Fencing Numbers

Group 8-03

Hu Buyan (3S2)

Daniel Lau (3A3)

Ai Zi Qi (3S1)

He Yue (3S2)

Hwa Chong Institution (High School)

1. Introduction

Fencing Numbers consists of n × n dots with the digits 0, 1, 2, or 3 written between within a cell formed by 2x2 vertices (Fig 1).

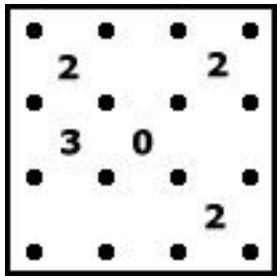


Fig 1

Each digit represents the number of line segments that will surround that square in a valid solution. For example, a square that contains a 3 would have line segments on three sides, and a square that contains a 2 would have line segments on two sides, and so on.

The following are the 3 fundamental rules in solving Fencing Number problems:

1.The value of the cell must be the same as the number of surrounding edges.

2. The loop can't leave the game field and can't cross itself.

3. There has to be exactly one loop

Fig 1a, Fig 1b and Fig 1c are some examples of solutions to Fig 1.

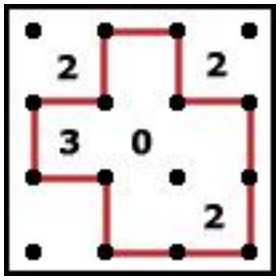


Fig 1a

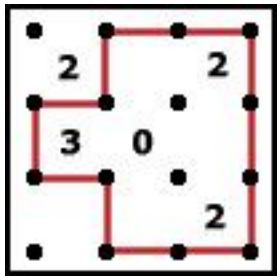


Fig 1b

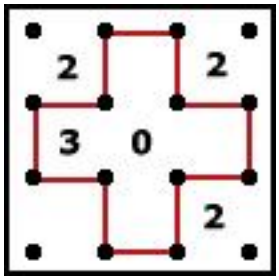


Fig 1c

1.1 Rationale

This topic has been researched on and the original question has been solved, however no general algorithm has been developed. As this topic offers many possibilities for expansion that is hardly researched upon, we can develop the problem according to our skill level.

1.2 Literature Review

Herting (n.d) presents an approach to Fencing Numbers based on precomputable rules, and is transferable to other problems. It also has a mathematical method of modelling Fencing Numbers so we can perform operations on it easier.

Susanti, Supatra, Lukas (2014) have also developed an algorithm for solving Fencing Number puzzles by using simplified mathematical models that separates each case into symmetrical segments represented by edges and vertices. Through imposing rules in increasingly restrictive order a solution could be obtained. However, it is only limited to a 2D equilateral triangular shaped cells in a rectangular grid. A similar approach for developing algorithms, where simplified representations are tackled systematically can be applied onto our research on an $n \times n$ grid, and other representations.

Liu, Wu, Sun (2012) managed to come up with an efficient algorithm for solving 40×35 puzzles in 1.7 seconds with programs. This was achieved by studying the aforementioned Herting's paper and optimising his algorithm via optimising deduction methods and more efficient methods for checking corners. However, it only works for specific grid sizes like 40×35, it provides insight on how we can optimise methodology used in solving fencing numbers in grid graph representation and even other graphical representations for our research. It also provides other methods, albeit less efficient, for solving it.

Ho, Tan (2019) presented finding matchings via graph colouring and the four-colour theorem. Once they had proven the chromatic number of their graphs, they could derive a pattern by solving using related theories. Although our fencing number problem is about closed paths and not matchings, we can look out for graph colouring solutions for our own problem to derive an algorithm too.

1.3 Terminology

Grid	Limited area of dimensions $n \times n$ where digits can be placed
n	Number of vertices every row/column of the grid, i.e. a grid has n x n vertices
Digit	A number placed in the grid that dictates the number of adjacent edges to it
Cell	Area where a digit is placed
Solve	Forming a closed loop without any violations

1.4 Objectives

1. To investigate the limitations of the digits placed in an $n \times n$ grid

2. To develop a strategy to solve fencing numbers problems

3. To expand the fencing numbers problems to different graph dimensions

1.5 Research Questions

1. To find the maximum possible sum of digits in an $n \times n$ grid such that it has at least one solution

2. To apply the strategies to solve an $n \times n$ grid of any size

3. To develop a rule-based approach to solve for Fencing Number problems on equilateral triangle lattice graph and king's graph

1.6 Fields of Math

- Geometry
- Graph theory

2. Methodology

Firstly we will research techniques and methods, including related theories and formulas. Then by applying the knowledge onto Fencing numbers we will solve each case systematically to observe generals patterns. Lastly, from our results, we can derive an algorithm by coding a programme using C++.

3. Results

3.1 Research Set 1

To achieve the maximal sum of digits in any valid Fencing number problem, a pattern of looping was derived by logical deduction. Vertices can only be connected to 2 edges for any closed loop to be formed. Since each inner edge increases the total sum of digits by 2, while an outer edge only accounts for an increase of 1 (Fig 2), inner edges are more important in this pattern.

