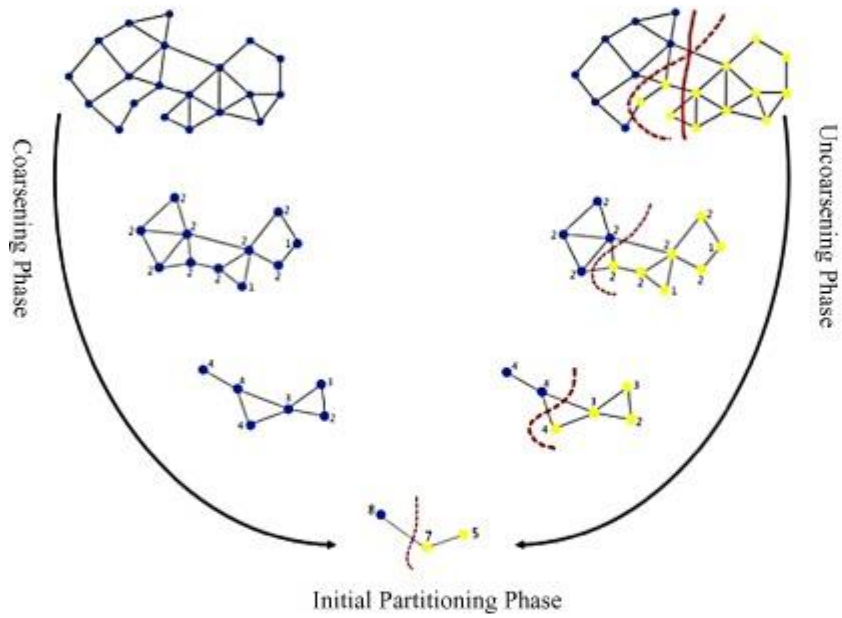
Multilevel Graph Partitioning

George Karypis and Vipin Kumar

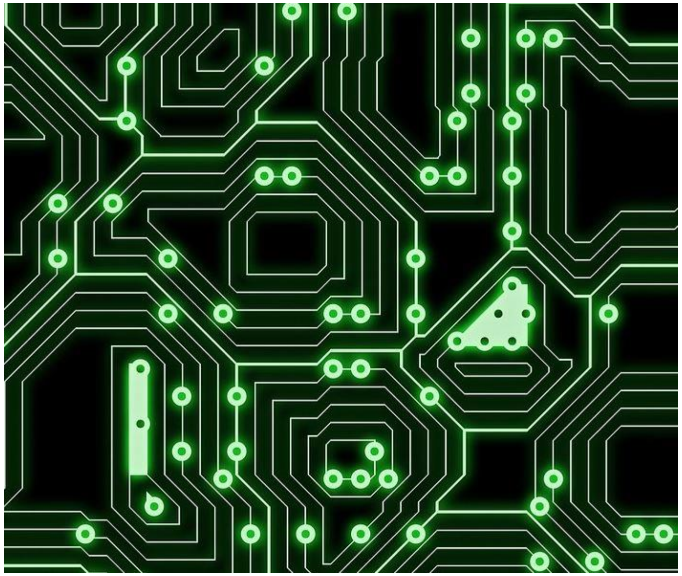


Adapted from Jmes Demmel's slide (UC-Berkely 2009) and Wasim Mohiuddin (2011)

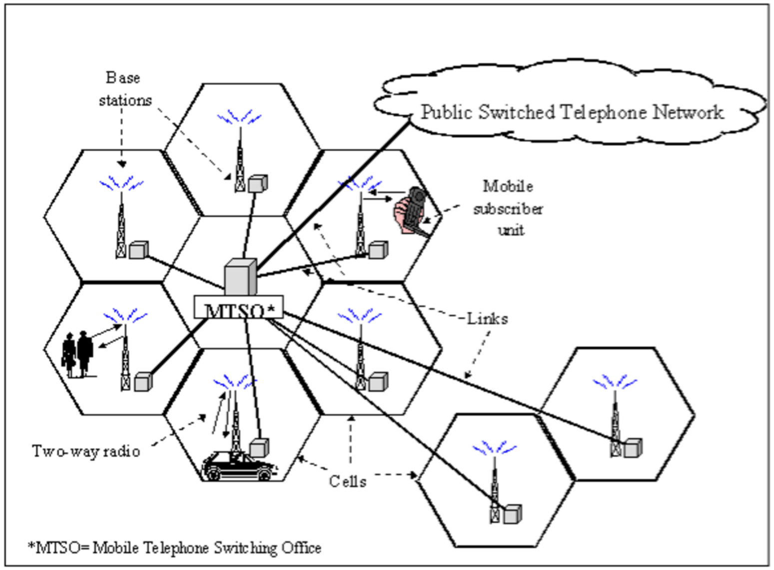
Cover image from:
Wang, Wanyi, et al. "Polygonal Clustering Analysis Using Multilevel Graph-Partition." *Transactions in GIS* 19.5 (2015): 716-736.

Presented by Yijiang Huang
4-11-2018

Introduction – Graph Partitioning is important!



VLSI design



Telephone Network design

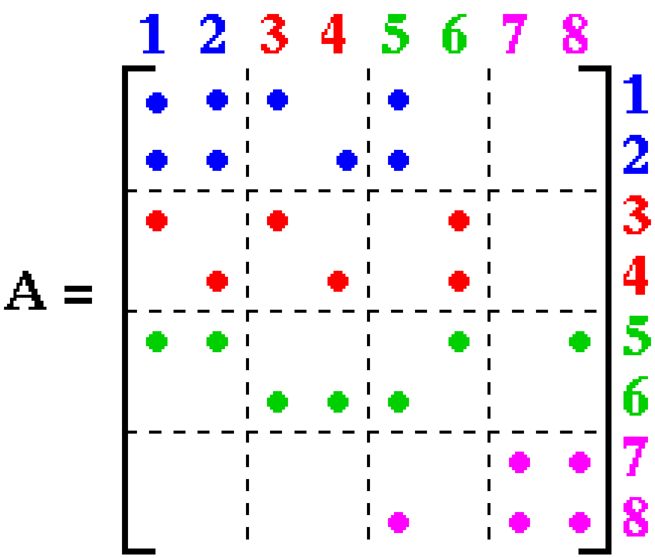
N = {units on chip}, E = {wires}, $W_E(j,k)$ = wire length

