

Improved Mapping Strategies for Integrated Oscillator-Based Ising Machine Networks

Boyang Li

Technical University of Darmstadt, Darmstadt, 64295, Germany

boyanglitud@outlook.com

Abstract. Oscillator-based Ising machines (OIMs) are computational devices designed to tackle complex optimization problems by utilizing an array of interconnected oscillators. OIMs have the potential to outperform classical computing methods in both speed and energy efficiency, particularly when addressing optimization problems that prove challenging for traditional computers. One optimization problem involves mapping the nodes and edges of a random problem graph onto the Ising network. This paper aims to find a proper mapping strategy to solve this optimization problem. The proposed strategy supports large problem graph sizes and demonstrates a high success rate for medium-sized graphs, all while preserving the speed advantage inherent in oscillator-based implementations. The mapping process will be implemented using software-based methods and executed with precise visualization in MATLAB.

Keywords: Oscillator-based Ising machines, Ising network, Mapping strategy, Place and Route Strategy, MATLAB.

1. Introduction

Optimization problems play a crucial role in many applications, where the goal is often to find the best possible solution. These tasks typically include challenges like graph partitioning and finding the maximum cut [1, 2]. For problems with discrete variables, the set of possible solutions is countable. Although it is theoretically possible to test all solutions, the sheer number of possibilities makes this impractical in reality. Even advanced algorithms can struggle with such problems, especially NP-complete problems, which cannot be solved in polynomial time [3]. Heuristics offer a reasonable solution within a reasonable time frame, even though they may not be optimal. This approach is sufficient for many applications [4]. However, solving these problems remains computationally intensive. It requires significant time and energy resources. The new algorithm, which directly maps the problem graph nodes and the edge onto the instant network, still preserves the original graph parameters. It provides some exception-handling methods, which help the mapping strategy support large problem node sizes and increase the mapping success rate. The mapping algorithm is intended for OIMs but is also suitable for general use.

2. Related works

The mapping algorithm for an OIM network is designed to allow an almost 1-to-1 representation of the input problem nodes and hardware problem nodes. Some current mapping algorithms rely on heuristic

techniques and graph simplifications. The problem graph can be simplified by merging two one-hop neighbor nodes into one. The problem graph can also be simplified by splitting a single node with a degree of 1. If a path exists in the problem graph, the whole path graph will be replaced by a single edge [5]. Although these simplifications reduce the graph size and increase computational speed, defining the edge weights in the simplified graph remains a challenge.

Some algorithms rely on the spin value of the problem node. These algorithms partition the problem nodes into two sets of spin values (+1, -1) to minimize the Ising Hamiltonian function [6, 7]. While effective for small problem graphs, as the number of nodes increases, the algorithm needs more computer power to calculate the spin of each node, greatly increasing the execution time.

Other algorithms use neural networks to solve the optimization problem. Neural networks are promising for intelligent information processing because of their characteristics, such as generalization ability, parallel processing, and learning ability. An algorithm combined with an artificial neural network effectively solves such problems [8]. However, this approach requires extensive data to train the neural network and substantial computing power for the training process.

3. 2-D Ising network

An Ising network, or Ising model network, is inspired by the Ising model in statistical physics. In an Ising network, each graph node is represented by a physical oscillator [9]. The signal attached to the nodes represents the spin states of magnetic moments in the Ising model. These spin states are binary variables, either +1 or -1, where -1 represents a spin pointing down and +1 represents a spin pointing up. The algorithm focusses on finding the same structure on the Ising network with the problem graph and ignores each node's spin value at first.

This paper will introduce two common kinds of Ising networks, which are often used to test the mapping algorithm [4, 9]. The first one is the king's graph, which depicts the permissible movements of a king chess piece on a chessboard. The second network is a graph similar to the King's graph. In the advanced network, four additional edges between nodes with a distance of 2 in the horizontal and vertical direction of a King's graph are added. The advanced network has four additional edges compared to the king's graph. These additional edges will also decrease the disconnected region and increase local connections. The topology structures of the two common types of Ising networks with 36 nodes are shown in Figure 1. In the figure, the edge size of the king's graph is 110, while the edge size of the advanced network is 158.