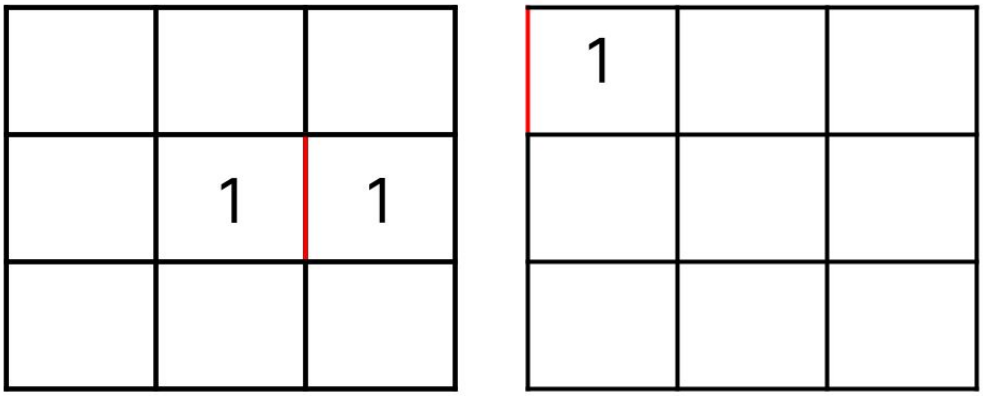


Fig 2

3.1a Odd

Corner vertices (in red) are connected to 2 outer edges thus accounting for 2 edges, border vertices (in yellow) are connected to 1 inner and 1 outer edge, thus accounting for a maximum of 3 edges while inner vertices (in green) are connected to 2 inner edges, thus accounting for a total of 4 edges. After calculation, the ideal total sum of digits is derived as $2n^2 + 2n$. (Fig 3)

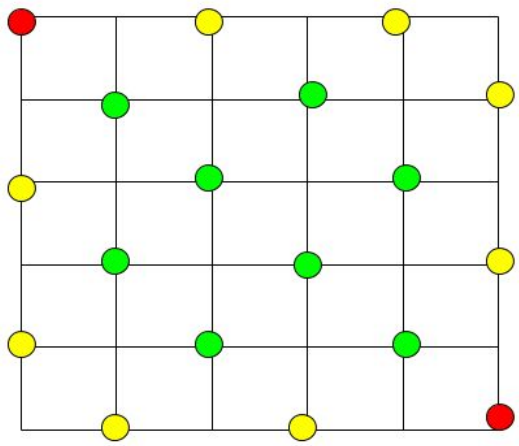


Fig 3

To avoid double counting, only alternating vertices are considered, where all vertices have to be connected by the closed loop. Therefore a horned pattern is the most optimal for odd sized grids (Fig 4)

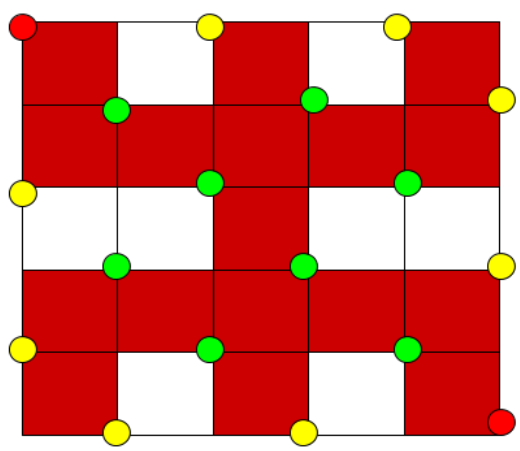


Fig 4

For two border vertices however, it can be seen that they cover only 2 outer edges instead of 1 outer edge and one inner edge. Hence an imperfection of -2 is inevitable for a closed loop to form, giving a final formula of

$$2n^2+2n-2$$

Proof

To prove the pattern we developed is the best, it suffices to prove that the 2 imperfect border vertices mentioned is unavoidable.

First, it is established that corner cells are necessary, to ensure that corner vertices have 2 outer edges connected to it, which adds 2 edges.

If a corner cell is present, there has to be an adjacent cell connecting it to other cells in the grid for a closed loop to be formed, which causes the imperfection. For all odd nxn grids, there will be 2 of those imperfections.

