

Algoritmos Evolutivos Aplicados no Aprendizado em Jogos de Estratégia em Tempo Real



Rodrigo de Freitas Pereira
Claudio Fabiano Motta Toledo

Marcio Kassouf Crocomo
Eduardo do Valle Simões

Sumário

- ▶ Trabalhos Relacionados
- ▶ O jogo: BOS Wars
- ▶ O Algoritmo Evolutivo
- ▶ Resultados Computacionais
- ▶ Conclusões
- ▶ Referências
- ▶ Agradecimentos

Trabalhos Relacionados

- ▶ CROCOMO et al (2008) - SBGames
 - ▶ Algoritmo Genético aplicado a um RPG
 - ▶ O jogo desenvolvido por Crocomo, onde dois times compostos por dois magos e dois guerreiros se enfrentam.
 - ▶ Cada personagem possui um *script* definido como um conjunto de ações executadas em sequência.
 - ▶ As regras são formadas por cinco ações básicas: atacar, beber poção, mover-se, executar magia e passar a vez.
 - ▶ Ganha o time que matar todos os personagens do time adversário



Trabalhos Relacionados

- ▶ Spronck et al (2007)
 - ▶ Algoritmo Genético aplicado a um RTS
 - ▶ Wargus: baseado no Warcraft II
 - ▶ AG usado para gerar *scripts* (estratégia de jogo)
 - ▶ Ações de jogo: criar batalhões de guerra e construções, atacar inimigo.



O jogo: BOS Wars

- ▶ RTS Futurista
- ▶ Manter equilíbrio entre coleta de recursos e criação de exército para ataque
- ▶ Objetivo: destruir time adversário
- ▶ IA: ações codificadas em *scripts*
- ▶ Cinco tipos de estratégias para jogar: três ofensivas, uma defensiva e uma balanceada entre ações de ataque e defesa.
- ▶ Desenvolvido em C++ e LUA



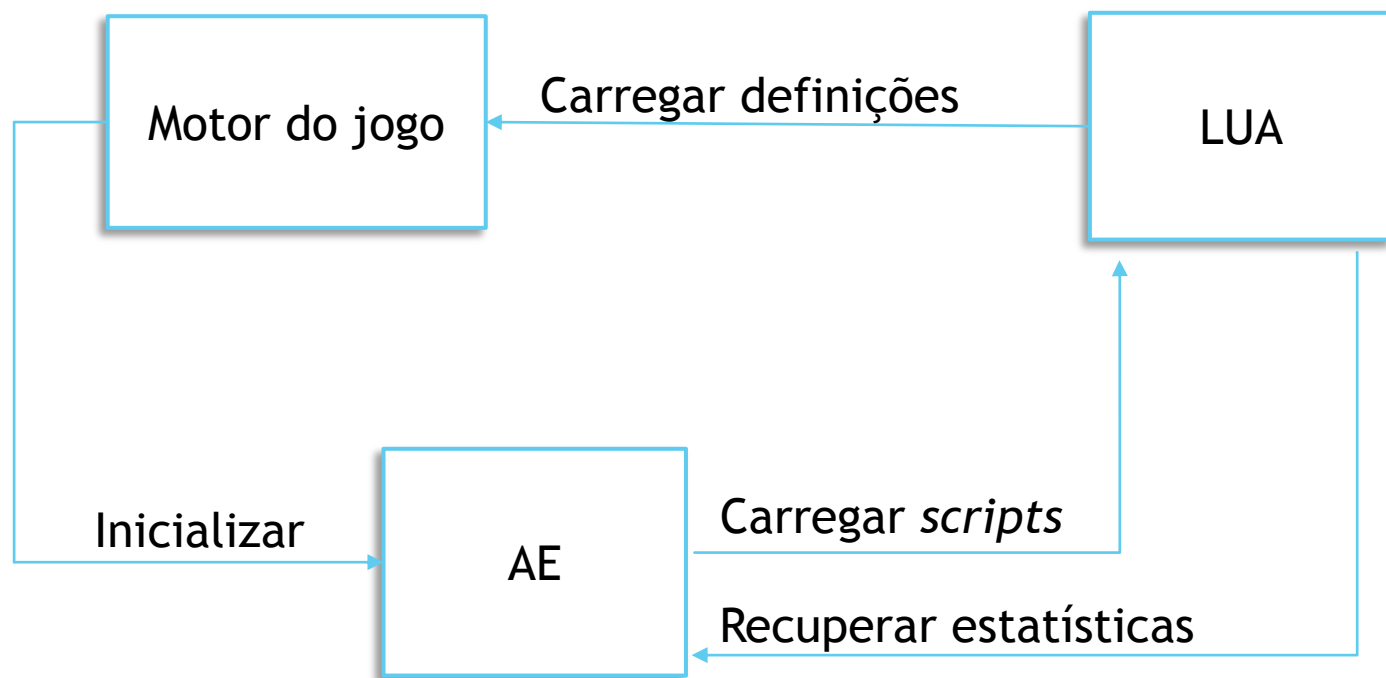
O jogo: BOS Wars

▶ Exemplo de um *script* de jogo

```
function() return AiNeed("unit-vault") end,  
function() return AiNeed("unit-camp") end,  
function() return AiForce(1,{"unit-assault",1}) end,  
function() return AiWaitForce(1) end,  
function() return AiAttackWithForce(1) end ,  
function() return AiNeed("unit-aircraftfactory") end,  
function() return AiForce(5,{"unit-heli",3}) end,  
function() return AiForce(5,{"unit-chopper",4}) end,  
function() return AiNeed("unit-vfac") end,  
function() return AiForce(5,{"unit-buggy",5}) end,  
function() return AiWaitForce(5) end,  
function() return AiAttackWithForce(5) end ,
```

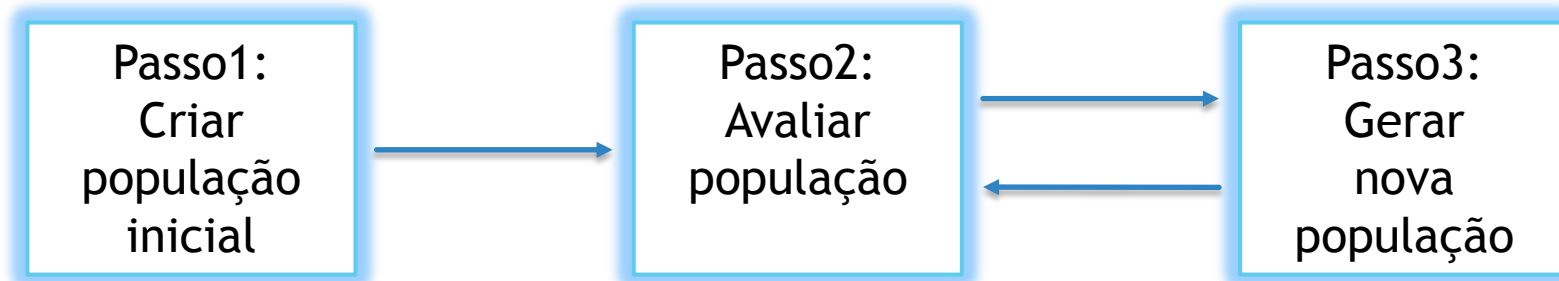
O jogo: BOS Wars

► Comunicação AE - BOS Wars



O Algoritmo Evolutivo

- ▶ Baseado no AG desenvolvido por Crocomo et al (2008).
- ▶ Principais diferenças estão na codificação e decodificação de um indivíduo (*script*)



