

# Utilizando Robótica Evolutiva para o Desenvolvimento da Morfologia de Robôs

Jesimar da Silva Arantes

Orientador: Tales Heimfarth  
Co-orientador: Raphael W. de Bettio

Bacharelado em Ciência da Computação  
Universidade Federal de Lavras

Fevereiro – 2014

GRUBi 

# Sumário

- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia
- 4 Resultados
- 5 Considerações Finais

## 1 Introdução

- Contextualização
- Objetivos

## 2 Referencial Teórico

## 3 Metodologia

## 4 Resultados

## 5 Considerações Finais

### Ideia Básica

- Este trabalho apresenta o desenvolvimento um sistema para Robótica Evolutiva (RE) para a evolução da morfologia de robôs
- Nesse sistema as peças do robô foram modeladas com base em algumas peças para construção de robôs
- Os controles do robô (comportamentos) foram definidas de forma genérica baseadas na arquitetura de subsunção
- A evolução dos robôs se dará a partir de regras definidas em Algoritmos Genéticos (AG) a partir de uma função de avaliação

# Introdução

## Objetivos

### Objetivo Geral

Construir um módulo de robótica evolutiva que utilizará AG para evoluir a morfologia de robôs em um ambiente virtual

### Objetivo Específicos

- 1 Construir o módulo de evolução de robôs na plataforma *GrubiBots*
- 2 Modelar as peças do robô
- 3 Definição da função objetivo
- 4 Desenvolvimento do algoritmo genético
- 5 Definir os parâmetros do algoritmo genético

## 1 Introdução

## 2 Referencial Teórico

- Algoritmos Genéticos
- Sistema de Controle de Robôs
- GrubiBots
- Biblioteca ODE

## 3 Metodologia

## 4 Resultados

## 5 Considerações Finais

### Visão Geral

- Os Algoritmos Genéticos (AGs) são algoritmo de computação evolutiva inicialmente proposto por Holland (1975)
- AGs seguem o princípio da seleção natural
- O conceito de seleção natural foi proposto Darwin em 1859
- A ideia consiste em que espécies com características mais adaptadas ao meio tendem a sobreviver
- Já as espécies pouco adaptadas tendem a ser extintas
- Hereditariedade é a capacidade dos seres vivos de transmitirem características genéticas aos seus descendentes

### Terminologia

- **População:** Conjunto de indivíduos
- **Indivíduo:** É uma solução do problema
- **Função de Fitness:** Avaliar as soluções produzidas pelos indivíduos
- **Seleção:** Consiste em selecionar os indivíduos para reprodução
- **Torneio- $q$ :** Consiste no sorteio aleatório de  $q$  indivíduos e seleciona-se o melhor
- **Crossover:** São escolhidos dois indivíduos para se reproduzirem
- **Mutação:** É um operador que auxilia a garantir variabilidade genética
- **Elitismo:** Copia os melhores indivíduos para a nova população

### Funcionamento

- O AG inicialmente define uma população de indivíduos
- Cada indivíduo representa uma solução do problema em questão
- A população tem indivíduos inicializados de forma aleatória
- Primeiramente se seleciona dois indivíduos da população para se realizar o *crossover*
- Em seguida, ele é processado por uma rotina que realiza mutações em sua codificação
- Esse procedimento é executado até que toda a população da próxima geração seja totalmente gerada
- Esse procedimento de geração das gerações seguintes são executados um número determinado de vezes

# Referencial Teórico

## Sistema de Controle de Robôs

### Visão Geral

A tarefa do sistema de controle é fazer com que o sistema alcance um determinado estado, obtendo informações através de sensores e transmite comandos para os atuadores

### Arquitetura de Subsunção

- 1 Proposta por Brooks em 1986
- 2 Os comportamentos são definidos por leituras dos sensores e envio de comandos aos atuadores
- 3 Os comportamentos são conectados uns aos outros formando uma rede em camadas
- 4 Cada camada é responsável por uma atividade do robô

### Visão Geral

- 1 A plataforma *GrubiBots* foi desenvolvida no GRUBI/UFLA
- 2 Voltada para a programação de robôs móveis autônomos baseados em sensores
- 3 Alguns robôs reais podem ser controlados pela plataforma

