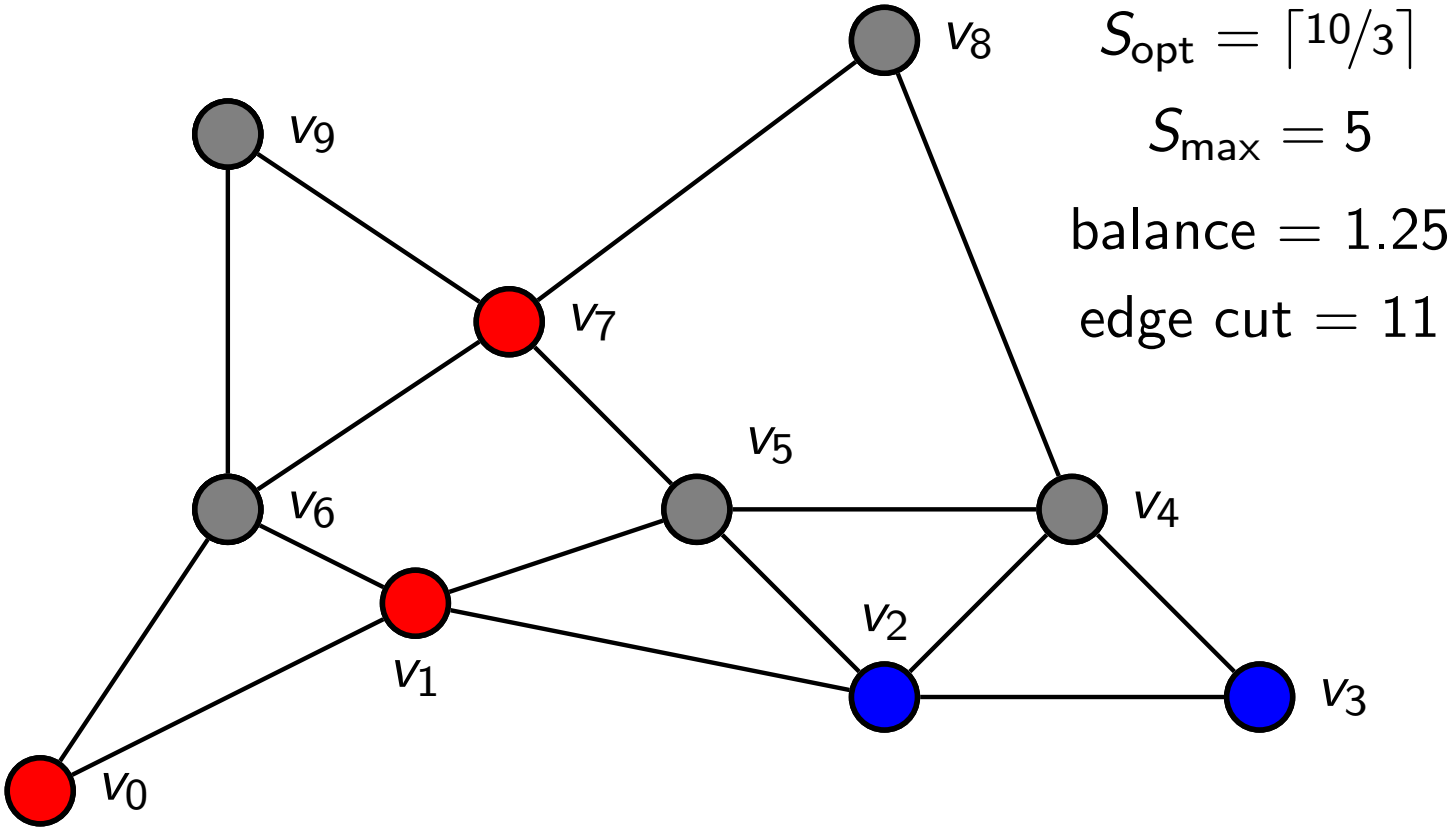
Engineering a Scalable High Quality Graph Partitioner

