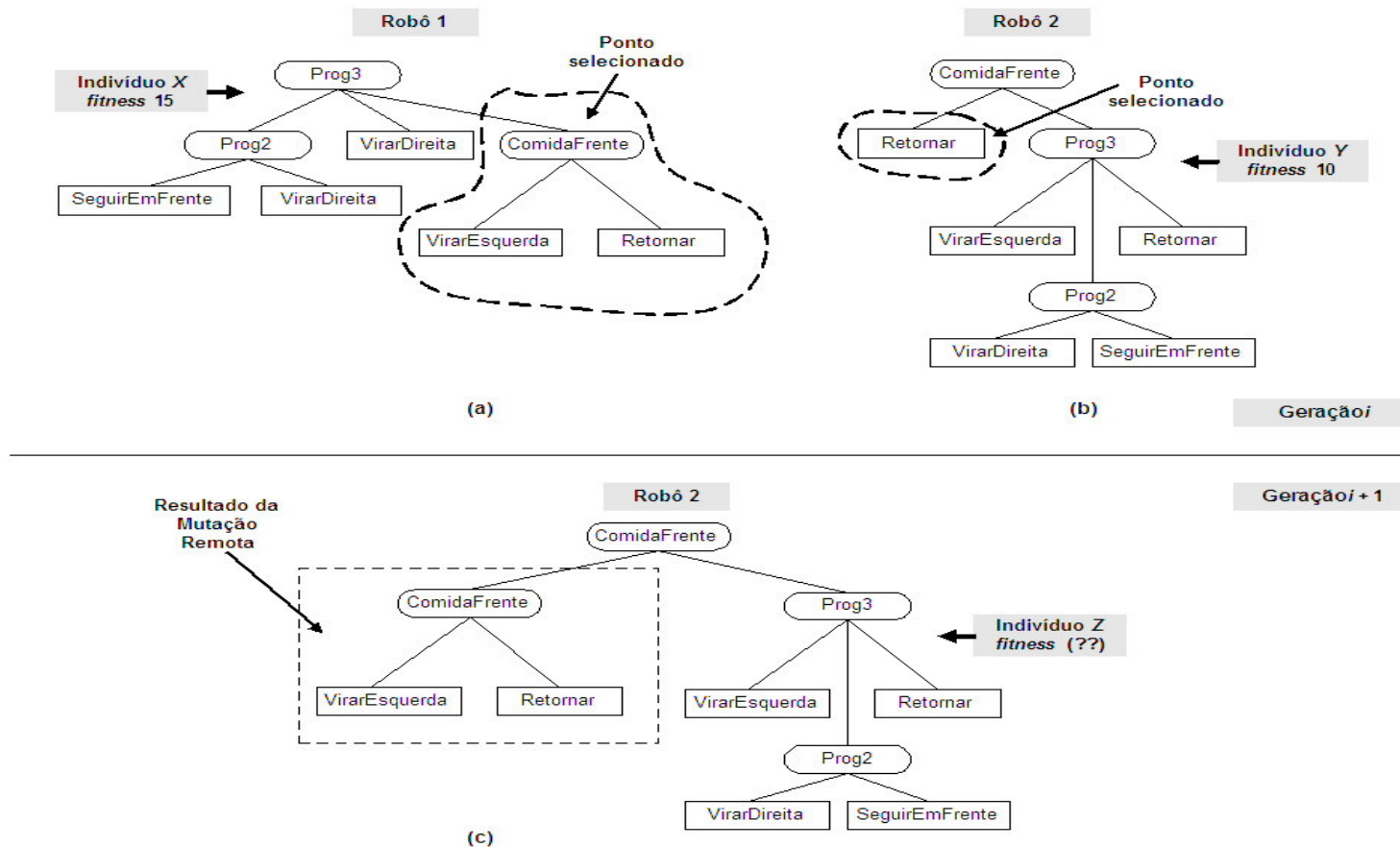
# Programação Genética Distribuída

- Exemplo: **tarefa de forrageamento**
  - Conjunto de funções e terminais

Funções			
Nome	Aridade	Id.	Definição
ComidaFrente	2	2	Se encontrou comida, executa nodo da esquerda; senão, executa nodo da direita.
Prog2	2	4	Executa dois ramos da árvore
Prog3	3	6	Executa três ramos da árvore
Terminais			
VirarDireita	0	1	Faz o robô virar a direita (15 graus).
VirarEsquerda	0	3	Faz o robô virar a esquerda (15 graus).
SeguirEmFrente	0	5	Faz o robô seguir em frente (300ms).
Retornar	0	7	Faz o robô retornar, dar a ré (300ms).