O Algoritmo Evolutivo

- ▶ Indivíduo
 - ▶ Estrutura formada por um conjunto de genes
 - ▶ Codifica uma estratégia de jogo (*script*)
 - ▶ Representação:



O Algoritmo Evolutivo

▶ Gene

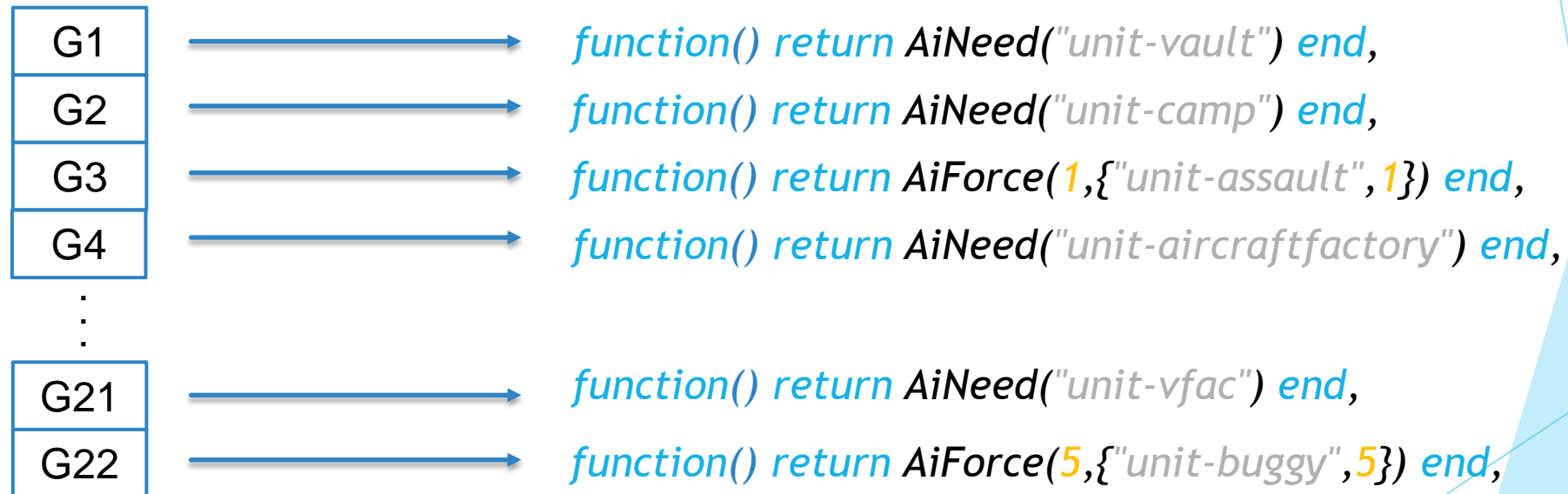
- ▶ Menor estrutura do indivíduo que armazena uma ação de jogo.
- ▶ Dois tipos de gene foram definidos: Batalhão, Construção

Gene Batalhão
Força Unidade Quantidade

Gene Construção
Construção

O Algoritmo Evolutivo

► Decodificação de um indivíduo em *script*



O Algoritmo Evolutivo

- ▶ Fitness : nota de avaliação de um indivíduo em uma partida

$$F(t) = \begin{cases} 1 - \frac{t}{2 * T_{\max}} & \text{Se indivíduo ganhar a partida} \\ \frac{t}{2 * T_{\max}} & \text{Se indivíduo perder a partida} \end{cases}$$

- ▶ $F(t)$: Função de fitness
- ▶ t : Tempo gasto em uma partida
- ▶ T_{\max} : Tempo limite de uma partida

O Algoritmo Evolutivo

▶ Torneio



O Algoritmo Evolutivo

► Crossover Uniforme

Pai 1 - Melhor Indivíduo

Ga	Gb	Gc	Gd
----	----	----	----

Pai 2

Gx	Gy	Gz	Gw
----	----	----	----

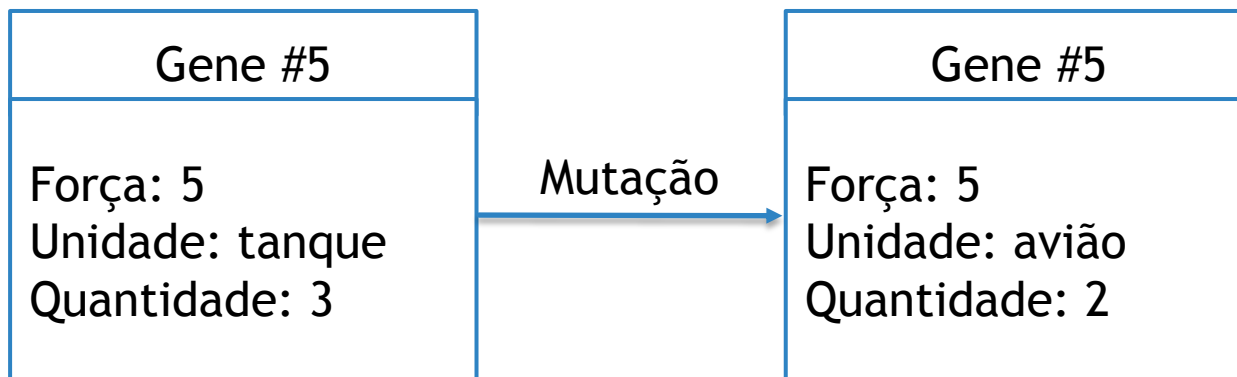
Filho Gerado

Ga	Gy	Gc	Gw
----	----	----	----



O Algoritmo Evolutivo

- ▶ Mutação : troca parâmetro de um gene



Resultados Computacionais

- ▶ Objetivo: Calibrar o AE e avaliar contra scripts disponíveis no jogo.
- ▶ 1º: Avaliar tamanho de cromossomo (AE vs *Script Default* - Balanceado)
- ▶ 2º: Avaliar taxa de mutação (AE vs *Script Default* - Balanceado)
- ▶ 3º: Avaliar contra os 5 *scripts* do jogo