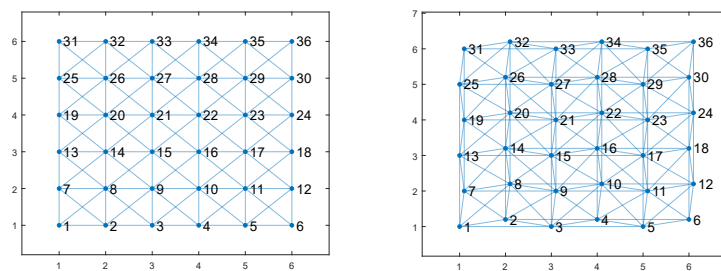


Figure 1. Two different kinds of Ising networks

4. Optimization problem

A random problem graph can represent optimization problems [10]. This section introduces the methods of generating a problem graph.

The first method is to generate a problem graph in which each node has a random number of degrees. The optimization problem graph should be simple and connected. Another method to generate a random problem graph is to get the sub-graph from the Ising network [11]. One advantage is that this problem graph can be mapped onto the Ising network, which allows for testing the effectiveness of the mapping algorithm.

Figure 2 shows examples of problem graphs with ten nodes and twenty edges. The left problem graph is generated entirely randomly, whereas the right problem graph is a sub-graph of the king's graph.

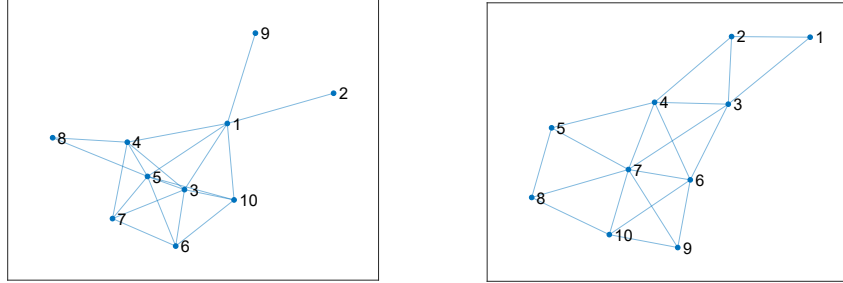


Figure 2. Two different kinds of problem graphs

In the next chapter, the mapping algorithm will be introduced, detailing how it maps the nodes and edges from the problem graph to the Ising network.

5. Mapping algorithm

The mapping process is mainly divided into two parts. In the first part, the algorithm attempts to map all the nodes and edges directly onto the Ising network. However, some nodes and edges might not be mapped onto the Ising network because of the approximation of the one-on-one mapping process. So, in the second part, the exception handling process and routing channel are used to map these unmapped nodes and edges onto the Ising network.

5.1. Basic mapping algorithm

The mapping process starts with finding the starting points for the problem graph. This node should have the highest centrality $C(v)$. The Centrality of node v is defined as:

$$C(v) = \frac{N-1}{\sum_u d(u,v)} \quad (1)$$

Where $d(u,v)$ represents the distance between node u and v , N represents the number of nodes of the problem graph. The starting node should be mapped in the center of the Ising network.

Each block of the king's graph is a complete graph k_4 , which consists of four nodes where each node is connected to every other node. The complete graph k_4 can be mapped onto the Ising network in one way. So, another option for choosing the starting points is to choose the complete graph k_4 with the highest sum of normalized closeness centrality to its neighbors.

After finding and mapping the starting points, the rest of the problem graph nodes will be mapped onto the Ising network through iteration. The edges between the mapped problem nodes should also be mapped at the same time. During this process, the method maps as many edges as possible.

- First, find the one-hop-neighbors of the mapped problem structure onto the Ising network.
- If there are multiple one-hop neighbors, the node with the largest number of connections to the mapped structure should be mapped first.
- If more than one node has the same number of connections to the mapped structure, the node with a higher degree should be mapped onto the Ising network first. If all one-hop neighbors have been mapped, calculate the new one-hop neighbors of the current mapped structure. In the problem graph, these new one-hop-neighbors of the mapped structure will be mapped onto the Ising network. Repeat the step above until all the nodes have been tried to be mapped onto the network.
- Due to the approximation mapping method, some nodes may not find a proper corresponding network node. These nodes will be mapped through the exception-handling process. If some edges

cannot be mapped onto the Ising network, these edges will be mapped through the routing channel on the Ising network.