# Programação Genética Distribuída

- Exemplo: tarefa de forrageamento



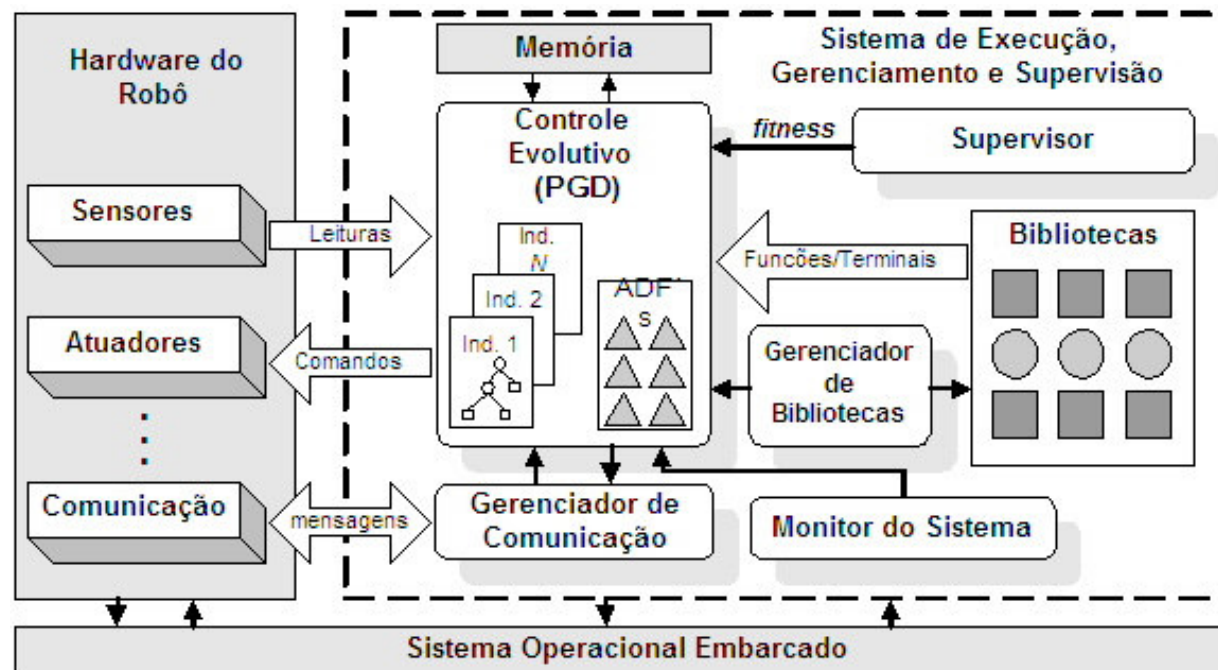


# Programação Genética Distribuída

- Sistema de Execução, Gerenciamento e Supervisão (SEGS)
  - Sistema que executa e gerencia o processo evolutivo em cada robô.
  - O SEGS é um sistema embarcado que controla as ações do robô.
  - É composto por seis módulos
    - Controle Evolutivo, Memória, Gerenciador de Comunicação, Gerenciador de Bibliotecas, Supervisor e Monitor.