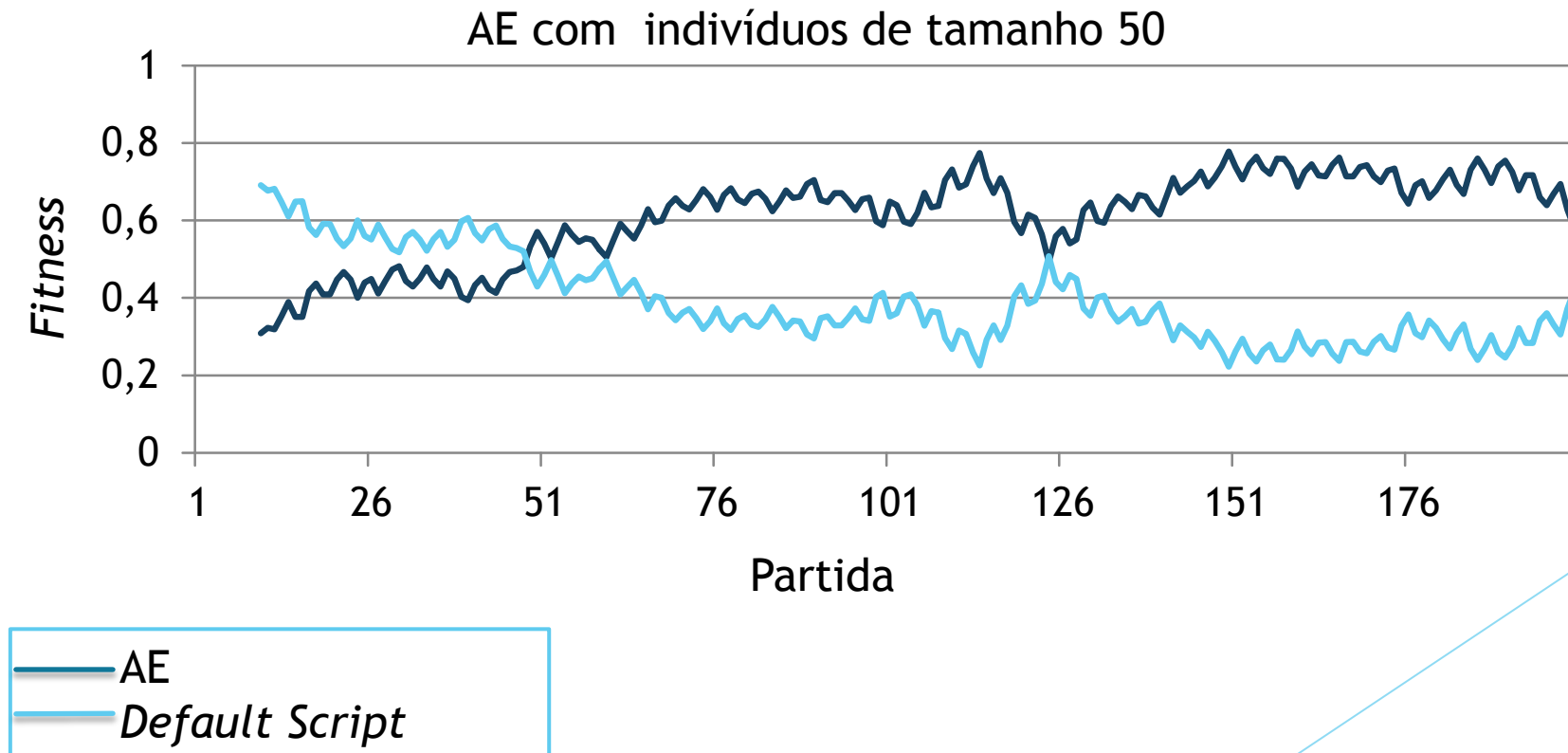
Resultados Computacionais

▶ Métricas

- ▶ Média Móvel: cada ponto nos gráficos representam a média dos 10 últimos valores
 - ▶ Ponto de Equilíbrio Médio (P.E.M.): número da primeira partida em que o NPC controlado pelo AE venceu X partidas consecutivas.
 - ▶ Porcentagem de vitórias
 - ▶ Para cada avaliação foram realizados 10 testes
 - ▶ Número de partidas por teste: 200
 - ▶ Tempo máximo adotado para cada partida: 5 min.
 - ▶ Empate = vitória do inimigo.
- ▶ AE evolui com uma população formada por 4 indivíduos.