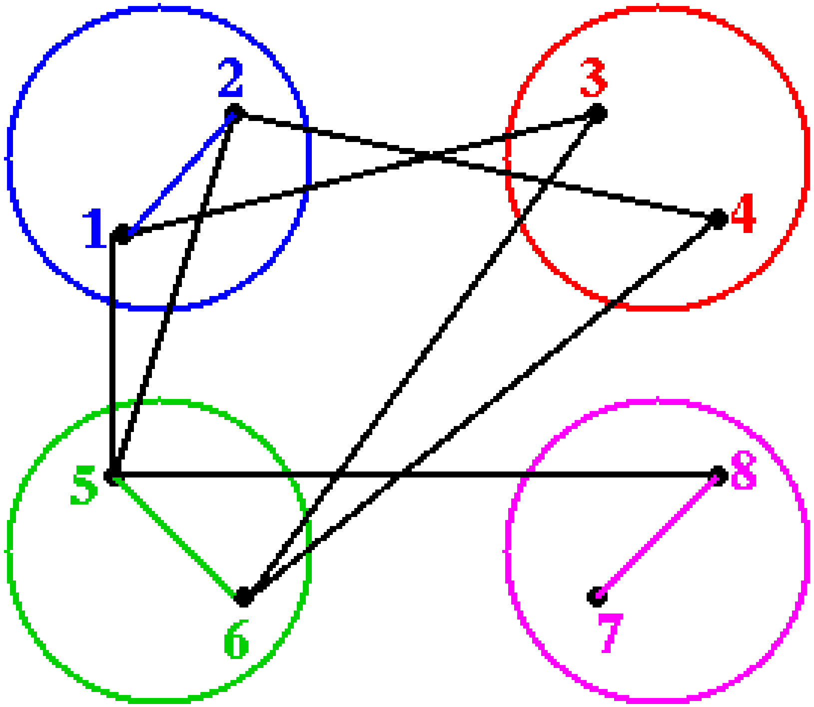
Original application, algorithm due to Kernighan

Load Balancing while Minimizing Communication

Sparse Matrix Vector Multiplication y = y +A*x

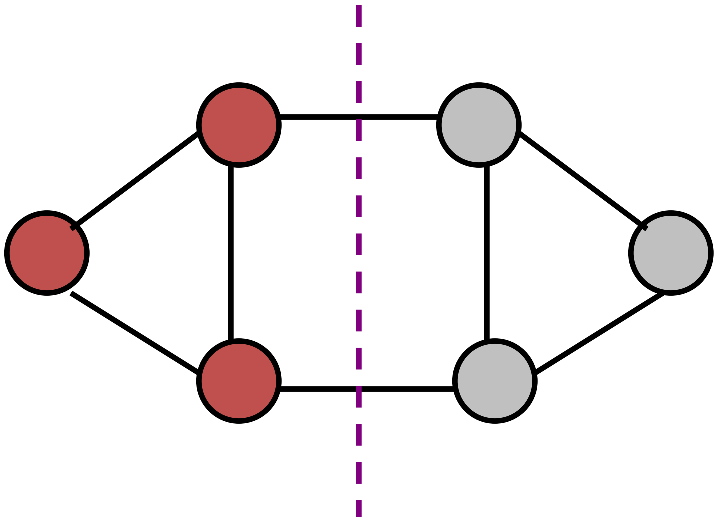
Partitioning a Sparse Symmetric Matrix





Introduction

- Vertices are assigned a weight proportional to their task
- Edges are assigned weights that reflect the amount of data that needs to be exchanged



Introduction – Graph Partitioning is important!

Prefabricated Construction

N = {Structural nodes},

E = {Steel elements},

$W_E(j,k)$ = “onsite welding difficulty”

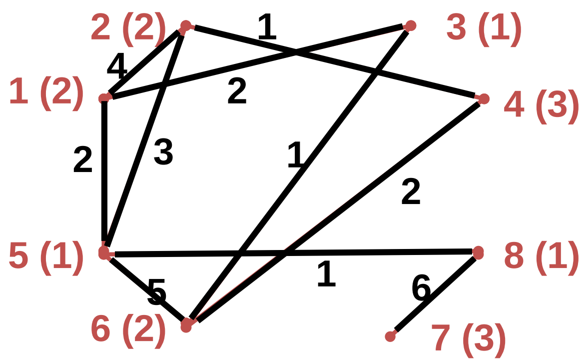
Chord, Antony Gormley, MIT 2015

Outlines

- Introduction
- Definition of Graph Partitioning
- Literature Review
- Multilevel Partitioning - Overview
- Phase 1: Coarsening phase
- Phase 2: Partitioning phase
- Phase 3: Uncoarsening phase
- Experimental Results

Problem Definition

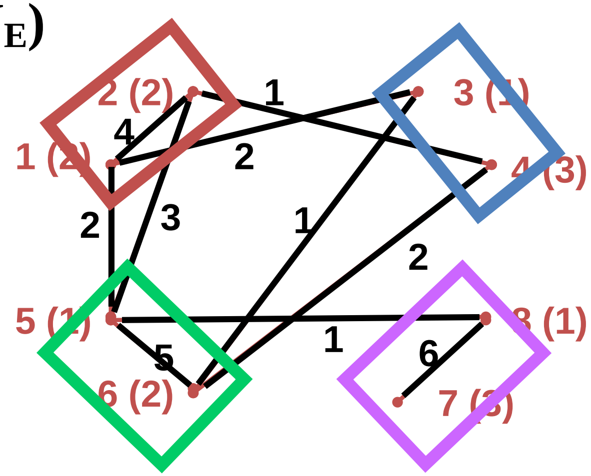
- Given a graph $G = (N, E, W_N, W_E)$
  - N = nodes (or vertices),
  - W_N = node weights
  - E = edges
  - W_E = edge weights



- Ex: N = {tasks}, W_N = {task costs}, edge (j,k) in E means task j sends $W_E(j,k)$ words to task k
- Choose a partition $N = N_1 \cup N_2 \cup \ldots \cup N_P$ such that
  - The sum of the node weights in each N_j is "about the same"
  - The sum of all edge weights of edges connecting all different pairs N_j and N_k is minimized
- Ex: balance the work load, while minimizing communication
- Special case of $N = N_1 \cup N_2$: Graph Bisection

Problem Definition

- Given a graph $G = (N, E, W_N, W_E)$
  - N = nodes (or vertices),
  - W_N = node weights
  - E = edges
  - W_E = edge weights