# Programação Genética Distribuída

- Sistema de Execução, Gerenciamento e Supervisão

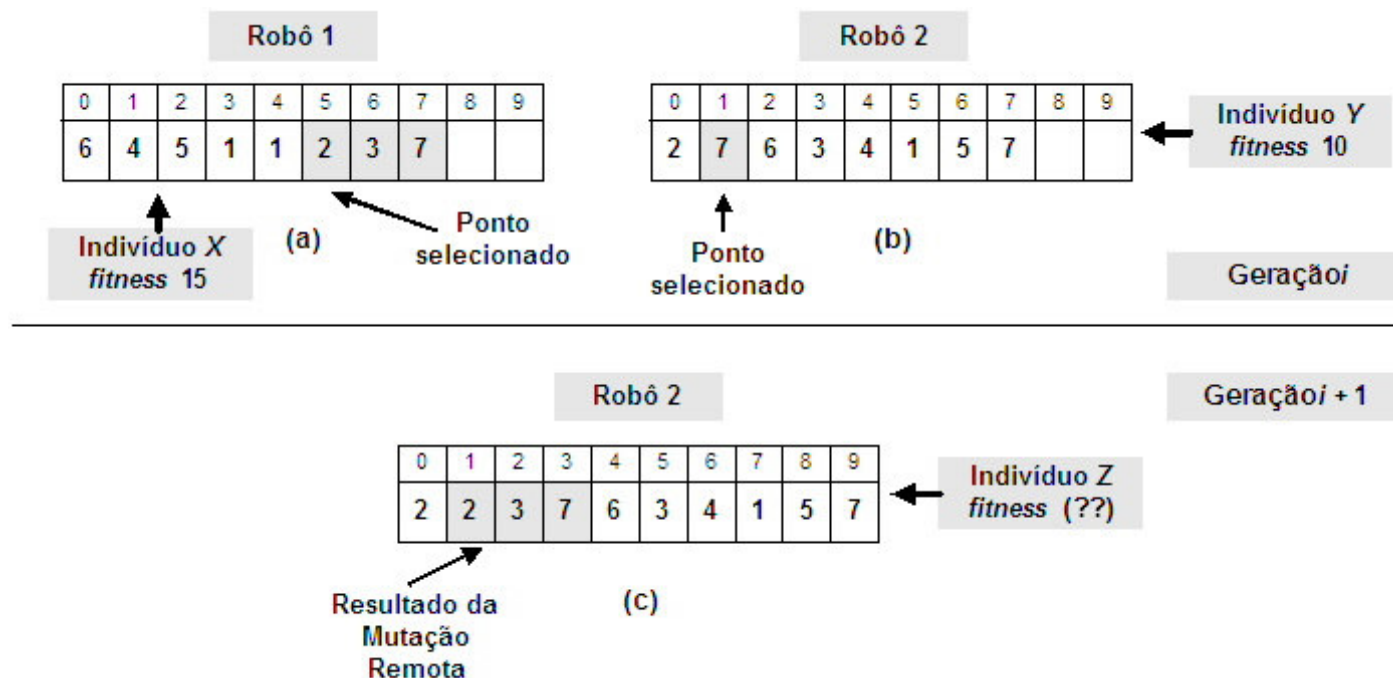


# Programação Genética Distribuída

- Implementação da PGD e do SEGS
  - A PGD foi implementada como um biblioteca.
  - Na PGD os indivíduos são representados por um vetor (Programação Genética Linear).
  - Tanto a PGD como o SEGS foram implementados em C.

# Programação Genética Distribuída

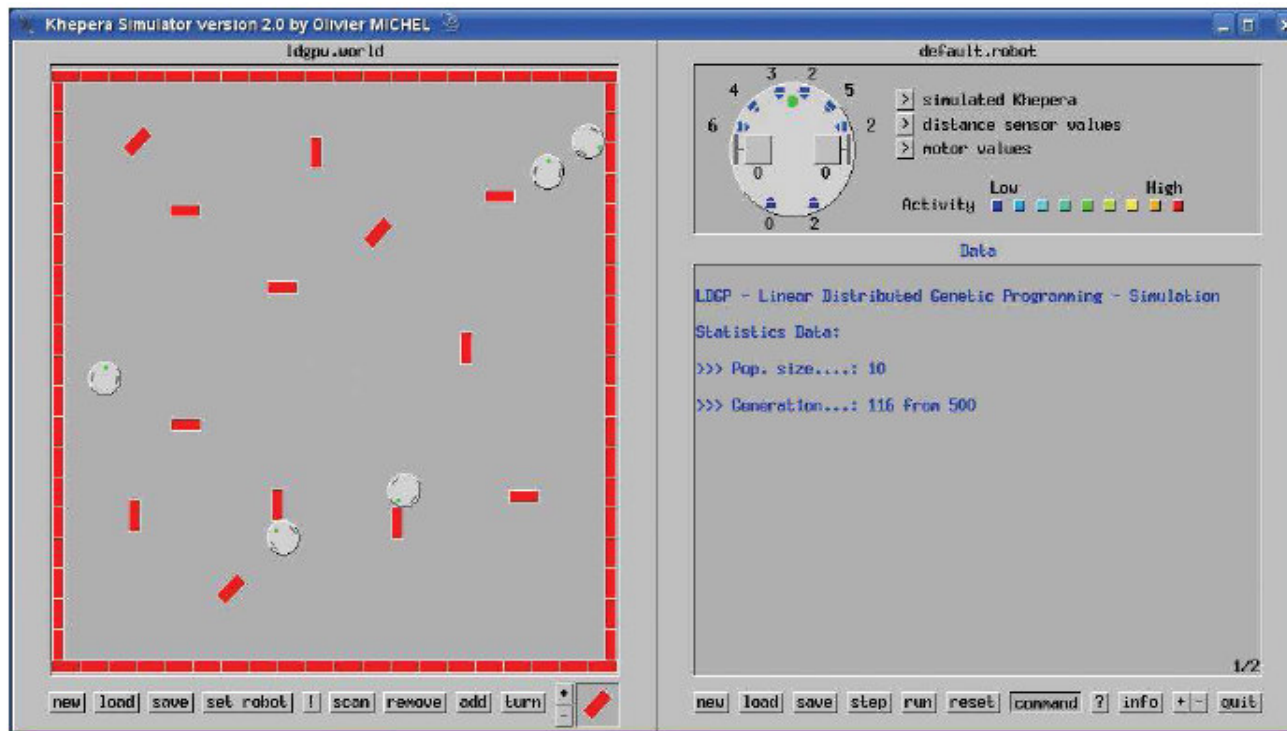
- Representação vetorial dos indivíduos na PGD



