A Method of Initial Partition Selection for Kernighan-Lin Algorithm

Mikhail Sheblaev
Faculty of Space Research
Lomonosov Moscow State University
sheblaev@gmail.com

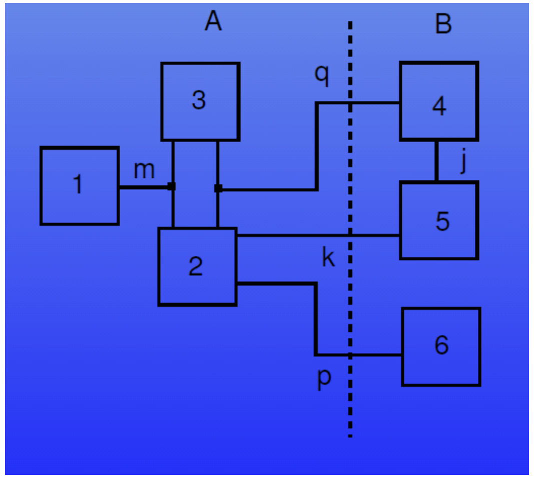
Mincut problem

- *H(V, E)* – (hyper)graph, *V* — vertices, *E* – (hyper)edges,
- $w(e_i)$, $w(v_i)$ - weights for vertices and edges.
- Mincut problem – balanced vertices set partition:
  - $A \cup B = V, A \cap B = \emptyset$
  - $\Psi = \{ e \in E | \, e \; connects \; A \text{ и } B\} \; - Cut(H),$
  - Minimize graph cut weight

$$CutCost = \sum_{e_i \in \Psi} w(e_i) \rightarrow min$$

  - Balance conditions:

$$1 - \varepsilon \leq \sum_{v_i \in A} w(v_i) \; / \sum_{v_i \in B} w(v_i) \leq 1 + \varepsilon$$



Applications

- HPC:
  - critical areas
  - load balance
  - task assignment
- Linear Algebra:
  - matrix operations
  - clusters
- Discrete optimization:
  - SAT solvers
  - Traveling Salesman Problem
- EDA problems:
  - Logical synthesis
  - Physical layout
- Packages: MLPart, hMetis, ZOLTAN, PaToH, SCOTCH

