Dominosa: A Interactive Game and Problem Solver

Madison Everett and Omar Martinez

Department of Mathematics and Computer Science

BELMONT UNIVERSITY

Background

Dominosa is one of many variations of Domino related games and puzzles. Dominosa is a single-player game that consists of a grid of numbers. Within the grid of numbers, there are unique Dominos pieces that correspond with certain pairs of numbers throughout the grid. The goal of Dominosa is to fill the board with the corresponding Domino pieces, without having two of the same domino on the grid.

Dominosa is also related to a Constraint-Satisfaction Problem, or CSP. CSPs are problems that have a constraint that needs to be satisfied in order for the problem to be solved.

Although the general formats of a CSP increase the efficiency of solving a problem, there are several algorithms that can be implemented to further increase the efficiency of finding a solution to a specific problem.

One of these main algorithms is the Backtracking Search algorithm, pseudocode shown below:

```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
  return BACKTRACK({ }, csp)

function BACKTRACK(assignment, csp) returns a solution, or failure
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(csp, assignment)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
     if value is consistent with assignment then
        add {var = value} to assignment
        inferences ← INFERENCE(csp, var, assignment)
        if inferences ≠ failure then
           add inferences to assignment
           result ← BACKTRACK(assignment, csp)
           if result ≠ failure then
              return result
     remove {var = value} and inferences from assignment
  return failure
```

This algorithm is able to use several efficiency based heuristics such as: Least-Constraining Value(LCV), Minimum Remaining Values, Maximum Arc Consistency(MAC), and forward-checking to further maximize the results of the algorithm.

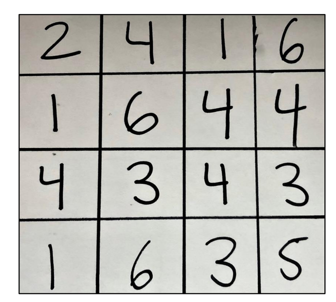
Purpose

Although a CSP toy problem, Dominosa holds real-world significance because of its relation to other real-world constraint-satisfaction problems, such as nurse scheduling, chemical storage, and even configuration of computer systems. Successfully creating a program that solves Dominosa efficiently could possibly give way to real-world implications and create a basis for solving CSPs.

Design

We chose to build this project within the programming language Python because of the ease of reusing of code. After this decision, we first created the interactive game, which contains a menu to give the player an option on what they would like do within the game. In addition to the menu, the interactive game also includes several predetermined Dominosa boards to be used in interactive play, as well. These boards consist of a grid of numbers similar to the physical game.

However, in order for the Dominosa board to be correctly solved by the Backtracking Search algorithm, the grid had to be converted to a readable data structure. This was done by creating a list of the possible pairs within the domino grid. The process can be seen below.

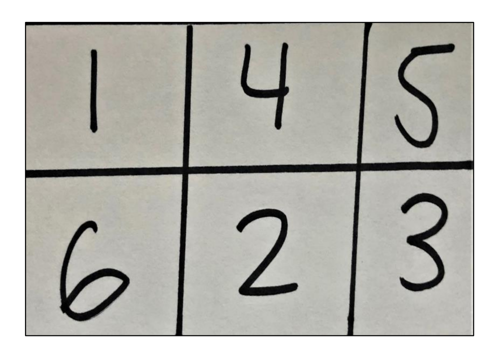



```
Harder(Medium)
2 4 1 6
1 6 4 4
4 3 4 3
1 6 3 5
```

```
harder = ['1|2','3|5','1|6','3|4','4|6','1|4','4|6',
          '4|6','3|6','4|6','3|6','4|6','3|6','3|6',
          '1|4','1|6','1|4','4|4', '3|4','3|4','3|5'
          ,'4|6']
```

Results

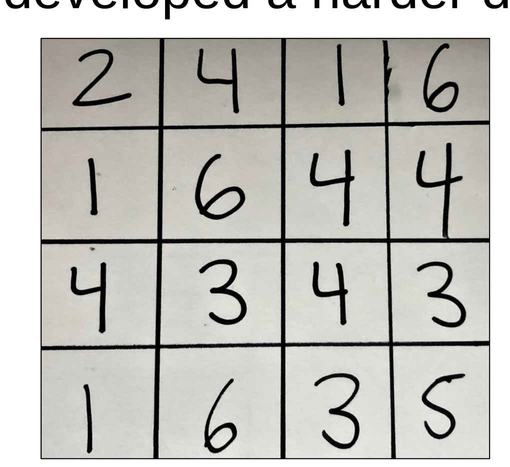
After changing the format of the Dominosa puzzle, we began comparing the result of the Backtracking Search algorithm to the problem solving speed of a human.
We first started this testing with an easy Dominosa puzzle consisting of three dominos, shown below with the solution:



Due to the ease of the puzzle, the human player and Backtracking Search algorithm, without heuristics, performed at the same level with about 9 assignments/moves for each player.

However, with the added heuristics of LCV, the Backtracking Search algorithm managed to beat the human player with just 4 assignments/moves.

To continue testing, we then developed a harder difficulty, involving eight dominos pieces, shown below with the solution:



This puzzle difficulty continued to show improvement on the performance of the Backtracking Search algorithm compared to the human player.

The Backtracking Search algorithm without any heuristics, found the solution in just 58 assignments/moves, which is still less than the human player's assignments, which rounded out about 70 assignment/moves.

With the added heuristics, such as forward-chaining, MRV, LCV, and MAC, the efficiency of the algorithm drastically increased, taking the number of overall assignments from 58 assignments to just 8 assignments/moves, which completely outperforms the human player.

Conclusion

From this program, we have discovered that without heuristics, the Backtracking Search algorithm performs similarly to a smart human player. However, using added heuristics allows the Backtracking Search algorithm to perform more optimally and efficiently than a human player. From this discovery, this program can further adapted to be used as a bases for solving other constraint satisfaction problems that exist within the real-world.

Future Improvements

Despite successful results, there are several improvements that can be done to further develop this program. Some of these improvements are:

- Allowing the Dominosa Solver to actually create boards based on user's input to increase game choices
- Improving the Dominosa Solver to allow it to find a solution from any viable set of domino related numbers.
- Implementing the forward-checking algorithm and the Conflict-Directed Backjumping algorithm to the Dominosa Solver.
- Continuing to maximize the Backtracking Search algorithm to find the least amount of assignments for any given board.

References

Aggoun, Abderrahmane. "Panorama of Real - Life Applications in Logistics Embedding Bin Packing Optimization Algorithms, Robotics and Cloud Computing Technologies." *HAL Archives*, 30 Nov. 2016, https://hal.archives-ouvertes.fr/hal-01378469.

Brailsford, Sally C, et al. "Constraint Satisfaction Problems: Algorithms and Applications." *European Journal of Operational Research*, North-Holland, 13 Jan. 2011, www.sciencedirect.com/science/article/pii/S0377221798003646

Chen, Yarong, et al. "Technology and System of Constraint Programming for Industry Production Scheduling - Part I: A Brief Survey and Potential Directions." *SpringerLink*, SP Higher Education Press, 26 Aug. 2010, link.springer.com/article/10.1007%2Fs11465-010-0106-x#citeas.

Rossi, Francesca, and Toby Walsh. "Constraint Programming." *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*, edited by Peter van Beek, 1st ed., Elsevier, 2006, pp. 83–117.

Tatham, Simon. "Chapter 17." *Dominosa*, Simon Tatham's Portable Puzzle Collection, 2014, www.chiark.greenend.org.uk/~sgtatham/puzzles/doc/dominosa.html#dominosa.