# Resultados

# Resultados

- Tarefa 1: Desviar de obstáculos (navegação livre de colisões)



# Resultados

- Tarefa 1: **Desviar de obstáculos (navegação livre de colisões)**
  - Conjunto de funções e terminais

Funções			
Nome	Aridade	Id.	Definição
Prog1	1	2	Executa um ramo da árvore.
Prog2	2	4	Executa dois ramos da árvore.
Prog3	3	6	Executa três ramos da árvore.

Terminais			
Nome	Aridade	Id.	Definição
TurnRight	0	1	Virar a direita (15 graus).
TurnLeft	0	3	Virar a esquerda (15 graus).
GoForward	0	5	Seguir em frente (300 ms).
Return	0	7	Retornar (300ms).

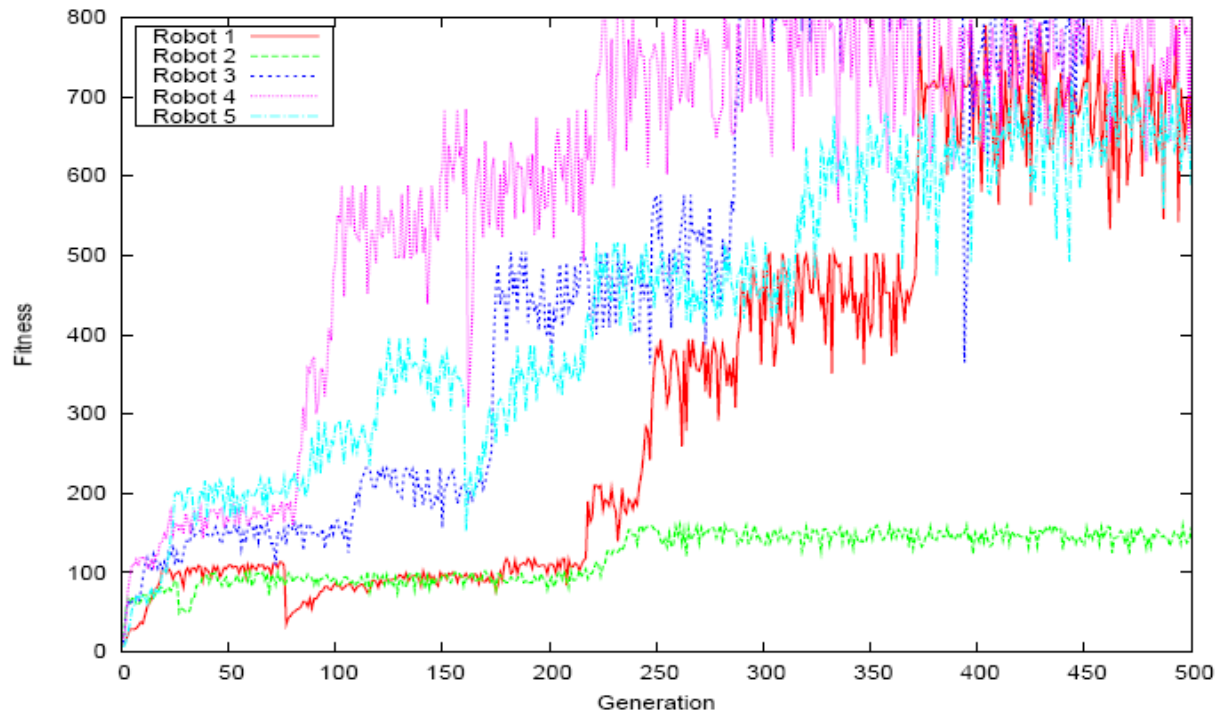
# Resultados

- Tarefa 1: **Desviar de obstáculos (navegação livre de colisões)**