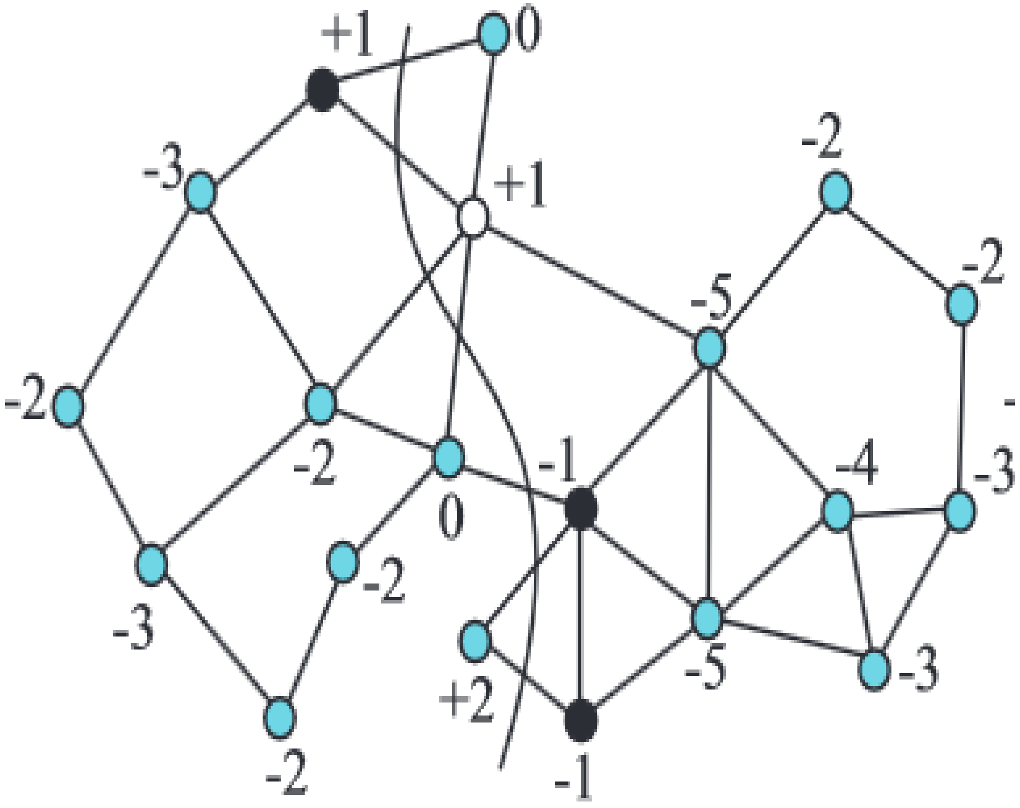
How to solve

- NP-hard problem and *really* big graphs
- Combinatorial optimization: branch and bounds, bisection
- Spectral methods: eigenvectors for Laplasian
- Simulated annealing, genetic algorithms, ant colonies and other black magic
- Local search heuristics: Kernighan-Lin & Fiduccia-Mattheyses

KLFM-алгоритм

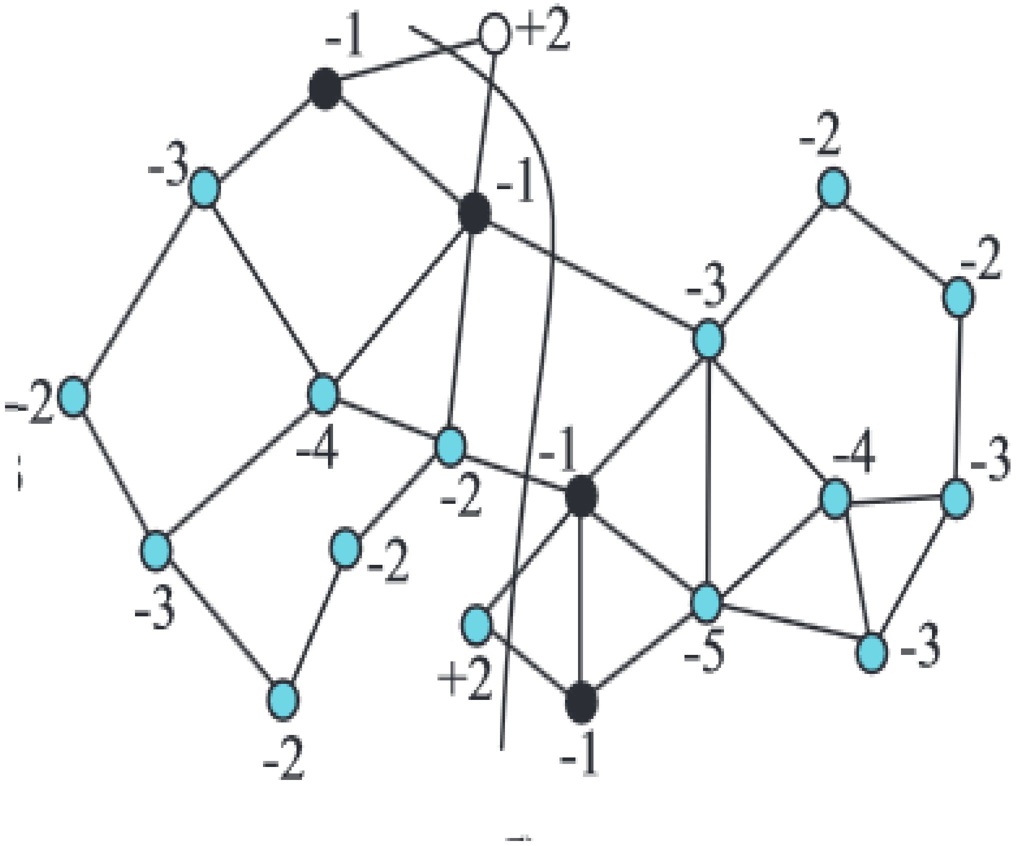
- Initial Partition
- Base vertex selection
- FS(v) - “move” forces
- TE(v) – “tense” forces
- $\Delta g(v)$ = FS(V) – TE(V) - move cost,
  gain
- “Gain bucket” - special data