3.1b Even

For even, it is much more complicated as it contains 2 different possible vertice arrangements. (Fig 5a, Fig 5b)

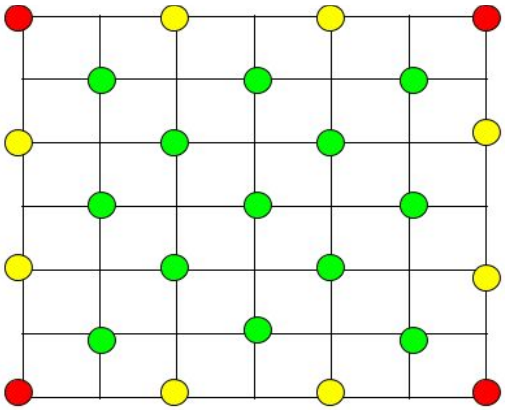


Fig 5a

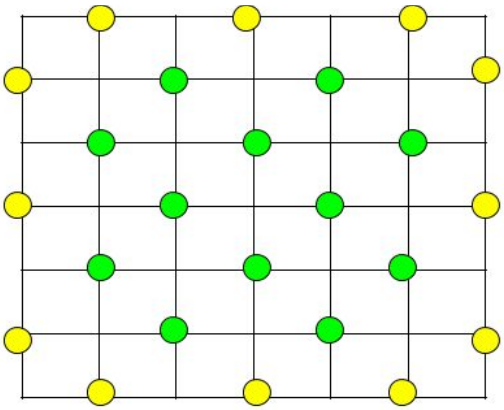


Fig 5b

The ideal sum is $2n^2 + 2n$ for both arrangements, as expected. However, there are some inevitable flaws for the corner and border vertices, since only closed loops are valid solutions.

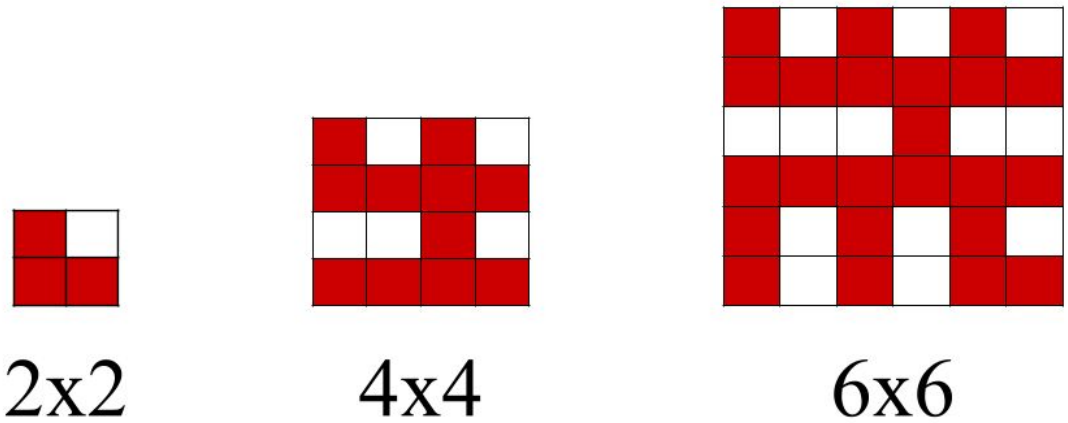


Fig 6

Based on the pattern we developed (Fig 6), the maximum possible sum of digits for an even nxn grid is:

$$2n^2 + 2n - 3$$

Proof

To prove that it is the best possible pattern, it suffices to prove that there are a minimum of 3 flaws, which are at the corner cells. Assuming that all corner cells are filled up, the vertice arrangement on the left (Fig 5) will be perfect, but it sacrifices on the vertice arrangement on the right (Fig 5), as each corner cell needs an adjacent cell, forcing each yellow vertice to have 2 outer edges, which amounts to a subtraction of 4 from the perfect arrangement. If 3 corner cells are filled up, the left arrangement (Fig 5) will have subtracted by 2, while the right will be subtracted by 3. If 2 or less corner cells are filled up, it will be subtracted by 4 or more. Thus, the best case scenario is with 3 corner cells filled up, where it is only subtracted by 3 for the worse arrangement. Hence, the equation will be 3 less than $2n^2 + 2n$, which is $2n^2 + 2n - 3$.

3.2 Research Set 2

Herting (n.d) suggested a rule-based approach which this study expanded upon. Different rules and conditions were developed which have varying dimensions and preconditions. By imposing rules onto any *n*x*n* fencing number problem together with a simple brute force algorithm, in increasing dimensions and conditions, an algorithm for solving can be formed that functions on both odd and even grid sizes.

There are 2 types of rules,
1. Static rules which do not require pre-existing edges nor crosses to apply
2. Preconditional rules which require pre-existing edges or crosses to apply

Since rules are applied in increasing conditions and dimensions, static rules are computed first, fulfilling prerequisites for the pre conditional rules to apply.

Some examples of the rules developed are as below.