5.2. Exception handling process

Because the size and connectivity of the Ising network are not infinitely large, some problem nodes may not find suitable network nodes for mapping. The exception-handling process will provide the non-mapped problem nodes with more free network nodes that can be used for mapping.

The exception-handling process is divided into three parts. First, move the whole mapped structure in vertical, horizontal, diagonal, and off-diagonal directions. If the mapped structure reaches the border of the Ising network, moving it can create new free spaces for mapping. This process helps generate additional mapping opportunities when a new problem node needs to be mapped as a neighbor of the existing structure.

Second, swap two one-hop neighbors with only one connection to the mapped structure. For example, if nodes u and v each have only one connection to the mapped structure on the Ising network, and node u is surrounded by mapped nodes while node v has more free nodes nearby, swap the positions of nodes u and v if another node needs to be mapped as a neighbor of node u .

Third, choose a K -hop neighbor. If node u is blocked by other nodes and no free neighbor nodes are available for mapping, choose a k -hop neighbor on the Ising network as the mapped node. The k -hop neighbor should have the lowest sum of distances to the mapped structure.

The three methods mentioned above can be used to handle exceptions without introducing new issues. The structure maintains the same topological relationships and connection positions, while spin is also unaffected by the change of the mapped positions. In other words, the exception-handling process provides greater flexibility without altering the overall topological structure.

5.3. Routing channel

A node may get blocked by other nodes and be unable to find a free edge during the mapping process. This problem can be solved by adding a routing channel between the network nodes. The routing channel will provide free edges for each node, allowing paths in the routing channel to represent non-mapped problem edges [12, 13].

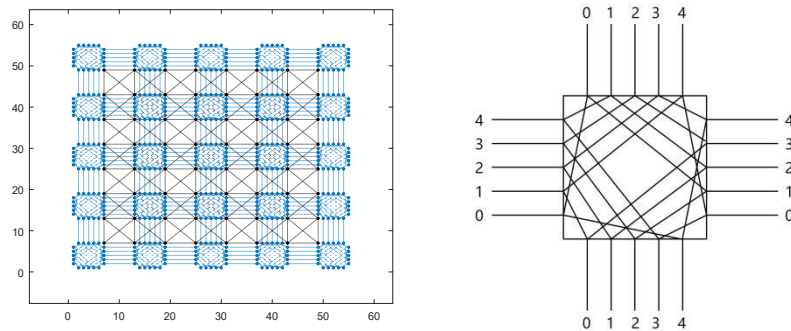


Figure 3. Ising network with routing channels and routing blocks

Adding a routing channel to the network significantly increases the number of nodes within the network, as shown in Figure 3. The black-colored nodes are the network nodes, while the blue nodes denote routing channel nodes. In a graph, the nodes' labels are also changed when nodes are added or removed. The routing blocks ensure the routing path can either go horizontally or vertically. The routing block is designed by Steven J.E. Wilton [14]. The set of edges in the routing block S is defined as follows:

$$S = \prod_{i=0}^{W-1} \{ (t_{0,i}, t_{2,i}), (t_{0,i}, t_{1,(W-i) \bmod W}), (t_{1,i}, t_{2,(i+1) \bmod W}), (t_{1,i}, t_{3,i}), \\ (t_{2,i}, t_{3,(2W-2-i) \bmod W}), (t_{3,i}, t_{0,(i+1) \bmod W}) \} \quad (2)$$

Where W represents the width of the routing channel W and t represents the channel ordinal number.

To prevent congestion, a routing penalty is introduced. Routing penalty refers to the additional time or delay incurred in transmitting data between two points due to the presence of intermediate nodes or routers in a network. A routing penalty occurs when the path between two nodes requires a longer route or involves more interconnects than necessary. In the network, the delay can be ignored. A routing penalty will be added to the routing channel when one of the channels is used for mapping. The weight of other edges in the channel will be determined by a certain number related to Length_x .