  - Parâmetros usados no experimento:
    - Número de robôs: **5**
    - Número de gerações: **500**
    - Tamanho da população local: **10 indivíduos**
    - Tamanho do vetor: **50 posições**
    - Probabilidade de cruzamento: **60%**
    - Probabilidade de mutação: **10%**
    - Método de seleção: **seleção por torneio ( $t = 6$ )**
    - Função de avaliação: método de punição e recompensa: se CHOCOU COM OBSTÁCULO; ***fitness -= 3***; senão ***fitness += 2***



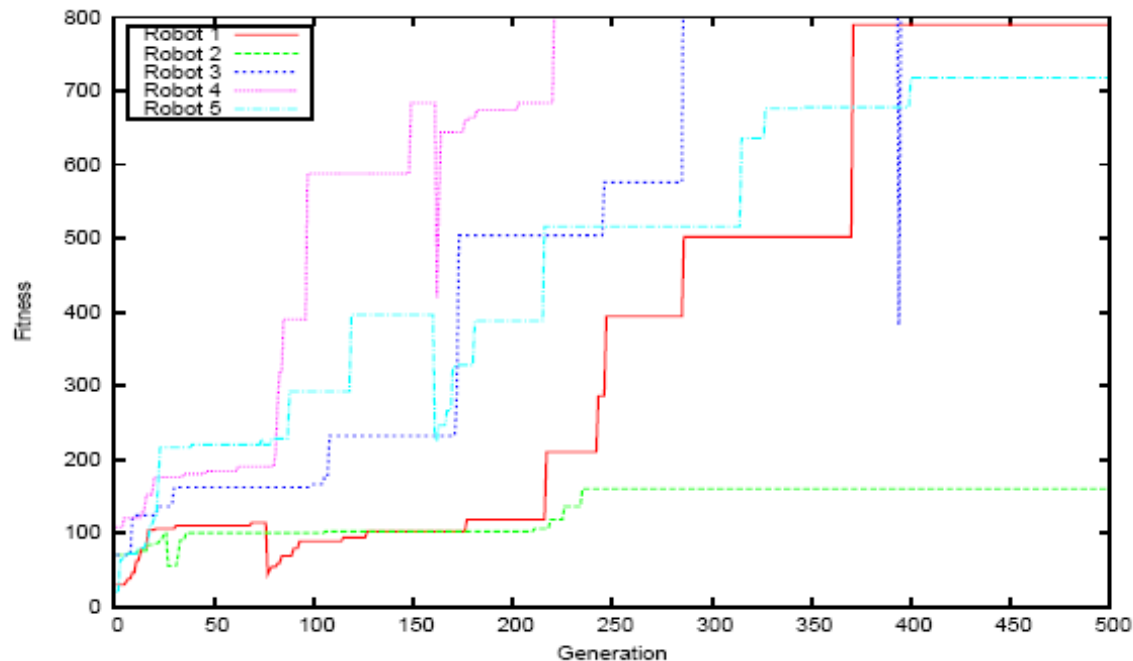
# Resultados

- Tarefa 1: **Desviar de obstáculos (navegação livre de colisões)**
  - Gráfico da média da aptidão da população



