

CS 6511 - AI
Project #2 - CSP Tile Placement
Ryan Gross

GW ID: G47667332

<https://github.com/ryangrossGitHub/gw-cs6511-constraint-satisfaction-tile-placement>

Introduction

The solution at the github link above solves all 10 example problems that had provided answer keys from <https://github.com/amrinderarora/ai/tree/master/src/main/resources/csp/tileplacement> in under 35 seconds, with 8 of 10 solving in under 5 second. The remainder of this document will detail the key parts of the solution and show results.

Problem Representation

The tile placement problem can simply be thought of as trying to find the combinations of el and outer tiles that add up to the desired bush count across 25 sets of bushes. Initially I wanted to represent variables as 4x4 grids, and values as the tile shape, but after struggling to make this work I settled on the following representation:

- **Variables:** Tile types (el, outer block, full)
- **Values:** Bush counts per 4x4 grid
- **Minimum Remaining Value (MRV):** Least abundant tile type
- **Least Constrained Value (LCV):** 4x4 grid that keeps average bush counts as far away from targets as possible. Put another way, favor grids that evenly reduce all bush counts rather than going for grids that reduce the maximum number of target bushes. This even reduction leaves more options available.

```
min_delta = min([targets[0] - actuals[0], targets[1] - actuals[1], targets[2] - actuals[2],  
                targets[3] - actuals[3]])
```

Time and Space Saving Tips Prior to Search

Here are a few ways the problem can be simplified to make solving easier:

- **Full blocks can be ignored.** They will just fill in the remaining tile locations at the end. This strategy allows us to reduce complexity by only considering two tile types in our calculations.
- **Only read the bushes within the el shape into memory.** No other tile type will use the bushes hidden by the el shape. This saves calculation time in heuristics, and increases accuracy as those bushes will never be a meaningful part of the solution.
- **Cache el and outer bush totals.** Caching after reading the file avoids re-calculating these static numbers during our search.

Search Steps

The coded arc consistency steps are as follows:

1. Determine MRV ordering (mrv[0] is most mrv).

```
Total Combos EL_SHAPE: 53,130
Total Combos OUTER_BOUNDARY: 5,200,300

if el_combo_counts > outer_combo_counts:
    mrv = ['OUTER_BOUNDARY', 'EL_SHAPE']
else:
    mrv = ['EL_SHAPE', 'OUTER_BOUNDARY']
```

2. Generate all combinations for the MRV, which is the least abundant tile type.

```
> 00000 = {tuple: 5} (0, 1, 2, 3, 4)
> 00001 = {tuple: 5} (0, 1, 2, 3, 5)
> 00002 = {tuple: 5} (0, 1, 2, 3, 6)
> 00003 = {tuple: 5} (0, 1, 2, 3, 7)
> 00004 = {tuple: 5} (0, 1, 2, 3, 8)
> 00005 = {tuple: 5} (0, 1, 2, 3, 9)
> 00006 = {tuple: 5} (0, 1, 2, 3, 10)
> 00007 = {tuple: 5} (0, 1, 2, 3, 11)
> 00008 = {tuple: 5} (0, 1, 2, 3, 12)
> 00009 = {tuple: 5} (0, 1, 2, 3, 13)
```

3. Get MRV tile type combo bush totals, and while we are already iterating, forward check to filter out any combos already violating constraints to reduce domain.