structure with O(1) update time



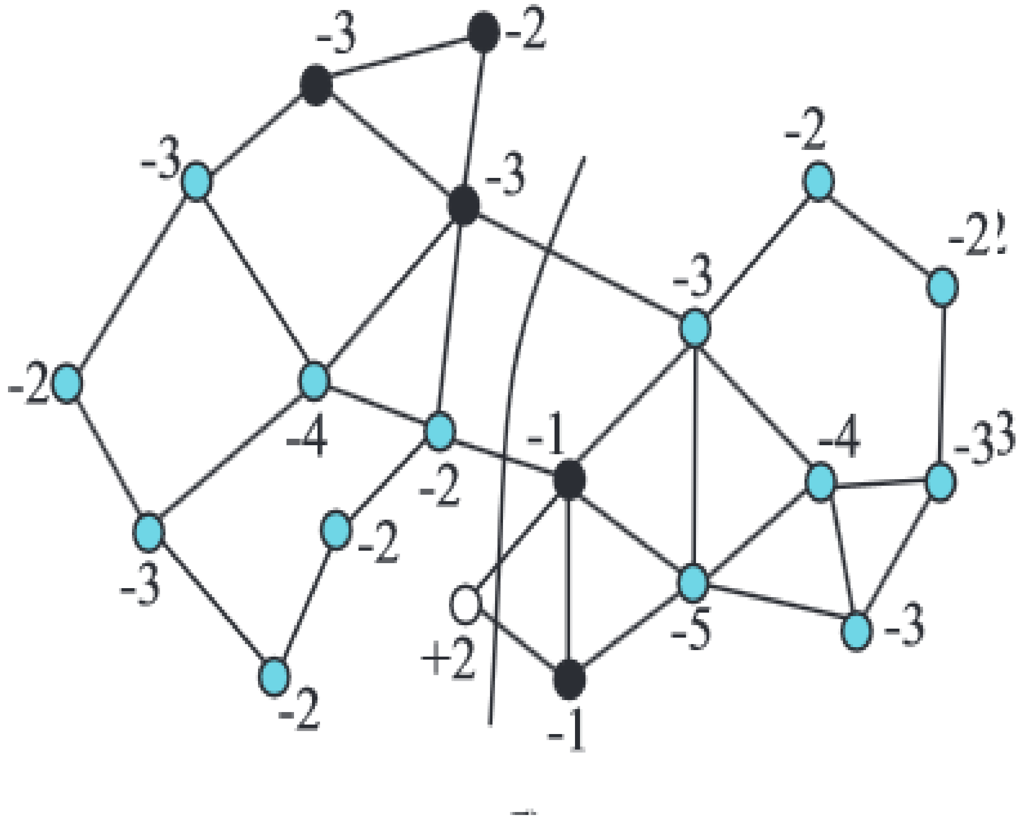
KLFM heuristic

- Step-by-step move of base vertex
- Gain recalculation
- Check of balance condition