Manuel Holtgrewe, Peter Sanders, Christian Schulz

Institute for Theoretical Computer Science, Algorithmics II

Novemeber 24, 2009

Graph Partitioning



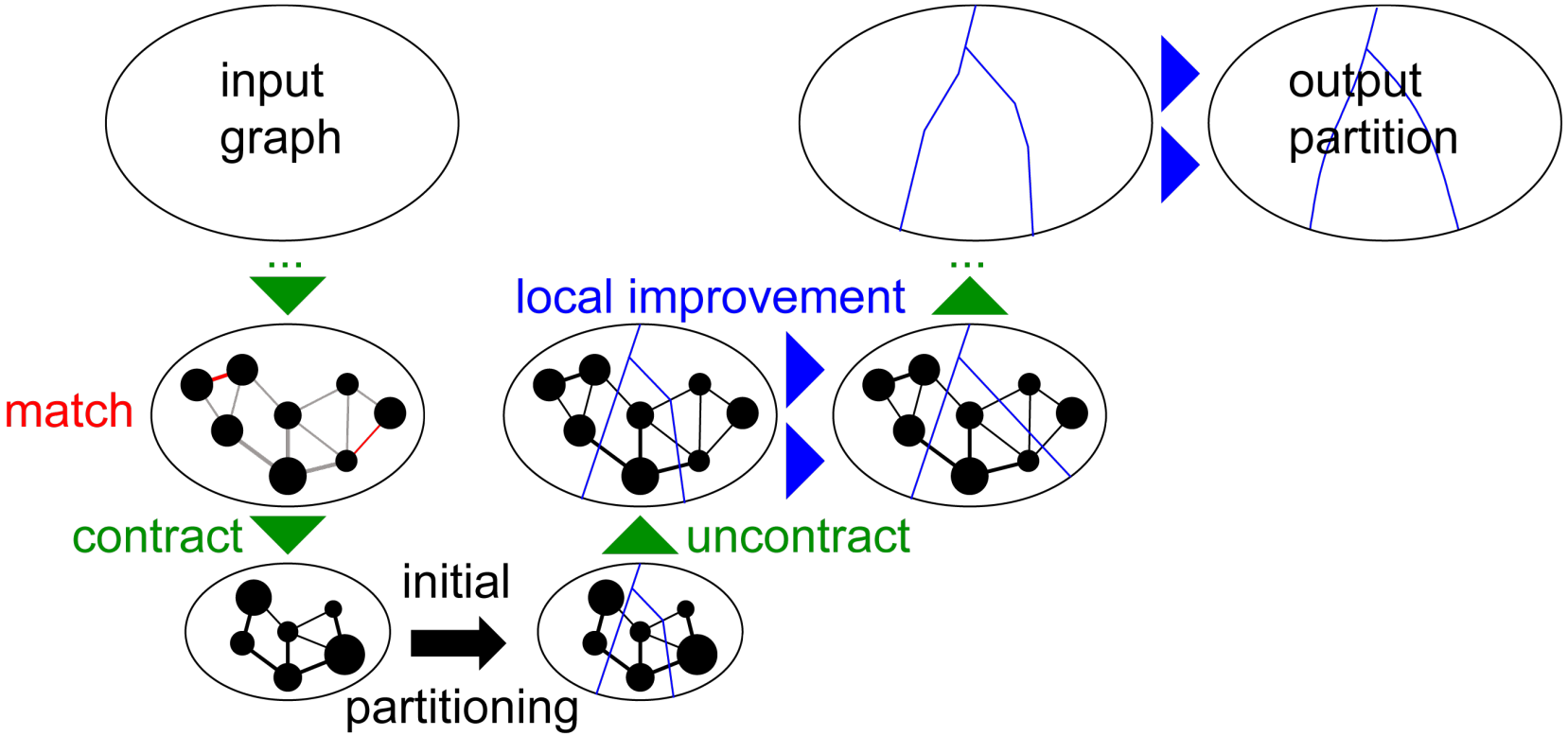
$S_{\text{opt}} = \lceil 10/3 \rceil$

$S_{\text{max}} = 5$

balance $= 1.25$

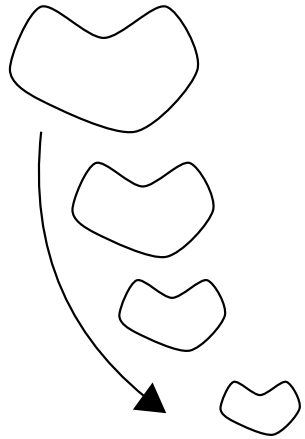
edge cut $= 11$

Well-Known Multi-Level Approach



Match and Contract

- Sequential Matching: GPA [Maue, Sanders, 2007].
  1. 1st: greedy algorithm finds heavy paths and cycles
  2. 2nd: optimal matching on those using dynamic programming
- GPA, Greedy, HEM achieve half-approx. in worst case
- GPA gives considerably better results (empirically)
- Parallel Matching running on the gap graph: Implementation of Manne and Bisseling's sketch [Manne, Bisseling, 2007].
- Use edge ratings for priorization



Edge Ratings - Few Basic Principles

- good to contract heavy edges because this decreases the cut size
- avoid clusters with many outgoing edges