## Características Tecnológicas/Arquitetônicas

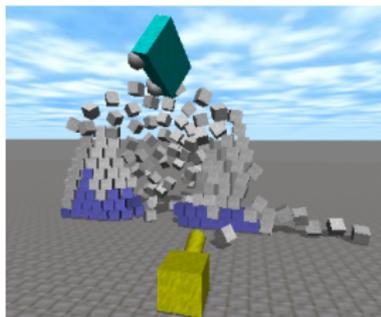
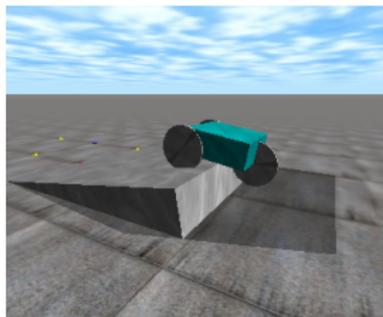
- 1 Implementada na linguagem Java
- 2 Capacidades de Simulação
- 3 Extensibilidade
- 4 Modelo Arquitetônico Distribuído
- 5 Arquitetura de Subsunção

# Referencial Teórico

## Biblioteca ODE

### Visão Geral

- 1 ODE - *Open Dynamics Engine*
- 2 Esta biblioteca é um motor de física *Open Source*
- 3 A ode4j é a implementação da ODE para Java



## 1 Introdução

## 2 Referencial Teórico

## 3 Metodologia

- Ferramentas Utilizadas
- Morfologia do Robô
- Pontos do Conexão do Robô
- Montagem da Morfologia do Robô
- Operadores de Crossover e Mutação
- Cenários Simulados e Função de Fitness

## 4 Resultados

## 5 Considerações Finais

# Metodologia

## Ferramentas Utilizadas

### Plataforma de Robótica

- O módulo de robótica implementado funciona como uma extensão da plataforma *GrubiBots*

### Linguagem Utilizada

- Foi desenvolvido em Java
- Suporta os principais conceitos de orientação a objetos
- Independente de plataforma

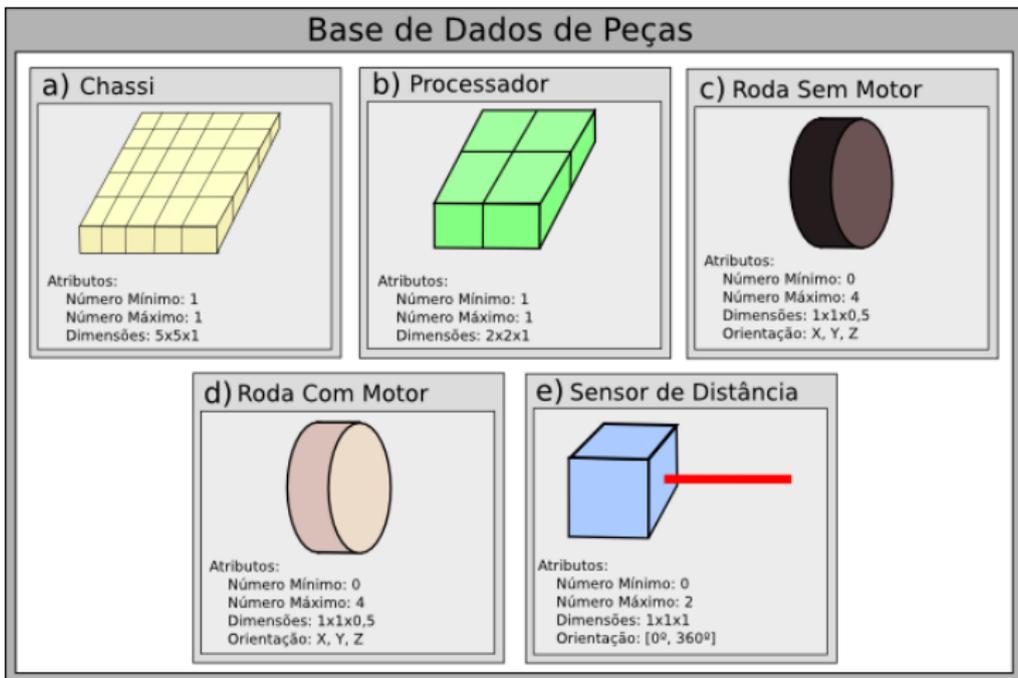
### Biblioteca de Física

- A construção do módulo vai utilizar a biblioteca ODE
- Justificativa a plataforma *GrubiBots* a utiliza como motor de física