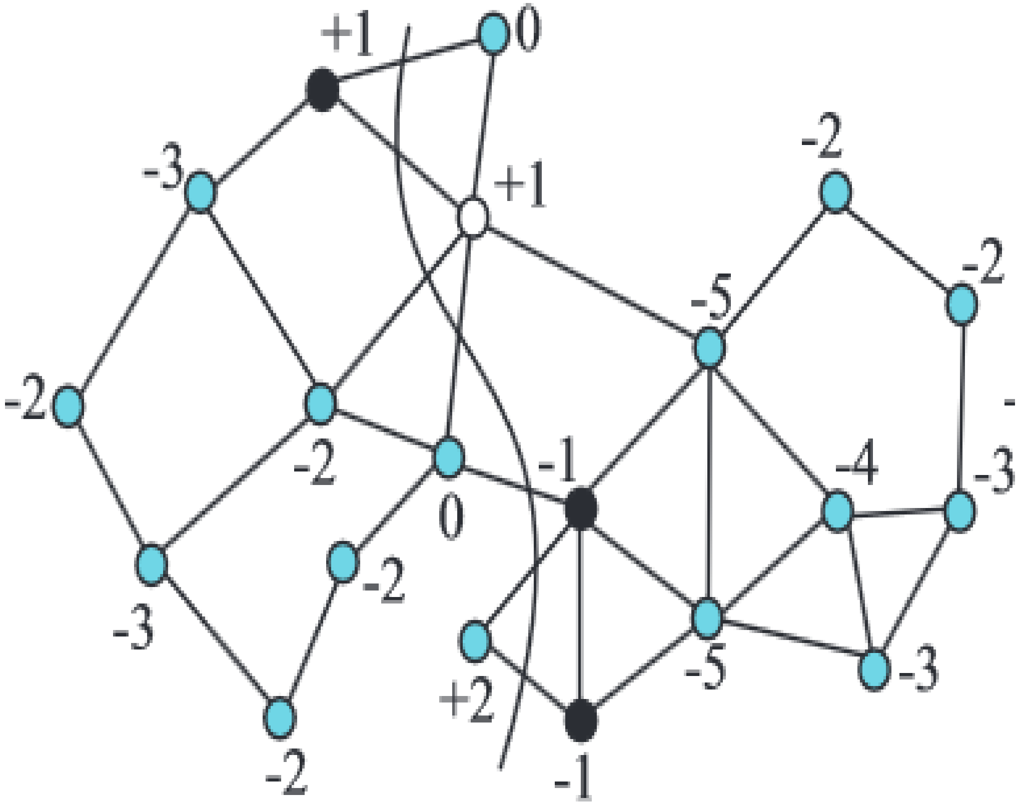
KLFM heuristic

- Step-by-step move of base vertex
- Gain recalculation
- Check of balance condition



KLFM heuristic

- Local search heuristic
- Converges to local optimum due to "greedy nature"
- ***Depends of initial partition selection!***



Initial partition generation

- $H(V) = A \cup B$
  - map initial partition to index vector V at $\{0,1\}^{|H(V)|}$
  - $v_i = 0 \leftrightarrow v_i \in A, \quad v_i = 1 \leftrightarrow v_i \in B$
- Metric induced by KLFM:

$$D_{FM}(V_0, V_1) = \sum_i |FM(V_0)_i - FM(V_1)_i|.$$

- Approximate metric:

$$D_k{}'_{FM}(V_0, V_1) = \sum_i |FM_k{}'(V_0)_i - FM_k{}'(V_1)_i| \qquad \text{– k steps of KLFM}$$

Dimension reduction

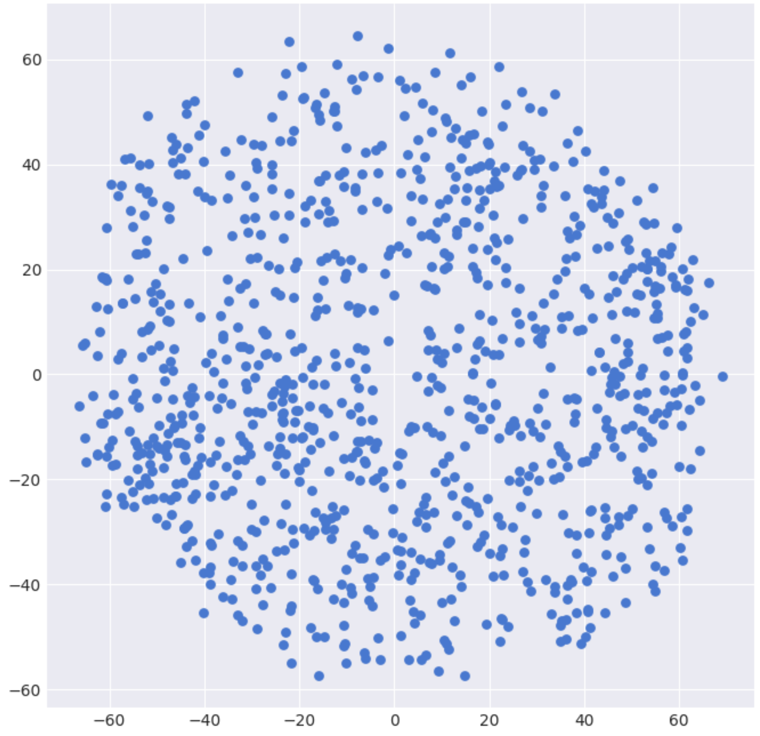
- t-SNE, method from data science, maps high dimension space to 2D
  - Maps neighbors to neighbors, keeps proximity structure for closest objects.
  - Localizes isolated structures
- Generate initial random partitions and apply t-SNE dimension reduction

