

Algoritmos Genéticos

Luigi Muller Sousa Linhares
Departamento de Ciência da
Computação
Universidade Federal de Roraima
Boa Vista, Roraima, Brasil
sluigimuller@gmail.com

Tarlison Sander Lima Brito
Departamento de Ciência da
Computação
Universidade Federal de Roraima
Boa Vista, Roraima, Brasil
britotarlison@gmail.com

Abstract—Esse relatório tem uma análise do artigo “Simulating nature's methods of evolving the best design solution” de Cezary Janikow e Daniel Clair, um resumo sobre o vídeo da implementação de Ivan Seidel de um algoritmo genético para jogar com o dinossauro do Google Chrome, e por último uma implementação de algoritmo genético e rede neural para jogar Flappy Bird.

Link do repositório desse trabalho:
https://github.com/luigim1998/LuigiTarlison_finalProject_AA_RR_2020

Keywords—Genetic Algorithms, Game, Neural Network

I. INTRODUÇÃO

Neste trabalho serão analisados dois trabalhos. O primeiro é “Simulating nature's methods of evolving the best design solution” de Cezary Janikow e Daniel Clair, que descreve algoritmos genéticos, abordando as estratégias para utilização desse método, incluindo como representar os dados, as operações que podem ser feitas para buscar resultados melhores e as desvantagens dessa abordagem. O segundo é um resumo do vídeo “Inteligência Artificial com Dinossauro da Google” do canal do YouTube do Ivan Seidel que aborda o assunto e fala sobre sua implementação de algoritmo genético e redes neurais para que ele jogue como o dinossauro no jogo disponível no navegador Chrome.

Por último será descrito uma implementação modificada do jogo Flappy Bird que utiliza redes neurais e algoritmos genéticos para busca de pássaros com resultados ótimos. Será abordado as modificações feitas no jogo, a rede neural usada, quais dados são usados para a interação com o ambiente e as técnicas utilizadas para otimização.

II. ARTIGO DO CEZARY JANIKOW E DANIEL CLAIR

Os problemas computacionais estão sendo resolvidos cada vez mais com agentes dinâmicos interagindo no meio tentando reproduzir como os organismos interagem no ambiente, ou seja, competindo e cooperando pelos recursos. Os algoritmos genéticos é uma das abordagens que possuem mais sucesso nesse quesito.

Os algoritmos genéticos simulam agentes individuais que são organizados em populações onde tentam sobreviver e cooperar para que cheguem numa melhor adaptação. Os agentes são chamados de cromossomos, esses cromossomos são compostos de genes e o significado para o cromossomo é definido pelo usuário de modo que ele represente uma possível solução para o problema.

Utilizando de seleção natural e herança de informações os algoritmos genéticos conseguem prover um comportamento de adaptação. A seleção é um processo estocástico onde a chance de sobrevivência é proporcional ao nível de adaptação do agente, adaptação essa que é dada

pela avaliação do fenótipo no ambiente do problema. A cooperação é alcançada através da junção de informações, usualmente, entre dois agentes que é dada o nome de crossover. A mutação ocorre quando é introduzido uma informação diferente no agente mudando sua estrutura inicial. Algoritmos que usam esses mecanismos são robustos, pois, conseguem equilibrar eficiência e eficácia.

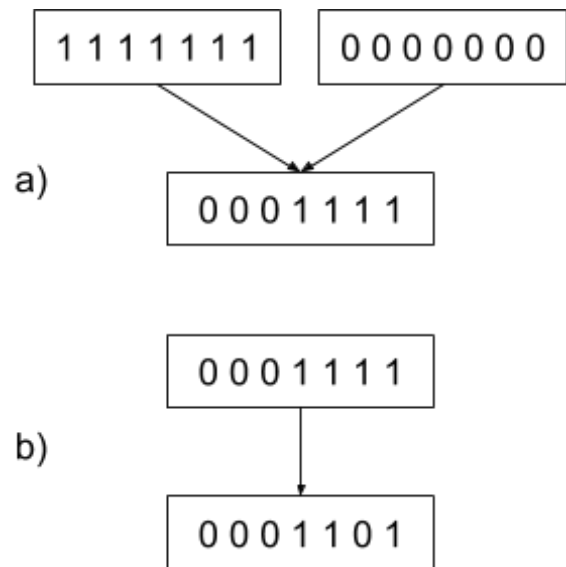


Fig. 1. A figura a) mostra um exemplo de crossover entre dois cromossomos que geram um descendente. A figura b) mostra um exemplo de mutação em que o sexto gene muda de 1 para 0.