GRUBIBOTS

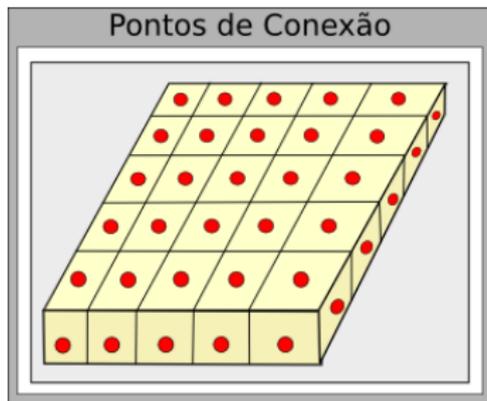
# Metodologia

## Morfologia do Robô



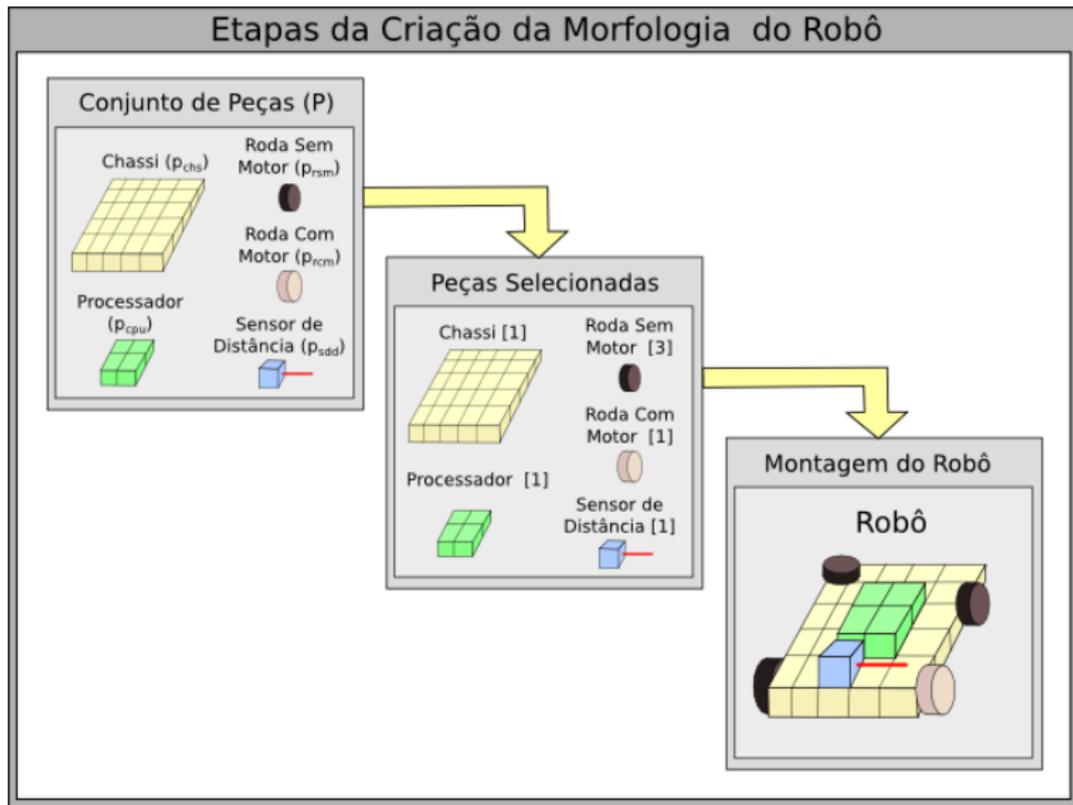
### Visão Geral

- 1 Definir no simulador o local dos pontos de acoplamento de peças
- 2 O *Chassi* é constituído de um conjunto de células
- 3 Em cada célula foi colocação um ponto de junção
- 4 Na parte de baixo do *Chassi* não é permitido acoplar peças



