Potts Model Formulations of Combinatorial Optimization Problems

-Could include some results, not comparative, just to show that it works.

-In that case, could include FPGA placement tasks

-Should it include comparisons to ising model formulations? No… would complicate the message, create an opportunity for conflict, and besides, I don’t even want to think about all the Ising model issues. Yuck. Well you can mention it, but don’t bother explaining it or showing “results” for the ising model. Yeah, that sounds good. Focus on you.

# Abstract

It is well known that combinatorial optimization problems can be solved by encoding constraints into the weights of an Ising model []. This principle has been the foundation of much research into combinatorial solver accelerators based on various principles [], all creating stochastically bistable circuit constructs called P-Bits. Solving combinatorial optimization problems by encoding constraints into the weights of a Potts model is also possible [], but less commonly understood. In this paper we demonstrate how several different optimization problems can be encoded as a Potts model, starting with simple problems like graph coloring and working up to a more complex, real-world example consisting of placing logic during FPGA synthesis, targeting the ICE-40 line of FPGAs. A theme of “weight kernels” emerges as the various combinatorial applications are considered, providing a natural mechanism for compressing the size of Potts model weight matrices in order to support efficient computation.

# Introduction

The ability to encode combinatorial optimization problems in the weights of an Ising model was first realized in [].

# A general approach to Potts model encodings

Each pair of nodes in a Potts model is connected by not just a single weight, as in an Ising model, but a whole matrix of weights.

# Graph Coloring

Graph coloring is perhaps the simplest and easiest Potts model problem to solve.

# Traveling Salesman

As with the

# Reversible Logic

Reversible logic is our first example of a Potts model formulation that has an inhomogeneous collection of q-sizes.

# FPGA Placement

The FPGA placement task provides an example of how Potts models can be applied to a real optimization problem.

# Conclusion

The Potts model representation of combinatorial problems is a very flexible one. Its inherent multi-valued variables allows a clear pathway to encode an optimization problem as a Potts model.

# Code Availability

Examples of all discussed Potts model encodings are available in a published python package, “PottsPlayground”, which can be installed with pip. The package independently generates graph coloring and traveling salesman problem instances, and can work cooperatively with the ICESTORM toolchain to generate Reversible Logic instances and to solve real FPGA placement tasks.

References