The routing path should have the lowest value Cost, which is related to the sum of edge weight and the number of nodes used for the routing path.

$$\text{Cost} = \text{Length}_x \times M + \sum_{i=1}^M e_i \quad (3)$$

The set of edge weights is E , where the edge is the routing path between two network nodes that use the routing channel for mapping, and the element of the set E is e . The number of edges in the routing channel used for mapping is M .

Figure 4 shows a mapping result with a random generated problem graph with 30 nodes and 50 edges.

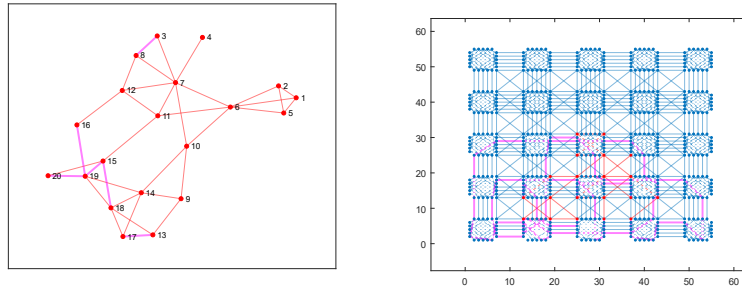


Figure 4. Routing results of a randomly generated problem graph

The red-colored edges in the problem graph indicate those mapped through the initial part of the mapping process based on connectivity. The magenta-colored edges represent those mapped through the routing channel. In the Ising network, the magenta-colored edges represent the mapped edges and the routing paths. The red-colored node is the corresponding mapped network node. The mapping process is considered successful based on this result.

6. Benchmark

Compared with the previous mapping strategy, which depends on heuristic techniques and graph simplifications, the current mapping strategy has dramatically improved its success rate. As shown in Figure 5, when the number of nodes in the randomly generated problem graph is lower than 140, all the nodes and edges can be mapped onto the Ising network (King's graph with four routing channels) using the current mapping strategy. When the number of nodes increases from 36 to 150, the success rate of the current mapping strategy is always 100%. In contrast, the previous success rate drops to 0 when the number of nodes reaches 50 for the heuristic mapping strategy. The current mapping strategy enormously increases the mapping success rate when the number of problem nodes is limited to 140.

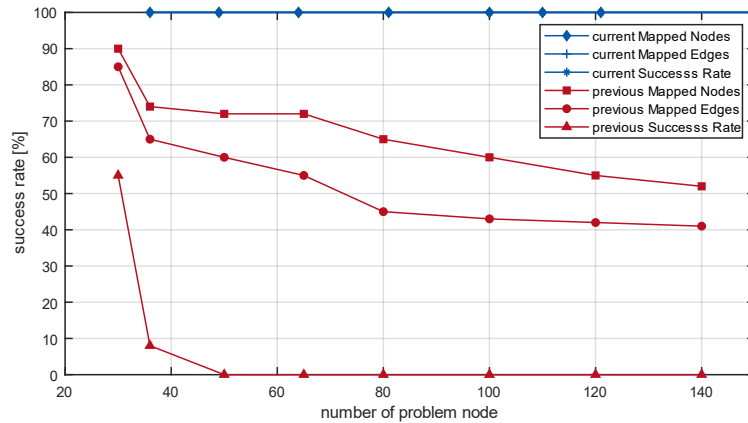


Figure 5. The comparison between the new and previous heuristic mapping strategy

This paper introduces two types of problem graphs — the randomly generated problem graph and a sub-graph of the Ising network. It is meaningful to compare the success rate for different kinds of problem graphs for the same number of nodes and edges. The success rate when the sub-graph is taken as the problem graph should be larger than the success rate if the problem graph is randomly generated. A good mapping algorithm should have similar success rates for these two problem graphs. For this evaluation, the width of the routing channel is set to 12.