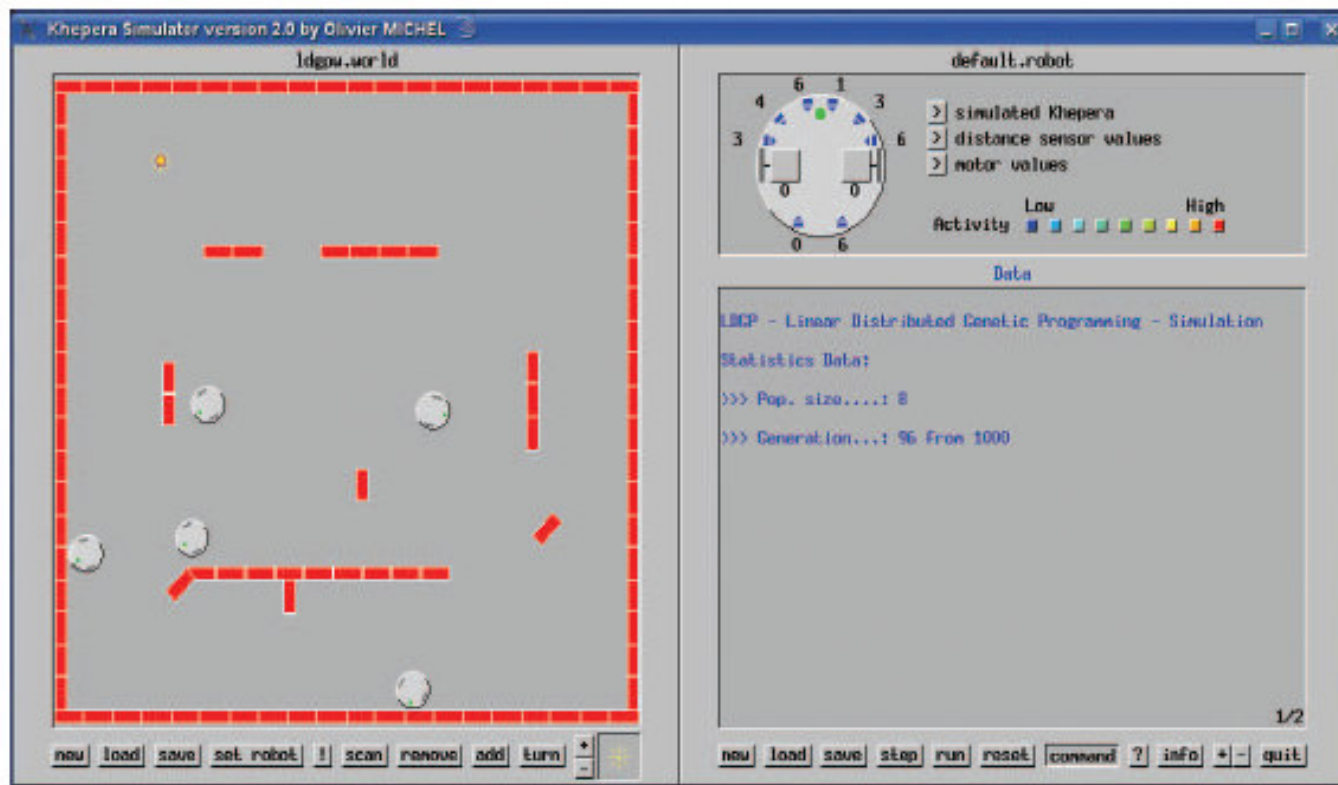
# Resultados

- Tarefa 1: **Desviar de obstáculos (navegação livre de colisões)**
  - Gráfico da média do melhor indivíduo em cada geração



# Resultados

- Tarefa 2: **Forrageamento**



# Resultados

- Tarefa 2: **Forrageamento**
  - Conjunto de funções e terminais

Funções			
Nome	Aridade	Id.	Definição
FoodAhead	2	2	Se encontrou comida, executa o nodo da esquerda; senão, executa o nodo da direita.
Prog2	2	4	Executa dois ramos da árvore.
Prog3	3	6	Executa três ramos da árvore.

