Quadratic Placement Revisited

- Charles Alpert, IBM Austin Research Lab
- Tony Chan, UCLA Mathematics Department
- Dennis Huang, UCLA Computer Science/Synopsys
- Igor Markov, UCLA Mathematics Department
- Kenneth Yan, UCLA Computer Science

Partially supported by a grant from Cadence Design Systems

Motivations

- "Quadratic placement" is dominant
- Effectiveness of QP depends on context
  - solver numerics
  - min-cut partitioner technology
  - problem size, constraints
- What are futures for QP approach?

Outline of Talk

- Motivations and background
- Improved numerical solvers
  - Gains from preconditioners and Krylov subspace solvers
- Analyses of solver-partitioner interaction
  - Order convergence criteria
  - Do modern partitioners need solvers?
- Futures and conclusions

The Placement Problem

- Given:
  - Vertex-, edge-weighted graph
  - Constraints on vertex locations
- Find:
  - vertex locations to minimize placement objective subject to constraints
- Applications:
  - timing-, routability-driven placement

Squared-Wirelength Objective

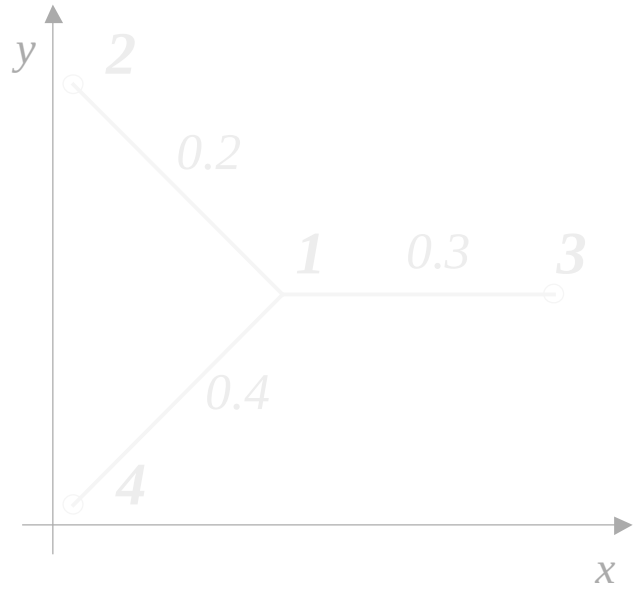
- Classic 1-dim quadratic objective

  $\Phi_{\mathbf{q}}(\mathbf{x}) = 0.5\,\mathbf{x}^{T}\mathbf{Q}\,\mathbf{x} = \Sigma_{i>j} w_{ij} (x_i - x_j)^2$

  - $\mathbf{Q}$ - the Laplacian
  - x_i - locations of vertices (modules)
  - w_{ij} - edge weights
- Constraints: $\mathbf{Hx}=\mathbf{b}$

Laplacian Example

$$Q = \begin{pmatrix} -0.9 & 0.2 & 0.3 & 0.4 \\ 0.2 & -0.2 & 0 & 0 \\ 0.3 & 0 & -0.3 & 0 \\ 0.4 & 0 & 0 & -0.4 \end{pmatrix}$$



Analytic Placement

- To minimize $\Phi_q(x) = 0.5x^TQx$, solve $Qx=0$
- Constraints pre-solved to reduce system
- Issues
  - Find most effective linear solver
  - If iterative, when should it stop?
  - Honor discrete (slot) constraints

Prototypical “Quadratic Placement”

- Analytic placement clumps cells in center
- Partitioner used for “spreading”
  - split of 1-dim analytic placement is seed for partitioning step
- Result: a top-down bipartitioning method
- Easy to implement, handles large instances

Outline of Talk

- Motivations and background
- Improved numerical solvers
  - Gains from preconditioners and Krylov subspace solvers
- Analyses of solver-partitioner interaction
  - Order convergence criteria
  - Do modern partitioners need solvers?
- Futures and conclusions

Linear Solver Implementation

- Handle large problems with iterative solvers
  - *SOR/SSOR (1950s)*
  - *Krylov subspace* (KSP) methods: *CG*, *GMRes*, *QMR*, *BiCGStab*, etc.
- Preconditioners improve solver performance
  - fewer solver iterations needed
  - cost tradeoff with partitioner overhead
  - *ILU*, *SOR*, *SSOR*, etc.

