Search Strategy Implementation

For the single player, SPREAD-only version of Infexion, we utilised the A* search strategy to find an optimal solution. Our decision was made by comparing all the uninformed and informed search strategies. The game has a finite number of solutions and each step cost is always equal to 1; hence, only A*, breadth-first search (BFS) and iterative deepening search (IDS) are guaranteed to be complete and optimal out of all the algorithms and were explored. Uniform-cost search also fulfils this criteria, but since it can be reduced to BFS, we decided to disregard it.

For our A* search, we chose data structures that would improve our time and space complexities. The structures are continuously updated as the search progresses:

1. All expanded nodes are stored in a dictionary which takes O(1) time for insertion and lookup.
2. Nodes to expand are stored in a priority queue ordered in ascending *f(n)* (evaluation function) values, taking O(*n* log *n*) time for enqueue and O(1) for dequeue.

Time and Space Complexity

In regard to time and space complexities, we define the following terms which are based on the entire search (in other words, while search has not found the optimal solution yet):

- b: Max. branching factor of search tree; the max. number of possible actions RED can take
- d: Depth of least cost solution; the min. number of moves to get to the goal state
- ϵ: Relative error in the heuristic $h(n)$

Theoretically, A* would perform better than BFS and IDS as long as it uses an admissible heuristic, pruning board states to explore and therefore improving space and time complexity. The performance of the algorithms would be A* $O(b^{\epsilon d})$ < BFS $O(b^{d})$ == IDS $O(b^{d})$ and A* $O(b^{\epsilon d})$ < IDS $O(bd)$ < BFS $O(b^{d})$ based on time and space complexities respectively. Note that A*'s space complexity is $O(b^{\epsilon d})$ since all expanded nodes are kept in memory.

Empirically, A* also had the best performance. As shown in Figure 1, it was able to find a solution in all test cases. The test cases are ordered left to right with ascending difficulty, and while it may appear that BFS stopped after test6-1 and IDS after test6-4, this merely means they had very long running times. While A* initially performed the worst for both time and space complexity, it improved with each level of difficulty, especially over IDS. Though BFS appears to have a better space complexity, the fact that A* is far quicker in finding a solution for more complex boards would indicate that it is still the best choice of search strategy.

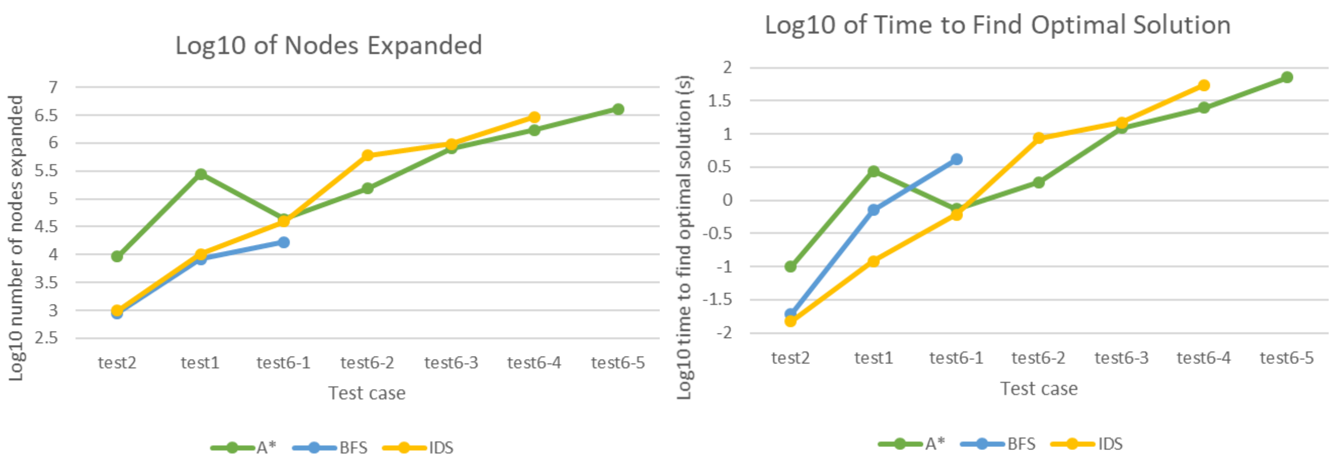

Figure 1: Results of experiments with A, IDS, and BFS for test cases of increasing complexity from left to right of the x-axis. Tests can be found in the 'tests' folder in Gradescope.*