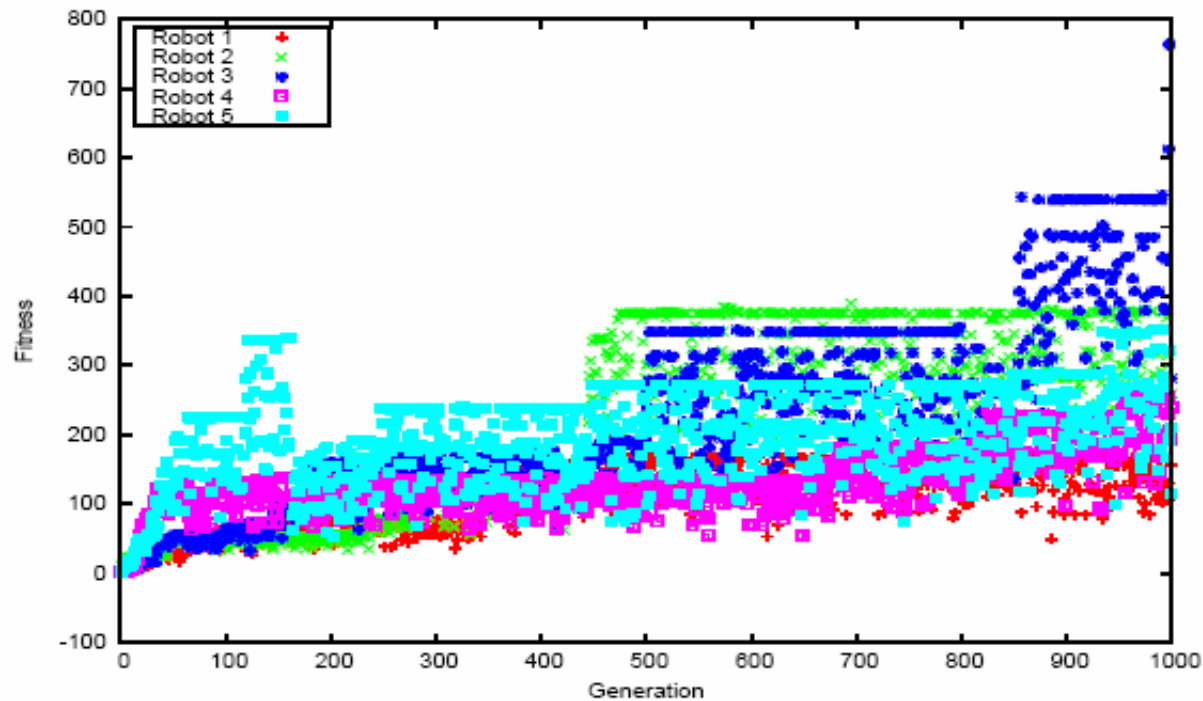
Terminais			
Nome	Aridade	Id.	Definição
TurnRight	0	1	Virar a direita (15 graus).
TurnLeft	0	3	Virar a esquerda (15 graus).
GoForward	0	5	Siguir em frente (300 ms).
Return	0	7	Retornar (300ms).

# Resultados

- Tarefa 2: **Forrageamento**
  - Parâmetros usados no experimento:
    - Número de robôs: **5**
    - Número de gerações: **1000**
    - Tamanho da população local: **8 indivíduos**
    - Tamanho do vetor: **50 posições**
    - Probabilidade de cruzamento: **50%**
    - Probabilidade de mutação: **20%**
    - Método de seleção: **seleção por torneio ( $t = 6$ )**
    - Função de avaliação: método de punição e recompensa: se FOODAHEAD; ***fitness += 10***; se CHOCOU COM OBSTÁCULO; ***fitness -= 1***; senão ***fitness += 1***