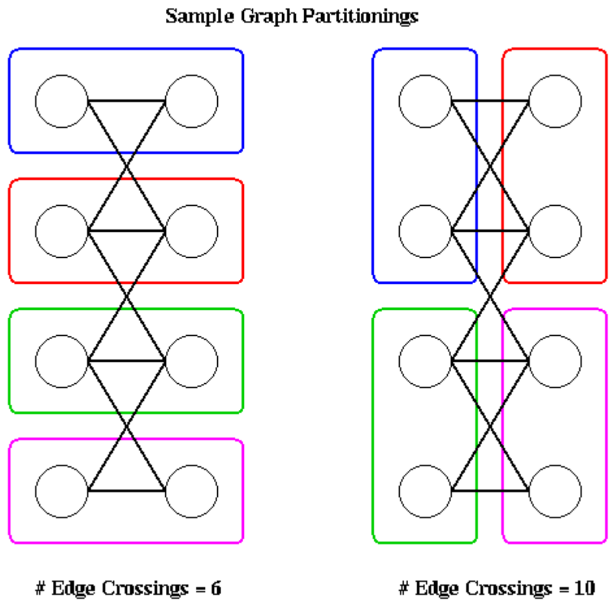


- Ex: N = {tasks}, W_N = {task costs}, edge (j,k) in E means task j sends $W_E(j,k)$ words to task k
- Choose a partition $N = N_1 \cup N_2 \cup \ldots \cup N_P$ such that
  - The sum of the node weights in each N_j is "about the same"
  - The sum of all edge weights of edges connecting all different pairs N_j and N_k is minimized
- Ex: balance the work load, while minimizing communication
- Special case of $N = N_1 \cup N_2$: Graph Bisection

Cost of Graph Partitioning

- Many possible partitionings to search
- Just to divide in 2 parts there are:

  n choose n/2 = $n!/((n/2)!)^2 \sim$ $\text{sqrt}(2/(np))*2^{\mathbf{n}}$ possibilities



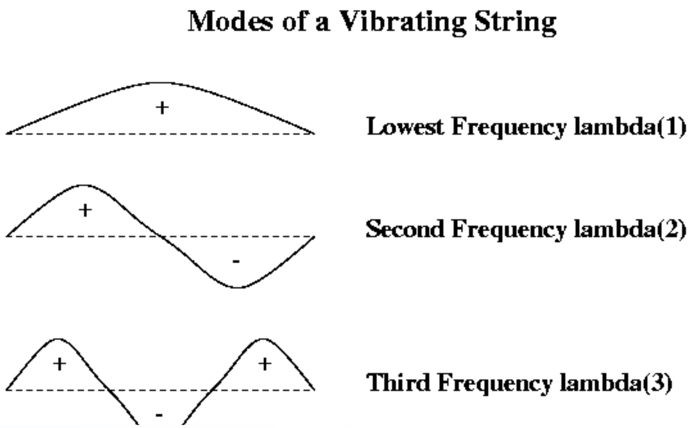
- Choosing optimal partitioning is NP-complete
- We need good heuristics

Outlines

- Introduction
- Definition of Graph Partitioning
- Literature Review
- Multilevel Partitioning - Overview
- Phase 1: Coarsening phase
- Phase 2: Partitioning phase
- Phase 3: Uncoarsening phase
- Experimental Results

Existing methods

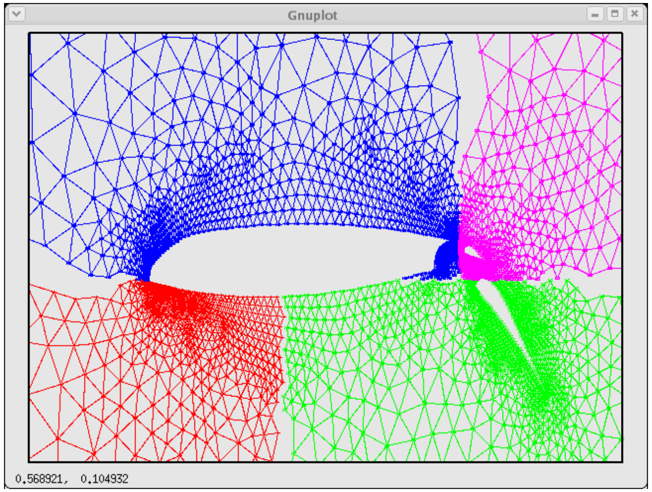
Spectral Partitioning



Intuition: planar ~ trampoline

Not requiring nodal coordinate
Good partition
Computation overhead

Geometric Partitioning



Successful in graphs with nodal coordinates

Existing methods

- Multilevel Spectral Bisection [Barnard and Simon 1993]

- Multilevel Graph Partition [Hendrickson and Leland 1995]

- Fast and Good
  Multilevel Graph Partition [Karypis and Kumar 1998]

Main contribution

Compared to previous multilevel partition work, this paper:

1. Builds on [Hendrickson and Leland 1995] work, uses the **same** overall scheme but proposes different algorithms in each of the subcomponent in the scheme, does detailed **comparison**, and makes **improvements**.

2. Give a good analysis and insight on graph partitioning algorithm based on the presented comparison.

Outlines

Multilevel Partitioning - Overview

If we want to partition G(N,E), but it is too big to do efficiently, what can we do?

1) Replace G(N,E) by a **coarse approximation** $G_c(N_c,E_c)$, and partition G_c instead

2) Use partition of G_c to get a rough partitioning of G, and then iteratively **improve** it

What if G_c still too big?

Apply same idea recursively (recursive bisection)