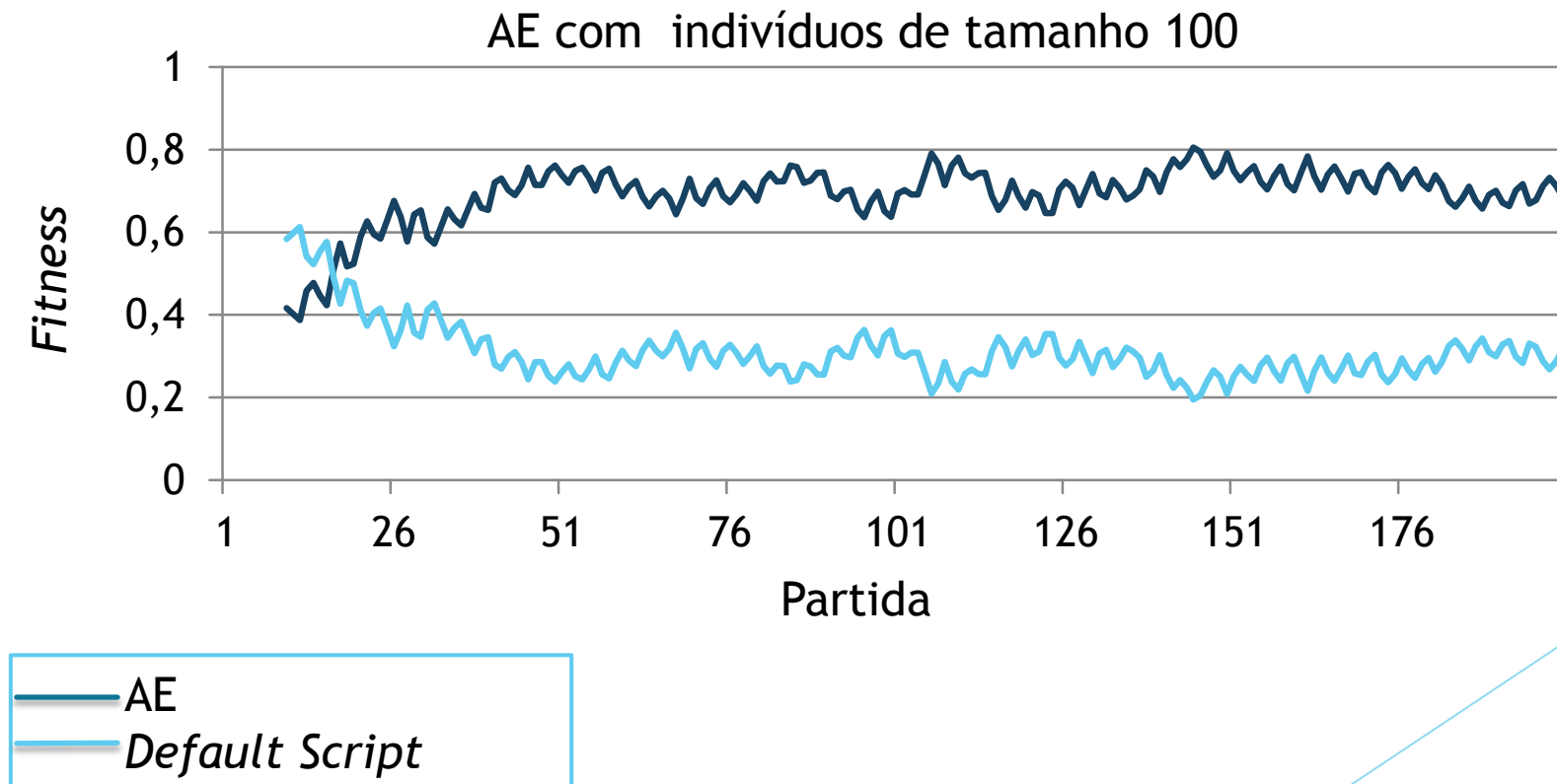
Resultados Computacionais

- ▶ Avaliando tamanho do indivíduo



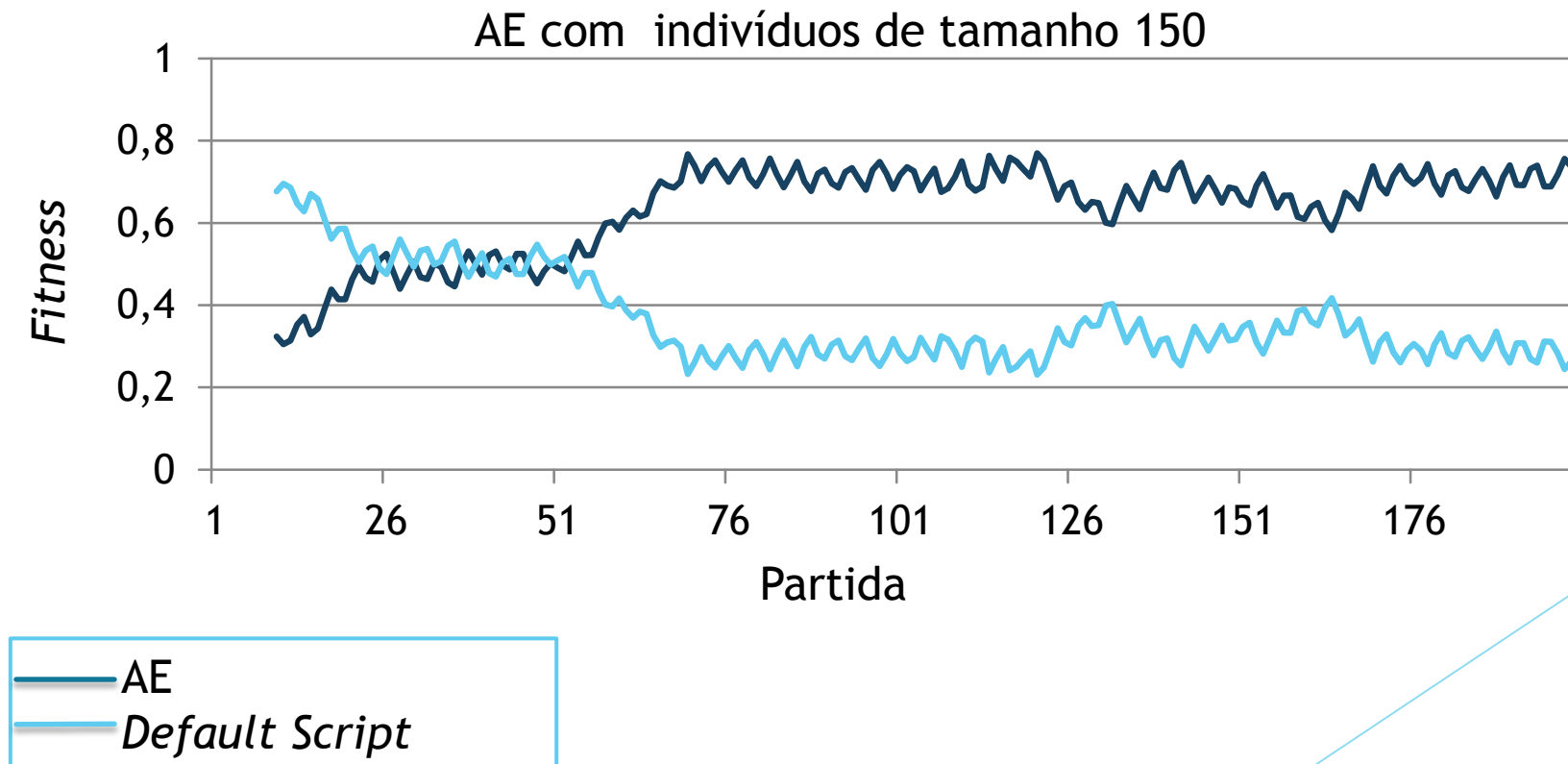
Resultados Computacionais

- ▶ Avaliando tamanho do indivíduo



Resultados Computacionais

- ▶ Avaliando tamanho do indivíduo



Resultados Computacionais

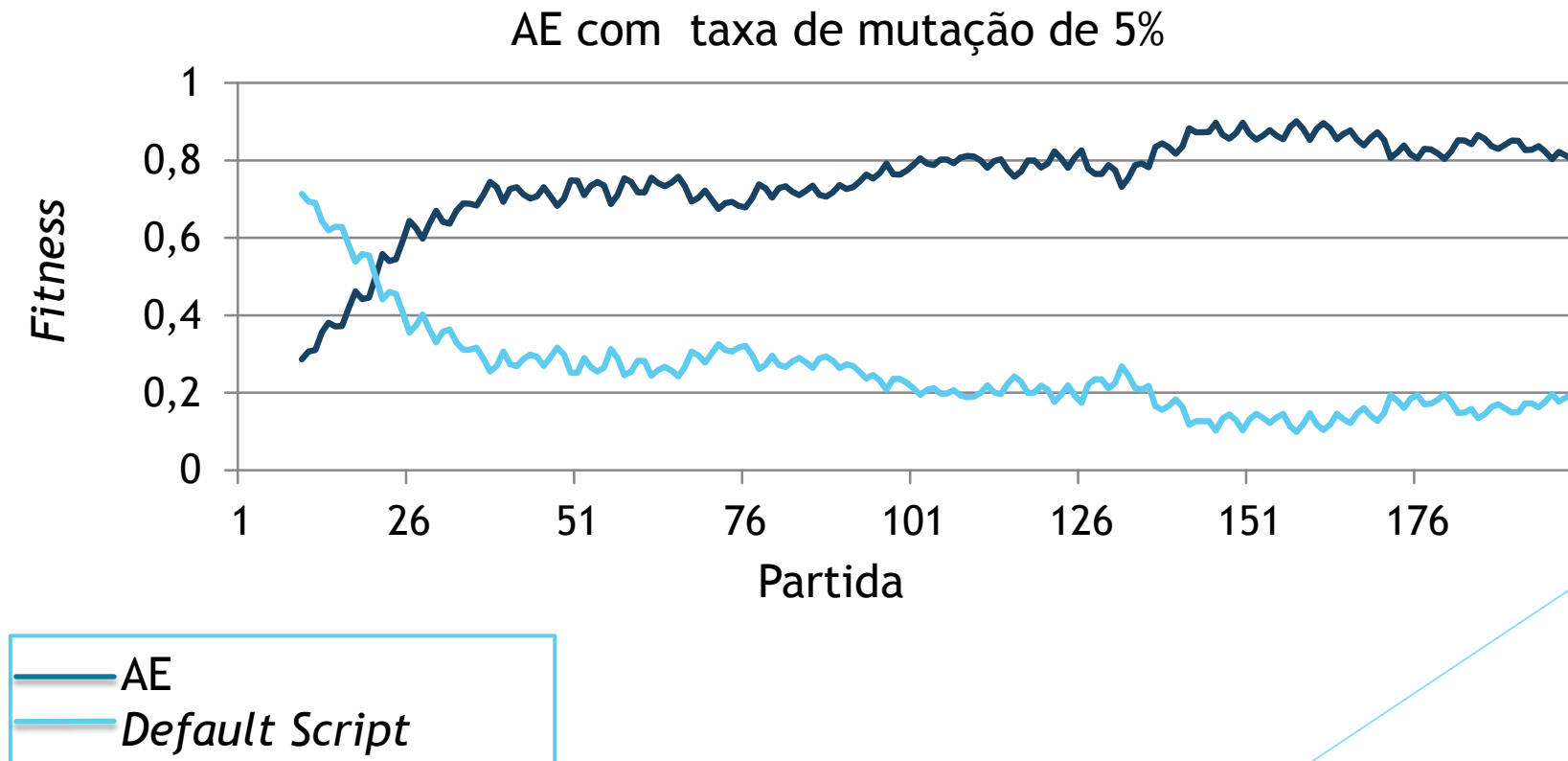
- ▶ Avaliando tamanho do indivíduo

Ponto de Equilíbrio Médio

TAMANHO	% VITÓRIAS	P.E.M. 5	P.E.M. 10
50	58,85	54	73
100	64,5	35	71