- contract light nodes to keep the node weight at any level reasonably uniform.

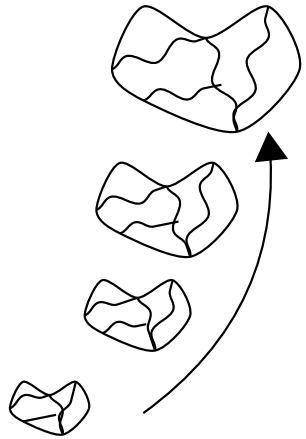
weight	"number of edges in original graph"
expansion	edge weight divided by the number of vertices
expansion*	edge weight divided by the product of number of vertices
inner-outer	edge weight divided by number of edges out of cluster

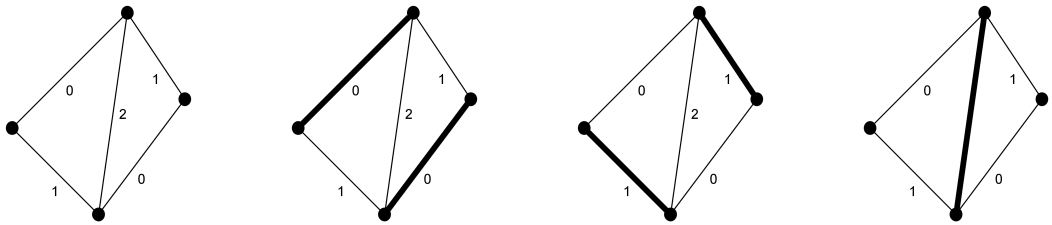
Initial Partitioning

- Every PE can perform initial partitioning using different seeds
- Compared PARTY, PMETIS, and SCOTCH
- SCOTCH yielded best results
- we will try other partitioners as well

Uncontract and Local Improvement

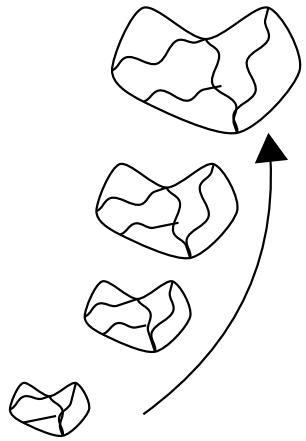
- we adopt basic approach from [Fiduccia, Mettheyses, 1982] (linear time)
- basic idea: at any time, each PE may work on one pair of neighboring blocks
- local search constrained to moving nodes between these two blocks graph
- assign pairs of blocks to PEs, through edge colorings of *quotient graph Q*



