

Uma avaliação do método de Algoritmos Genéticos para Angry Birds

Raul Freire Aguiar

Universidade Federal do ABC, Santo André, SP

f.raul@ufabc.edu.br

1 Introdução

Angry Birds está entre um dos jogos mais populares ao redor do mundo, faz parte de uma série desenvolvida pela empresa finlandesa Rovio Entertainment. O jogo consiste em realizar lançamentos de aves utilizando um estilingue afim de destruir estruturas e matar porcos. Quanto menos aves utilizar e mais estruturas de objetos destruir, maior é a pontuação do jogador [Ferreira et al., 2013].

Jogadores humanos, compreendem facilmente como jogar o jogo, pois só precisam de um pouco de raciocínio visual e intuição para escolher bons tiros, entretanto um agente inteligente não carrega esta intuição e faz uso de técnicas de visão computacional para identificar as estruturas de objetos presentes em seu campo visual.

A construção de um programa de computador capaz de jogar o angry birds assim como os seres humanos é uma tarefa bastante desafiadora. Isto é em parte, consequência do grande espaço de ações do jogo, as aves podem ser atiradas em diferentes ângulos e o resultado de cada lançamento é desconhecido antes de cada tiro, sendo assim necessário realmente executar ou simular. A Angry Birds AI Competition (ABAIC) é um programa de inovação criado com a finalidade de premiar competidores que conseguiram desenvolver um programa de computador capaz de jogar com sucesso Angry Birds. Na página do evento encontra-se disponível um software básico que joga o jogo, website, <http://aibirds.org/>, neste incluem diversos módulos, entre eles o de visão computacional, módulo de planejamento de trajetória e interface de jogo compatível com a versão Chrome de Angry birds.

O objetivo deste artigo é apresentar como requisito parcial da disciplina de Inteligência Artificial, o desenvolvimento de um sistema computacional capaz de jogar Angry Birds utilizando métodos de Algoritmos Genéticos. A intenção é apresentar uma melhoria nos resultados em relação ao software básico disponibilizado na página da ABAIC.

2 Abordagem utilizada

O conjunto de testes utilizado foi o da jornada “Poached Eggs” que possui 63 fases e é compatível com o plugin Angry Birds Chrome (chrome.angrybirds.com). Cada fase é composta por um campo de jogo diferente, contendo

estruturas de objetos de madeira, gelo, rochas, colinas e bomba TNT, podendo conter um ou mais porcos. Para completar a fase é necessário destruir todos os porcos disparando um número limitado de aves. As aves são disparadas através do estilingue e possuem diferentes cores e tamanhos, indicando características específicas para cada um, são eles: *RedBird*, *YellowBird*, *BlueBird*, *WhiteBird* e *BlackBird* [Narayan-Chen et al., 2013].

A abordagem para avaliação foi superar o maior número de fases dado um critério de no máximo 40 repetições. O foco escolhido não foi maximizar os pontos e sim fazer o reconhecimento do maior número de ambientes, aprender com o novo campo desconhecido e ir mais longe que o agente básico.

Os resultados alcançados descritos neste artigo não foram gerados através de várias execuções do algoritmo afim de obter o melhor resultado em cada fase. O algoritmo rodou da primeira à vigésima primeira fase sem interrupções em cada etapa. A cada execução do programa os *scores* e o número de reinícios podem sofrer alterações, pois algumas jogadas são baseadas na aleatoriedade.

3 Critérios estabelecidos

Foi observado logo de início, que apenas alterando as chamadas de método de visão para `RealShape()` já representavam melhorias em relação ao agente básico, sendo assim foi acrescentado método para captura de objetos e detecção de estilingue em `RealShape()`. Obtivemos como referência a equipe DATALAB vencedora da competição em 2014.

Para este experimento, foi considerado o tempo de Tap para as aves brancas como um recurso separado, ou seja, contruímos um estimador simples que mantém o controle de Tap para determinar os melhores intervalos em aves brancas. Utilizamos como referência a equipe S-BIRDS AVENGERS, uma das finalistas na competição ABAIC em 2014.

3.1 Critério para seleção do alvo

Para seleção do alvo, alguns critérios foram estabelecidos afim de aumentar a performance e diminuir o tempo de aprendizagem do algoritmo Genético, a seguir:

- Se existir apenas um porco, capturar suas coordenadas e definir como alvo.
- Se existir mais de um porco, basear suas escolhas da seguinte forma:
 1. A primeira execução da fase será sempre aleatória, definindo como alvo um objeto aleatório em maior quantidade;
 2. caso houver reinicialização, a segunda execução da fase será parte aleatória, parte selecionando o indivíduo mais apto da população do algoritmo Genético;
 3. a cada três reinicializações todos os disparos serão feitos capturando indivíduos mais aptos da população e;
 4. A cada cinco reinicializações, basear todos os alvos em objetos aleatórios que estão em maior quantidade.