Last number is LCV calc, representing
min delta from targets



```
> 00000 = {list: 3} [(0, 1, 2, 3, 4), [9, 9, 11, 8], 8]
> 00001 = {list: 3} [(0, 1, 2, 3, 5), [6, 12, 12, 8], 7]
> 00002 = {list: 3} [(0, 1, 2, 3, 6), [6, 9, 13, 9], 6]
> 00003 = {list: 3} [(0, 1, 2, 3, 7), [6, 11, 12, 6], 7]
> 00004 = {list: 3} [(0, 1, 2, 3, 8), [9, 9, 10, 10], 8]
> 00005 = {list: 3} [(0, 1, 2, 3, 9), [7, 11, 10, 7], 9]
> 00006 = {list: 3} [(0, 1, 2, 3, 10), [8, 11, 10, 9], 9]
> 00007 = {list: 3} [(0, 1, 2, 3, 11), [7, 11, 11, 8], 8]
> 00008 = {list: 3} [(0, 1, 2, 3, 12), [9, 11, 10, 9], 8]
> 00009 = {list: 3} [(0, 1, 2, 3, 13), [9, 12, 11, 7], 8]
```

```
min_delta = min([targets[0] - actuals[0], targets[1] - actuals[1], targets[2] - actuals[2],
                 targets[3] - actuals[3]])
```

4. Sort MRV tile type combos by LCV as described in the problem representation section.

Largest delta leaves greatest options open, least constrained.



```
> 00000 = (list: 3) [(3, 9, 14, 16, 21), [5, 8, 7, 6], 12]
> 00001 = (list: 3) [(7, 9, 14, 16, 21), [4, 7, 7, 6], 12]
> 00002 = (list: 3) [(7, 9, 16, 18, 21), [5, 8, 7, 6], 12]
> 00003 = (list: 3) [(1, 3, 7, 14, 16), [6, 9, 8, 5], 11]
> 00004 = (list: 3) [(1, 3, 9, 14, 21), [6, 8, 8, 6], 11]
> 00005 = (list: 3) [(1, 5, 7, 14, 16), [5, 9, 8, 7], 11]
> 00006 = (list: 3) [(1, 7, 9, 14, 16), [6, 8, 6, 6], 11]
> 00007 = (list: 3) [(1, 7, 9, 14, 17), [6, 9, 8, 5], 11]
> 00008 = (list: 3) [(1, 7, 9, 14, 21), [5, 7, 8, 6], 11]
> 00009 = (list: 3) [(1, 7, 9, 18, 21), [6, 8, 8, 6], 11]
> 00010 = (list: 3) [(1, 7, 11, 14, 16), [6, 8, 7, 7], 11]
```

5. Generate all combinations for the non-MRV tile type.

```
> 0000000 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
> 0000001 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12)
> 0000002 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13)
> 0000003 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14)
> 0000004 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15)
> 0000005 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16)
> 0000006 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 17)
> 0000007 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 18)
> 0000008 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 19)
> 0000009 = (tuple: 12) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20)
```

6. Propagate constraints to enforce arc consistency by filtering any combination where MRV and non-MRV tile types occupy the same space, or exceed any bush targets.

Lessons Learned

I started with a small grid and proceeded by implementing DFS, then DFS + MRV + LCV, then DFS + MRV + LCV + Forward Checking, and then wanted to end with arc consistency. I found that while I was getting correct answers on small problems, my solutions were incredibly slow. To my dismay, after thinking I was 95% done, I ran a few examples overnight that never returned. I had to go back to the drawing board, and that's when I created the solution described in this paper. If I had to do it all over again, I would **start with implementing arc consistency at the beginning rather than try to add it to a DFS based algorithm at the end**. This code is in `/old_solution/backtrack.py`.

Results

On remaining pages, 1 test per page.

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658913086500.txt

```
Processing: test1.txt
Total Combos EL_SHAPE: 480,700
Total Combos OUTER_BOUNDARY: 177,100
Minimum Remaining Values (MRV): OUTER_BOUNDARY
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 177,100
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 1.1219990253448486s
0 4 OUTER_BOUNDARY
1 4 FULL_BLOCK
2 4 FULL_BLOCK
3 4 FULL_BLOCK
4 4 FULL_BLOCK
5 4 OUTER_BOUNDARY
6 4 EL_SHAPE
7 4 EL_SHAPE
8 4 EL_SHAPE
9 4 FULL_BLOCK
10 4 OUTER_BOUNDARY
11 4 FULL_BLOCK
12 4 OUTER_BOUNDARY
13 4 EL_SHAPE
14 4 FULL_BLOCK
15 4 EL_SHAPE
16 4 EL_SHAPE
17 4 FULL_BLOCK
18 4 FULL_BLOCK
19 4 OUTER_BOUNDARY
20 4 OUTER_BOUNDARY
21 4 FULL_BLOCK
22 4 EL_SHAPE
23 4 FULL_BLOCK
24 4 FULL_BLOCK

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658918378200.txt