Static rules (Fig 7)

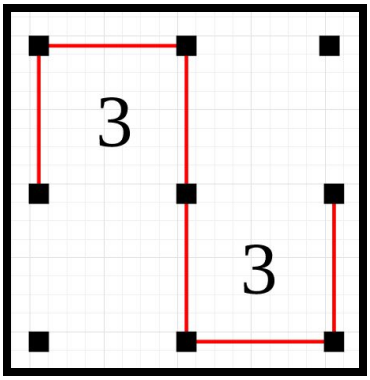


Fig 7i

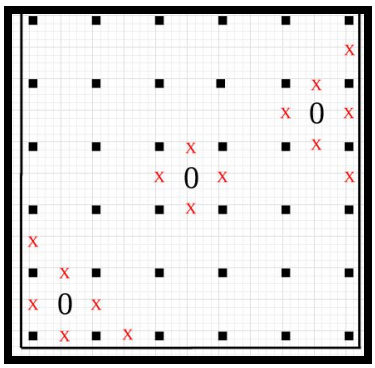


Fig7ii

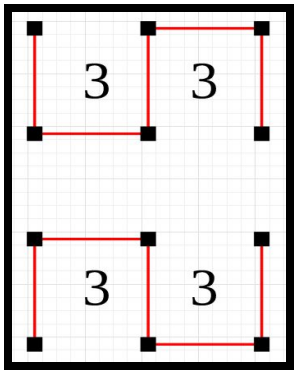


Fig 7iii

Static-corner rules (Fig 8)

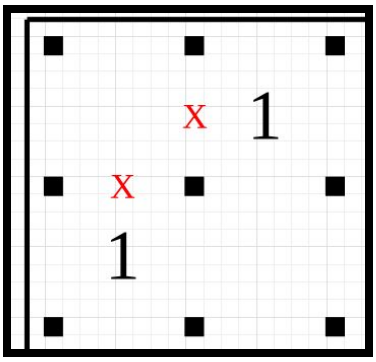


Fig 8i

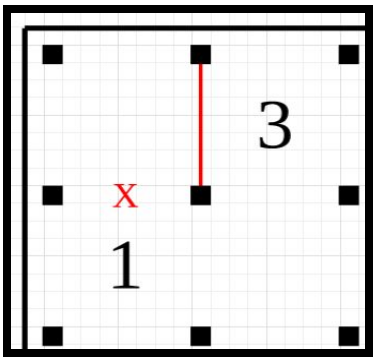


Fig 8ii

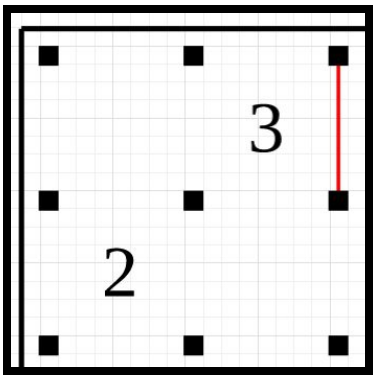


Fig 8iii

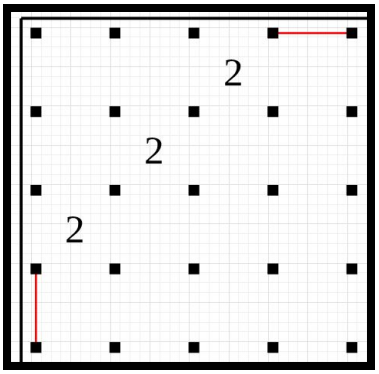


Fig 8iv

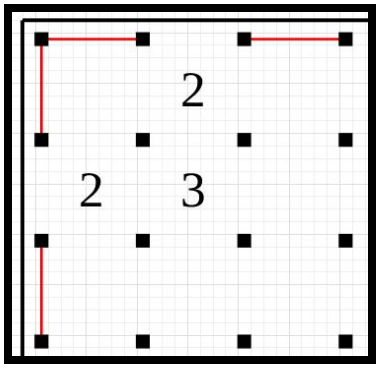


Fig 8v

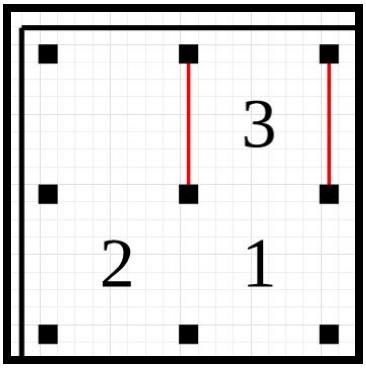


Fig 8vi

Pre-conditional rules(Fig 9)

