Resolução do Klotski utilizando Métodos de Pesquisa em Python (Tema 1/ Grupo 31)

André Lopes dos Santos (200505634)
Departamento de Engenharia Informática
Faculdade de Engenharia da Universidade do Porto
Porto, Portugal
up200505634@fe.up.pt

Bernardo Oliveira Teixeira Santos (201504711)
Departamento de Engenharia Informática
Faculdade de Engenharia da Universidade do Porto
Porto, Portugal
up201504711@fe.up.pt

Miguel Rossi Seabra (200604224)
Departamento de Engenharia Informática
Faculdade de Engenharia da Universidade do Porto
Porto, Portugal
ei06054@fe.up.pt

***Abstract*—Abordagem a diversos métodos de pesquisa e seus algoritmos, que são uma componente importante da inteligência artificial, assim como à utilização dos mesmos na resolução do Klotski, um conhecido puzzle de deslizamento de blocos. Para a resolução deste puzzle, são aplicados diversos métodos de pesquisa que são adaptados a este puzzle de modo a resolver o mesmo. A linguagem utilizada para implementar estes métodos é Python e verificou-se que os métodos de Pesquisas Gulosa e Pesquisa A* foram aqueles que se mostraram mais eficazes na resolução destes puzzles.**

***Index Terms*—Inteligência Artificial, Pesquisa, Algoritmo A*, Klotski**

I. Introdução

Neste trabalho abordar-se-ão diversos métodos de pesquisa em Python aplicados à resolução de puzzles de blocos conhecidos pelo nome "Klotski", mas mais especificamente a iteração deste que está presente na Google Play Store [1] Discutir-se-á a história do jogo, as suas regras e a formulação formal do problema. Também falar-se-á dos diferentes métodos de pesquisa, da sua implementação em Python e a eficiência dos mesmos para o jogo em questão.

II. Descrição do Problema

Os puzzles Klotski são puzzles de movimento de blocos em que o objetivo é mover um bloco específico para um sítio específico, movendo todas as outras peças nesse processo. Na Figura 1 pode-se ver um exemplo de um puzzle.

A origem deste puzzle é ainda incerta, mas pensa-se que uma primeira iteração do mesmo tenha sido patenteada por Henry Walton em 1893, embora existam diversos países (Inglaterra, Japão) que reclamam a autoria do puzzle "Original", sendo que ainda hoje é desconhecida (ou não existe consenso) quanto à origem do mesmo. [2] A variante aqui estudada é a iteração presente na Google Play Store [1] que acrescenta regras como blocos que não se podem mover (i.e. paredes) e outros blocos azuis que se portam como paredes até que o bloco vermelho toque em todos eles, fazendo-os desaparecer. Na Figura 2 pode-se ver um exemplo de um puzzle que contém paredes (representadas por blocos castanhos) e os já mencionados blocos azuis.

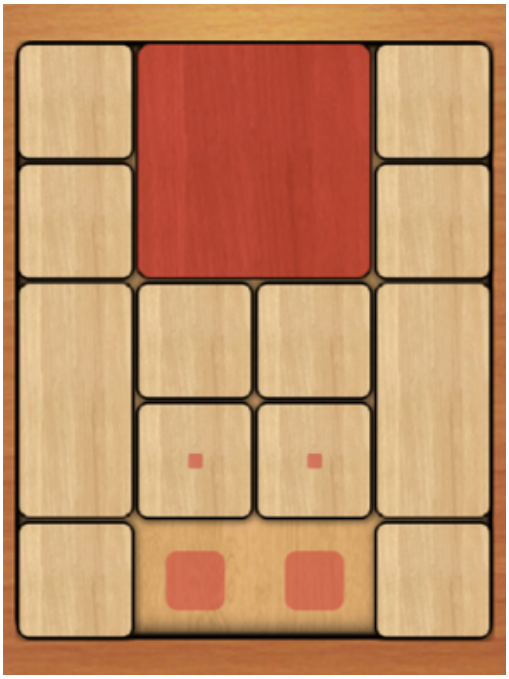

Fig. 1. Puzzle básico de Klotski, sendo o bloco vermelho aquele que tem de ir para um local específico (pontos vermelhos)

III. Formulação do Problema

Representação do Estado: O estado é representado por uma Matriz de tamanho variável (a * b) [0 – espaço livre], e pela posição das diferentes peças que são numeradas (1, 2, 3, etc.). Tanto as células do puzzle como os as diferentes peças, têm uma característica extra: n- normal; f-final. No caso das células estes representam as células normais do jogo enquanto as células finais representam os sítios onde se devem encontrar as peças finais. No caso das diferentes peças, as peças normais

Fig. 2. Puzzle de Klotski com paredes (representado com blocos castanhos) e com blocos azuis.