# Metodologia

## Montagem da Morfologia do Robô

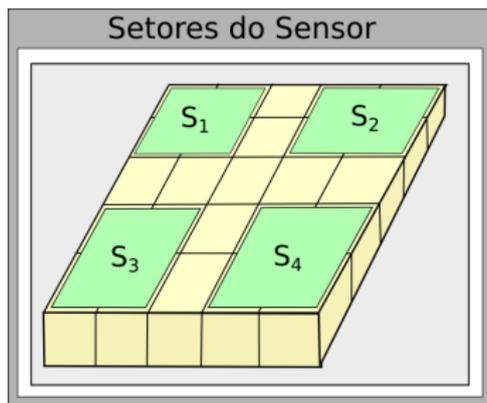


# Metodologia

## Operadores de Crossover e Mutação

### Operadores de Crossover

- 1 *Crossover* de Processador
- 2 *Crossover* de Roda Sem Motor
- 3 *Crossover* de Roda Com Motor
- 4 *Crossover* de Sensor de Distância



### Operadores de Mutação

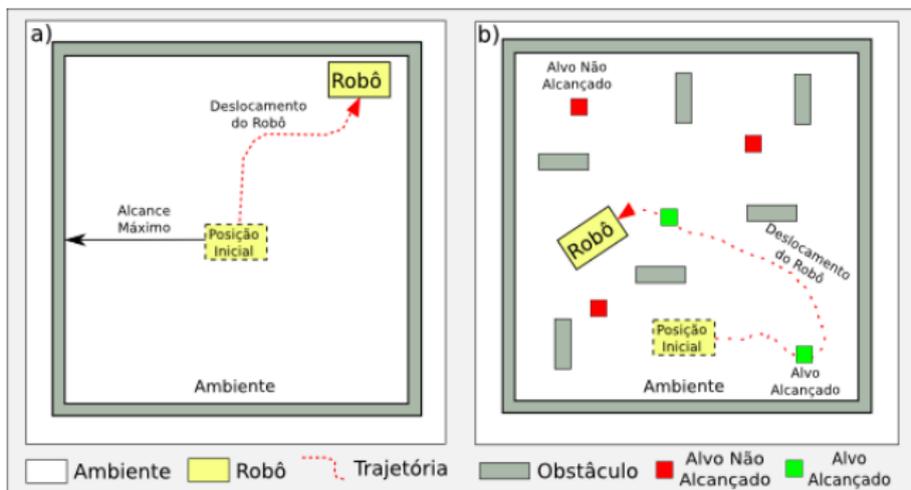
- 1 Adicionar Peça
- 2 Remover Peça
- 3 Trocar Peças

# Metodologia

## Cenários Simulados e Função de Fitness

### Cenários Simulados

- 1  $Problema_1$ : Problema da locomoção do robô no ambiente livre de obstáculos
- 2  $Problema_2$ : Problema de locomoção do robô no ambiente com obstáculos e alvos



GRUBi

# Metodologia

## Cenários Simulados e Função de Fitness

### Função de Fitness do Problema<sub>1</sub>

$$\text{Min}F(\dots) = w_{hw} * \sum_{p \in P} (w_p * q_p) + w_{sw} * (C_{lado}/2 - R_{desl}) \quad (1)$$

onde,

$$\sum_{p \in P} (w_p * q_p) = w_{chs} * q_{chs} + w_{cpu} * q_{cpu} + w_{rsm} * q_{rsm} + w_{rcm} * q_{rcm} + w_{sdd} * q_{sdd} \quad (2)$$

### Função de Fitness do Problema<sub>2</sub>

$$\text{Min}F(\dots) = w_{hw} * \sum_{p \in P} (w_p * q_p) + w_{alv} * (q_{alvT} - q_{alvC}) + w_{col} * q_{col} \quad (3)$$

- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia
- 4 Resultados**
  - Parâmetros Utilizados
  - Gráficos
  - Morfologia dos Robôs
- 5 Considerações Finais

