

Artificial intelligence - Project 3

- AllOut -

George Botis
Daria Francioli

20/01/2023

1 Introducere

Scopul acestui proiect este acela de a implementa:

1. un program care rezolvă diferite lvl pentru jocul AllOut.
2. un program care încearcă să rezolve puzzle-ul într-un timp cât mai scurt și cu mișcări cât mai puține.

1.1 Ideea jocului

"AllOut" este un joc de tip puzzle în care jucătorul trebuie să stingă toate luminițele din matrice prin selectarea celulelor în ordine corectă. Scopul este de a stinge toate luminițele în cel mai scurt timp posibil și cu cele mai puține mișcări. Jocul este disponibil pentru mai multe platforme, inclusiv PC, telefoane mobile și tablete.

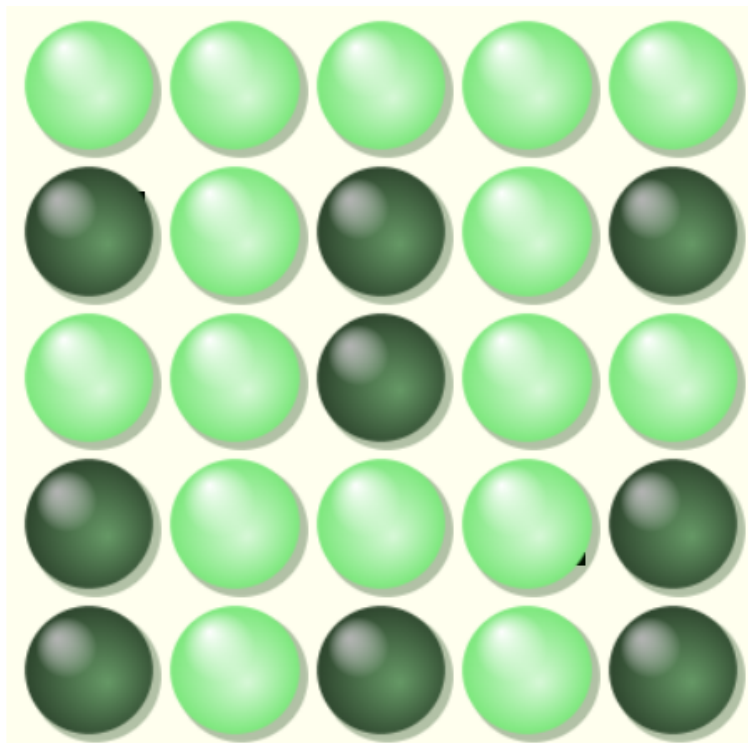


Figure 1: AllOut

Avem nevoie o matrice de $n \times n$ (fiecare n reprezintă un lvl) lumini aprinse sau stinse. Este, de asemenea, nevoie de o strategie și gândire logică pentru a rezolva puzzle-urile mai dificile. Am folosit algoritmul de căutare astar.

2 Definirea problemei

În această secțiune vom prezenta problema fiecărui puzzle.

2.1 AllOut - 4x4

Ne este dată o matrice de 4 coloane și 4 rânduri, în care unele dintre celule sunt luminate. Scopul este de a stinge toate celulele prin selectarea celulelor în ordine corectă. Când o celulă este selectată, cele din jurul ei (vecinii) se vor schimba din starea lor curentă. Jucătorul trebuie să folosească această caracteristică pentru a stinge toate celulele în cel mai scurt timp posibil și cu cele mai puține mișcări posibile.

Am creat 2 astfel de lvl pentru a le compara timpul de execuție.

