HYPERGRAPH PARTITIONING

▶ STUDENTS / UNIVERSITIES

Ali Yıldırım/SU
Cavit Çakır/SU
Hüseyin Ahmet Cemil Özaslan/SU
Kaya Kapağan/SU

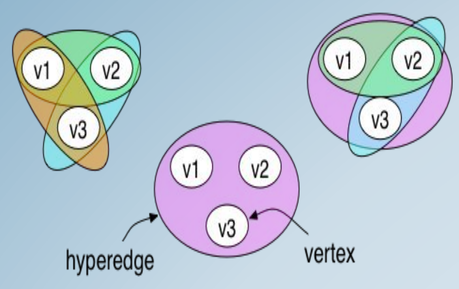
▶ SUPERVISOR(S)

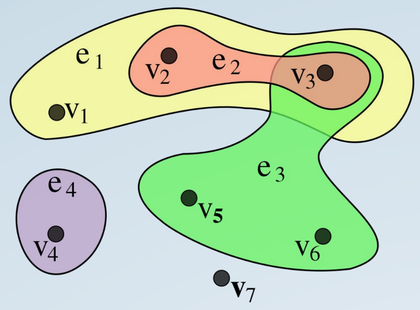
Kamer Kaya
Erdinç Öztürk

Sabancı Üniversitesi

pure PROGRAM FOR UNDERGRADUATE RESEARCH

WHAT IS 'HYPERGRAPH' ?





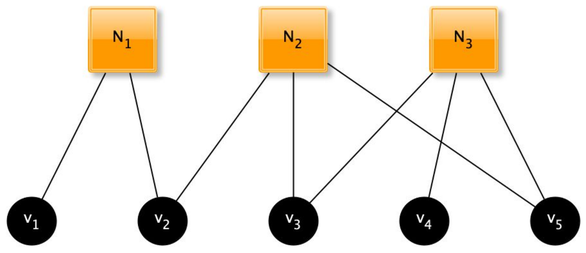
A *hypergraph* H = (V, N) is defined as a set of vertices V and a set of nets (hyperedges) N among those vertices. A net n ∈ N is a subset of vertices and the vertices in n are called its pins.

Figure 1: A hypergraph representation with 3 nets (N1, N2, N3) and 5 vertices (v1, v2, v3, v4, v5). Pin sets of the nets are given below.

N1 = {v1, v2}, N2 = {v2, v3, v5},

N3= {v3, v4, v5}

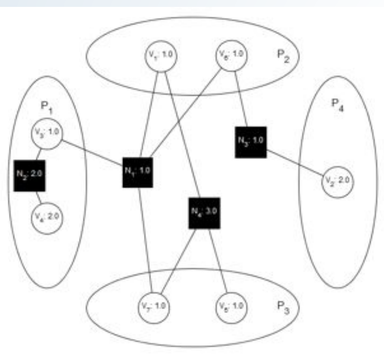
Figure 1



OBJECTIVE AND AIM OF THE RESEARCH

- Partitioning of a hypergraph
- 2 different research areas:
  - Fast Hypergraph partitioning (large Scale)
    - Kaya, Cavit
  - Optimal Hypergraph partitioning (small scale)
    - Cemil, Ali

METRICS



1. Cut Net
2. Total Message
3. Total Volume
4. Max send message
5. Max send Volume
6. Max send receive volume

HYPERGRAPH PARTITIONING

A K-way partition of a hypergraph H is a partition of its vertex set, which is denoted as ∏ = {V1, V2, . . . , Vk}, where