150	62,5	62	62

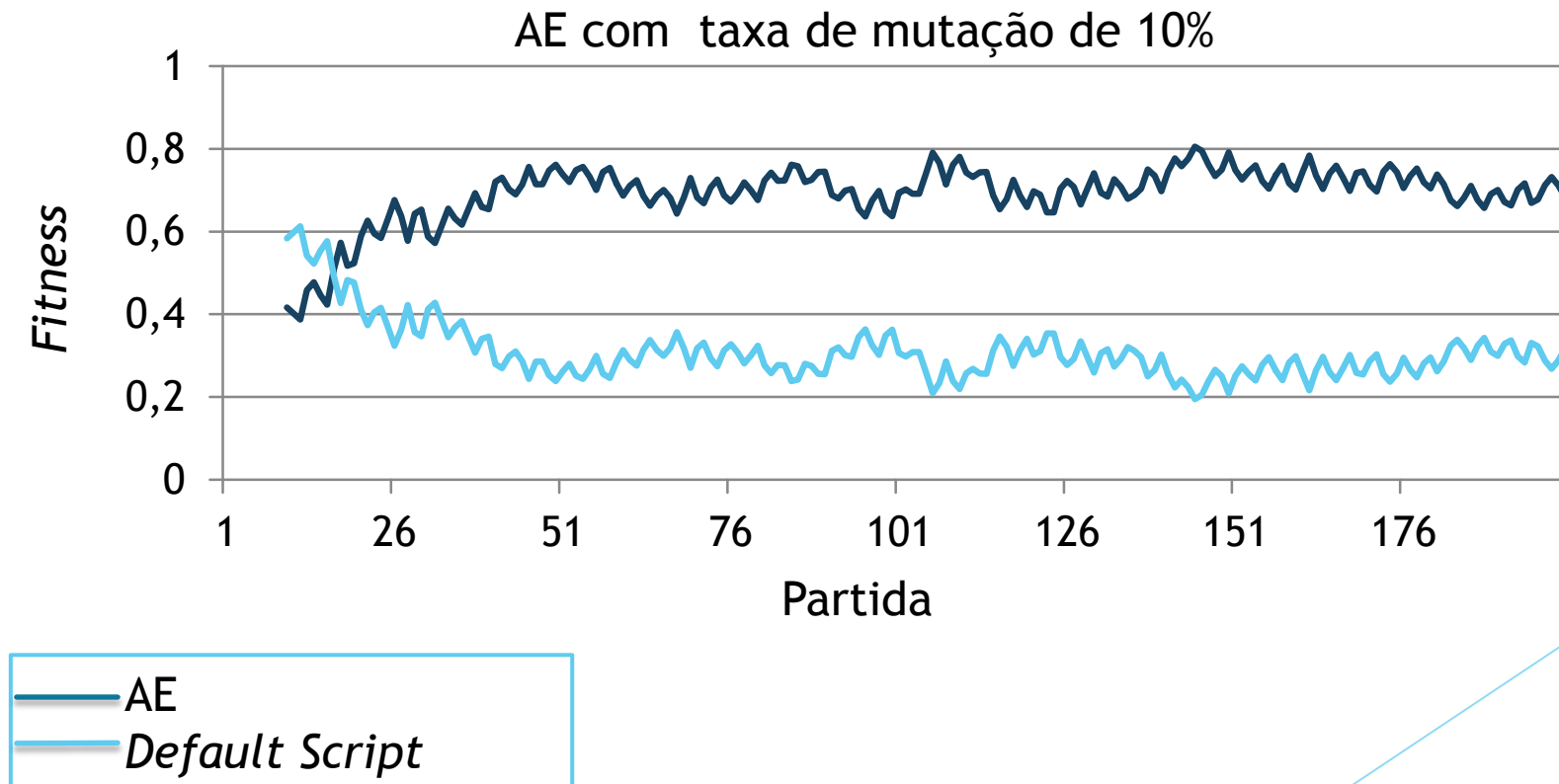
Resultados Computacionais

- ▶ Avaliando taxa de mutação



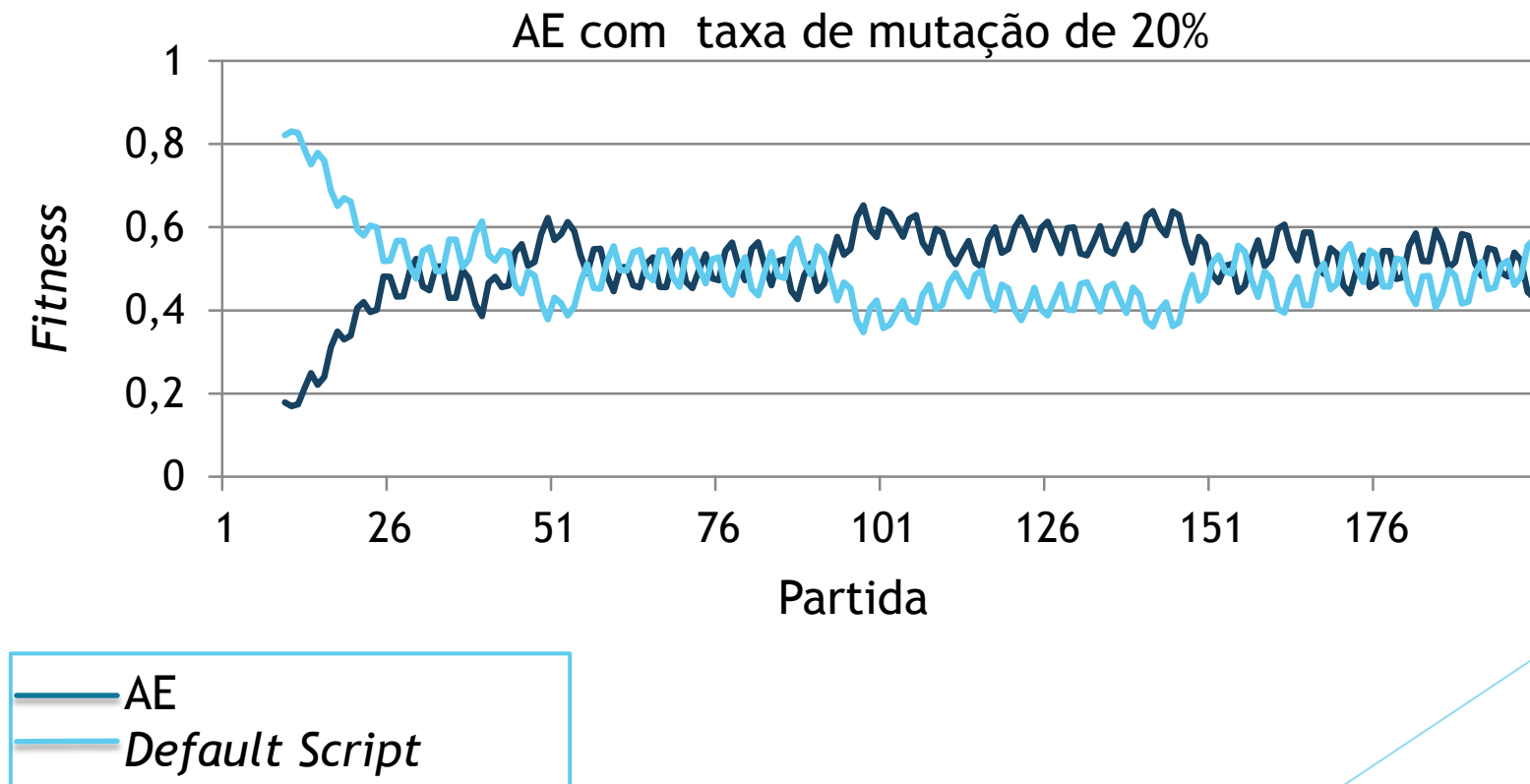
Resultados Computacionais

- ▶ Avaliando taxa de mutação



Resultados Computacionais

- ▶ Avaliando taxa de mutação



Resultados Computacionais

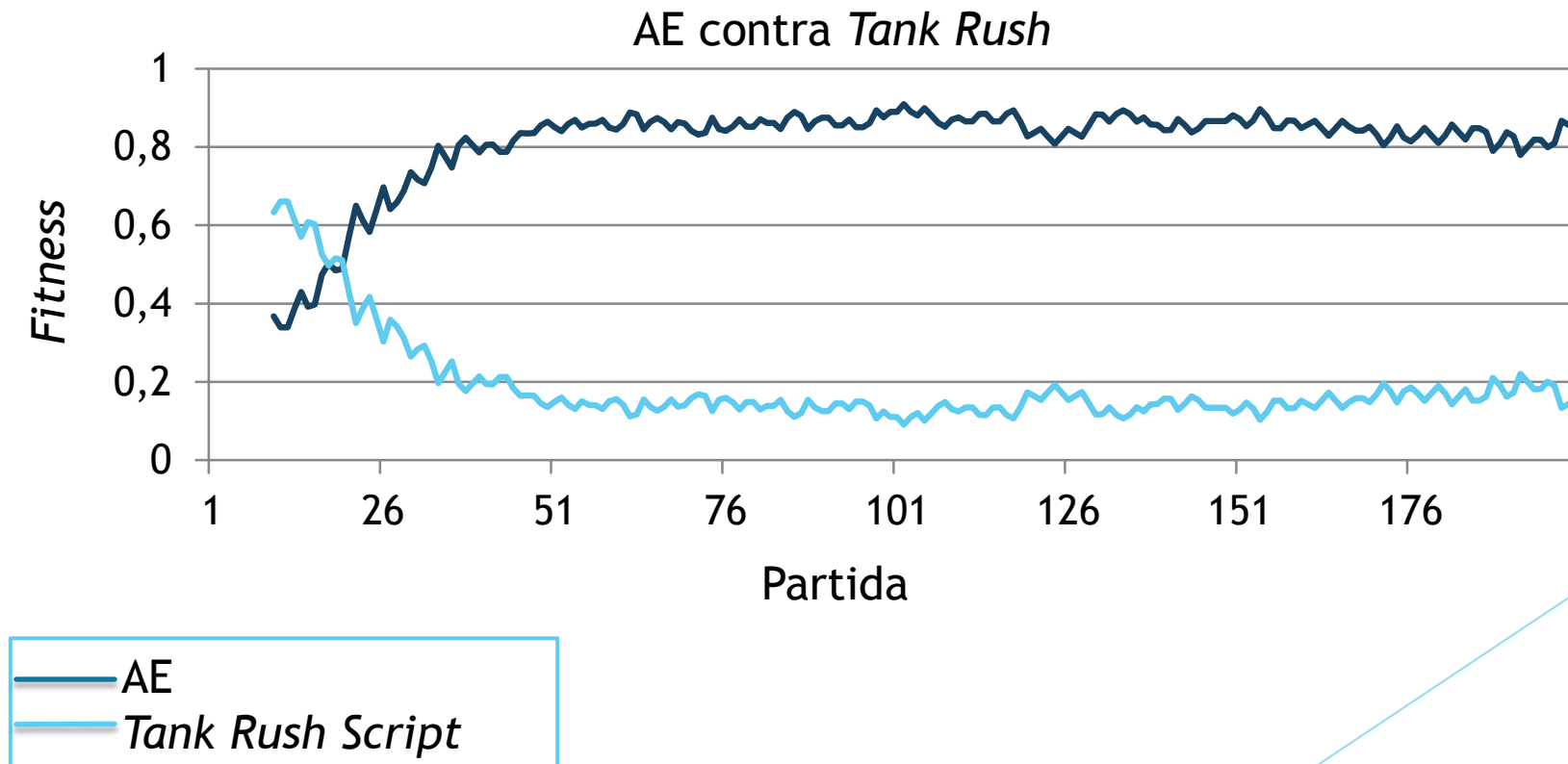
- ▶ Avaliando taxa de mutação

Ponto de Equilíbrio Médio

TAXA(%)	% VITÓRIAS	P.E.M. 5	P.E.M. 10
5	72	44	68
10	64,5	35	71