Os algoritmos genéticos não são buscas aleatórias, eles exploram regularidades nas informações que aquele cromossomo representa.

Alguns cromossomos podem apresentar níveis de adaptação muito parecidos se eles representarem espécies similares, mas podem ter uma avaliação muito diferente um do outro e para explorar as semelhanças entre os cromossomos schematas são usadas que são ilustrações usadas para estudar o que está acontecendo com o cromossomo.

Um dos problemas maiores nos algoritmos genéticos são as restrições que a solução deve satisfazer, independente do domínio, são elas que causam os problemas mais sérios. Como, por exemplo, fazer com que a pesquisa pelo melhor fenótipo se deslize para regiões impróprias trazendo resultados errados. A maneira mais comum de lidar com esses problemas é penalizando os cromossomos que não cumprir com o objetivo, mas essa abordagem só funciona bem em algoritmos genéticos com restrições simples. Outra abordagem é ter rotinas de acompanhamento para todos os operadores. Isso “corrige” os cromossomos gerados, trazendo-os para o espaço correto, porém, para muitas

aplicações, essas rotinas não são triviais e introduzem complexidade computacional adicional.

III. ALGORITMO GENÉTICO

Algoritmos genéticos (AG) são métodos de busca que se baseiam nos mecanismos de seleção natural e da genética para encontrar resultados ótimos para um determinado problema.

As técnicas de busca tradicionais operam sobre um único candidato que, iterativamente, é manipulado seguindo heurísticas associadas ao problema.

Porém, AG utiliza estratégias de busca paralela para uma busca mais ampla no espaço de solução utilizando populações. Cada cromossomo nessa população terá um significado (fenótipo) que será dado pelo usuário de modo que cada um represente uma solução potencial.

O fenótipo será utilizado para fazer a avaliação do cromossomo no ambiente do problema, com a pressão seletiva, os cromossomos com melhores desempenhos são escolhidos para ‘sobreviverem’ e criarem descendentes. Através de cooperação, os sobreviventes podem gerar descendentes através da junção de seus cromossomos (crossover). Mutações adicionais podem acontecer. Através da iteração desses passos, é possível que atinja o resultado desejado.

Um aplicação de AGs está na otimização de extração de óleo no reservatório no campo Cymric, no Vale de San Joaquin. O ciclo de extração consiste em:

- Injeção de vapor na rocha de diatomita (de alta porosidade), durante 3 a 4 dias;
- Por alguns dias o vapor penetra na rocha e o óleo vai para a fratura gerada pelo vapor;
- Após isso, o óleo pode ser extraído, durante 20 a 25 dias;

Todo o processo pode demorar de 26 a 30 dias. Devido ao período que não há extração, há um incentivo para reduzir o período de vaporização, mas é notado que a produção é mais alta após a vaporização e decai rapidamente, subentendendo que é melhor aumentar a frequência de vaporização, logo é possível usar AG para descobrir um ciclo ideal para maximizar a produção

IV. ALGORITMO DO DINO

A ideia é simples, criar um programa que, sem informação prévia do que é certo ou errado, capaz de “aprender” a jogar e marcar a maior quantidade de pontos possível no jogo do dinossauro no navegador google chrome. O jogo consiste em desviar dos cactos e dos pássaros que vão em direção ao dinossauro, seja pulando ou abaixando. Para isso, será utilizado redes neurais e algoritmos genéticos.

Então, como o dinossauro irá “aprender” o que deve fazer? que informações utilizar para que seja possível a realização do projeto? Com sensores e atuadores, os sensores irão passar informações como distância, tamanho e velocidade. A distância que o dinossauro está do cacto (ou pássaro), o tamanho do cacto e a que velocidade o cacto está indo em direção do dinossauro. Já os atuadores serão as saídas da rede, a partir das informações dos sensores a rede

terá que decidir se o dinossauro deverá pular (tecla upArrow) ou abaixar-se (tecla downArrow).

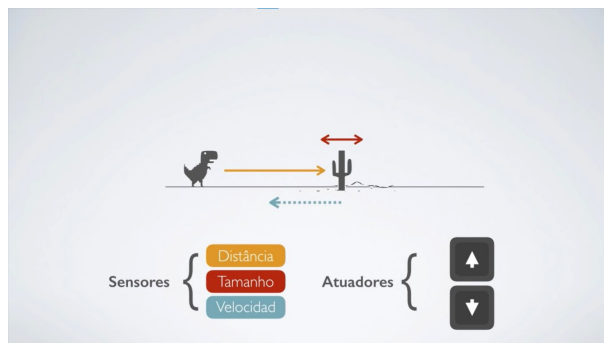


Fig. 2. Representação dos sensores e atuadores que o dinossauro deve ter para interagir com o ambiente.