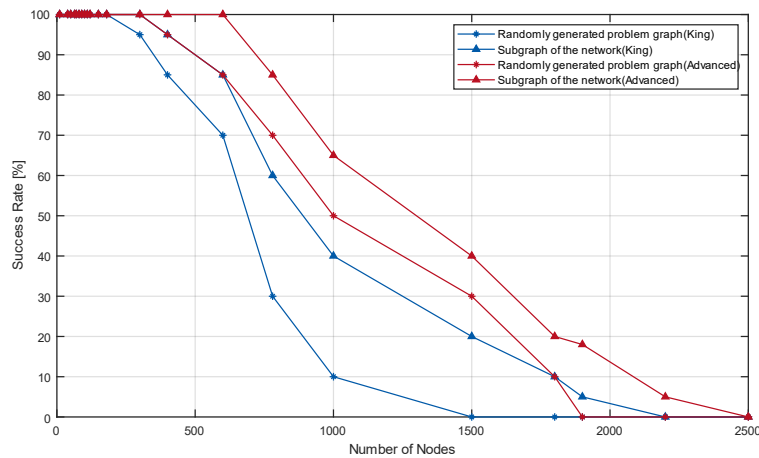


Figure 6. Success Rate for different kinds of Ising networks and problem graphs

Figure 6 shows the success rate for different kinds of problem graphs. The success rate for the sub-graph is higher than that for the randomly generated problem graph. However, the gap between these two success rates widens as the number of nodes increases, indicating that the mapping algorithm is not perfect and still has room for improvement.

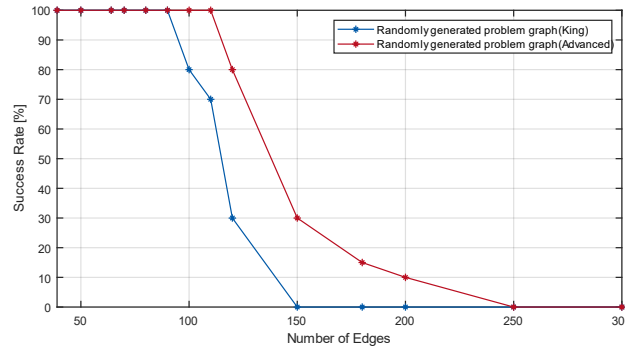


Figure 7. The success rate for different densities of the problem graph

The success rate for dense and sparse graphs is also different. Figure 7 shows that, with a set number of 40 problem nodes, the success rate of the mapping algorithm decreases as the number of edges increases.

Regarding execution time, when the number of nodes is fewer than 200, the average execution time is under 30 seconds, which is acceptable. However, when the number of nodes is larger than 300, the average execution time is considerable, which still needs improvement. Figure 8 demonstrates that the execution time for the randomly generated problem graph is similar to that for the network sub-graph. When the number of problem nodes is lower than 400, the network structure does not affect the runtime. The run time depends on the problem node size. When the number of problem nodes exceeds 400, the advanced network has a significantly higher number of nodes compared to the king's graph, resulting in increased execution time for the advanced network.

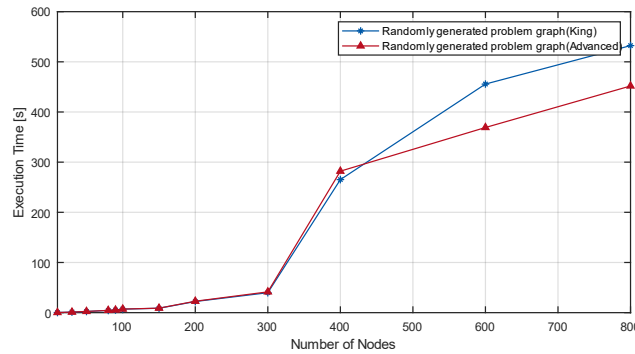


Figure 8. The execution time for different kinds of Ising network

7. Conclusion

This paper introduces an algorithm to map a given optimization problem onto a specific hardware structure. However, the success rate of the mapping process decreases as the problem node size increases. Compared with the previous mapping strategy, the new mapping algorithm has shown significant improvement in the success rate, enabling more nodes and edges from the problem graph to be successfully mapped onto the Ising network. Additionally, the mapping algorithm performs well when dealing with sparse graphs but encounters difficulties when mapping dense graphs onto the Ising network due to limited routing channels. However, the mapping strategy still requires improvement. The node selection process significantly limits the success rate of the new algorithm. Moreover, while the runtime of the mapping process is manageable with a small number of nodes, it increases significantly as the number of nodes grows, which is undesirable. Therefore, improvements are needed to address runtime issues. One potential enhancement for the mapping algorithm is to find more accurate ways of mapping nodes onto the Ising network. Currently, most nodes are mapped using approximations,