# Resultados

## Parâmetros Utilizados

Problema	Parâmetro	Valor
Ambos	Tamanho da População	100
Ambos	Número de Gerações	50
Ambos	Tamanho do Torneio	3
Ambos	Elitismo	Sim
<i>Problema<sub>1</sub></i>	Número de <i>Steps</i> (Tempo)	600
<i>Problema<sub>2</sub></i>	Número de <i>Steps</i> (Tempo)	6000
<i>Problema<sub>1</sub></i>	Número de Simulações	3
<i>Problema<sub>2</sub></i>	Número de Simulações	2

Problema	Parâmetro	Valor
<i>Problema<sub>1</sub> Conf<sub>1</sub></i>	Taxa <i>Crossover</i>	75%
<i>Problema<sub>1</sub> Conf<sub>1</sub></i>	Taxa de Mutação	85%
<i>Problema<sub>1</sub> Conf<sub>2</sub></i>	Taxa <i>Crossover</i>	80%
<i>Problema<sub>1</sub> Conf<sub>2</sub></i>	Taxa de Mutação	10%
<i>Problema<sub>2</sub></i>	Taxa <i>Crossover</i>	75%
<i>Problema<sub>2</sub></i>	Taxa de Mutação	85%

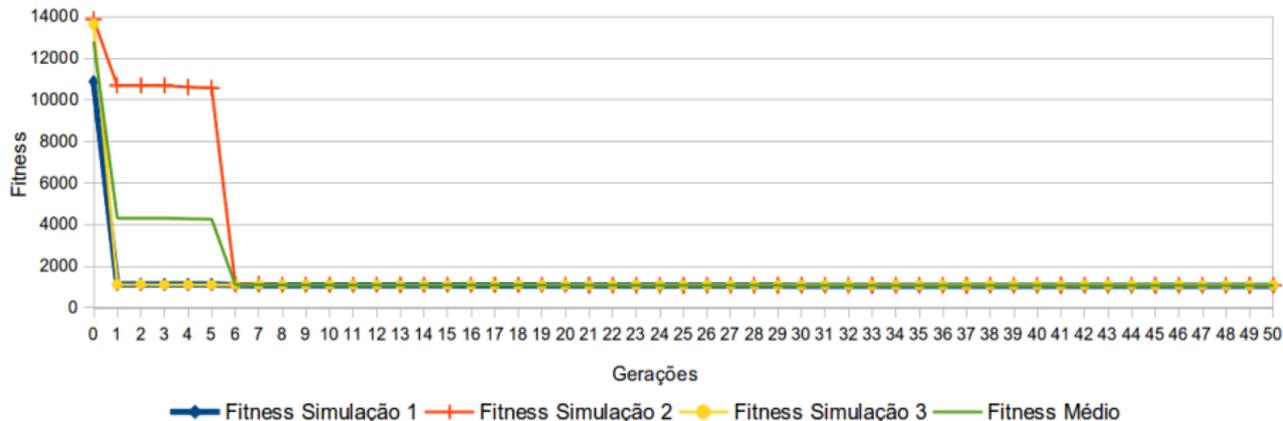
# Resultados

## Gráficos

### Dados

- 1 Melhor *Fitness*: 1048,92
- 2 *Fitness* Médio: 1077,65
- 3 Tempo médio gasto: 1 hora 45 minutos