Os critérios acima foram definidos para que o ambiente fosse explorado e coordenadas com melhor *score* fossem descobertos. Como referência para definição destes critérios, utilizou-se um pouco dos conceitos de aprendizagem por reforço, pois quando o ambiente é desconhecido, uma boa estratégia é basear-se em movimentos que maximizem seus valores e movimentos puramente aleatórios afim de explorar caminhos ainda não explorados.

4 Implementação

O agente foi implementado utilizando a versão 1.32 do framework fornecido pela Universidade da Austrália (ANU) para a competição ABAIC, disponível em <http://aibirds.org/>.

A proposta foi implementar o método de algoritmos Genéticos para Angry Birds e observar se o método apresenta melhoras em relação ao agente básico (Burro).

Para definição de um indivíduo, utilizou-se os dados das estruturas de objetos presentes no campo de visão do agente, a cada criação de novo indivíduo é escolhido um objeto aleatoriamente para representá-lo e um *score* de 4300 estático para todos eles. Criou-se também método para inserir um indivíduo definido, com objeto e *score* definidos.

No método de *fitness*, a aptidão do indivíduo é representado por seu *score*, e acaso o objeto for um porco é alimentado em seu *score* mais 50 e acaso for um TNT é acrescido 70, dando com isso mais aptidão a estes tipos de objetos.

No método de normalização linear os indivíduos são ordenados de acordo com a função de *fitness*.

Para o método de *crossover* e afim de mutar os indivíduos, foi escolhido os atributos das coordenadas x e y, e o ângulo do objeto, com intuito de criar novos indivíduos através da combinação de dois melhores.

A taxa de mutação foi fixada em 2 por cento, tamanho da população em 300 indivíduos, com elitismo, e número máximo de gerações em 25.

5 Resultados

Os resultados foram otimistas ao que era esperado e serão descritos a seguir:

O agente descrito neste artigo, tinha o objetivo de superar o agente básico (Burro) no número de fases que era capaz de passar dado um critério de no máximo 40 reinicializações.

O agente básico falhou em passar do level 4 da segunda etapa, totalizando 24 fases conquistadas, enquanto que o nosso agente passou de todas as fases exceto as fases 15 e 20 da terceira etapa da jornada “*Poached Eggs*”, totalizando 55 fases conquistadas até a primeira falha e 60 fases completadas no total.

Tabela 1: scores encontrados pelo agente inteligente referente à primeira etapa (1/3) de fases 1-21, formatado como: fase (número de reinicializações) e score.

1 (0)	28550	8 (0)	24790	15 (0)	36650
2 (0)	41750	9 (1)	41560	16 (1)	46290
3 (0)	41790	10 (0)	50800	17 (1)	47830
4 (0)	28710	11 (0)	52920	18 (0)	43410
5 (0)	53490	12 (0)	50930	19 (0)	39380
6 (0)	24140	13 (3)	47820	20 (1)	59430
7 (1)	25190	14 (2)	68760	21 (2)	66960

Total Score: 921150

Tabela 2: scores encontrados pelo agente básico referente à primeira etapa (1/3) de fases 1-21, formatado como: fase (número de reinicializações) e score.

1 (0)	28960	8 (0)	39150	15 (1)	33100
2 (0)	34560	9 (1)	35100	16 (0)	65370
3 (0)	40370	10 (3)	34610	17 (1)	54570
4 (0)	10640	11 (0)	39260	18 (2)	43580
5 (1)	56140	12 (0)	52640	19 (0)	29270
6 (1)	24800	13 (1)	23550	20 (2)	36130
7 (0)	29000	14 (1)	53790	21 (9)	60990

Total Score: 825580



Figura 1: Gráfico comparativo entre os agentes correspondente à primeira etapa de fases 1/3.

A Tabela 1 apresenta os *scores* de 1/3 das fases da jornada *Poached Eggs* executadas pelo nosso agente, a Tabela 2 representa os *scores* das mesmas fases da Tabela 1 porém, executadas pelo agente básico. Observamos uma melhora no *score* total de 95570 pontos, o número de reiniciações também apresentou em menor quantidade, uma diferença de 11 reinicializações no total.

Tabela 3: *scores* encontrados pelo agente inteligente referente à segunda etapa (2/3) de fases 1-21, formatado como: fase (número de reinícios) e *score*.

