Distance in the Graph Model of a Sudoku Grid

Julia Beilke with Dr. Kim Factor
Department of Math, Statistics, and Computer Science, Marquette University

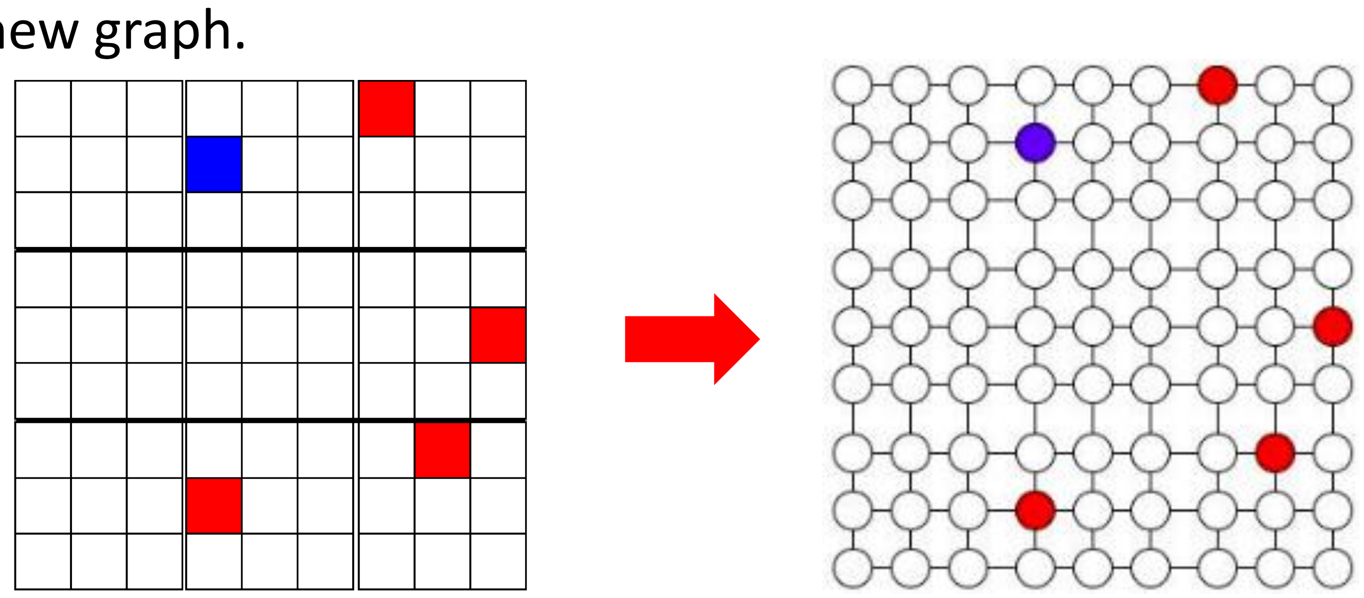
Introduction

Problem Definition:
Consider a 9 by 9 Sudoku grid.
If we know the positions of *k* instances of the same digit to be placed on the grid, in what order should we visit the squares such that we minimize the total distance covered in the path that is created?

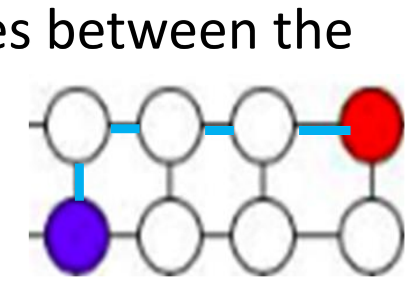
Background:
This problem is a subset of the Traveling Salesman Problem (TSP). There is no known algorithm to solve the TSP aside from an exhaustive search. We have utilized the constraints of a Sudoku grid to create a more approachable problem with fewer possible configurations.