Experiments with Linear Solvers

BiCGS/ SOR 1.0	SOR 1.95
10.0	29.5
76.03	98.9
115.6	479.4
54.0	95.6

Experiments with Linear Solvers

Solvers	Preconditioners				
	SOR 1.0	**SSOR 1.0**			**ILU**
	39.6	*---*	*37.2*	*40.7*	***34.3***
	134.4	*79.2*	*67.4*	*133.1*	*57.1*
BiCGS					
	38.9	***36.4***	*37.3*	*39.3*	***35.7***
TFQMR					
	59.2	*45.7*	***34.2***	*60.0*	*41.4*

Outline of Talk

- Motivations and background
- Improved numerical solvers
- Analyses of solver-partitioner interaction
  - Order convergence criteria
  - Do modern partitioners need solvers?
- Futures and conclusions

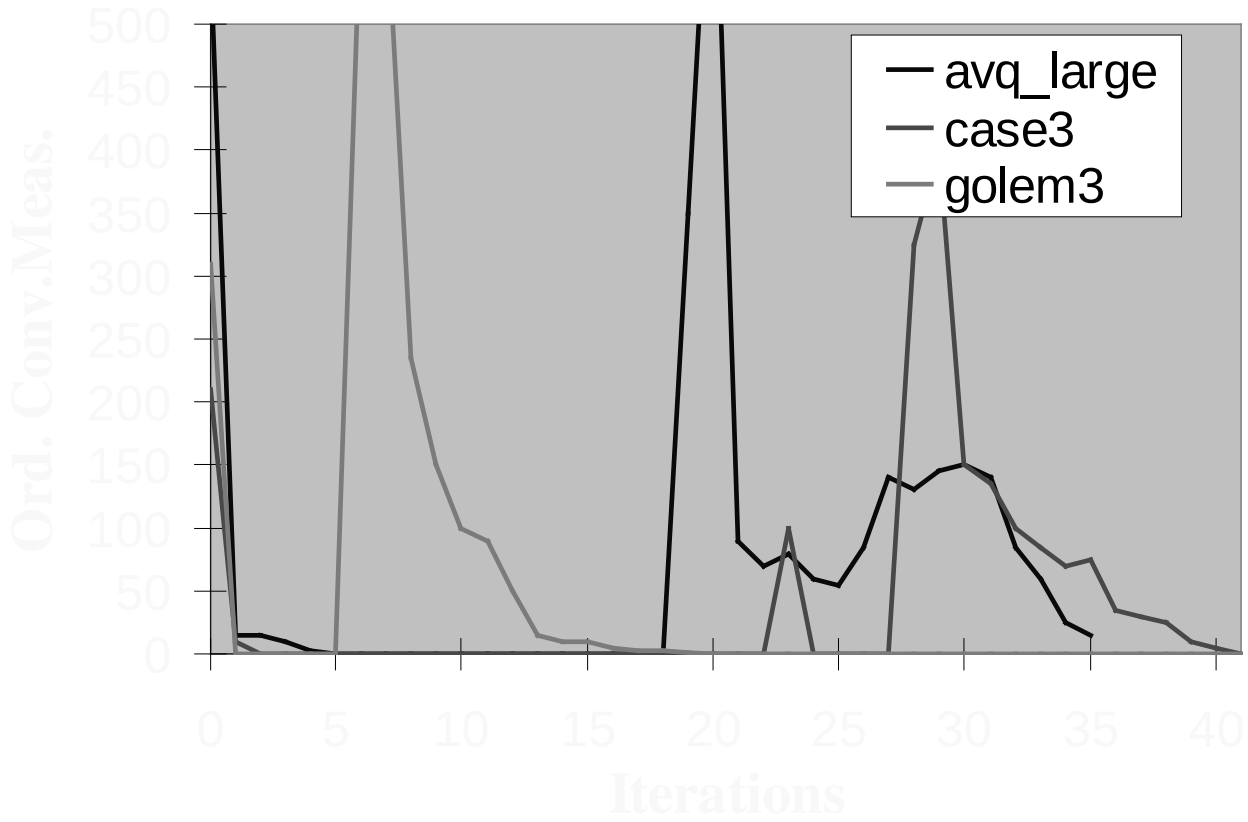
Order Convergence

- Iteratively solving $Qx=c$ yields a sequence of approximate solutions (iterates)
- Seeding partitioner from split of analytic placement means only order of locations matters
- Stop iterations when order “stabilizes” !

Order Convergence

- New permutation-based order convergence measures
- Four measures proposed:
  - FLOWNUM %
  - Hamming distance
  - Δ-related
  - FLOW%

Order Convergence Experiments



Order Convergence Stages

- Relative order of cells
  - Rapid initial stabilization
  - Period of relative stability
  - Large changes when modules clumped in center (inessential to partitioner)
  - Convergence in norm
- Can stop iterations early !