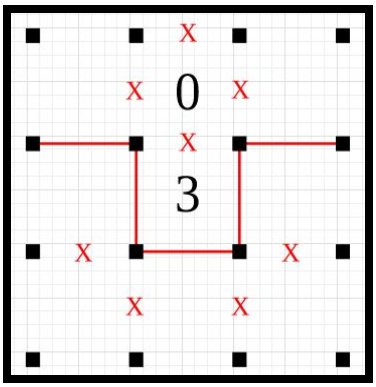


Fig 9i

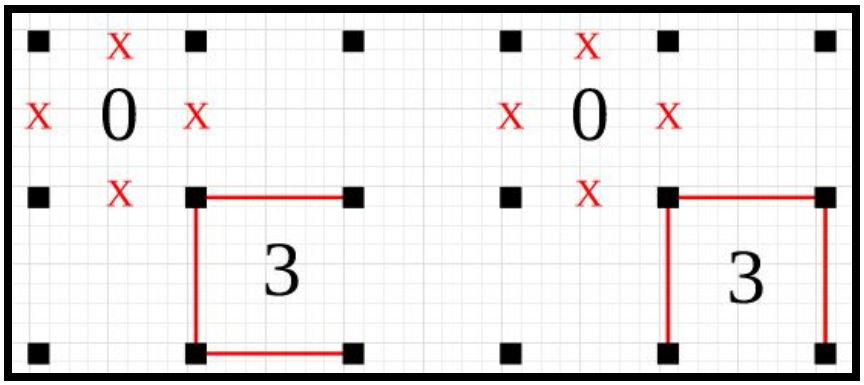


Fig 9ii

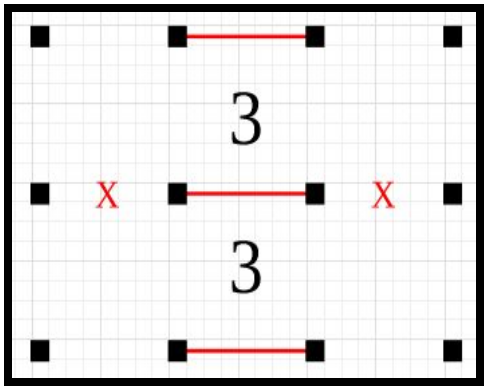


Fig 9iii

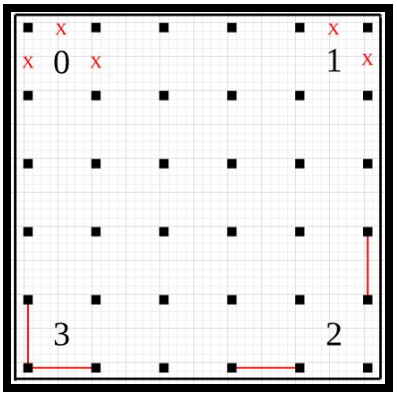


Fig 9iv

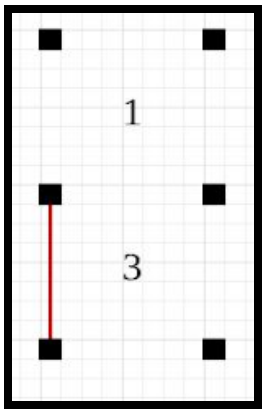


Fig 9v

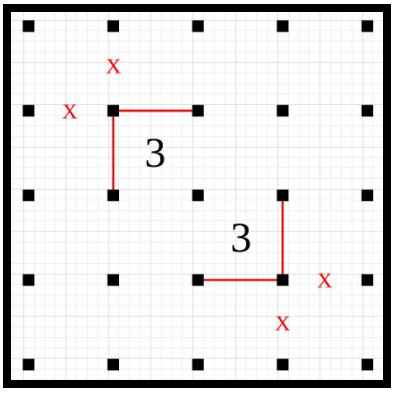


Fig 9vi

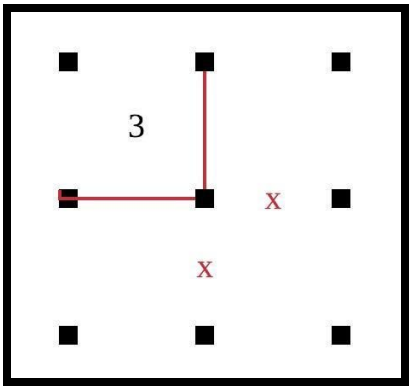


Fig 9vii

3.3 Research Set 3

3.3.1 Regular shapes

This study expanded fencing numbers and our algorithms to different graph dimensions and shapes. Any tesselatable regular shape and graphs with a fixed vertice to edge ratio are viable.

Equilateral triangles:

Using the same method of accounting via vertices, the optimal pattern for equilateral triangle graphs was found (Fig 10)

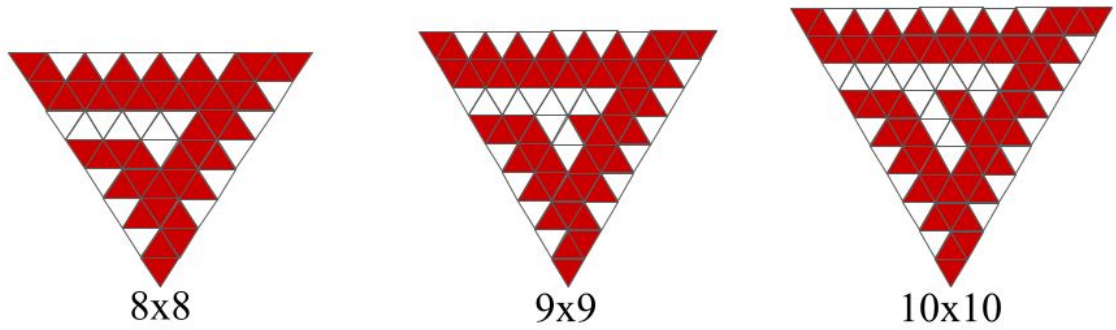


Fig 10

Proof

First, in order to get the most optimal shape for the equilateral triangles, we need to ensure that all vertices are connected to 2 edges. In the shapes above, all the vertices have 2 edges

