

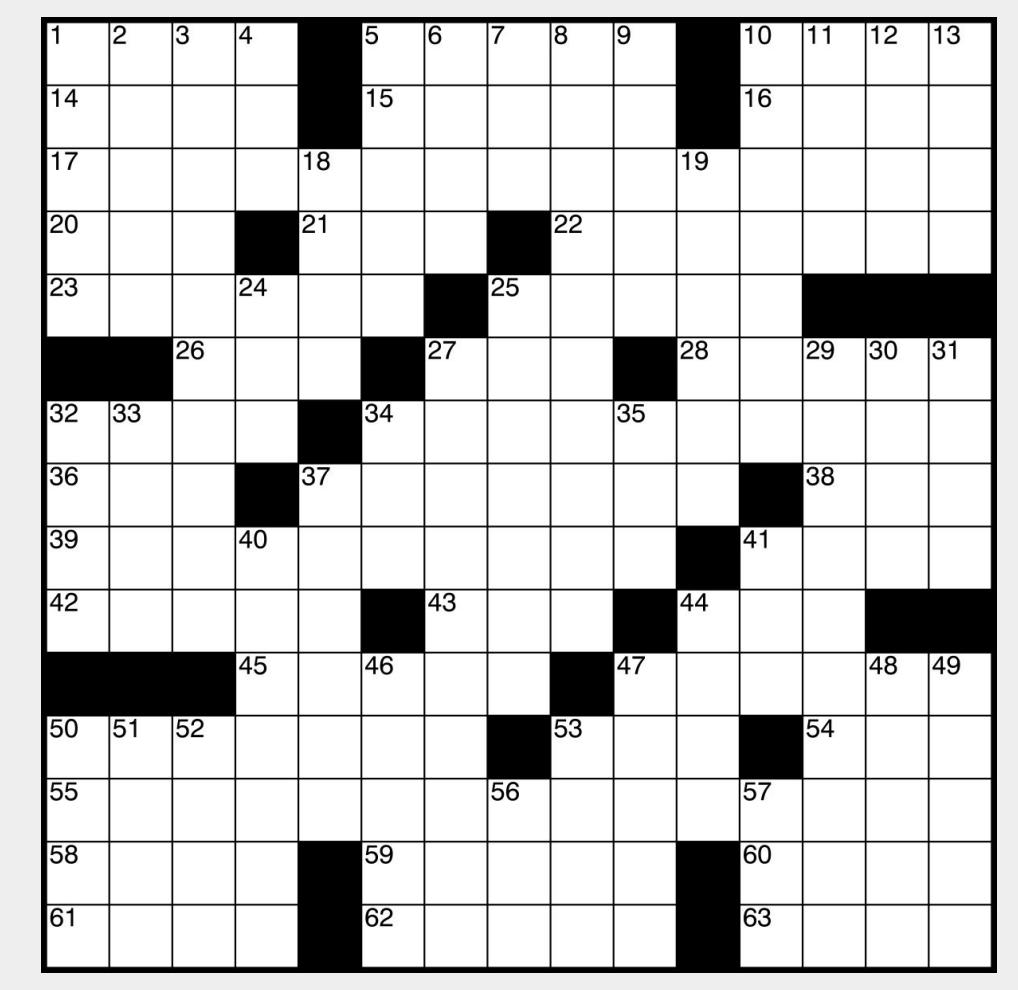
Automatic Crossword Puzzle Construction



What is a Crossword?

There are several common general limitations put on crosswords by most crossword publishers. These include:

- Crosswords ought to have 180-degree rotational symmetry
- All answers must be at least 3 letters long
- Words shouldn't be used more than once in the fill of the crossword
- All letters used must be found in both across AND down answers
- All words need to be connected



Modern Crossword Construction Software

Current software to construct Crossword Puzzles has many limitations:

- They require a large word list - usually well over 100,000 words
- There aren't enough performance metrics and public datasets for testing
- When asked to fill a grid - some may not find a fill even when one exists
- Not all fill is created equal - multiple potential fills for a crossword grid may exist, and some may be "more fun" to solve by requiring less extraneous vocabulary

References

1. M. Ginsberg, M. Frank, M. Halpin, and M. Torrance, Search Lessons Learned from Crossword Puzzles, AAAI 1990.

Otis Peterson

Supervised by Michael Wehar



Crossword Puzzle Generation Example

- Generated in 4.04 seconds
- Both brute force methods failed to terminate after 5 minutes
- Existing software, on the same grid, generated a grid in around 6 seconds

Try our demo app at:
crosswordconstruction.com



Crossword Puzzles are Hard to Construct - Defining the Problem

Crossword Puzzle Construction Problem

- Input
 - A finite alphabet Σ
 - A set of words W over Σ
 - An n by n Boolean matrix A (called the grid)
- Decision Question
 - Can we fill all 0's of the matrix A with words from W so that every maximally contiguous horizontal and vertical sequence forms a word from W ?
- Optimization Question
 - Also given some "weight" on each word in W
 - How can we maximize the grid's weight by filling all 0's of the matrix A with words from W so that every maximally contiguous horizontal and vertical sequence forms a word from W ?

Intelligent Heuristic Explanation

(Based on concepts and algorithms presented in [1])

1. Store a list of potential words for each word space
2. Insert a word into the word space with the smallest word list, update all word lists
3. If any word space has no words remaining in its word list, remove the most recently added word, remove that word from the word list for that space, revert the other word lists
4. Repeat steps 2 and 3 until the grid is filled with valid words

Advantages

- Demonstrated extreme performance improvements
- Prunes the search space much more quickly
 - A brute force approach can get stuck making thousands of changes to an already invalid filling
- The intelligent algorithm will always address a problem as soon as it arises

Disadvantages

- More complicated algorithm that is more difficult to implement and debug
- More challenging to verify that the algorithm will always find a solution if one exists and that the implementation is correct
- Unclear if an elegant formula for expressing the number of filled grids that we check using this approach exists

Further Research

- Test out different strategies to speed up our solving heuristic
 - Examine existing strategies and compare them to ours
 - Explore alternative ways to store the word lists
- Construct a solving algorithm that may attempt to take weighting of words into account
 - Implement heuristics to choose which words to insert considering preferred word lists from crossword publishers

Acknowledgements: We thank Swarthmore College, Swarthmore College Research Fund, and Individual Donors to Swarthmore College for supporting our research work during 2020 - 2021.