Algorithm

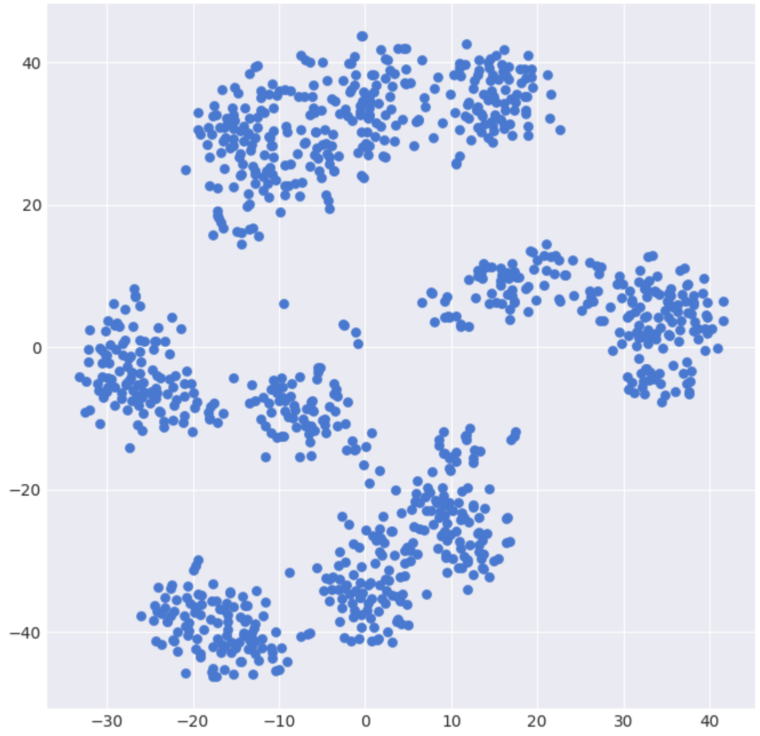
1. Set parameters:
   1. n - # of steps for metric D' approximation
   2. N – size of random initial partitions set - I_N
2. $FM_n(I_N)$ - image of I_N after application of FM_n.
3. Reduce dimension of $FM_n(I_N)$, using Hamming metric as proximity measure at source space. The result is $\mathrm{T}(I_N) \subset R^2$.
4. Clusterize $T(I_N)$.
5. Select representatives $\{V_1, \ldots, V_k\}$ of each cluster, run full-flow KLFM with initial partitions $\{V_1, \ldots, V_k\}$.
6. Get the best solution from step 5.