• parts are pairwise disjoint, i.e., Vk ∩ Vl = ∅ for all 1 ≤ k ≤ l ≤ K

• each part Vk is a nonempty subset of V, i.e., Vk ⊆ V and Vk ≠ ∅ for 1 ≤ k ≤ K,

• the union of K parts is equal to V,

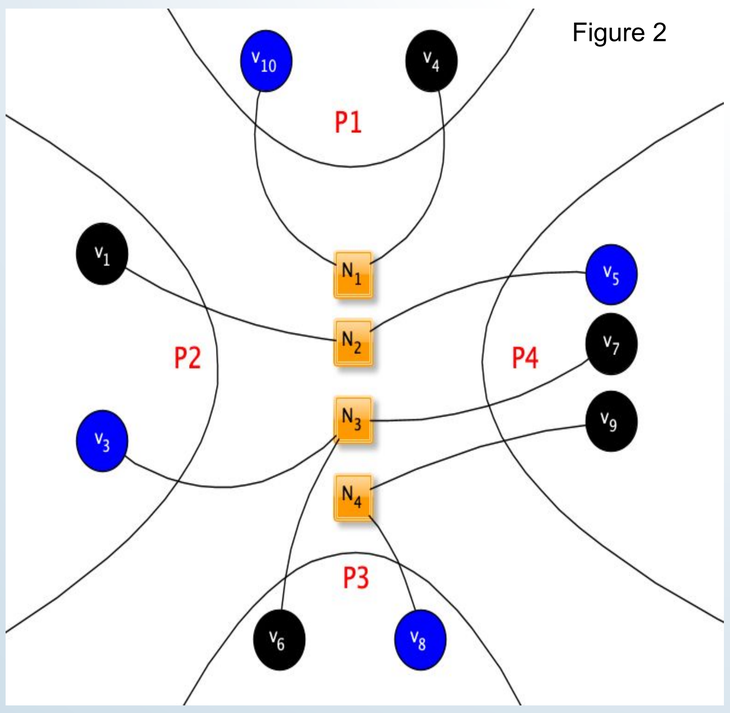
Let Wk denote the total vertex weight in Vk, that is Wk = ∑ v∈Vk w[v] and Wavg denote the weight of each part when the total vertex weight is equally distributed, that is Wavg = ∑ v∈Vk w[v] / K. If each part Vk ∈ Π satisfies the balance criterion Wk ≤ Wavg(1 + ε), for k = 1, 2, . . . , K

We say that Π is ε-balanced where ε is called the maximum allowed imbalance ratio.

Figure 2: A partitioning result of a hypergraph with 4 nets (N1, N2, N3, N4) and 10 vertices (v1, v2, .., v10) into 4 parts. Resulting partitions are given below.

P1 = {v4, v10}, P2 = {v1, v3}, P3= {v6, v8}, P4= {v5, v7, v9}

Figure 2



FAST HYPERGRAPH PARTITIONING

There are some tools in the literature which can partition a given hypergraph very slow for large hypergraphs and that is the reason we are working on fast hypergraph partitioning.

- Our approach to partition a hypergraph is ; We take a vertex, after that look at it's edges, after that we choose current optimal location to put it in. There is two way of choosing vertex to partition first one is take them randomly, other one is apply breadth first search and after that take vertices according to BFS.
- We use greedy method which do not perform good when we are partitioning at first, because there is no data to compare and choose current optimal place. To overcome this problem, we applied refinement method. In refinement we look at the vertices again and check if we have better part to put vertices in.

OPTIMAL HYPERGRAPH PARTITIONING

There are various tools proposed in the literature which can partition a given hypergraph very fast for small hypergraphs. However, none of the traditional methods provides an optimal partitioning pattern, these tools do not provide an optimal partition always. Therefore, we will in pursuit for optimal solution. Table shows PaToh results and our programs results.

File Name	Number of Parts	Imbalance Ratio	Number of Pins	Number of Nets	PaToH Cost	Our Cost
normalCNet-ibm32	4	1,03	32	32	284	270
normalCNet-can_24	4	1,03	24	24	148	148
normalCNet-GD01	4	1,03	73	73	154	154
unitCNet-ibm32	4	1,03	32	32	190	180
unitCNet-can_24	4	1,03	24	24	120	110
unitCNet-GD01	4	1,03	33	33	120	110
normalFG-bcspwr	4	1,03	85	78	59	59
unifFG-bcspwr	4	1,03	85	78	50	50

PART ASSIGNMENT ALGORITHM

The main purpose of this algorithm is that we need to eliminate some partitions which will cause symmetry. If it is not done, same result will be obtained without getting any efficiency.