A* Search Heuristic

Our A* approach uses an evaluation function $f(n) = g(n) + h(n)$ where:

- n = Board state of interest
- $g(n)$ = Number of moves to reach board state *n*
- $h(n)$ = Estimated shortest number of moves to reach a goal state from board state *n*

To obtain $h(n)$, we assume the board state is in a relaxed version of Infexion and find the shortest number of moves under those conditions. In this version, when we SPREAD a red cell with **power *p***, this **directly** infects ***p* blue cells** on the board in **one** move and stacks their power by one (they won't spread over the board as they would in normal Infexion).

Simulating playing *n* with the steps below intuitively always results in the optimal solution for this relaxed version, avoiding using complex search trees to find the shortest number of moves:

1) Spread the red cell with the highest power to maximise the number of blue cells infected.
2) With the available power the red cell has, infect blue cells with highest powers first so they'll in turn have high power as red cells to maximise the number of blue cells infected.
3) Repeat until there are no more blue cells.

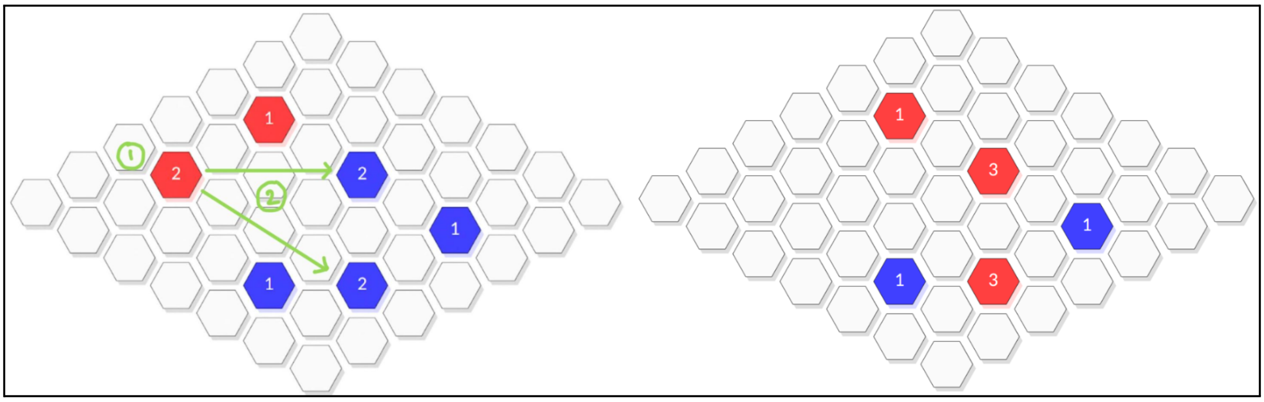

Figure 2: Example of the most optimal move performed in a game of relaxed infexion with the outlined steps. Left board state will go to right board state in one move.

Thus red cells in this relaxed Infexion will always infect all blue cells in fewer or equal moves than in the real game, making this heuristic admissible and therefore the A* search optimal. Performing A* search with it prunes high-*f(n)* nodes unlikely to result in optimal solutions early on, reducing both space and time costs of the search as evidenced in Figure 1.

SPAWN Extension

If SPAWN actions were to be allowed in single player Infexion, this would intuitively cause a goal state to be reached in fewer moves than in the SPREAD-only version, causing *d* to be slightly smaller. However, *b* would greatly increase since there are more possible actions a board could do, producing more child nodes when a board state is expanded. Thus, search trees for this version would be larger in width and shorter in height compared to the SPREAD-only counterpart.

This means the heuristic we use for this new problem must prune more nodes at a time to accommodate for the increase in the maximum number of child nodes per expansion. This may require us to devise another heuristic which comes closer to the true cost of getting to the goal state from a board state.