Graphical Representation

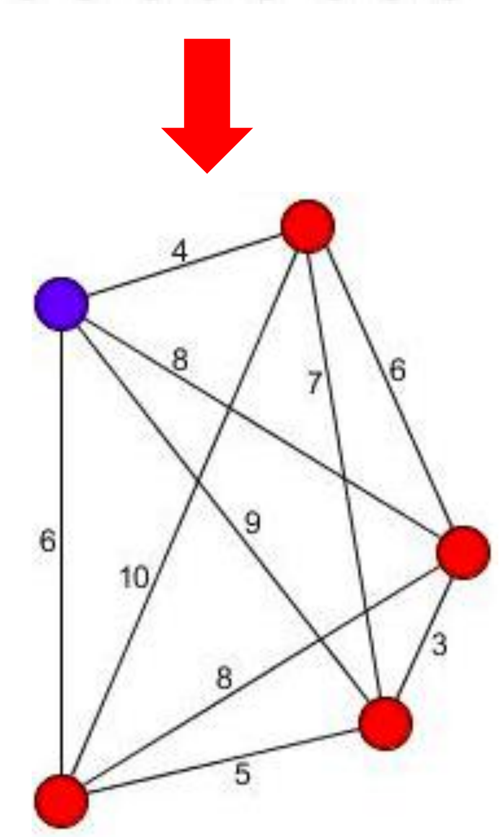
Sudoku grids are first represented as weighted graphs, where each square in the grid is a vertex and edges connect vertices whose corresponding squares share a side. Then, a weighted graph is created using only the filled in squares as vertices and the minimum distance between each pair of vertices in the first graph as the weights of the edges in the new graph.



Calculating distance: The weight of the edges in the final graph is calculated by taking the minimum number of edges between the vertices in the first graph.



Here, we must walk across at least 4 edges from the blue vertex to get to the red vertex, so the total distance is 4.

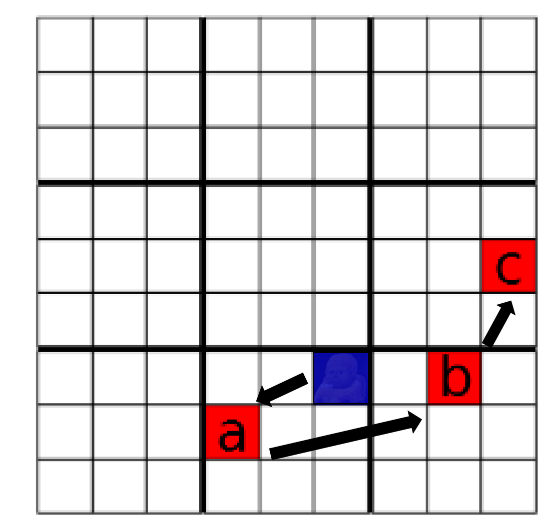


Heuristic Techniques with *k=3* Squares

Squares *in line*

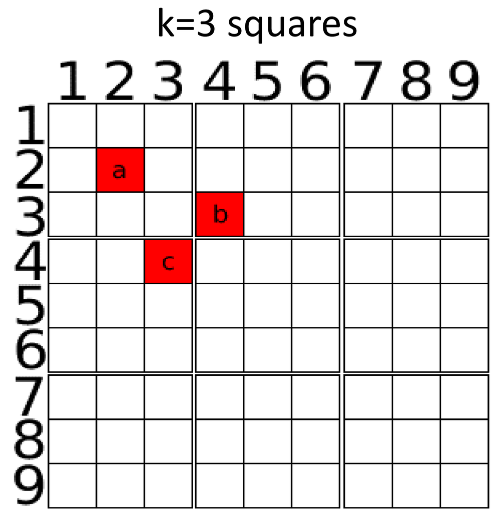
Definition. Squares are *in line* if the Cartesian x and y values of the squares in the configuration are either in ascending or descending order.

Heuristic. From any starting point s, visit the closest corner square first, then follow the in line path, which is also the greedy path, through the remaining points.