Uma forma de fazer este programa é mapeamento entradas para as saídas, uma coisa que a rede neural executará bem de uma forma que crie uma função (função matemática) que generalize tudo o que acontece no meio entre o mapeamento da entrada para a saída.

Em uma rede neural cada camada possui uma quantidade de neurônios pré-determinada onde a entrada para cada um deles é multiplicada por um peso, peso este que auxilia para que a rede consiga chegar no resultado final.

No nosso caso, a partir dos valores que chegarem a camada final será decidido se será pressionado a tecla para pular ou baixar. Se o valor for maior do que 0.55 será para pular e caso menos do que 0.45 será para abaixar e valores intermediários entre isso irá fazer nada.

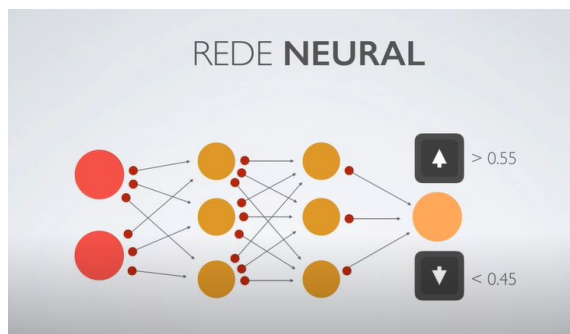


Fig. 3. Representação da rede neural utilizada na implementação de Ivan Seidel.

Agora considerando que todos os pesos e colocando eles em sequência, transformando ele em uma sequência linear, como se fosse um DNA (uma sequência de informações) onde cada informação também possui uma informação dentro dela. Desta forma temos uma configuração de um algoritmo genético.

Para fazer funcionar, será preciso pegar um DNA inicial e criar N-cópias com valores aleatórios em cada gene, pois, a intenção aqui é que ele aprenda sozinho do zero. Cada DNA com os seus respectivos genes será chamado de Genoma.

Os genomas são aplicados no programa em tempo real, a cada entrada dos sensores do programa os genomas irão gerar uma respectiva saída. Assim podemos verificar qual se saiu melhor, no caso aqui abordado é o genoma capaz de pular mais cactos. Assim, conseguimos fazer uma ‘seleção

natural' selecionando sempre os melhores de cada geração (cada iteração do programa) e a partir destes melhores selecionados é feito o crossover que é juntar os genes em outros genomas assim criando novos. Além de fazer mutações em seus genes e modificando alguns genes aleatoriamente a partir de uma probabilidade de mutação pré-definida.

Assim utilizando o programa na prática ele irá aprender a como se portar e assim sempre melhorando utilizando as técnicas aqui abordada. No vídeo a rede do Ivan Seidel foi capaz de fazer até 2532 pontos no jogos, o que seria muito mais se algumas alterações na rede fossem feitas e mais tempo para a execução do programa também.

V. IMPLEMENTAÇÃO DO TRABALHO

A. Jogo Escolhido

O jogo implementado é baseado no Flappy Bird. A versão original consiste em um pássaro que o jogador controla sua altura através de cliques na tela que o fazem saltar. Ele deve passar entre pares de canos que possuem lacunas de tamanho igual e dispostas de forma aleatórias, um ponto é ganhado toda vez que é passado entre lacunas, esses pares avançam na frente do pássaro. O objetivo é ganhar o máximo de pontos possíveis. O jogador morre ao colidir nos canos ou chão, ou ultrapassar o limite da tela.

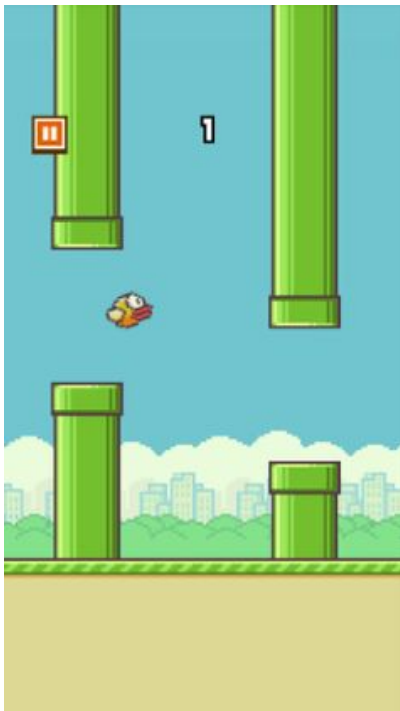


Fig. 4. Captura de tela da versão original do jogo.