```
Processing: test2.txt
Total Combos EL_SHAPE: 4,457,400
Total Combos OUTER_BOUNDARY: 177,100
Minimum Remaining Values (MRV): OUTER_BOUNDARY
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 177,100
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 3.946040391921997s
0 4 FULL_BLOCK
1 4 EL_SHAPE
2 4 EL_SHAPE
3 4 EL_SHAPE
4 4 EL_SHAPE
5 4 OUTER_BOUNDARY
6 4 OUTER_BOUNDARY
7 4 FULL_BLOCK
8 4 FULL_BLOCK
9 4 FULL_BLOCK
10 4 EL_SHAPE
11 4 OUTER_BOUNDARY
12 4 OUTER_BOUNDARY
13 4 EL_SHAPE
14 4 FULL_BLOCK
15 4 EL_SHAPE
16 4 OUTER_BOUNDARY
17 4 FULL_BLOCK
18 4 EL_SHAPE
19 4 EL_SHAPE
20 4 OUTER_BOUNDARY
21 4 FULL_BLOCK
22 4 FULL_BLOCK
23 4 EL_SHAPE
24 4 EL_SHAPE

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658921400300.txt

```
Processing: test3.txt
Total Combos EL_SHAPE: 177,100
Total Combos OUTER_BOUNDARY: 1,081,575
Minimum Remaining Values (MRV): EL_SHAPE
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 115,881
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 2.2570323944091797s
0 4 OUTER_BOUNDARY
1 4 EL_SHAPE
2 4 FULL_BLOCK
3 4 OUTER_BOUNDARY
4 4 FULL_BLOCK
5 4 FULL_BLOCK
6 4 OUTER_BOUNDARY
7 4 FULL_BLOCK
8 4 OUTER_BOUNDARY
9 4 OUTER_BOUNDARY
10 4 EL_SHAPE
11 4 FULL_BLOCK
12 4 FULL_BLOCK
13 4 FULL_BLOCK
14 4 EL_SHAPE
15 4 OUTER_BOUNDARY
16 4 EL_SHAPE
17 4 FULL_BLOCK
18 4 EL_SHAPE
19 4 OUTER_BOUNDARY
20 4 EL_SHAPE
21 4 OUTER_BOUNDARY
22 4 FULL_BLOCK
23 4 FULL_BLOCK
24 4 FULL_BLOCK

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658924404900.txt

```
Processing: test4.txt
Total Combos EL_SHAPE: 177,100
Total Combos OUTER_BOUNDARY: 2,042,975
Minimum Remaining Values (MRV): EL_SHAPE
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 150,545
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 2.560030698776245s
0 4 EL_SHAPE
1 4 FULL_BLOCK
2 4 OUTER_BOUNDARY
3 4 FULL_BLOCK
4 4 FULL_BLOCK
5 4 OUTER_BOUNDARY
6 4 OUTER_BOUNDARY
7 4 EL_SHAPE
8 4 OUTER_BOUNDARY
9 4 FULL_BLOCK
10 4 EL_SHAPE
11 4 FULL_BLOCK
12 4 FULL_BLOCK
13 4 FULL_BLOCK
14 4 EL_SHAPE
15 4 FULL_BLOCK
16 4 OUTER_BOUNDARY
17 4 FULL_BLOCK
18 4 OUTER_BOUNDARY
19 4 OUTER_BOUNDARY
20 4 OUTER_BOUNDARY
21 4 OUTER_BOUNDARY
22 4 EL_SHAPE
23 4 FULL_BLOCK
24 4 EL_SHAPE