resulting in some edges being unable to be mapped directly onto the network. These not-mapped edges have to be routed through the routing channels, which decreases the overall success rate of the mapping algorithm and increases the runtime.

References

- [1] T. Wang, L. Wu and J. Roychowdhury, "Late Breaking Results: New Computational Results and Hardware Prototypes for Oscillator-based Ising Machines," 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2019, pp. 1-2.
- [2] A. Mallick, M. K. Bashar and N. Shukla, "Oscillator-based Dynamical Computing Platforms to Solve Combinatorial Optimization," 2022 6th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), Oita, Japan, 2022, pp. 125-127.
- [3] J. Lamers, G. Verschaffelt and G. Van der Sande, "How to Boost Ising Machines' Ability to Find Optimum Solutions: a Bifurcation Analysis," 2023 International Conference on Photonics in Switching and Computing (PSC), Mantova, Italy, 2023, pp. 1-3.
- [4] M. Graber and K. Hofmann, "An Enhanced 1440 Coupled CMOS Oscillator Network to Solve Combinatorial Optimization Problems," 2023 IEEE 36th International System-on-Chip Conference (SOCC), Santa Clara, CA, USA, 2023, pp. 1-6.
- [5] M. Graber, M. Wesp and K. Hofmann, "A Fast Graph Minor Embedding Heuristic for Oscillator Based Ising Machines," 2022 Austrochip Workshop on Microelectronics (Austrochip), Villach, Austria, 2022, pp. 41-44.
- [6] A. Mallick, M. K. Bashar and N. Shukla, "Oscillator-based Dynamical Computing Platforms to Solve Combinatorial Optimization," 2022 6th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), Oita, Japan, 2022, pp. 125-127.
- [7] Y. Su, T. T. -H. Kim and B. Kim, "A Reconfigurable CMOS Ising Machine With Three-Body Spin Interactions for Solving Boolean Satisfiability With Direct Mapping," in *IEEE Solid-State Circuits Letters*, vol. 6, pp. 221-224, 2023.
- [8] Y. Shao et al., "Multi-Objective Neural Evolutionary Algorithm for Combinatorial Optimization Problems," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 2133-2143, April 2023, doi: 10.1109/TNNLS.2021.3105937.
- [9] M. Graber and K. Hofmann, "A Versatile & Adjustable 400 Node CMOS Oscillator Based Ising Machine to Investigate and Optimize the Internal Computing Principle," 2022 IEEE 35th International System-on-Chip Conference (SOCC), Belfast, United Kingdom, 2022, pp. 1-6.
- [10] I. Ahmed, P. -W. Chiu, W. Moy and C. H. Kim, "A Probabilistic Compute Fabric Based on Coupled Ring Oscillators for Solving Combinatorial Optimization Problems," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 9, pp. 2870-2880, Sept. 2021.
- [11] N. Yoshimura et al., "Efficient Ising Model Mapping for Induced Subgraph Isomorphism Problems Using Ising Machines," 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 2019, pp. 227-232.
- [12] M. -H. Chen, Y. -W. Chang and J. -J. Wang, "Performance-Driven Simultaneous Partitioning and Routing for Multi-FPGA Systems," 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2021.
- [13] M. A. Zapletina and D. A. Zheleznikov, "The Acceleration Techniques for the Modified Pathfinder Routing Algorithm on an Island-Style FPGA," 2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), Saint Petersburg, Russian Federation, 2022.
- [14] Wilton S J E. Architectures and algorithms for field-programmable gate arrays with embedded memory[M]. Toronto, Ontario, Canada: University of Toronto, 1997.