3.3.2 King's graph:

King's graph feature a central vertices allowing diagonal edges (Fig 11)

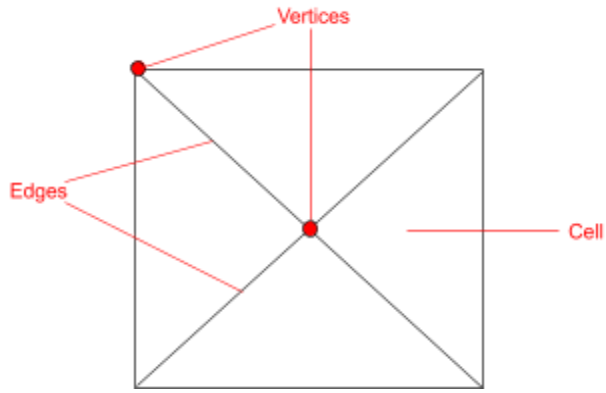


Fig 11

Once again, by accounting for vertices, the optimal pattern for King's graphs was found (Fig 12)

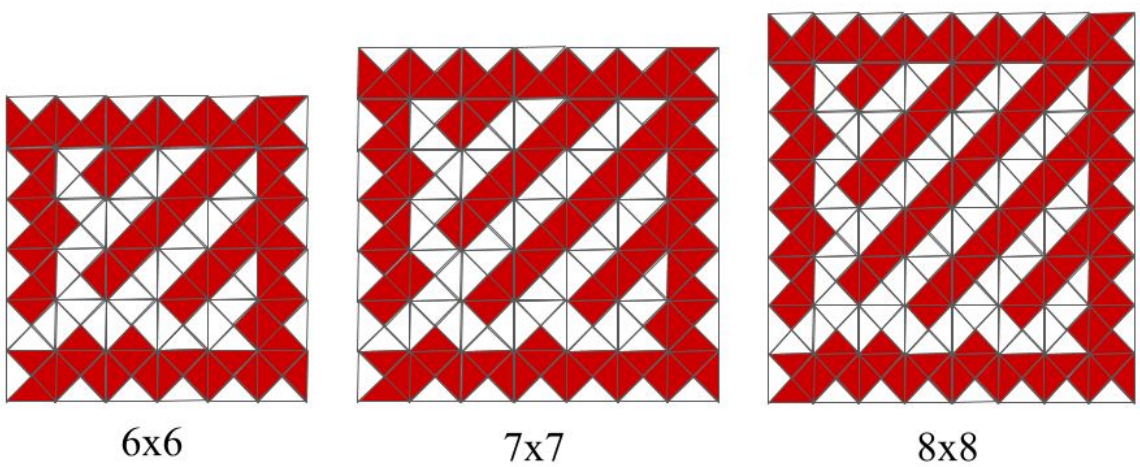


Fig 12

The formula for the border inner edges is $16n - 16$, as each side, there are sets of outer inner edges, and there are 4 sets of them, we are able to get, as each set have 2 inner edges and each inner edge is worth 2 points, we are able to get $16n - 16$.

Secondly, there are 4 outer edges, adding a constant 4 to the overall formula

Thirdly, the number of abnormal triangle increase with the formula $4n - 8$ as the number of abnormal triangle is in each graph is $n - 2$ and as each abnormal triangle have 2 inner edges and each inner edge has 2 points, thus the formula will be $4n - 8$.

Fourthly, the number of stripes is $2(n-3)(n-2)$. For 4x4, the number of inner edges of the stripes is 4, for 5x5, it is 4+8, for 6x6, it is 4+8+12, for 7x7, it is 4+8+12+16, and so on. This can be represented as 4(1+2+...+k), where k is a variable. From the pattern observed, it can be seen that $k = n$-3. Thus, it is 4(1+2+...+[n-4]+[n-3]), which is equal to 2(n-3)(n-2). Since each inner edge is 2 points, the formula is 4(n-3)(n-2).

Lastly, the number of uncounted edges equates to $4n-6$.