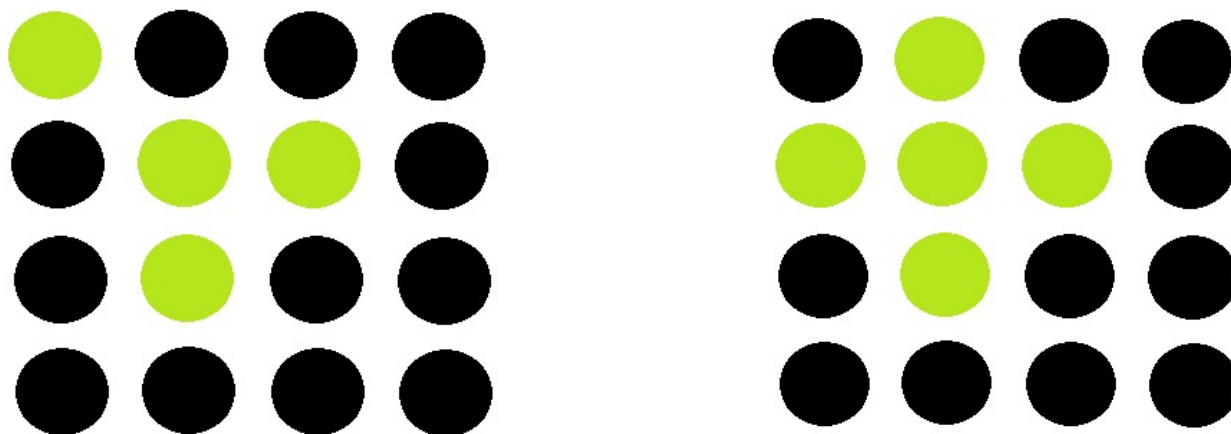


Figure 2: AllOut 4x4

După cum se observă și în poze, avem 2 situații cu câte 5 lumini aprinse, iar restul sunt stinse. Se poate găsi o posibilitate pentru a stinge toate luminile? Vom afla la implementare.

2.2 AllOut - 5x5

Ne este dată o matrice de 5 coloane și 5 rânduri. Acest lvl este mai greu de implementat și de rezolvat. Implicit timpul va fi mai mare, de aceea ne trebuie o strategie cât mai bună.

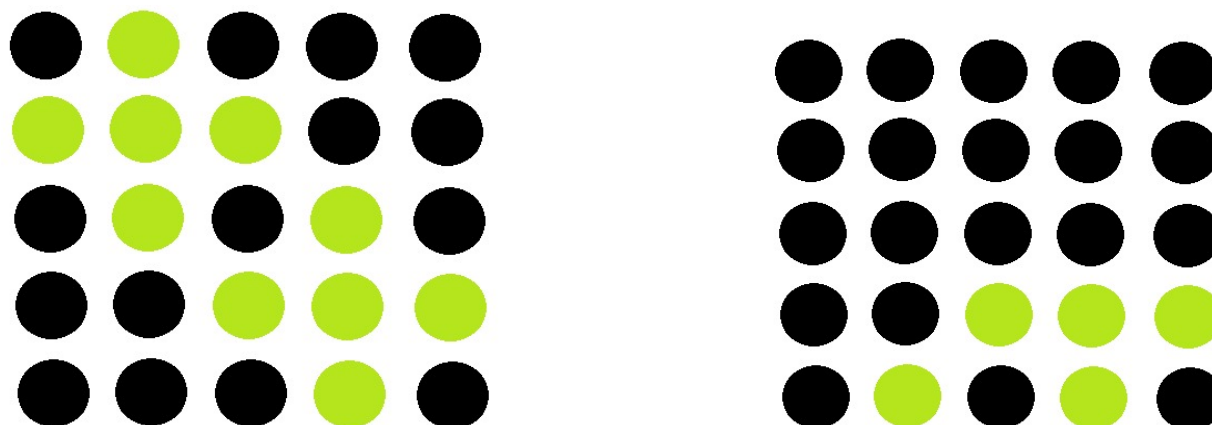


Figure 3: AllOut 5x5

După cum se observă și în poze, avem 2 situații: prima cu 10 lumini aprinse, iar cealaltă cu 5 lumini aprinse.

Diferența principală dintre un puzzle "Allout" de dimensiune 4x4 și unul de dimensiune 5x5 este mărimea grid-ului și numărul de celule care trebuie stinse. Într-un puzzle de dimensiune 4x4, există 16 celule în total, în timp ce într-un puzzle de dimensiune 5x5 există 25 de celule.

Acest lucru înseamnă că un puzzle de dimensiune 5x5 va fi mai dificil de rezolvat și va necesita mai multă gândire strategică din partea jucătorului. De asemenea, un puzzle de dimensiune 5x5 poate include celule suplimentare care au reguli speciale sau comportamente care îl fac mai complex și mai dificil de rezolvat decât 4x4.