Multilevel Partitioning - Overview

3 Phases

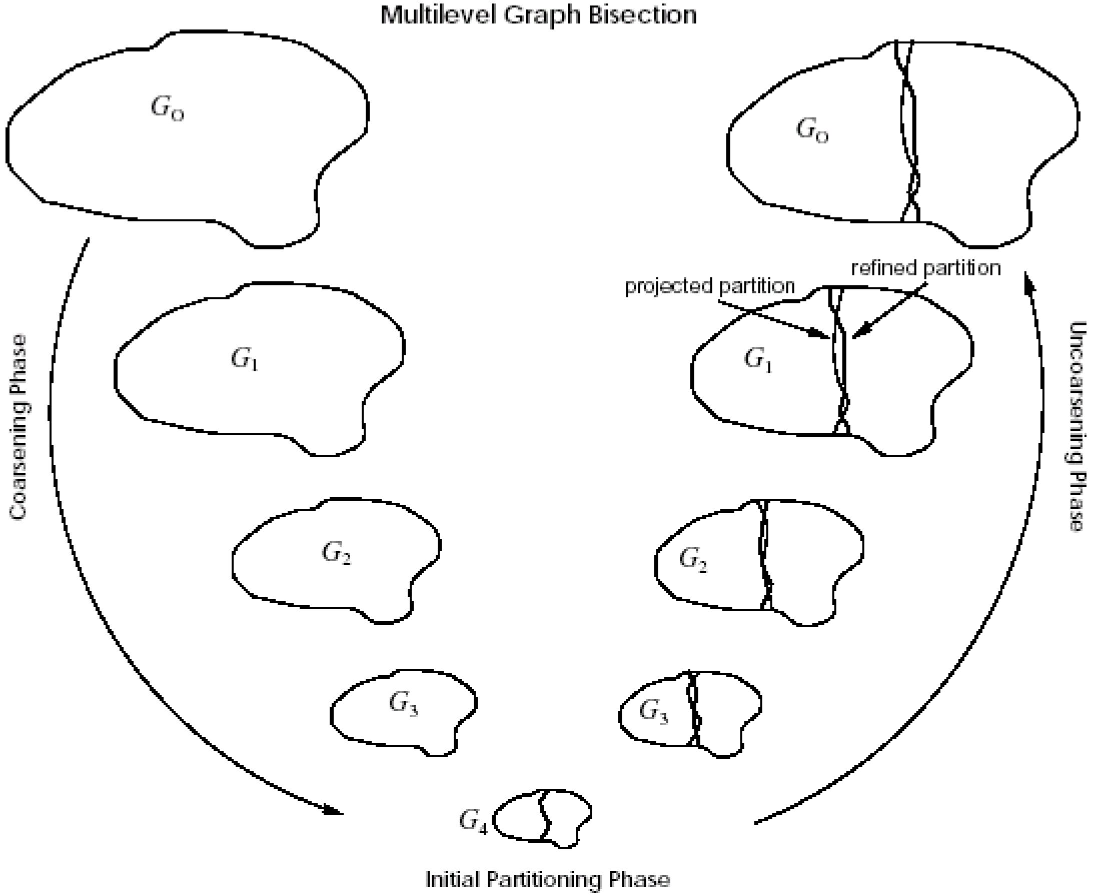
Coarsen

maximal matchings

Partition

Uncoarsen

Refinement



Existing methods

Coarsening **Initial Parition** **Uncoarsening**

Multilevel Graph Partition [Hendrickson and Leland 1995]

Random Matching(RM) Spectral Bisection Kernighan-Lin (KL)

Fast and Good
Multilevel Graph Partition [Karypis and Kumar 1998]

RM + heavy-edge heuristic **Greedy-Graph growing** **Boundary KL**

Outlines

- Introduction
- Definition of Graph Partitioning
- Literature Review
- Multilevel Partitioning - Overview
- Phase 1: Coarsening phase
- Phase 2: Partitioning phase
- Phase 3: Uncoarsening phase
- Experimental Results

Different Ways to Coarsen

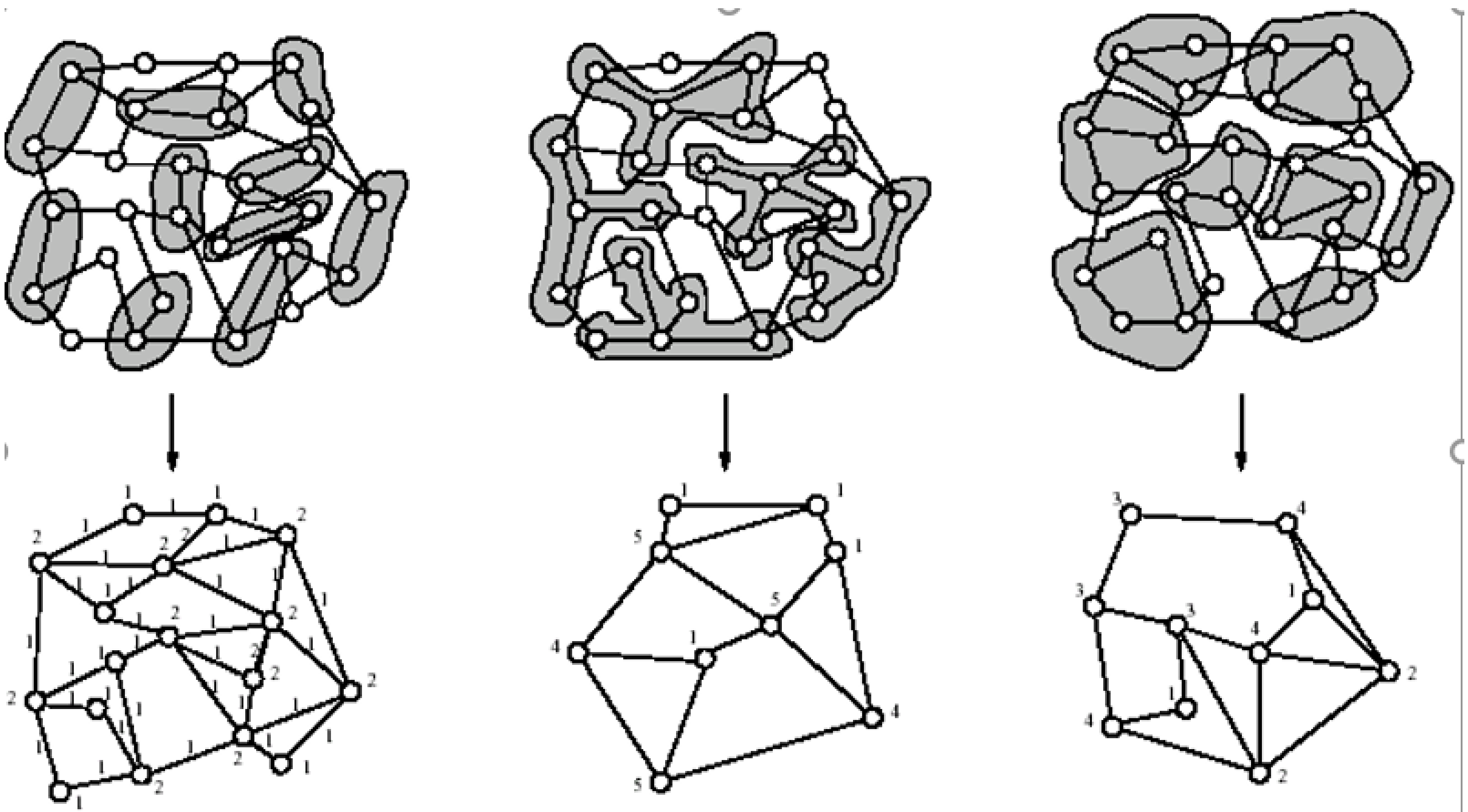


Figure 2: Different ways to coarsen a graph.