Gráfico de Resultados dos Fitness do Problema 1

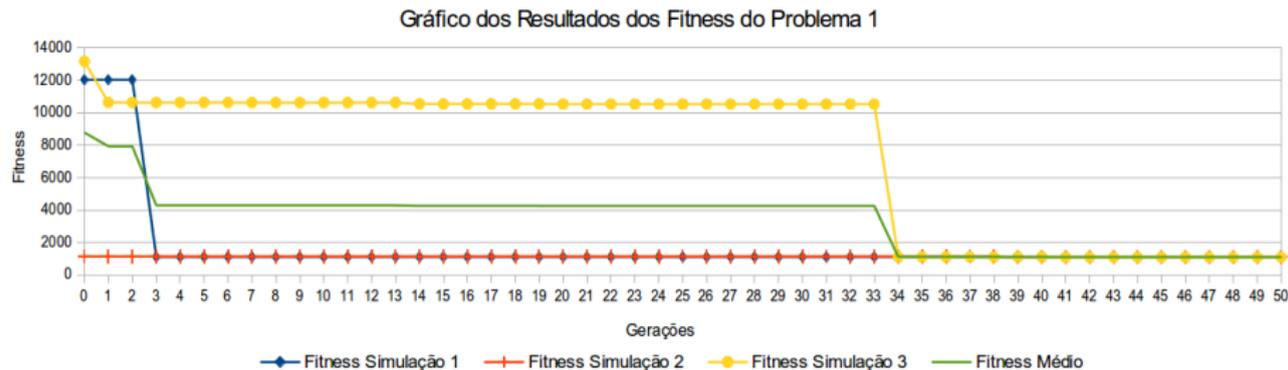


# Resultados

## Gráficos

### Dados

- 1 Melhor *Fitness*: 1087,81
- 2 *Fitness* Médio: 1099,09
- 3 Tempo médio gasto: 1 hora 44 minutos



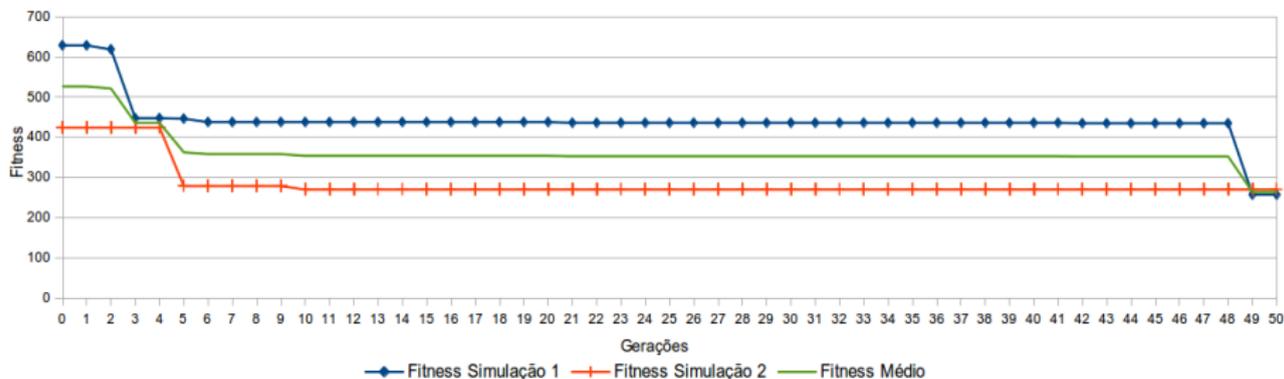
# Resultados

## Gráficos

### Dados

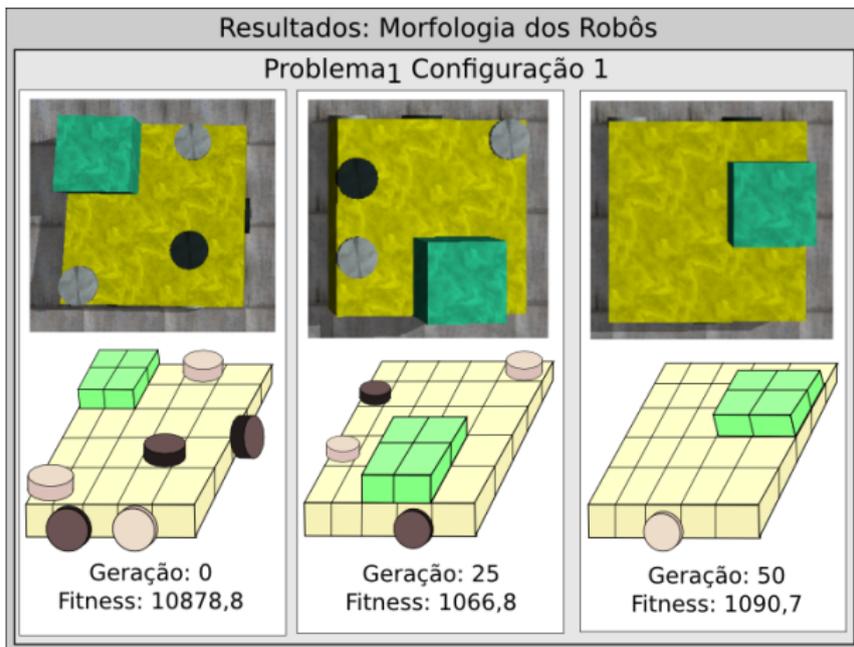
- 1 Melhor *Fitness*: 257,2
- 2 *Fitness* Médio: 263,55
- 3 Tempo médio gasto: 7 horas 18 minutos
- 4 Alvos atendidos: 3 de 5