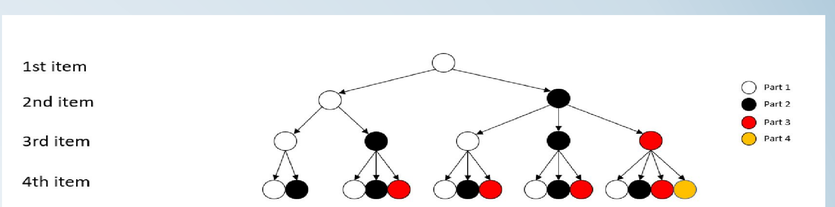


Figure 3: A decision tree according to Part Assigment Algorithm

BRANCH AND BOUND ALGORITHM

This is an algorithm for reducing time consumption by decreasing the number of iterations. Future cost is estimated by looking at the current partial partitioning.

PARALLELIZATION OF THE PROGRAM

The program is parallelized in both fast hypergraph partitioning research and optimal hypergraph partitioning. Separating workloads to threads and decreasing time consumption is aimed.

CONCLUSION AND RESULTS

File Name	Metric	Thread Count	Branch and Bound	Iteration	Time
normalCNet-bcspwr	Cutnet	1	not used	17.8581 million times	0.710339 seconds
normalCNet-bcspwr	Cutnet	1	used	5.90644 million times	0.44587 seconds
normalCNet-bcspwr	Cutnet	4	not used	15.0074 million times	0.172892 seconds
normalCNet-bcspwr	Cutnet	4	used	4.71702 million times	0.143692 seconds
normalCNet-GD01	Total Message	1	not used	0.176918 million times	0.0152337 seconds
normalCNet-GD01	Total Message	1	used	0.176078 million times	0.0358949 seconds
normalCNet-GD01	Total Message	4	not used	0.017856 million times	0.0130031 seconds
normalCNet-GD01	Total Message	4	used	0.006171 million times	0.023381 seconds
unitCNet-ibm32	Cutnet	1	not used	10386.9 million times	377.117 seconds
unitCNet-ibm32	Cutnet	1	used	1077 million times	95.5905 seconds
unitCNet-ibm32	Cutnet	4	not used	8051.84 million times	98.8092 seconds
unitCNet-ibm32	Cutnet	4	used	486.704 million times	13.1381 seconds
unitCNet-can_24	Total Message	1	not used	0.101399 million times	0.00388396 seconds
unitCNet-can_24	Total Message	1	used	0.099973 million times	0.00776162 seconds
unitCNet-can_24	Total Message	4	not used	0.04339 million times	0.0213624 seconds
unitCNet-can_24	Total Message	4	used	0.048067 million times	0.0219297 seconds

Optimal Hypergraph partitioning results with different number of threads, and branch & bound algorithm for CutNet and Total Message metrics.

MatName	K	Patoh_Met.testres	Patoh_Met.testr	TV0.testres	TV0.testres6
gen-amazon-2008.mtx	256	431574	18.83	716815	0.281218
gen-amazon-2008.mtx	512	513514	20.679	760125	0.306436
gen-amazon-2008.mtx	1024	603682	22.93	818972	0.352746
gen-amazon-2008.mtx	2048	702651	24.935	897783	0.442809
gen-wiki-Talk.mtx	256	195803	99.488	559567	0.877854
gen-wiki-Talk.mtx	512	314311	86.357	820574	1.3502
gen-wiki-Talk.mtx	1024	530053	66.333	1112175	2.26649
gen-wiki-Talk.mtx	2048	757756	71.508	1362830	4.04573
sym-coPapersDBLP.mtx	256	1441179	104.313	1999675	1.2475
sym-coPapersDBLP.mtx	512	1633388	109.043	2419240	1.44603
sym-coPapersDBLP.mtx	1024	1820244	117.987	2686218	1.63259
sym-coPapersDBLP.mtx	2048	2085266	121.515	3016240	1.80875
sym-delaunay_n23.mtx	256	166577	114.996	7483623	2.01325
sym-delaunay_n23.mtx	512	237889	125.378	7503797	2.05749
sym-delaunay_n23.mtx	1024	339904	133.389	7512954	2.19763
sym-delaunay_n23.mtx	2048	484397	143.289	7522103	2.5064
sym-hugetrace-00020.mt	256	144295	208.803	11653460	3.77227
sym-hugetrace-00020.mt	512	209695	201.564	11685101	4.37191
sym-hugetrace-00020.mt	1024	300944	~~266.325~~	11702047	5.5097
sym-hugetrace-00020.mt	2048	423641	278.797	11710148	7.75539
sym-road_usa.mtx	256	26035	178.38	10746448	3.46394
sym-road_usa.mtx	512	41466	184.786	10763016	4.41903
sym-road_usa.mtx	1024	68423	180.845	10773275	6.24693
sym-road_usa.mtx	2048	108341	192.202	10780030	9.95086