20	48	51	110

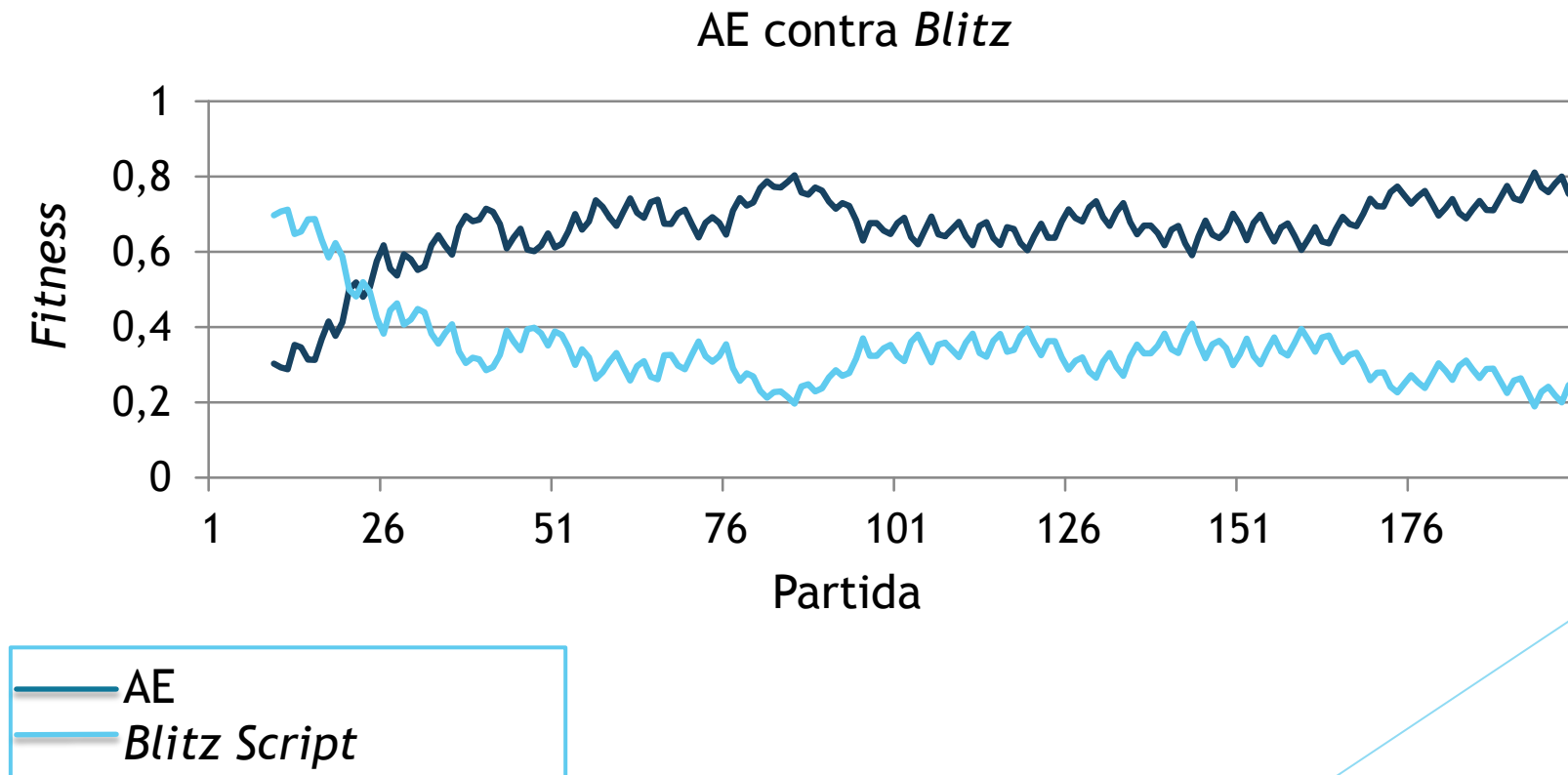
Resultados Computacionais

▶ AE contra *Scripts do BOS Wars*



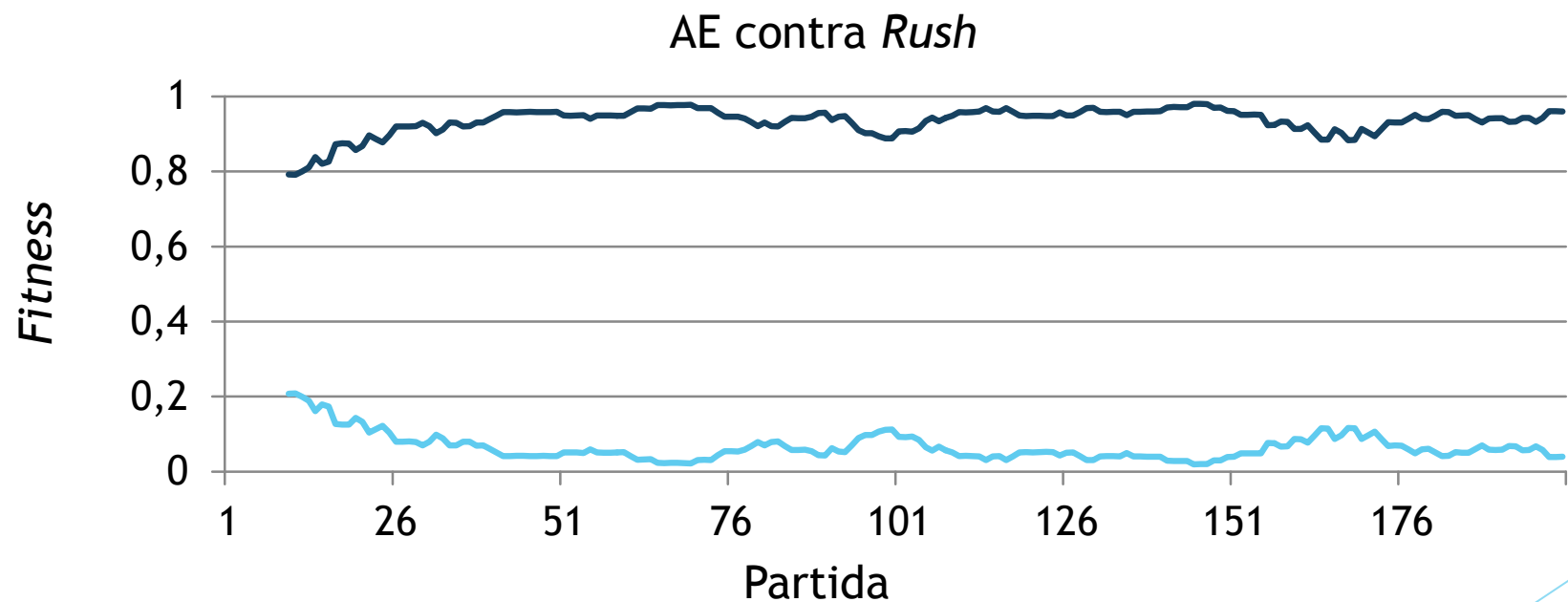
Resultados Computacionais

▶ AE contra *Scripts do BOS Wars*



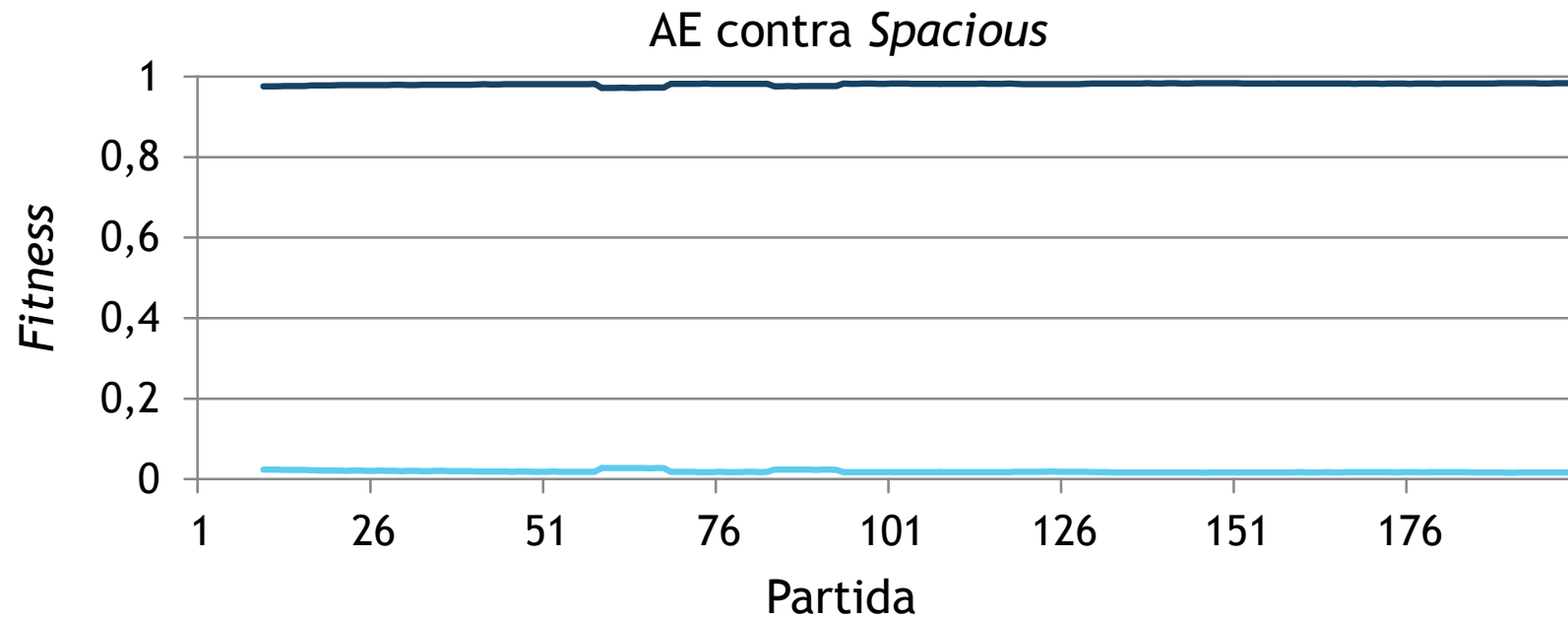
Resultados Computacionais

▶ AE contra *Scripts do BOS Wars*



Resultados Computacionais

▶ AE contra *Scripts do BOS Wars*



Resultados Computacionais

▶ AE contra *Scripts do BOS Wars*

