Project Proposals

1 Qualitative usibility analysis

Investigate, use, and document the hypergraph packages, specifically their ability represent dense calculated hypergraphs. Benchmark the known hypergraph packages against baseline data, problems, and algorithm. Perform a qualitative assessment for ease of use, features, and provided algorithms. Ensure new algorithms can be easily implemented in the software package.

2 Area survey of structural representation

Survey all the known hypergraph algorithms: geometric reduction, min-cut partitioning, shortest path and compare the results of using the hypergraph version of algorithms against standard graph versions of the same algorithm. Investigate the implementation source to determine representations and document the differing characteristics of representation. Sparse, Dense, and Matrix representations internally change the possible algorithms that can be written against the structures.

3 History of hypergraphs

Profile all the known usages of hypergraphs, problems, representations and nomenclature used to describe hypergraphs and related technology. There are multiple problems from many domains that map directly to hypergraphs. Construction of a document detailing the discovery and usage of hypergraphs in the multiple domains would be the artifact of work. The effort would also detail the future directions that research could take that has not yet been explored.

4 Reaction network crawlers

Investigate, build and train reaction network crawlers to find reactions that have higher reaction energy. These crawlers could leverage the odometer / hyper-edge transition, using the hyperedge to create a heuristic to guide the odometer path search. The measurable results would be to compare the best high-energy reactions against the limited-BFS currently implemented.

5 Neural network simulation of dense hypergraph

Use the dense hypergraph to train a neural network to simulate the reaction calculations. The coefficients must be perfect whole round numbers so the ratios that come back must carry the coefficient perfectly. This difference between the neural network output and coefficient would be used to calculate the exact training data. Using this technique, the network could be trained to 99.9999

6 Implement parallel hypergraphs in openCL

Implement a parallel version of hypergraphs in C/openCL using odometers for representation. Demonstrate that hyperedge access time is a function of the number of nodes in a given hyperedge. Show that this hyperedge access time is reducible over the number of processors via parallelization benchmarking.

7 Battery measurments and modeling

Investigate the physical different points and characteristics to measure a battery. Build a model that has sample points at each of the key points. This will be used to compare models of batteries against in-situ batteries measurements. This training information would be used to condition a neural network to blend the model of the battery and the measurements to create a simulation that is closer to reality. The end artifact would be a system that is capable of accepting measured inputs and calculated model inputs and comparing the difference between them.