3 Soluțiile problemelor

Domain file

```
1  ;;; Domain file: all-out.pddl
2  (define (domain all-out)
3    (:predicates
4      (light-on ?x)
5      (neighbor ?x ?y)
6    )
7    (:action toggle
8      :parameters (?x)
9      :precondition (light-on ?x)
10     :effect (and
11       (not (light-on ?x))
12       (forall (?y) (when (neighbor ?x ?y) (not (light-on ?y)))))
13    )
14  )
15 )
```

În domain definim matricea de lumini și vecini: light-on sunt luminile aprinse(cu un singur număr), au not în față dacă sunt stinse, iar neighbor este format din 2 numere deoarece conține numărul luminii căreia este vecin și numărul său. Acțiunea este aceea de a stinge lumina: not(light-on ?x) pentru toate luminile și vecinii lor.

AllOut 4x4:

```
1  ;; Problem file: all-out-p02.pddl
2  (define (problem all-out-p02)
3    (:domain all-out)
4    (:objects
5      light1 light2 light3 light4 light5 light6 light7 light8 light9 light10 light11 light12
6      light13 light14 light15 light16
7    )
8    (:init
9      (not(light-on light1)) (not(light-on light2))
10     (not(light-on light3)) (not(light-on light4))
11     (not(light-on light5)) (not(light-on light6))
12     (not(light-on light7)) (not(light-on light8))
13     (not(light-on light9)) (not(light-on light10))
14     (not(light-on light11)) (not(light-on light12))
15     (not(light-on light13)) (not(light-on light14))
16     (not(light-on light15)) (not(light-on light16))
17
18     (neighbor light1 light2) (neighbor light1 light5)
19     (neighbor light2 light1) (neighbor light2 light3)
20     (neighbor light2 light6)
21     (neighbor light3 light2) (neighbor light3 light4)
22     (neighbor light3 light7)
23     (neighbor light4 light3) (neighbor light4 light8)
24     (neighbor light5 light1) (neighbor light5 light6)
25     (neighbor light5 light9)
26     (neighbor light6 light2) (neighbor light6 light5)
27     (neighbor light6 light7) (neighbor light6 light10)
28     (neighbor light7 light3) (neighbor light7 light6)
29     (neighbor light7 light8) (neighbor light7 light11)
30     (neighbor light8 light4) (neighbor light8 light7)
31     (neighbor light8 light12)
32     (neighbor light9 light5) (neighbor light9 light10)
33     (neighbor light9 light13)
34     (neighbor light10 light9) (neighbor light10 light6)
35     (neighbor light10 light11) (neighbor light10 light14)
36     (neighbor light11 light7) (neighbor light11 light10)
37     (neighbor light11 light12) (neighbor light11 light15)
38     (neighbor light12 light8) (neighbor light12 light11)
39     (neighbor light12 light16)
40     (neighbor light13 light9) (neighbor light13 light14)
41     (neighbor light14 light13) (neighbor light14 light10)
42     (neighbor light14 light15)
43     (neighbor light15 light14) (neighbor light15 light11)
44     (neighbor light15 light16)
45     (neighbor light16 light15) (neighbor light16 light12)
46
47   )
48   (:goal (forall (?x) (not (light-on ?x)))))
```

3.1 Case 1

(not(light-on light1)) (light-on light2) (not(light-on light3)) (not(light-on light4))
(light-on light5) (light-on light6) (light-on light7) (not(light-on light8))
(not(light-on light9)) (light-on light10) (not(light-on light11)) (not(light-on light12))
(not(light-on light13)) (not(light-on light14)) (not(light-on light15)) (not(light-on light16))