Coarsening methods

A coarser graph can be obtained by collapsing adjacent vertices

Matching, Maximal Matching

Different Ways to Coarsen

Random Matching (RM)

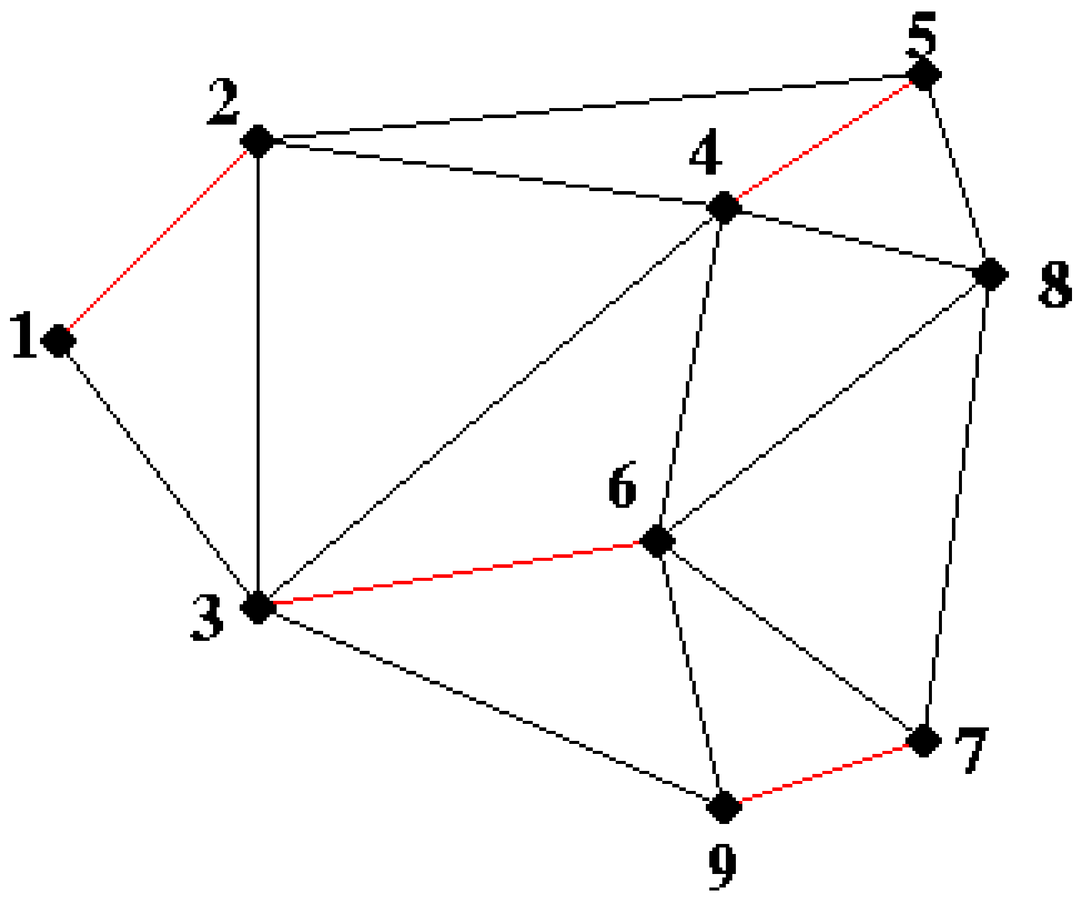
- with **Heavy Edge Matching (HEM)**
- with Light Edge Matching (LEM)
- with Heavy Clique Matching (HCM)

Coarsening phase Maximal Matching

- *Definition*: A matching of a graph G(N,E) is a subset E_m of E such that no two edges in E_m share an endpoint
- *Definition:* A maximal matching of a graph G(N,E) is a matching E_m to which no more edges can be added and remain a matching
- A simple greedy algorithm computes a maximal matching:

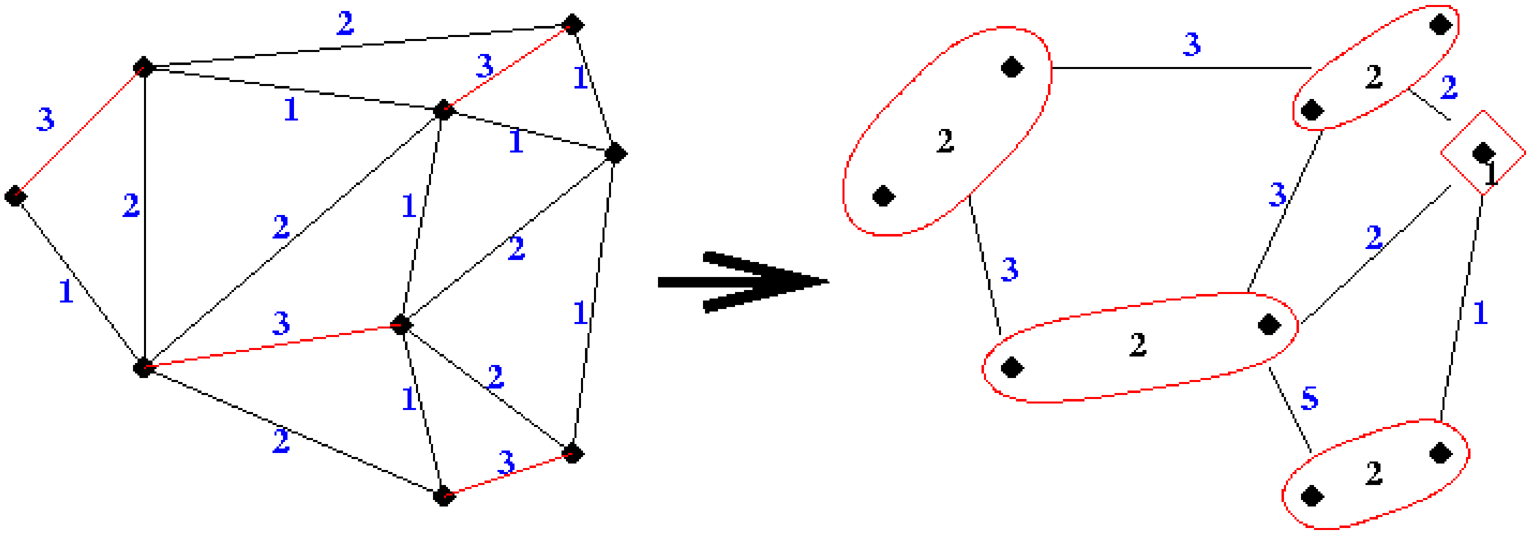
```
let Em be empty
mark all nodes in N as unmatched
for i = 1 to |N|      … visit the nodes in any order (random)
    if i has not been matched
         mark i as matched
         if there is an edge e=(i,j)  where j is also unmatched,
              add e to Em
              mark j as matched
          endif
    endif
endfor
```