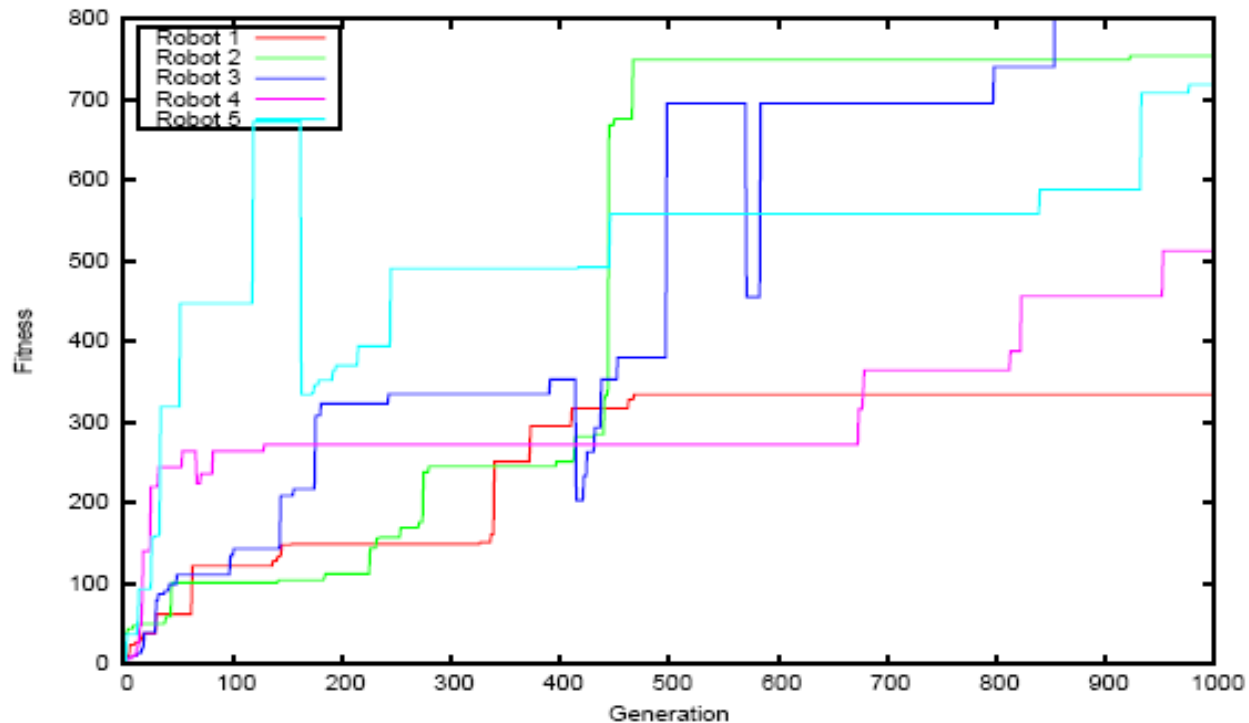
# Resultados

- Tarefa 2: **Forrageamento**
  - Gráfico da média da aptidão da população



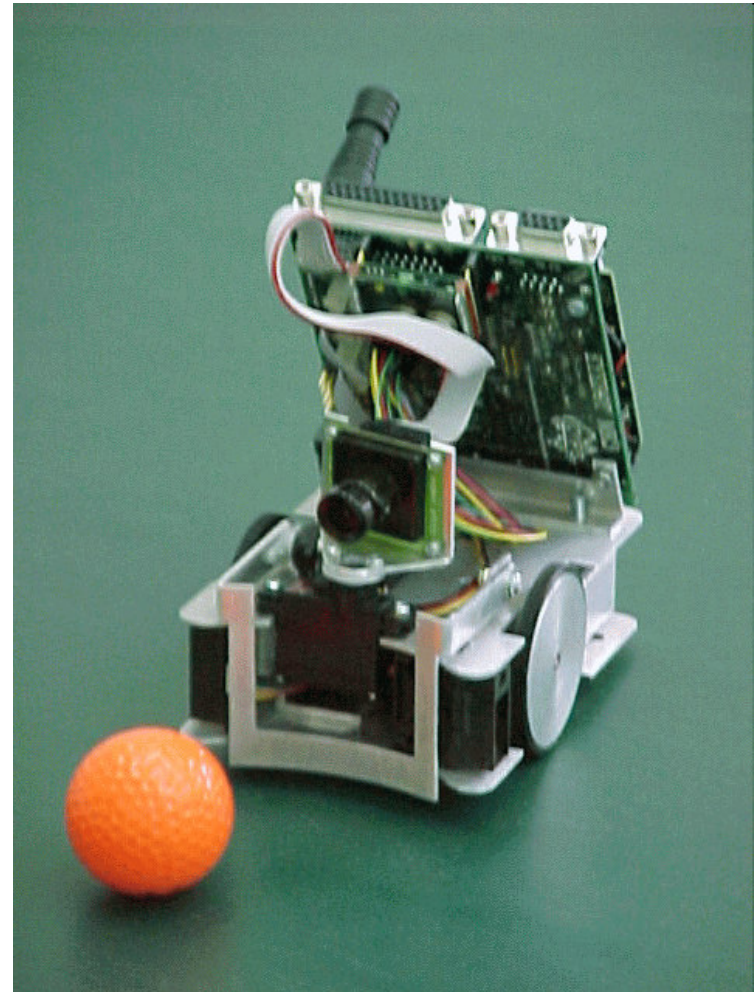
# Resultados

- Tarefa 2: **Forrageamento**
  - Gráfico da média do melhor indivíduo em cada geração



# Resultados

- Próximos experimentos:
  - Evolução em robôs reais
    - Robôs da família Eyebot (Soccerbot)





# Contato/Agradecimento

- Email para contato:
  - [anderson@das.ufsc.br](mailto:anderson@das.ufsc.br)

***OBRIGADO!***