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658926570700.txt

```
Processing: test5.txt
Total Combos EL_SHAPE: 4,457,400
Total Combos OUTER_BOUNDARY: 300
Minimum Remaining Values (MRV): OUTER_BOUNDARY
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 300
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 1.2900376319885254s
0 4 OUTER_BOUNDARY
1 4 EL_SHAPE
2 4 FULL_BLOCK
3 4 FULL_BLOCK
4 4 FULL_BLOCK
5 4 EL_SHAPE
6 4 EL_SHAPE
7 4 EL_SHAPE
8 4 EL_SHAPE
9 4 FULL_BLOCK
10 4 EL_SHAPE
11 4 EL_SHAPE
12 4 FULL_BLOCK
13 4 EL_SHAPE
14 4 FULL_BLOCK
15 4 EL_SHAPE
16 4 EL_SHAPE
17 4 FULL_BLOCK
18 4 FULL_BLOCK
19 4 FULL_BLOCK
20 4 EL_SHAPE
21 4 OUTER_BOUNDARY
22 4 FULL_BLOCK
23 4 FULL_BLOCK
24 4 FULL_BLOCK

Process finished with exit code 0
```


https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658928646700.txt

```
Processing: test6.txt
Total Combos EL_SHAPE: 1,081,575
Total Combos OUTER_BOUNDARY: 2,042,975
Minimum Remaining Values (MRV): EL_SHAPE
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 894,438
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 34.825140714645386s
0 4 EL_SHAPE
1 4 EL_SHAPE
2 4 EL_SHAPE
3 4 EL_SHAPE
4 4 OUTER_BOUNDARY
5 4 EL_SHAPE
6 4 OUTER_BOUNDARY
7 4 FULL_BLOCK
8 4 FULL_BLOCK
9 4 FULL_BLOCK
10 4 EL_SHAPE
11 4 OUTER_BOUNDARY
12 4 OUTER_BOUNDARY
13 4 OUTER_BOUNDARY
14 4 OUTER_BOUNDARY
15 4 OUTER_BOUNDARY
16 4 EL_SHAPE
17 4 FULL_BLOCK
18 4 OUTER_BOUNDARY
19 4 FULL_BLOCK
20 4 OUTER_BOUNDARY
21 4 EL_SHAPE
22 4 FULL_BLOCK
23 4 FULL_BLOCK
24 4 FULL_BLOCK

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658930331900.txt

```
Processing: test7.txt
Total Combos EL_SHAPE: 4,457,400
Total Combos OUTER_BOUNDARY: 480,700
Minimum Remaining Values (MRV): OUTER_BOUNDARY
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 480,700
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 4.258021593093872s
0 4 OUTER_BOUNDARY
1 4 FULL_BLOCK
2 4 OUTER_BOUNDARY
3 4 EL_SHAPE
4 4 FULL_BLOCK
5 4 EL_SHAPE
6 4 OUTER_BOUNDARY
7 4 EL_SHAPE
8 4 FULL_BLOCK
9 4 FULL_BLOCK
10 4 EL_SHAPE
11 4 FULL_BLOCK
12 4 OUTER_BOUNDARY
13 4 EL_SHAPE
14 4 OUTER_BOUNDARY
15 4 OUTER_BOUNDARY
16 4 OUTER_BOUNDARY
17 4 EL_SHAPE
18 4 EL_SHAPE
19 4 EL_SHAPE
20 4 EL_SHAPE
21 4 FULL_BLOCK
22 4 FULL_BLOCK
23 4 EL_SHAPE
24 4 EL_SHAPE