Gráfico dos Resultados dos Fitness do Problema 2



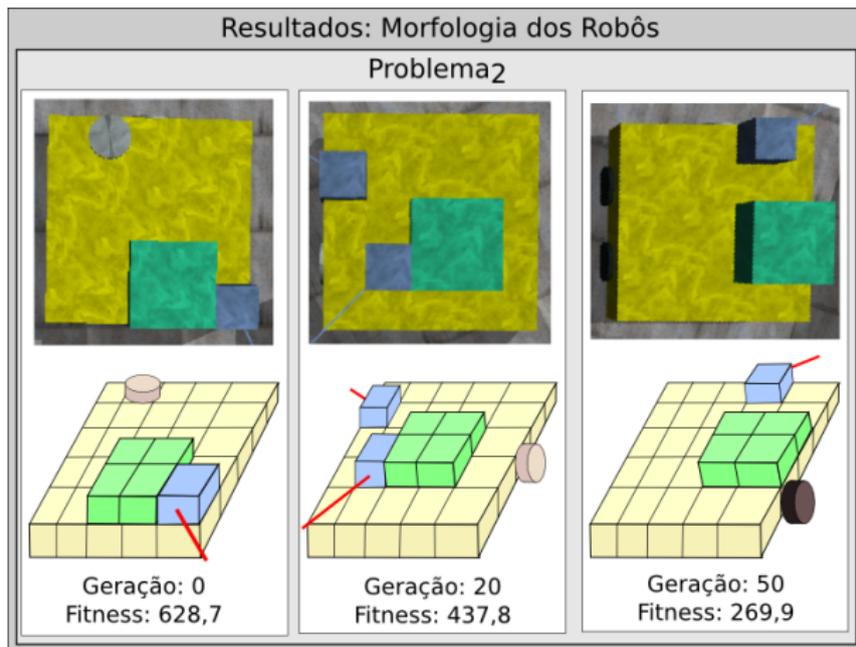
# Resultados

## Morfologia dos Robôs



# Resultados

## Morfologia dos Robôs





- 1 Introdução
- 2 Referencial Teórico
- 3 Metodologia
- 4 Resultados
- 5 Considerações Finais**
  - Conclusão
  - Trabalhos Futuros

# Considerações Finais

## Conclusão

### Recapitulando

- Este trabalho desenvolveu um módulo de robótica evolutiva
- Este módulo é capaz de evoluir a morfologia de um robô
- Com base num conjunto de peças, cenário e função objetivo
- Este módulo foi implementado dentro da plataforma *GrubiBots*

# Considerações Finais

## Conclusão

### Conclusão

- Solução ótima para o primeiro problema
- Solução razoável para o segundo problema
- No primeiro problema a segunda configuração mostra-se mais promissora (*Crossover* 75% e *Mutação* 85%)
- No entanto mais testes devem ser efetuados no segundo problema
- Novas funcionalidades incrementadas e melhoradas no *GrubiBots*

# Considerações Finais

## Trabalhos Futuros

### Trabalhos Futuros

- Efetuar mais testes variando melhor os parâmetros do AG
- Evoluir no segundo problema tanto a morfologia quanto o controle
- Resolver problemas em cenários mais complexos (cenários dinâmicos)
- Evolução de enxame de robôs
- Evolução colaborativa
- Evolução competitiva

Obrigado a Todos!!!  
Perguntas!!!