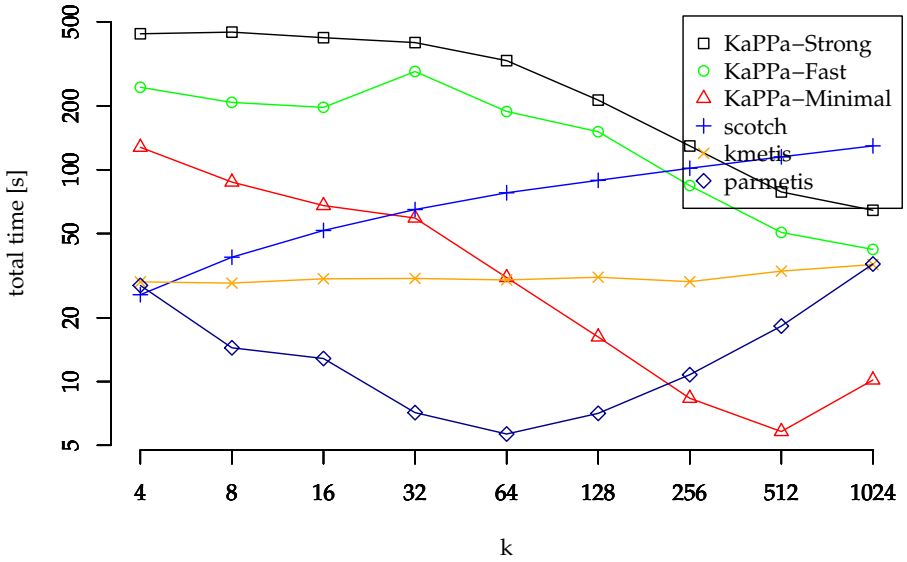


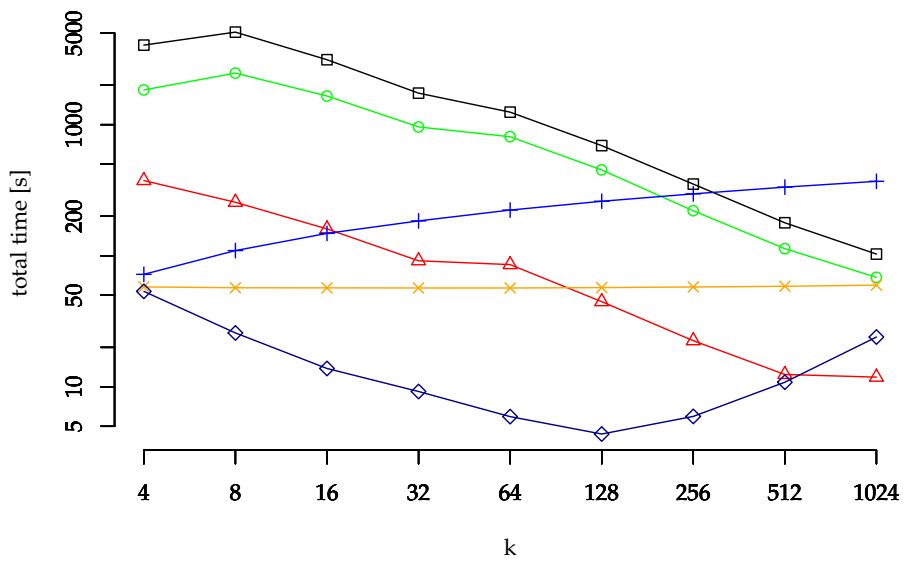
Uncontract and Local Improvement

- BFS to reduce communication volume
- Queue Selection Strategies of local search improved quality
  1. use queue promising larger gain (TopGain)
  2. use queue of heavier block (MaxLoad) if one block is overloaded



Scalablity





- Streetnetwork Europe ($|V| = 18M, |E| = 44M$)
- DelaunayGraph ($|V| = 33M, |E| = 100M$)

Quality - Walshaw Benchmark

Five percent case highlights

Graph	2		4		8		16		32		64	
wing	+ 826	**770**	** 1734	**1636**	* 2632	**2551**	* 4106	**4015**	* 6063	**5806**	+ 8300	**7991**
brack2	** **660**	660	** **2739**	2755	* **6776**	6883	+ **11557**	11558	* 17617	**17529**	+ 26555	**26281**
finan512	** **162**	162	** **324**	324	** **648**	648	* **1296**	1296	** **2592**	2592	** 10909	**10560**
fetooth	** 3785	**3773**	* **6863**	7027	+ **11498**	11957	* **17509**	18090	* 25641	**25624**	+ 35795	**35476**
ferotor	** **1955**	1957	+ **7031**	7520	* **12643**	12678	** **20098**	20263	+ 31611	**31576**	+ 47186	**46608**
598a	** 2344	**2336**	+ **7837**	7978	** **15794**	16031	* **25782**	26257	** **39478**	40179	** **58180**	58307
feocean	** **311**	311	** **1688**	1704	* **3952**	4019	+ **7671**	7838	+ 12953	**12746**	+ **20660**	21784
144	* 6502	**6362**	+ 15313	**15250**	** **25529**	25611	+ **38182**	38478	+ **57202**	57354	+ 80653	**80257**
wave	** 8613	**8563**	+ **16780**	17306	+ **28753**	29776	+ **42810**	43791	** **62382**	63675	** **86867**	87654
m14b	** 3844	**3802**	* **13124**	13285	** **25701**	26153	+ **42644**	43747	* **66845**	67551	+ **99460**	100183
auto	* 9587	**9450**	+ **25805**	26097	** **44915**	48174	+ **76500**	80495	** **121988**	124251	** **174173**	174904

