Project Proposal

Hypergraphs made their first appearances in the mid 1960's with the earliest books on the subject mater being published in the late 1980's [2]. Hypergraphs are increasingly relevant because of their ability to model complex subset interactions [12]. Hypergraphs can easily be shown to contain 2^N hyper-edges; the number of hyperedges is controlled algorithmicly and can vary over:

$c \leftrightarrow log(N) \leftrightarrow N \leftrightarrow Nlog(N) \leftrightarrow N^2 \leftrightarrow 2^N \leftrightarrow N^N \leftrightarrow N^M$

Hypergraphs can model every known graph, along with every path through the graph; the problem becomes finding the optimal hypergraph ordering and corresponding hyperedge [1] [5]. Odometers [6] play a key role in this implementation and explicitly allowing the implementer to control the enumeration places infinite power in the users hands. If the enumerator does not terminate, then the hypergraph enumerator will not terminate as well.

Hypergraphs and hyperedges are fully parallelizable and can even be used to model how to cluster parallelized computation [9] [15] [4]. Clustering graphs into subgraphs is known to be NP complete, yet any solutions to the problem is still treasured [14] [11] [13]. All of the solutions, although not optimal, can be shown to be derived from some type of branch and bound mechanism [3]. These solutions can further be shown to be related to some Hilbert-space filling curve that encounters a solution earlier as a concrete heuristic [8] [7] [10].

Hypergraphs have nomenclature that is widely varied. Multigraphs, hyperarc, hyper-edge, nodes and hyper-node are all similar yet distinctly different terms. Representations are just as varied with incident matrix, lists of nodes, and odometers. Odometer instances allow traversal of hypergraphs independently from other instances. Standard hypergraph mechanisms allow for conversion from Odometer to Hyperedge and vice-verse in constant time in the size(N)/size(processors(M)).

Dense Hypergraphs far larger then the known observable universe $2^{100} <= 2^{60,000}$ seem a rediculous notion until a question such as "return *ALL* of the possible unique reactions from a given set of compounds" as obeyed by mass laws ect. As it turns out it is possible to build a subset enumerator that simply instantiates a vector of integers that corrispond 1:1 with a hyperedge and vice versa as per the lookup table. A repeated subspace larger then the starter size corresponds to a N : 1 reduction. Repeats are akin to the unresticted permutations and combinations with repeats.

Build a hyper-graphs/hyper-net hyper-edge enumerator instanced with typed compounds using a breadth first search of the elemental compound indexs. This does not reduce the search space infact it adds $2^{Elements}!$. This is because the maximal combination of all $2^{Elements}$ is close to the size of the known universe we can assum that it will probably not complete, but rather be stopped by an external control once begun. But this sub enumeration arbitrarily divides the space with key knowledge. Fortunantly there is a branch and bound technique that uses matrix row reduced eschellon form over field of arbirary size ¡citation needed¿. After each node expansion if the equation that comes back as incomplete, unique, or linear combination allows the depth and bredth control to move forward over the hypernet of reactions. This is based purely on the coeffeciants of compunds in reactions.

Currently I would propose an in depth paper exploring the Odometer to Hyperedge to Hypergraph relationship. Odometers are numbered/indexed versions of hyperedges contained by a given hypergraph. These mechanisms allow for explicitly controlling enumeration, every hyperedge can be visited, a every hyperedge of a subset of hyperedges, or customized iterators based on an initial hyperedge can explore a dense hypergraph.

References

[1] Giorgio Ausiello, Giuseppe F. Italiano, and Umberto Nanni. Optimal traversal of directed hypergraphs. *University of Berkeley*, 1992.

[2] Claude Berge. Hypergraphs. *North-Holland Mathematical Library*, 1989.

[3] Jens Clausen. Branch and bound algorithms - principles and examples. 1999.

[4] Karen D. Devine, Erik G. Boman, Robert T. Heaphy, Rob H. Bisseling, and Umit V. Catalyurk. Parallel hypergraph partitioning for scientific computing. 2006.

[5] Aurélien Ducournau and Alain Bretto. Random walks in directed hypergraphs and application to semi-supervised image segmentation. *Computer Vision and Image Understanding*, 2014.

[6] Phillip P. Fuchs. Permutation odometers. 2016.

[7] J K Lawder. Calculation of mappings between one and n-dimensional values using the hilbert space-filling curve. 2000.

[8] Mohamed F. Mokbel, Walid G. Aref, and Ibrahim Kamel. Performance of multi-dimensional space-filling curves. 2003.

[9] Bálint Molnár. Applications of hypergraphs in informatics: A survey and opportunities for research. *Annales Univ. Sci. Budapest.*, 2014.

[10] Lars Relund Nielsen, Kim Allan Anderson, and Daniele Pretolani. Finding the k shortest hyperpaths: algorithms and applications. 2004.

[11] Communications of the ACM. How to partition a billion-node graph. 2014.

[12] Anna Rita and T.M. Murali. Pathway analysis with signaling hypergraphs. *Publications of the ACM*, 2014.

[13] Satu Elisa Schaeffer. Survey: Graph clustering. 2007.

[14] Zhao Sun, Hongzhi Wang, Haixun Wang, Bin Shao, and Jianzhong Li. Efficient subgraph matching on billion node graphs. 2012.

[15] Dengyong Zhou, Jiayuan Huang, and Bernard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *NEC Laboratories America, Inc.*, 2006.