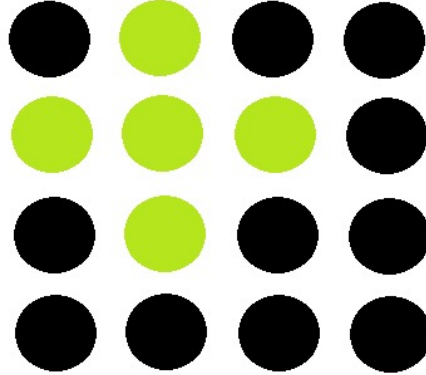


Figure 4: AllOut 4x4

3.2 Case 2

((light-on light1) (not(light-on light2)) (not(light-on light3)) (not(light-on light4))
(not(light-on light5)) (light-on light6) (light-on light7) (not(light-on light8))
(not(light-on light9)) (light-on light10) (not(light-on light11)) (not(light-on light12))
(not(light-on light13)) (not(light-on light14)) (not(light-on light15)) (not(light-on light16))

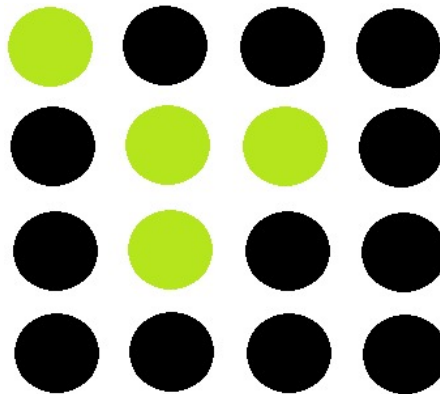


Figure 5: AllOut 4x4

AllOut 5x5:

```

1  ;; Problem file: all-out-p01.pddl
2  (define (problem all-out-p01)
3    (:domain all-out)
4    (:objects
5     light1 light2 light3 light4 light5 light6 light7 light8 light9 light10 light11 light12 light13
6     light14 light15 light16 light17 light18 light19 light20 light21 light22 light23 light24 light25
7    )
8    (:init
9     (not(light-on light1)) (light-on light2) (not(light-on light3))
10    (not(light-on light4)) (not(light-on light5))
11    (light-on light6) (light-on light7) (light-on light8)
12    (not(light-on light9)) (not(light-on light10))
13    (not(light-on light11)) (light-on light12) (not(light-on light13))
14    (light-on light14) (not(light-on light15))
15    (not(light-on light16)) (not(light-on light17)) (light-on light18)
16    (light-on light19) (light-on light20)
17    (not(light-on light21)) (not(light-on light22)) (not(light-on light23))
18    (light-on light24) (not(light-on light25))
19    (neighbor light1 light2) (neighbor light1 light6)
20    (neighbor light2 light1) (neighbor light2 light3) (neighbor light2 light7)
21    (neighbor light3 light2) (neighbor light3 light4) (neighbor light3 light8)
22    (neighbor light4 light3) (neighbor light4 light5) (neighbor light4 light9)
23    (neighbor light5 light4) (neighbor light5 light10)
24    (neighbor light6 light1) (neighbor light6 light7) (neighbor light6 light11)
25    (neighbor light7 light2) (neighbor light7 light6) (neighbor light7 light8) (neighbor light7 light12)
26    (neighbor light8 light3) (neighbor light8 light7) (neighbor light8 light9) (neighbor light8 light13)
27    (neighbor light9 light4) (neighbor light9 light8) (neighbor light9 light10) (neighbor light9 light14)
28    (neighbor light10 light5) (neighbor light10 light9) (neighbor light10 light15)
29    (neighbor light11 light6) (neighbor light11 light12) (neighbor light11 light16)
30    (neighbor light12 light7) (neighbor light12 light11) (neighbor light12 light13) (neighbor light12 light17)
31    (neighbor light13 light8) (neighbor light13 light12) (neighbor light13 light14) (neighbor light13 light18)
32    (neighbor light14 light9) (neighbor light14 light13) (neighbor light14 light15) (neighbor light14 light19)
33    (neighbor light15 light10) (neighbor light15 light14) (neighbor light15 light20)
34    (neighbor light16 light11) (neighbor light16 light17) (neighbor light16 light21)
35    (neighbor light17 light12) (neighbor light17 light16) (neighbor light17 light18) (neighbor light17 light22)
36    (neighbor light18 light13) (neighbor light18 light17) (neighbor light18 light19) (neighbor light18 light23)
37    (neighbor light19 light14) (neighbor light19 light18) (neighbor light19 light20) (neighbor light19 light24)
38    (neighbor light20 light15) (neighbor light20 light19) (neighbor light20 light25)
39    (neighbor light21 light16) (neighbor light21 light22)
40    (neighbor light22 light17) (neighbor light22 light21) (neighbor light22 light23)
41    (neighbor light23 light18) (neighbor light23 light22) (neighbor light23 light24)
42    (neighbor light24 light19) (neighbor light24 light23) (neighbor light24 light25)
43    (neighbor light25 light20) (neighbor light25 light24)
44    )
45    (:goal (forall (?x) (not (light-on ?x)))))