Future Work

- Base case $\Rightarrow$ Meyerhenke's code?
- Back to sequential partitioning
- Other pairwise improvers? Flow? Diffusion?
- Improve overall search strategies
- Put every aspect of multilevel graph partitioning on trial
- More information: http://arxiv.org/pdf/0910.2004

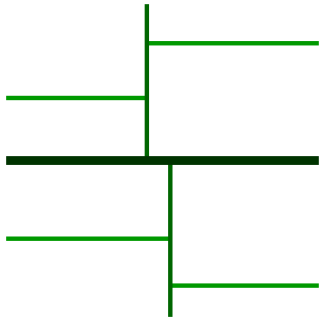
Thank you for your interest. Any questions?

Building Parts

- Recursive Coordinate Bisection as a prepartitioner
- Sequential and parallel algorithms for approximating maximal weight matchings
- Target functions for edges (*edge ratings*)
- Parallel contraction code
- Initial partitioning library (SCOTCH [Pellegrini, Roman, 1996])
- Pairwise refinement algorithm

Recursive Coordinate Bisection

- Similar to Bentley's *k*d-trees
- Parallel Quick-Select
- Could be improved using Parallel Randomized Selection algorithm by Blume et al.



References I


B. Hendrickson and R. W. Leland
A Multi-Level Algorithm For Partitioning Graphs
in Supercomputing 1995


G. Karypis and V. Kumar
Multilevel Graph Partitioning Schemes
ICCP (3) 1995, p. 113-122


C. Walshaw, M. Cross, M. G. Everett
Dynamic mesh partitioning: a unified optimisation and load-balancing algorithm
Tech. Rep. 95/IM/06, Univ. Greenwich, London


M. J. Berger and S. H. Bokhari
A Partitioning Strategy For Nonuniform Problems On Multiprocessors
IEEE Trans. Comput., 36(5):570580, 1987

References II


H. D. Simon
Partitioning of Unstructured Problems for Parallel Processing
Computing Systems in Engineering


A. Pothen, H. D. Simon, K. P. Liu
Partitioning Sparse Matrices with Eigenvectors of Graphs.
SIAM J. on Matrix Analysis and Applications, 11(3):430-452, 1990.


J. Maue and P. Sanders
Engineering Algorithms for Approximate Weighted Matching
Experimental Algorithms, p. 242-255


B. Monien, R. Preis
Quality Matching and Local Improvement for Multilevel Graph-Partitioning
Parallel Computing, 26(12):1609-1634, 2000

References III


B. W. Kernighan, S. Lin
An Efficient Heuristic Procedure for Partitioning Graphs
Bell Systems Technical Journal 14:291-307


C. Schulz
Scalable Parallel Refinement of Graph Partitions
Diplomarbeit, Universitt Karlsruhe (TH)


C. Fiduccia, R. Mattheyses
A Linear-Time Heuristic for Improving Network Partitions
Design Automation, 1982


M. Halappanavar, F. Dobrian, A. Pothen
Matchings in Massive Graphs of Terrascale Computers via Approximation
To appear, 2009

References IV


F. Pellegrini, J. Roman
Scotch: A Software Package for Static Mapping by Dual Recursive Bipartitioning of Process and Architecture Graphs
High-Performance Computing and Networking, p. 493-498


F. Manne and R. H. Bisseling
A Parallel Approximation Algorithm for the Weighted Maximum Matching Problem
7th International Conference on Parallel Processing and Applied Mathematics (PPAM), 2007,


Fiduccia, C. M. and Mattheyses, R. M.
A Linear-Time Heuristic for Improving Network Partitions
19th Conference on Design Automation, 1982