1 (1)	47230	8 (0)	40950	15 (13)	35470
2 (0)	45160	9 (0)	42330	16 (7)	56610
3 (2)	95170	10 (1)	28320	17 (9)	30750
4 (0)	52570	11 (3)	93820	18 (2)	52960
5 (0)	66220	12 (6)	45580	19 (0)	41170
6 (1)	57770	13 (0)	62930	20 (10)	27720
7 (1)	44370	14 (0)	49080	21 (36)	60110

Total Score: 1076290

Tabela 4: *scores* encontrados pelo agente básico referente à segunda etapa (2/3) de fases 1-21, formatado como: fase (número de reinicializações) e *score*.

1 (1)	48350	8 (1)	52080	15 (40)	0
2 (0)	45280	9 (5)	30940	16 (40)	0
3 (4)	105850	10 (2)	38250	17 (11)	29230
4 (40)	0	11 (12)	71290	18 (7)	50390
5 (0)	67850	12 (40)	0	19 (11)	39350
6 (40)	0	13 (0)	67050	20 (13)	53270
7 (5)	40280	14 (3)	38740	21 (40)	0

Total Score: 778200



Figura 2: Gráfico comparativo entre os agentes correspondente à segunda página de fases 2/3.

Observa-se na Tabela 4, que a complexidade aumenta a medida que vão passando as fases, e para a segunda etapa o agente básico demonstrou bastante dificuldade, e

considerado impossibilitado de passar do level quatro. Consideramos esta fase o limite conquistado pelo agente básico, mas pulamos afim de observar o comportamento do agente nas demais fases.

A Tabela 3 apresenta os *scores* obtidos pelo agente inteligente, novamente vê-se uma melhora nos *scores*, totalizando uma diferença de 298090 pontos, o número de reinícios mostrou uma diferença de 287 no total.

As Figuras 1 e 2, ilustram os *scores* obtidos pelos agentes em cada fase e observa-se que a medida que cresce a complexidade o agente inteligente se destaca em relação ao agente básico.

A Tabela 6 apresenta os *scores* obtidos pela última etapa de fases, percebemos que o número de reinicializações em cada level aumentou devido ao aumento da complexidade das fases, vale ressaltar também que a fase 3 da terceira etapa, nenhum dos agentes foi capaz de detectar o estilingue, sendo assim, esta foi desconsiderada de nossa análise.

Ocultamos as ilustrações do agente básico referente à última etapa de fases devido o agente não ter demonstrado um bom resultado na etapa dois, o agente demonstrou extrema dificuldade em matar todos os porcos devido ao aumento de estruturas como colinas que impossibilitam a contagem de pontos ao agente que sempre considera como alvo um porco aleatório.

Tabela 5: *scores* encontrados referente à terceira etapa (3/3) de fases 1-21, formatado como: fase (número de reinícios) *score*.

1 (0)	49810	8 (1)	81750	15 (40)	0
2 (0)	45420	9 (5)	58410	16 (26)	61430
3 --	--	10 (26)	44610	17 (28)	69480
4 (4)	31620	11 (4)	57230	18 (10)	96520
5 (1)	106690	12 (6)	47690	19 (3)	60470
6 (25)	64490	13 (6)	58020	20 (40)	0
7 (14)	58560	14 (3)	40480	21 (33)	111530

Total Score: 1212470

4 Conclusão

Reafirmamos a dificuldade em desenvolver sistemas computacionais capazes de jogar Angry Birds assim como os seres humanos porém, com os avanços da Inteligência Artificial percebemos que é possível. Este artigo demonstrou como uma ideia simplista utilizando métodos de Inteligência Artificial pode apresentar uma melhora em relação a um agente básico. Para quem demonstra interesse ou curiosidade, resultados realmente surpreendentes podem ser vistos na página da competição ABIAC.

Concluimos que a utilização de métodos de algoritmos Genéticos podem demonstrar uma melhora comparado a um agente básico, o algoritmo aprende a medida que joga o

jogo, entretanto vale lembrar que outras técnicas de Inteligência Artificial são capazes de diminuir esse período de aprendizagem sendo assim mais eficientes.

Referências

- [Ferreira et al., 2013] Leonardo A. Ferreira, Guilherme A. W. Lopes and Paulo E. Santos. Combining Qualitative Spatial Representation Utility Function and Decision Making under Uncertainty on the Angry Birds Domain. In: *International Joint Conference on Artificial Intelligence*. 2013.
- [Narayan-Chen et al., 2013] Anjali Narayan-Chen, Liqi Xu and Jude Shavlik. An Empirical Evaluation of Machine Learning Approaches for Angry Birds. In: *International Joint Conference on Artificial Intelligence*. 2013.