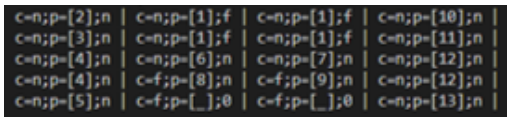

Fig. 3. Puzzle de Klotski mais simples: Representação no terminal

representam as peças móveis que se conhecem, e a peça final é aquela cujo objetivo é levar à(s) célula(s) final(ais).

Estados Iniciais: Os estados iniciais variam conforme o puzzle a ser resolvido. Na Figura 1 pode-se ver um exemplos de uma representação gráfica de um estado inicial de um puzzle. Na Figura 3 pode-se ver uma representação de um puzzle (mais simples) no terminal.

Teste Objetivo: A peça final deverá chegar à localização das células finais e chegar lá no número mínimo de jogadas possível.

Operadores/Jogadas: Mov(N,D) – Movimentação da peça N (sendo N um número acima de 1 mas igual ou inferior ao número mais alto),na direcção D (Cima, Baixo, Esquerda ou direita).

Pré-condições: Uma peça possa mover-se na direção desejada, i.e. ter espaço livre para todos os elementos da peça se poderem mover na direção desejada.

Efeitos: Mov(1, Cima) – A peça 1 move-se (se possível) uma casa para cima.

IV. Trabalho Relacionado

Foram encontradas diversas implementações em Python do puzzle Klotski, algumas com solvers já implementados [3] [4] [5] [6] [7]. A implementação do Klotski será baseada nestas mesmas fontes, sendo que serão realizadas diversas adaptações ao caso específico deste trabalho. Os métodos de pesquisa utilizados serão pesquisa em largura, pesquisa em profundidade, aprofundamento progressivo, custo uniforme, pesquisa gulosa e Algoritmo A*, sendo a implementação destes métodos baseados naqueles presentes no livro "Artificial Intelligence: A Modern Approach" (sendo este o livro principal da cadeira de Inteligência Artificial) [8], sendo que o código (em várias linguagens) para a implementação destes algoritmos está disponível online [9].

V. Implementação do Jogo

Como foi referido anteriormente, foram encontradas diversas implementações do jogo realizadas por diversas pessoas diferentes, no entanto a implementação do jogo que foi utilizada foi uma criada de raíz, sendo que as outras implementações encontradas serviram apenas como elemento de consulta quando necessário. O Programa utiliza ficheiros csv como meio para "receber" o estado do puzzle, seu tamanho, localização das diferentes peças e local final (onde deve ficar a peça objetivo). Tendo o estado do puzzle e o seu objetivo final, são aplicados os diferentes métodos de pesquisa para saber quais os passos necessários para chegar ao estado final no menor tempo possível.

Foram implementadas, entre outras, as seguintes funções: ler nível de ficheiro (função 'node_from_csv'), visualizar em modo de texto/gráfico um dado estado (métodos '_str_' que convertem os estados para representação String), validar uma dada jogada/operador (métodos 'move_up', 'move_down', 'move_right', 'move_left'), executar uma dada jogada/operador, num dado tabuleiro, tendo em conta os seus efeitos e gerando o respetivo estado sucessor (mesmos movimentos referidos anteriormente), listar todas as jogadas/operadores disponíveis num dado tabuleiro, avaliar um dado estado (idem), testar se um dado estado é solução (método 'terminal').

VI. Algoritmos de Pesquisa

Foram implementados os diferentes algoritmos de pesquisa para proceder à resolução deste puzzle. Os algoritmos implementados foram: Pesquisa Primeiro em Largura ("Breadth-First Search"), Pesquisa Primeiro em Profundidade ("Depth-First Search"), Pesquisa em Profundidade Iterativa, Pesquisa Gulosa ("Greedy Search") e Pesquisa A*.