```


3.3 Case 1

(not(light-on light1)) (not(light-on light2)) (not(light-on light3)) (not(light-on light4)) (not(light-on light5))
 (not(light-on light6)) (not(light-on light7)) (not(light-on light8)) (not(light-on light9)) (not(light-on light10))
 (not(light-on light11)) (not(light-on light12)) (not(light-on light13)) (not(light-on light14)) (not(light-on light15))
 (not(light-on light16)) (not(light-on light17)) (light-on light18) (light-on light19) (light-on light20)
 (not(light-on light21)) (light-on light22) (not(light-on light23)) (light-on light24) (not(light-on light25))

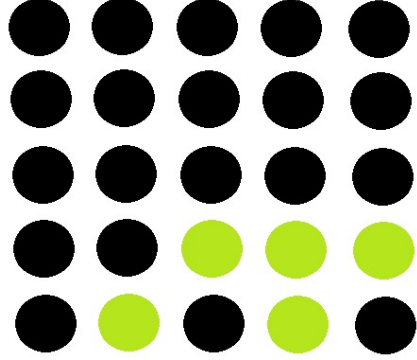


Figure 6: AllOut 5x5

3.4 Case 2

(not(light-on light1)) (light-on light2) (not(light-on light3)) (not(light-on light4)) (not(light-on light5))
 (light-on light6) (light-on light7) (light-on light8) (not(light-on light9)) (not(light-on light10))
 (not(light-on light11)) (light-on light12) (not(light-on light13)) (light-on light14) (not(light-on light15))
 (not(light-on light16)) (not(light-on light17)) (light-on light18) (light-on light19) (light-on light20)
 (not(light-on light21)) (not(light-on light22)) (not(light-on light23)) (light-on light24) (not(light-on light25))

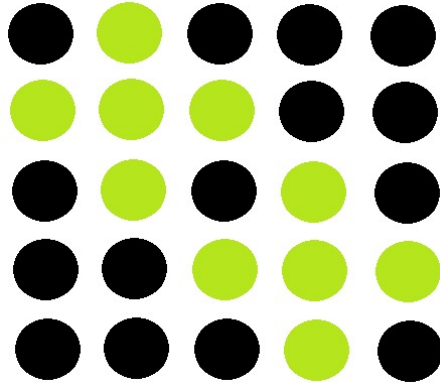


Figure 7: AllOut 5x5

4 Conluzii

După execuția testelor am obținut următoarele rezultate:

4.1 5x5 pic4

[t=0.00318114s, 10292 KB] Plan length: 2 step(s).
[t=0.00318114s, 10292 KB] Plan cost: 2
[t=0.00318114s, 10292 KB] Search time: 0.000477535s
[t=0.00318114s, 10292 KB] Total time: 0.00318114s
Solution found.

4.2 5x5 pic5

[t=0.00491012s, 10292 KB] Plan length: 2 step(s).
[t=0.00491012s, 10292 KB] Plan cost: 2
[t=0.00491012s, 10292 KB] Search time: 0.00118417s
[t=0.00491012s, 10292 KB] Total time: 0.00491012s
Solution found.