A versão modificada veio com a necessidade de dificultar para o jogador, já que ele conseguia atingir o resultado esperado em pouco tempo. As mudanças feitas foram:

- Os pares de canos passaram a se moverem também verticalmente, mudando a altura da lacuna mas mantendo seu tamanho;
- Foi adicionado um par de canos especiais que não se movem verticalmente mas possui lacuna menor e que não pode ser ultrapassado saltando, para isso o

jogador precisa usar um paraquedas que mantém sua altura durante a passagem. O paraquedas tem um tempo para reutilização que coincide com o tempo de ir desse cano a outro especial, forçando o usuário a usar somente no tempo certo.

Cada pássaro é um indivíduo que consiste de uma rede neural.

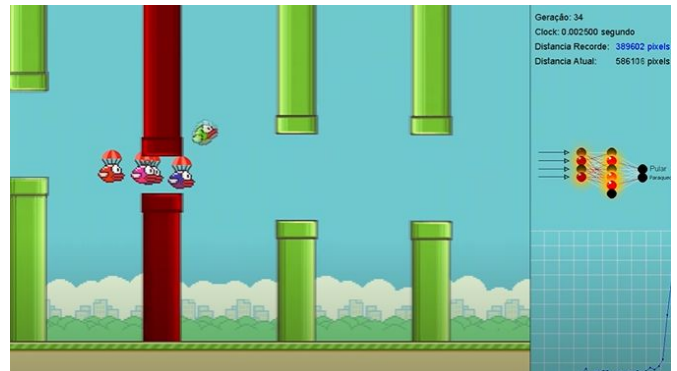


Fig. 5. Captura de tela da versão modificada do jogo

B. Rede Neural

A rede neural é uma rede densa, a camada de entrada possui um viés e quatro entradas que consistem em:

- Distância horizontal do pássaro ao cano mais próximo;
- Distância vertical do pássaro ao cano mais próximo;
- Velocidade do cano;
- Abertura dos canos.

Ela possui uma camada escondida com 6 neurônios mais um viés e a camada de saída com dois neurônios representando o comando de pular e utilizar o paraquedas.

Cada indivíduo tem sua rede neural iniciada com os pesos que são valores aleatórios entre -1000 a 1000. A função de ativação utilizada consiste na função RELU com a observação de que valores maiores que 10000 são convertidos em para o valor 10000. Cada peso na rede representa um gene do indivíduo.

O paraquedas e o salto são ativados quando o valor da saída da rede referente a eles for um número maior que zero.

C. Evolução e seleção

Na iteração responsável pela criação da nova geração, o método de seleção escolhe os cinco melhores indivíduos (que percorrem as maiores distâncias) e elimina os outros substituindo o DNA deles pelo DNA dos selecionados. Nessa implementação do Algoritmo Genético não existe cruzamento dos indivíduos, somente mutação dos genes dos indivíduos substituídos.

A quantidade de mutações que acontecem em cada indivíduos é aleatório e a mutação pode ser de três tipo:

- Um novo valor aleatório entre -1000 a 1000;
- Um novo valor igual ao peso anterior multiplicado por um valor entre 0,5 a 1,5;
- Um novo valor igual ao peso anterior mais um valor entre -10 a 10;

VI. COMPLEXIDADE DO ALGORITMO GENÉTICO

Considerando g o número de gerações, n o tamanho da população e m o tamanho do cromossomo, a complexidade para essa tarefa seria $O(gnm)$.

REFERENCES

- [1] CARVALHO, André Ponce de Leon F. de. **Algoritmos Genéticos**. Disponível em: <<https://sites.icmc.usp.br/andre/research/genetic/>>.
- [2] JANIKOW, Cezary Z.; CLAIR, Daniel St.. **Simulating nature's methods of evolving the best design solution**.
- [3] SIVANANDAM, S. N.; DEEPA S. N.. **Introduction to Genetic Algorithms**. Springer, 2008.
- [4] WIKIPEDIA. **Flappy Bird**. Disponível em: <https://en.wikipedia.org/wiki/Flappy_Bird>.
- [5] **Inteligência Artificial com Dinossauro da Google**. Disponível em: <<https://www.youtube.com/watch?v=P7XHzqZjXQs>>.
- [6] **Inteligência Artificial jogando Flappy Bird!!**. Disponível em: <https://www.youtube.com/watch?v=vavXvu_SMeM>.