Ponto de Equilíbrio Médio

<i>SCRIPT</i>	% VITÓRIAS	P.E.M. - 5	P.E.M. - 10
<i>DEFAULT</i>	72	44	68
<i>TANK RUSH</i>	81,1	26	42
<i>RUSH</i>	95	10	13
<i>SPACIOUS</i>	99,9	1	1
<i>BLITZ</i>	65,3	44	71

Conclusões

- ▶ AE obteve maior dificuldade em vencer inimigo que alterna as formas de ataque - *script Blitz*
- ▶ AE com menor dificuldade em vencer inimigo que ataca com um único tipo de unidade - *script Spacious*
- ▶ Em todos os casos o AE superou os *scripts* nativos de BOS Wars
- ▶ Trabalhos Futuros: Adaptação do AE para o jogo Wargus

Referências

- ▶ CROCOMO, M. K., 2008. Um Algoritmo Evolutivo para Aprendizado On-line em jogos Eletrônicos. Proceedings of SBGames 2008: Computing Track. Disponível em: http://www.sbgames.org/papers/sbgames08/computing/full/ct22_08.pdf
- ▶ PONSEN, M., SPRONCK, P., MUÑOZ-AVILA, H. AHA, D., 2007 *Knowledge Acquisition for Adaptive Game AI*. Science of Computer Programming, v.4, n.1, p. 59-75.

Agradecimentos