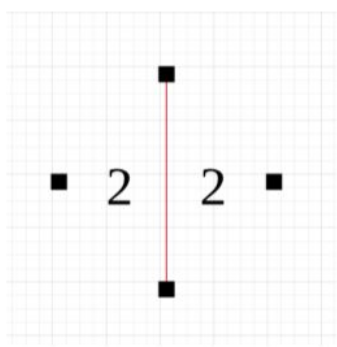
Taking the total sum, we are able to derive the formula

$$4n^2+4n-2.$$

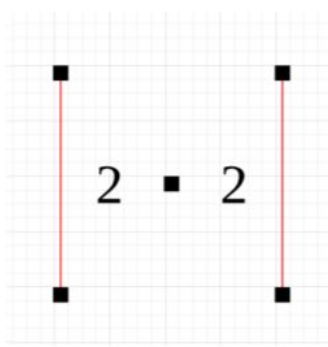
3.3.2 Rule-based approach (King's graph):

Rules for King's graph each come with multiple variations due to the fact that differing orientations of adjacent cells can be considered. The maximum digit placed in each cell is also limited to 2 instead of 3.

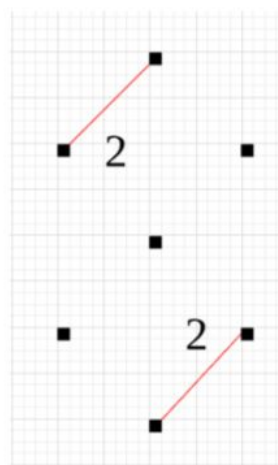
Static rules (Fig 13)



Horizontally Adjacent 2s #1



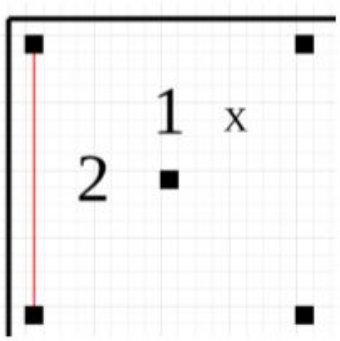
Horizontally Adjacent 2s #2



Diagonally Adjacent 2s

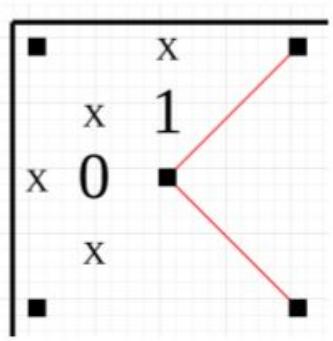
Fig 13 i Fig 13ii Fig13iii

Static corner rules (Fig 14)