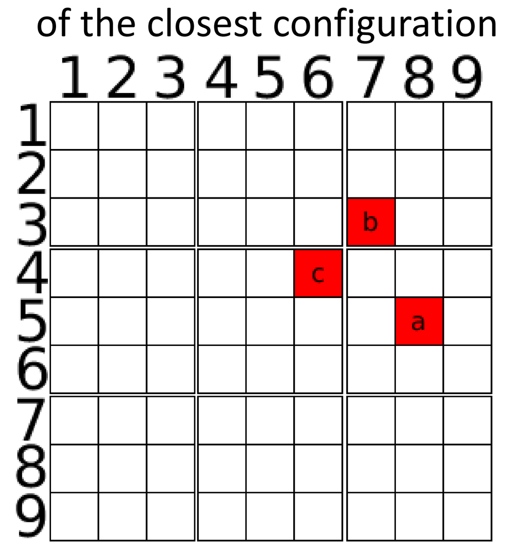


Closest Configuration and similar configurations

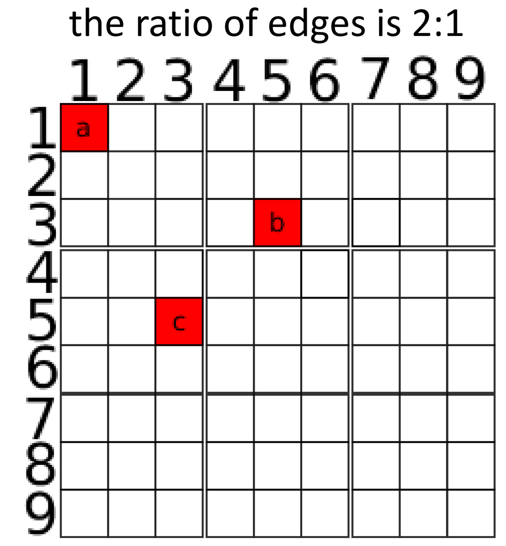
Closest Configuration of k=3 squares



Shifted and rotated version of the closest configuration



Similar triangle in which the ratio of edges is 2:1



Definitions. The *closest configuration* of *k=3* squares is the configuration in which the total distance between all three squares is as small as possible.

Similar configurations refer to *k=3* configurations in which the triangle formed by the squares is a shifted or rotated version of, or is a similar triangle to, the triangle formed by the squares in the *closest configuration.*

The *greedy approach* for choosing a path is the approach in which we continue to visit the nearest unvisited square to the square we are currently visiting, until all squares have been visited exactly once.

Theorem. For the *closest configuration* and all similar configurations of *k=3* squares, the greedy algorithm for selecting a path beginning at any starting point on the grid will yield the shortest path.

Configurations with *k=9* Squares

Findings
For any configuration of *k=9* squares, the sum of the weights of all edges in the weighted graph representation is 240. However, there are variances in the lengths of the shortest possible paths for different configurations.

Approaches
In order to prove that all configurations have the same sum of edges, the method used to date has been to try and show that no counterexample exists. This can be done with a proof by exhaustion. Because there are so many configurations, attempts have been made to show the statement is true with a computer- assisted proof. We have used a Java program that outputs every configuration, calculates the sum of the edges in the weighted graph, and returns false if any configuration has a sum other than 240.

Conclusions and Future Work

I have found algorithms for solving specific configurations of *k=3* squares. I have examined larger configurations, in particular configurations of *k=9* squares.
Future goals for this project include:

- Examining if all *k=9* configurations have the same sum of weights
- Classifying configurations of *k=9* squares
- Finding heuristic techniques for minimizing distance with *k=4,5,…,9* squares

Acknowledgements

I would like to thank my mentor, Dr. Kim Factor, for organizing this research project and offering her help and guidance this summer. I would also like to thank Dr. Dennis Brylow and Dr. Petra Eccarius Brylow for helping to create a positive summer research experience.

This research was funded by a National Science Foundation REU site grant, #ACI-1461264, 'Computation Across the Disciplines', at Marquette University.