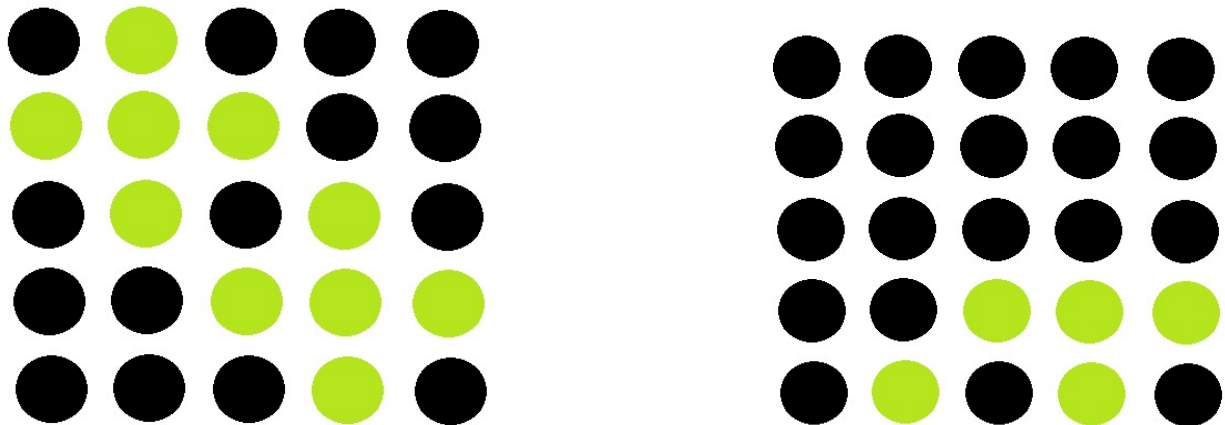


Figure 8: AllOut 5x5

Timpul de căutare pentru primul caz a fost mai mic deoarece sunt necesare doar 2 schimbări: stingerea luminii 7 și stingerea luminii 19.

4.3 4x4 pic2

[t=0.00261949s, 10292 KB] Plan length: 2 step(s).
[t=0.00261949s, 10292 KB] Plan cost: 2
[t=0.00261949s, 10292 KB] Search time: 0.000433352s
[t=0.00261949s, 10292 KB] Total time: 0.00261949s
Solution found.

4.4 4x4 pic3

[t=0.00431037s, 10292 KB] Plan length: 1 step(s)..
[t=0.00431037s, 10292 KB] Plan cost: 1
[t=0.00431037s, 10292 KB] Search time: 0.000444522s
[t=0.00431037s, 10292 KB] Total time: 0.00431037s
Solution found.

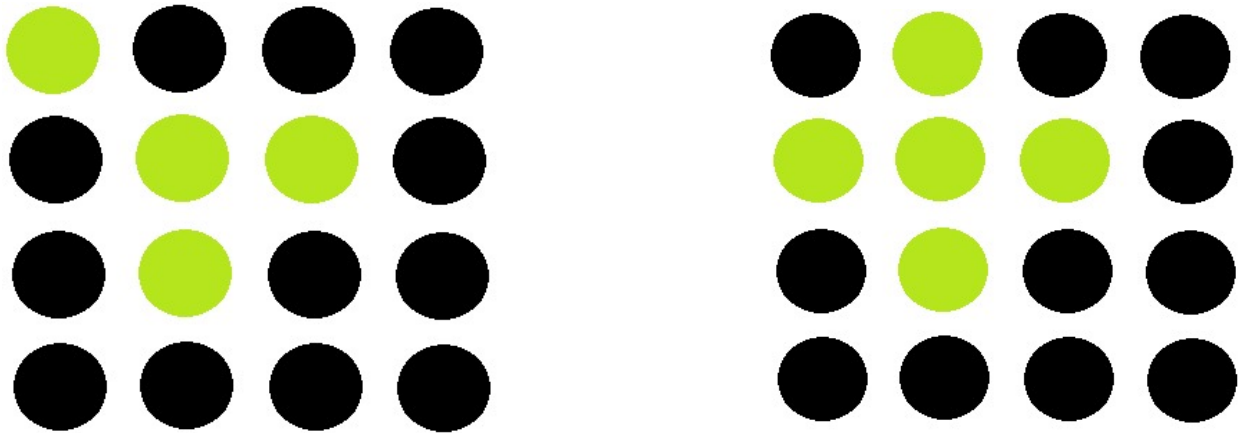


Figure 9: AllOut 4x4

Aici se poate observa că în primul caz este nevoie de 2 pași, iar în al doilea de unul singur.