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658931783100.txt

```
Processing: test8.txt
Total Combos EL_SHAPE: 5,200,300
Total Combos OUTER_BOUNDARY: 12,650
Minimum Remaining Values (MRV): OUTER_BOUNDARY
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 12,650
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 2.068998098373413s
0 4 EL_SHAPE
1 4 EL_SHAPE
2 4 EL_SHAPE
3 4 EL_SHAPE
4 4 FULL_BLOCK
5 4 OUTER_BOUNDARY
6 4 EL_SHAPE
7 4 EL_SHAPE
8 4 FULL_BLOCK
9 4 FULL_BLOCK
10 4 EL_SHAPE
11 4 FULL_BLOCK
12 4 FULL_BLOCK
13 4 EL_SHAPE
14 4 EL_SHAPE
15 4 OUTER_BOUNDARY
16 4 OUTER_BOUNDARY
17 4 EL_SHAPE
18 4 FULL_BLOCK
19 4 EL_SHAPE
20 4 OUTER_BOUNDARY
21 4 FULL_BLOCK
22 4 EL_SHAPE
23 4 FULL_BLOCK
24 4 EL_SHAPE

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658934155700.txt

```
Processing: test9.txt
Total Combos EL_SHAPE: 53,130
Total Combos OUTER_BOUNDARY: 4,457,400
Minimum Remaining Values (MRV): EL_SHAPE
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 53,043
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 1.8089983463287354s
0 4 OUTER_BOUNDARY
1 4 OUTER_BOUNDARY
2 4 EL_SHAPE
3 4 OUTER_BOUNDARY
4 4 FULL_BLOCK
5 4 EL_SHAPE
6 4 OUTER_BOUNDARY
7 4 OUTER_BOUNDARY
8 4 OUTER_BOUNDARY
9 4 FULL_BLOCK
10 4 OUTER_BOUNDARY
11 4 EL_SHAPE
12 4 OUTER_BOUNDARY
13 4 FULL_BLOCK
14 4 EL_SHAPE
15 4 OUTER_BOUNDARY
16 4 OUTER_BOUNDARY
17 4 EL_SHAPE
18 4 FULL_BLOCK
19 4 OUTER_BOUNDARY
20 4 OUTER_BOUNDARY
21 4 FULL_BLOCK
22 4 OUTER_BOUNDARY
23 4 FULL_BLOCK
24 4 OUTER_BOUNDARY

Process finished with exit code 0
```

https://github.com/amrinderarora/ai/blob/master/src/main/resources/csp/tileplacement/tilesproblem_1326658935930000.txt

```
Processing: test10.txt
Total Combos EL_SHAPE: 53,130
Total Combos OUTER_BOUNDARY: 5,200,300
Minimum Remaining Values (MRV): EL_SHAPE
Applying Forward Checking on MRV combos...
Total Combos MRV after Forward Check: 53,118
Sorting MRV combos into Least Constrained Value (LCV) order...
Applying Arc Consistency to remaining combos...
Solution found!
Time: 33.24703621864319s
0 4 FULL_BLOCK
1 4 OUTER_BOUNDARY
2 4 OUTER_BOUNDARY
3 4 OUTER_BOUNDARY
4 4 FULL_BLOCK
5 4 EL_SHAPE
6 4 FULL_BLOCK
7 4 EL_SHAPE
8 4 FULL_BLOCK
9 4 FULL_BLOCK
10 4 OUTER_BOUNDARY
11 4 FULL_BLOCK
12 4 OUTER_BOUNDARY
13 4 OUTER_BOUNDARY
14 4 OUTER_BOUNDARY
15 4 FULL_BLOCK
16 4 OUTER_BOUNDARY
17 4 OUTER_BOUNDARY
18 4 EL_SHAPE
19 4 FULL_BLOCK
20 4 OUTER_BOUNDARY
21 4 EL_SHAPE
22 4 EL_SHAPE
23 4 OUTER_BOUNDARY
24 4 OUTER_BOUNDARY

Process finished with exit code 0
```