Example of matching



Example of Coarsening

How to coarsen a graph using a maximal matching



G = (N, E)

E_m is shown in red

Edge weights shown in blue

Node weights are all one

$G_c = (N_c , E_c)$

N_c is shown in red

Edge weights shown in blue

Node weights shown in black

Outlines

- Introduction
- Definition of Graph Partitioning
- Literature Review
- Multilevel Partitioning - Overview
- Phase 1: Coarsening phase
- Phase 2: Partitioning phase
- Phase 3: Uncoarsening phase
- Experimental Results

Partitioning Algorithms

Spectral Bisection (SB)

Kernighan-Lin (KL)

- Fiduccia-Mattheyses (FM)

Graph Growing Algorithm (GGP)

Greedy Graph Growing Algorithm (GGGP)

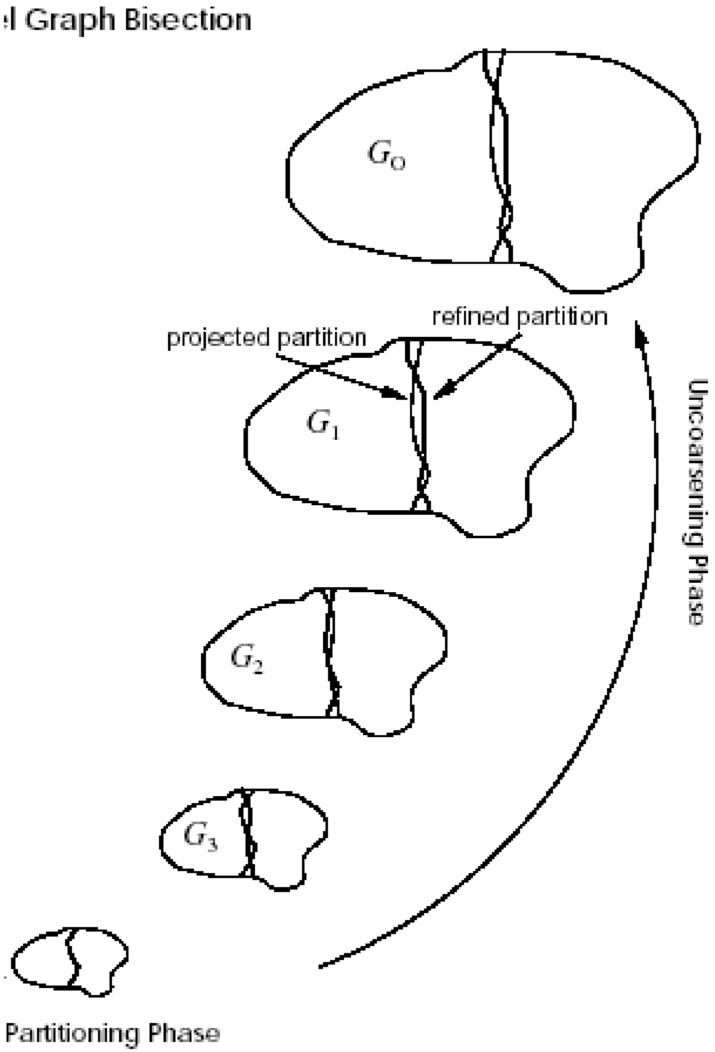
Kernighan/Lin

- Take a initial partition and iteratively improve it
  - Kernighan/Lin (1970), cost = $O(|N|^3)$ but easy to understand
  - Fiduccia/Mattheyses (1982), cost = $O(|E|)$, much better, but more complicated
- Given $G = (N,E,W_E)$ and a partitioning N = A U B, where |A| = |B|
  - T = cost(A,B) = S {W(e) where e connects nodes in A and B}
  - **Find subsets X of A and Y of B with |X| = |Y|**
  - Consider swapping X and Y **if it decreases cost:**
    - newA = (A – X) U Y   and   newB = (B – Y) U X
    - newT = cost(newA , newB) < T = cost(A,B)
- Need to compute new T efficiently for many possible X and Y, choose smallest (best)

Outlines

- Introduction
- Definition of Graph Partitioning
- Literature Review
- Multilevel Partitioning - Overview
- Phase 1: Coarsening phase
- Phase 2: Partitioning phase
- Phase 3: Uncoarsening phase
- Experimental Results

Phase 3: Uncoarsening phase



“Unshrink”