K : Number of parts divided
BFS (Breadth First Search) : Neighbor relationship
REF : Refinement
0 : Without
1 : Applied

Percentage

	matName	K	Patoh	PatohTime	TV_BFS0_REF0	TV_BFS0_REF0_Time	TV_BFS0_REF1	TV_BFS0_REF1_Per	TV_BFS1_REF0	TV_BFS1_REF0_Time	TV_BFS1_REF1	TV_BFS1_REF1_Per
SOCIAL	amazon-2008.hg	256	431574	18.830	796306	0.474725	738894	7.20979	1099298	1.43519	957852	12.8669
	amazon-2008.hg	512	513514	20.679	836395	0.450546	779564	6.79476	1034139	1.70693	909605	12.0423
	amazon-2008.hg	1024	603682	22.930	900766	0.61429	844001	6.30186	1285544	1.85634	1153497	10.2717
	amazon-2008.hg	2048	702651	24.935	973935	0.566248	915384	6.0118	1426014	1.72687	1286647	9.77319
	coPapersDBLP.hg	256	195803	99.488	1995642	2.1309	1944549	2.56023	2360162	21.7606	2247416	4.77704
	coPapersDBLP.hg	512	314311	86.357	2411964	2.27931	2322762	3.69831	2538146	20.3809	2426050	4.41645
	coPapersDBLP.hg	1024	530053	66.333	2685238	2.41265	2577186	4.02393	2760833	22.5302	2633815	4.60071
	coPapersDBLP.hg	2048	757756	71.508	2987828	2.84993	2858535	4.32732	3042743	22.3303	2889789	5.02685
STRUCTURAL	delaunay_n23.hg	256	1441179	104.313	7616106	3.31553	7159284	5.9981	641970	33.3556	590419	8.03013
	delaunay_n23.hg	512	1633388	109.043	7637998	3.25567	7183441	5.95126	777387	33.5101	713248	8.25059
	delaunay_n23.hg	1024	1820244	117.987	7649452	3.61385	7196520	5.9211	914044	31.3023	833911	8.76686
	delaunay_n23.hg	2048	2085266	121.515	7653974	4.15758	7199148	5.94235	1129794	31.7817	1022949	9.45703
	hugetrace-00020.hg	256	166577	114.996	12710550	6.27828	12600445	0.866249	570007	237.232	536475	5.88273
	hugetrace-00020.hg	512	237889	125.378	12745165	7.45644	12641080	0.816663	662525	232.055	620314	6.37123
	hugetrace-00020.hg	1024	339904	133.389	12761096	6.76619	12659283	0.797839	901725	215.873	843913	6.41127
	hugetrace-00020.hg	2048	484397	143.289	12760184	8.77364	12658926	0.793547	1205646	237.568	1126704	6.54769
	road_usa.hg	256	144295	208.803	10791425	5.19996	10223774	5.2602	376718	440.866	320379	14.9552
	road_usa.hg	512	209695	201.564	10807211	5.99123	10248790	5.16711	495851	419.821	424332	14.4235
	road_usa.hg	1024	300944	266.325	10812976	7.9263	10262683	5.08919	660481	435.933	565607	14.3644
	road_usa.hg	2048	423641	278.797	10817437	12.251	10273560	5.02778	878553	444.372	752183	14.3839

REFERENCES

Deveci, M., Kaya, K., Uçar, B., & Çatalyürek, Ü V. (2015). Hypergraph partitioning for multiple communication cost metrics: Model and methods. *Journal of Parallel and Distributed Computing*,77, 69-83. doi:10.1016/j.jpdc.2014.12.002

Papa, D., & Markov, I. (2007). Hypergraph Partitioning and Clustering. *Handbook of Approximation Algorithms and Metaheuristics Chapman & Hall/CRC Computer & Information Science Series*. doi:10.1201/9781420010749.ch61

Wenyin Yang, Guojun Wang, Md Zakirul Alam Bhuiyan, Kim-Kwang Raymond Choo,Hypergraph partitioning for social networks based on information entropy modularity,Journal of Network and Computer Applications,Volume 86,2017