- Steps 2, 3 and 5 may be executed in parallel

Examples

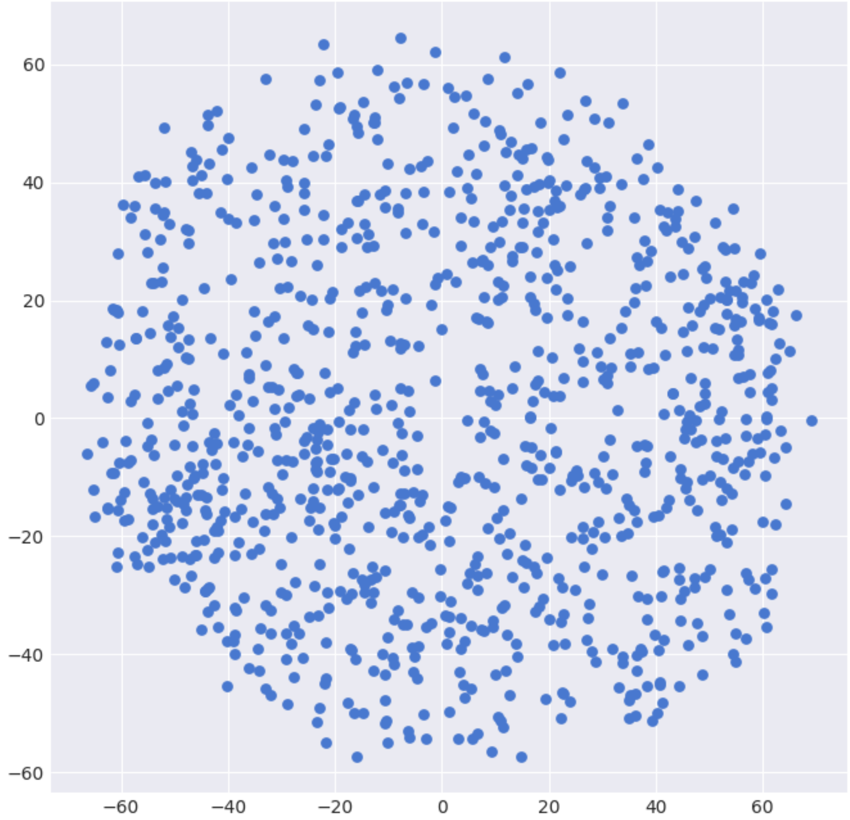


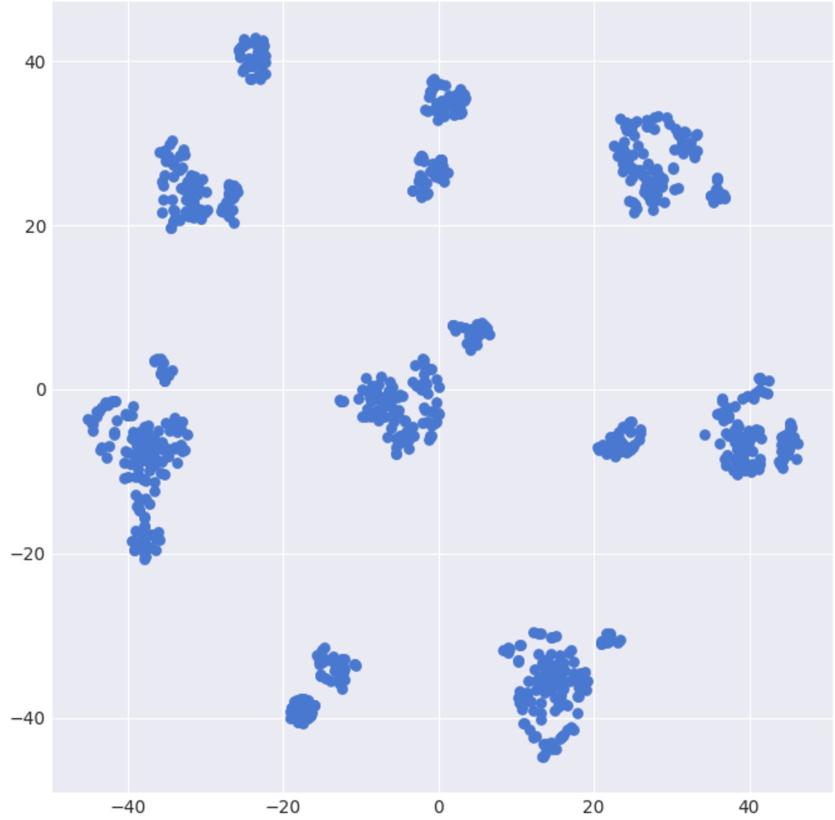
60
40
20
0
−20
−40
−60
−60 −40 −20 0 20 40 60



40
20
0
−20
−40
−30 −20 −10 0 10 20 30 40

Examples





Benchmarks ISPD98

Тест	Random cut size	"State of art " cut size - hMetis	New method, n=5, N = 100	Number of graph vertices
Ibm01	303	393	**288**	12752
Ibm02	446	373	**293**	19601
Ibm03	1734	1558	**885**	23136
Ibm04	1042	1093	**720**	27507
Ibm05	2010	**1878**	1971	29347
Ibm06	1076	1173	**775**	34498
Ibm07	1505	1332	**1074**	45926
Ibm08	1937	2491	**1531**	51309
Ibm09	1246	1457	**992**	53395
Ibm10	2638	2230	**1615**	69429

Chris Walsaw's benchmarks

Graph	|V|	|E|	Cut Size (new method)	Cut Size (Random)	Cut Size (BFS)	Best Cut Size [6]
add20	2395	7462	597	601	631	585
data	2851	15093	189	201	190	188
3elt	4720	13722	89	90	94	89
uk	4824	6837	21	25	32	19
add32	4960	9462	10	16	18	10
bcsstk33	8738	291583	10097	10097	10097	10097
whitaker3	9800	28989	126	126	129	126
crack	10240	30380	184	189	184	183
wing_nodal	10937	75488	1700	1723	1759	1695
fe_4elt2	11143	32818	130	130	130	130
vibrobox	12328	165250	10310	10310	11009	10310
bcsstk29	13992	302748	2818	2877	2895	2818
4elt	15606	45878	138	145	143	138
fe_sphere	16386	49152	386	398	430	386
cti	16840	48232	318	318	352	318
memplus	17758	54196	7250	7376	7053	5457
cs4	22499	43858	496	496	512	366
bcsstk30	28924	1007284	6335	6335	6336	6335
bcsstk31	35588	572914	2750	3342	3346	2699
fe_pwt	36519	144794	340	360	343	340
bcsstk32	44609	985046	4847	5688	5396	4667
fe_body	45087	163734	290	361	596	262
t60k	60005	89440	86	106	98	75
wing	62032	121544	1172	1509	1445	784
brack2	62631	366559	708	711	710	708
finan512	74752	261120	162	162	324	162