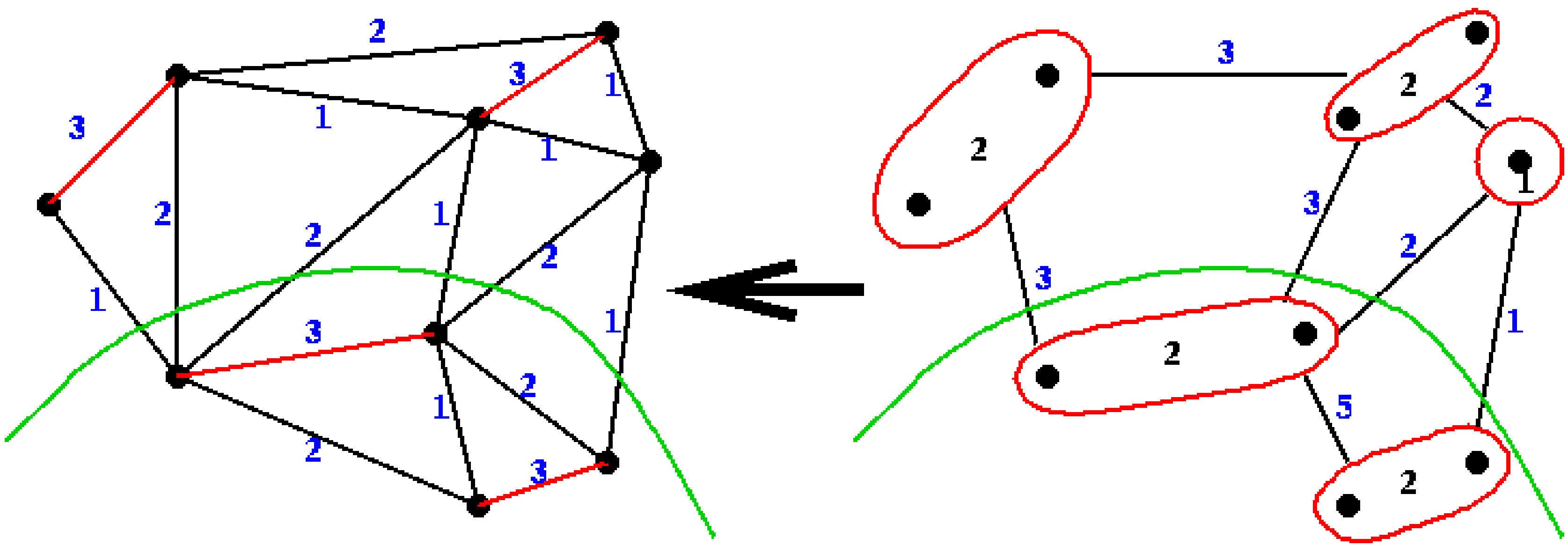
Refine edge cut
(we have more degrees of freedom!)

Kernighan-Lin refinement:
We have **good initial partition** from the uncoarsened graph. (so multiple trials!)

Only swap in boundary (Boundary-KL)

Phase 3: Uncoarsening phase

Converting a coarse partition to a fine partition



Partition shown in green

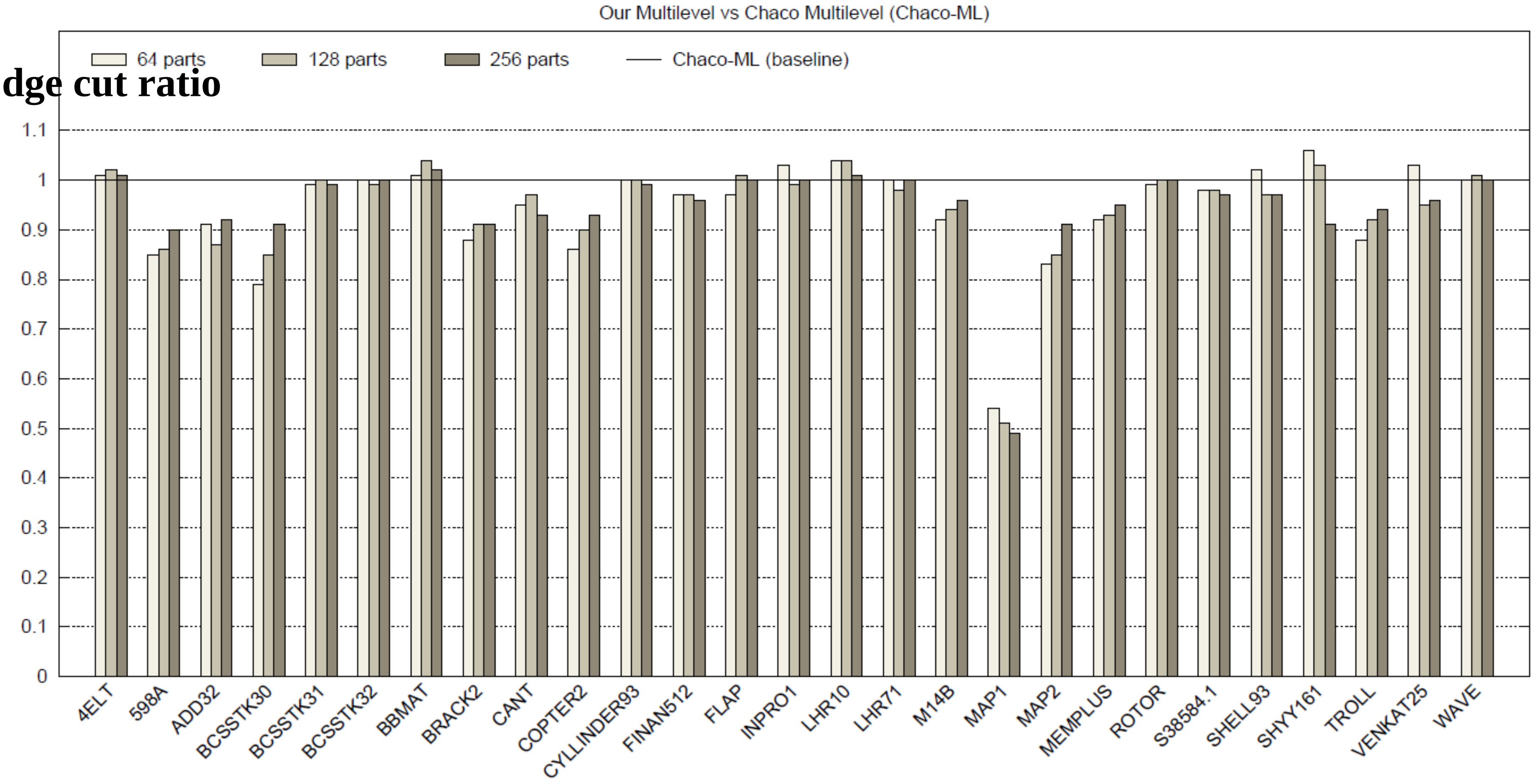
Outlines

- Introduction
- Definition of Graph Partitioning
- Literature Review
- Multilevel Partitioning - Overview
- Phase 1: Coarsening phase
- Phase 2: Partitioning phase
- Phase 3: Uncoarsening phase
- Experimental Results

This work (HEM + GGGP + BKL)

Multilevel Graph Partition [Hendrickson and Leland 1995] (RM + SB + KL)