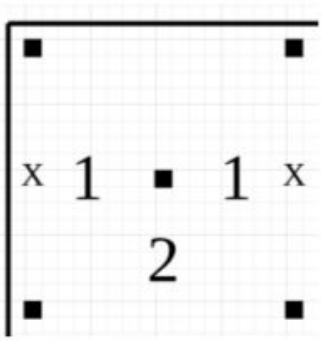
2-1 corner

Fig 14i



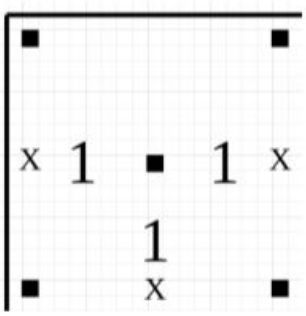
1-0 corner

Fig 14ii



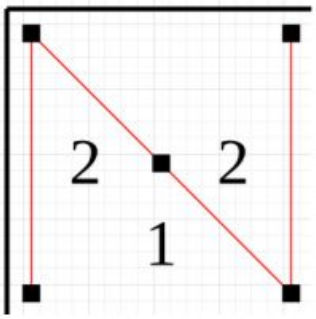
1-2-1 corner

Fig 14iii



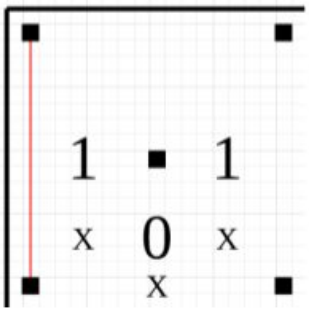
1-1-1 corner

Fig 14iv



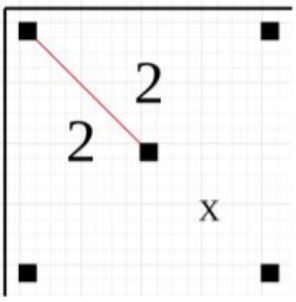
2-1-2 corner

Fig 14v



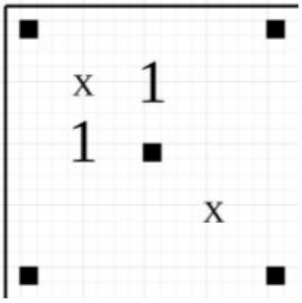
1-0-1 corner

Fig 14vi



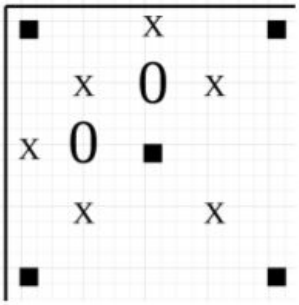
2-2 corner

Fig 14vii



1-1 corner

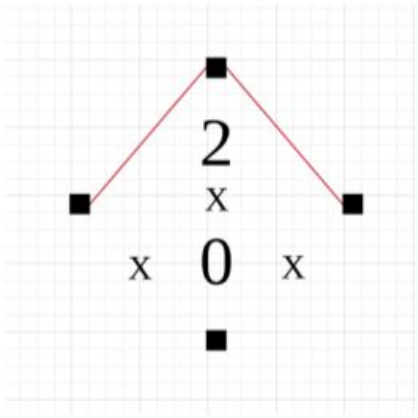
Fig 14viii



0-0 corner

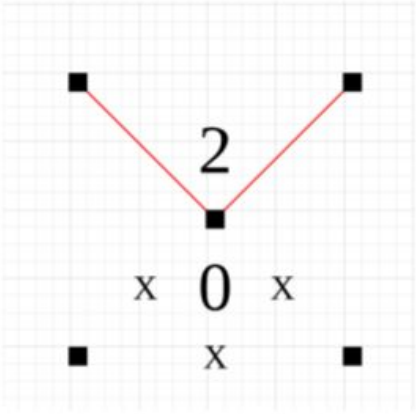
Fig 14ix

Pre-conditional rules (Fig 15)



Adjacent 0 and 2 #1

Fig 15i



Adjacent 0 and 2 #2

Fig 15ii

4. Applications

Algorithms derived in this study can be applied in ERP systems (enterprise resource planning) used by modern enterprises to compute and sort their resources. Manpower allocation can be sorted using this studies' algorithm.

Let the vertices represent each employee, arranged in order of hierarchy within management of the enterprise. Let the edges represent a unit of task and the digits represent the total units of task allocated to each chain of management. The maximum workload can be derived using our method from research set 1 and by using the rule based algorithm to solve the data given in a format of fencing numbers, the closed loop ensures even workload. (Fig 16)

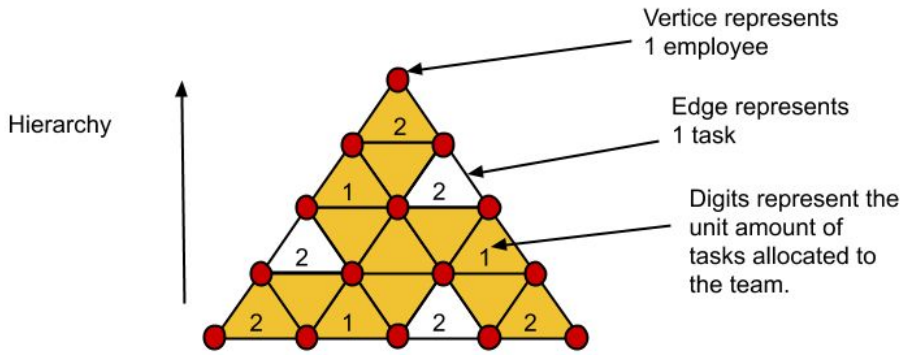


Fig 16

5. Limitations

Problem and algorithm can only be expanded to
1. Regular shapes that can tessellate
2. Specific graphs with a constant vertice to edge ratio

6. Conclusions and Future Extensions

By accounting for edges using vertices, the maximal sum of digits possible for each graph dimension can be easily derived, along with the pattern. By using a rule- based approach, fencing number problems of any viable graph and dimension can be solved when coupled with a brute force algorithm.

Some possible future extensions include:

- expanding to rectangular grids of n x m
- Multiple shapes for cells in a grid
- More efficient algorithms for solving each graph

7. Acknowledgements

We would like to thank our mentor Dr Ang Lai Chiang for his continuous support and feedback. We also express our gratitude towards our peers and the project judges for their mathematical inputs and constructive criticism.

8. References

Ho, L.I., Tan, K.H. (2019, 7 March) Regular Matchstick Graphs. Proceedings of Singapore Science and Engineering Fair 2019

Herting, S. (n.d.) A Rule Based Approach to the Puzzle of Slither Link. Retrieved from http://docplayer.net/79599331-A-rule-based-approach-to-the-puzzle-of-slither-link.html

Liu, T., Sun D.J., Wu, I. (2012, Nov 16) Solving the Slitherlink Problem. Retrieved from https://ir.nctu.edu.tw/bitstream/11536/20963/1/000313560200046.pdf

Samuel, L., Susanti, Supatra, K.V.I., (2014, May 30). Solving Logical Puzzles Using Mathematical Models. Retrieved from https://www.researchgate.net/profile/Samuel_Lukas/publication/232637138_Solving_Logical_Puzzles_Using_Mathematical_Models/links/09e4150889105550c1000000/Solving-Logical-Puzzles-Using-Mathematical-Models.pdf

Westreicher, D. (2011, Nov 16) Slitherlink Reloaded. Retrieved from https://david-westreicher.github.io/static/papers/ba-thesis.pdf