É preciso dizer que a pesquisa de custo uniforme não foi feita porque o custo é igual em todos os movimentos (custo 1), o que levaria a este algoritmo ser igual à pesquisa em largura, e este custo constante leva também a que a pesquisa A* seja igual à gulosa, pois apenas é utilizada uma heurística e não o custo da ação.

Foram criadas duas heurísticas para o Algoritmo A*. A primeira heurística calcula a distância de Manhattan (soma das distâncias entre as linhas e as colunas) entre a peça vermelha e e a posição objetivo). A segunda heurística calcula o número de peças localizadas entre a peça vermelha e a posição objetivo. Como se pode comprovar pelos resultados

da tabela em baixo, a primeira heurística revelou-se bastante mais eficiente.

Para implementar estes diferentes algoritmos no nosso jogo, foi adaptado a lógica de jogo aos algoritmos do livro [8], sendo que o código desses mesmos algoritmos é o mesmo presente no GitHub do livro [9].

VII. Experiências e Resultados

Realizaram-se diversas experiências, comparando o desempenho dos diferentes algoritmos na resolução dum puzzle Klotski mais simplicado. A utilização deste mesmo puzzle mais simples deve-se ao facto de a resolução deste levar menos tempo e assim ser um possivel testar o código desenvolvido em tempo útil. Na Figura 3 pode-se encontrar uma representação deste mesmo puzzle, tal como este aparece no terminal e cujo ficheiro csv é "easy.csv". Na tabela seguinte pode-se ver a comparação dos diferentes métodos de resolução

TABLE I
Comparação dos diferentes métodos de pesquisa

Método de pesquisa	Tempo (segundos)	Nós explorados
Pesquisa Primeiro em Largura	2,10389	8340
Pesquisa Primeiro em Profundidade	?	?
Pesquisa com Aprofundamento Progressivo	0,48823	41593
Pesquisa Gulosa	0,01297	94
Pesquisa com algoritmo A* (h1)	0,01293	94
Pesquisa com algoritmo A* (h2)	86,33182	31459

A Pesquisa Primeiro em Profundidade estava a demorar um tempo superior a 10 minutos, razão pela qual os seus resultados aparecem com um ponto de interrogação ("?").

VIII. Conclusões e Perspetivas de Desenvolvimento

Como se pode verificar na secção anterior, e tal como seria teoricamente esperado, o método de pesquisa A* e o da Pesquisa Gulosa foram aqueles que mostram ser o mais eficazes na resolução do puzzle Klotski. Esta similitude entre estes dois métodos, como foi explicado anteriormente, deve-se ao facto do custo ser igual em todos os movimentos de jogo.

Infelizmente ainda não foi possível implementar certas regras de jogo que são específicas em relação ao que existe na Google Play Store, mas sem dúvida que a adição das regras específicas desta variante de Klotski adicionaria mais complexidade ao problema, fazendo com que estes mesmos métodos de pesquisa levassem mais tempo a resolver estes puzzles que um puzzle klotski tradicional.

Referências Bibliográficas

References

[1] "Klotski - Aplicações no Google Play," [Online]. Available: https://play.google.com/store/apps/details?id=com.alcamasoft.juegos.klotski.android. [Acedido em 15 03 2019].

[2] "Klotski - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Klotski. [Acedido em 06 03 2019].

[3] Goshuujin, "Github - Goshuujin/Klotski," [Online]. Available: https://github.com/Goshuujin/Klotski. [Acedido em 16 03 2019].

[4] SamuelDSR, "Github - SamuelDSR/Klotski-python," [Online]. Available: https://github.com/SamuelDSR/Klotski-python. [Acedido em 16 03 2019].

[5] aschmied, "Github - aschmied/klotski-solver," [Online]. Available: https://github.com/aschmied/klotski-solver. [Acedido em 16 03 2019].

[6] jwmullally, "Github - jwmullally/klotski_solver," [Online]. Available: https://github.com/jwmullally/klotski_solver. [Acedido em 16 03 2019].

[7] D. R. Ying Dang, "Github," [Online]. Available: https://github.com/whydang/klotski.

[8] S. Russel e P. Norvig, Artificial Intelligence: A Modern Approach, Pearson Education Inc., 2010.

[9] S. Russel e P. Norvig, "AimaCode - Code for the Book Artificial Inteligence: A Modern Approach"," 2019. [Online]. Available: https://github.com/aimacode. [Acedido em Março 2019].