Edge cut ratio



Runtime ratio

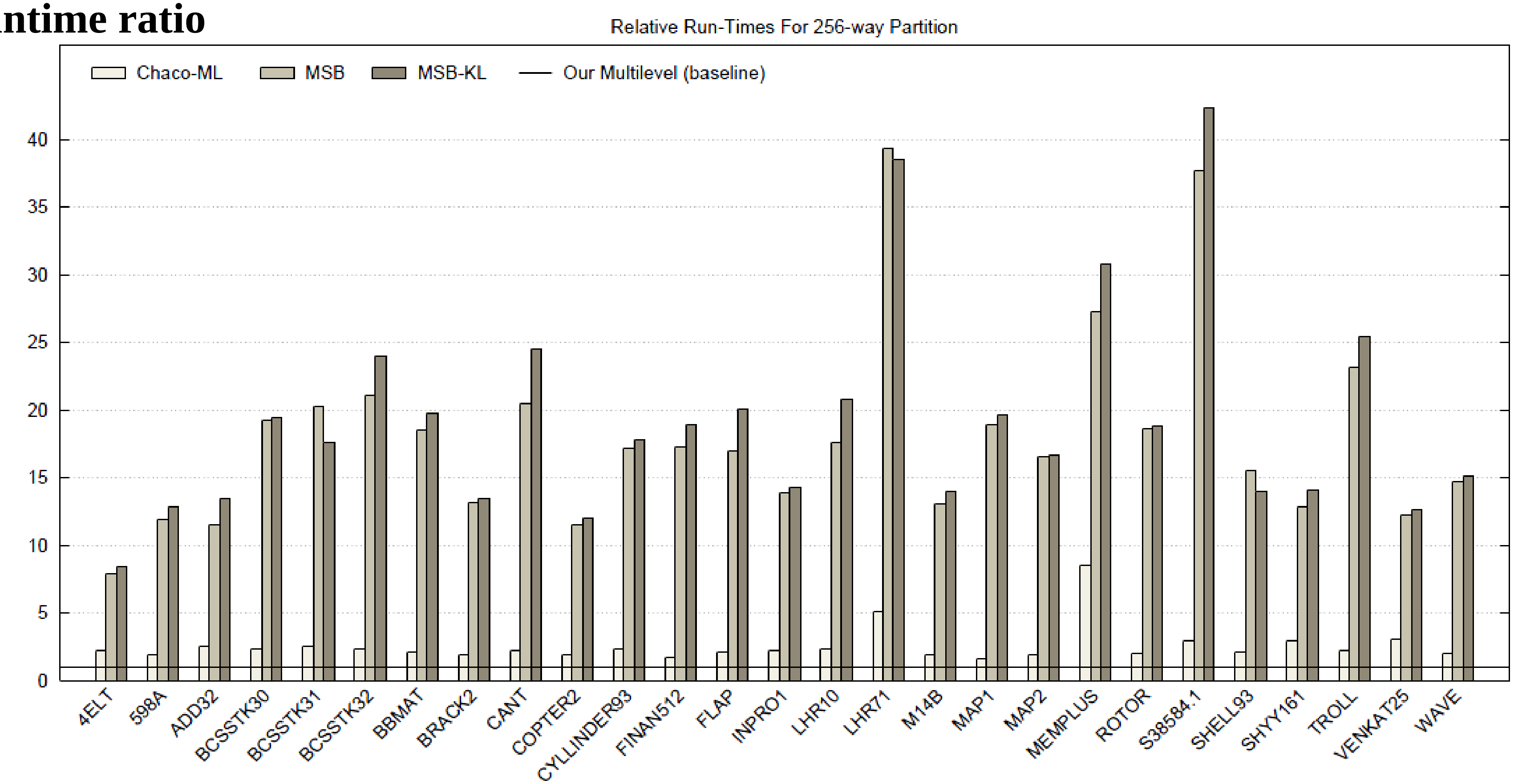


FIG. 6. *The time required to find a 256-way partition for Chaco-ML, MSB, and MSB-KL relative to the time required by our multilevel algorithm.*

	Number of Trials	Needs Coordinates	Quality	Local View	Global View	Run Time	Degree of Parallelism
Spectral Bisection	1	no	●●●●	○	●●●●	■■■■	▲▲
Multilevel Spectral Bisection	1	no	●●●●	○	●●●●	■■■	▲▲
Mulitlevel Spectral Bisection-KL	1	no	●●●●●●	●●	●●●●	■■■	▲▲
Multilevel Partitioning	1	no	●●●●●●	●●	●●●●	■■	▲▲
Levelized Nested Dissection	1	no	●●	○	●●	■■	▲▲
Kernighan-Lin	1	no	●●	●●	○	■■	▲
	10	no	●●●◐	●●	●◐	■■■	▲▲
	50	no	●●●●	●●	●●	■■■◧	▲▲
Coordinate Nested Dissection	1	yes	●	○	●	■	▲▲▲
Inertial	1	yes	●●	○	●●	■	▲▲▲
Inertial-KL	1	yes	●●●●	●●	●●	■■	▲
Geometric Partitioning	1	yes	●●	○	●●	■	▲▲▲
	10	yes	●●●◐	○	●●●◐	■■	▲▲▲
	50	yes	●●●●	○	●●●●	■■◧	▲▲▲
Geometric Partitioning-KL	1	yes	●●●●	●●	●●	■■	▲
	10	yes	●●●●●◐	●●	●●●◐	■■■	▲▲
	50	yes	●●●●●●	●●	●●●●	■■■◧	▲▲

Local view:
Localized refinement

Global view:
Takes into account the general structure of the graph

(each “dot” represents 10% improvement)

Available Implementations

Multilevel Graph Partitioning (still developing!)

METIS (www.cs.umn.edu/~metis)

ParMETIS - **parallel version**

Multilevel Spectral Bisection

S. Barnard and H. Simon, “A fast multilevel implementation of recursive spectral bisection …”, Proc. 6th SIAM Conf. On Parallel Processing, 1993

Chaco (www.cs.sandia.gov/CRF/papers_chaco.html)

Hybrids possible

Ex: Using Kernighan/Lin to improve a partition from spectral bisection

References

Karypis, George, and Vipin Kumar. "A fast and high quality multilevel scheme for partitioning irregular graphs." *SIAM Journal on scientific Computing* 20, no. 1 (1998): 359-392.

Slide from James Demmel
https://people.eecs.berkeley.edu/~demmel/cs267_Spr16/Lectures/lecture14_partition_jwd16_4pp.pdf

Slide from Wasim Mohiuddin
http://delab.csd.auth.gr/courses/c_mmdb/mmdb-2011-2012-metis.ppt