Outline of Talk

- Motivations and background
- Improved numerical solvers
- Analyses of solver-partitioner interaction
  - Order convergence criteria
  - Do modern partitioners need solvers?
- Futures and conclusions

Solver-Partitioner Interaction

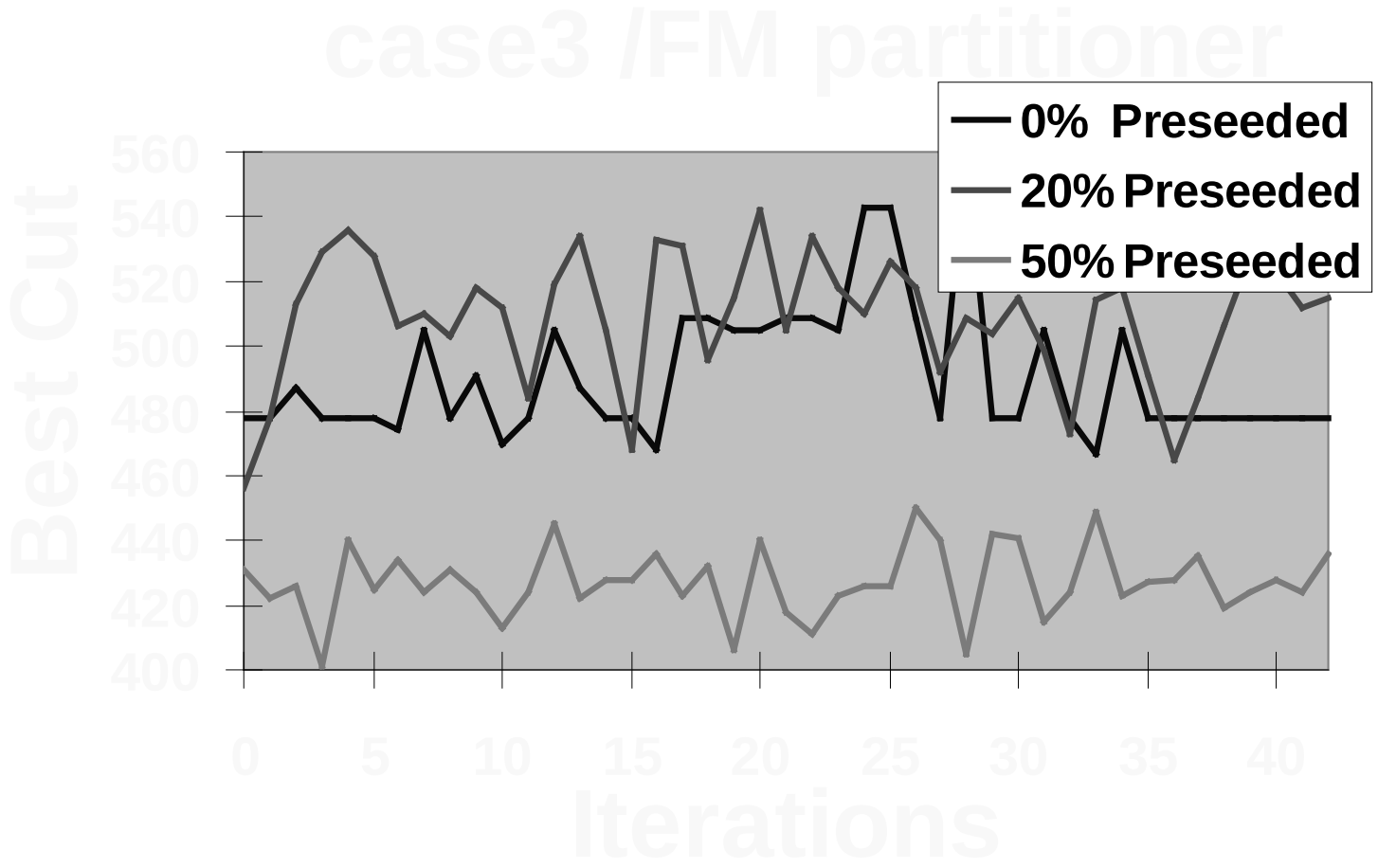
Experiment:

- Iterates of an *AP* solution used to seed min-cut partitioner
- Seeding: {*0%* | *20%* | *50%* } leftmost and rightmost cells into respective partitions
- Run *FM* or *ML-FM* (exact bisection)
- Fixed pads remain locked

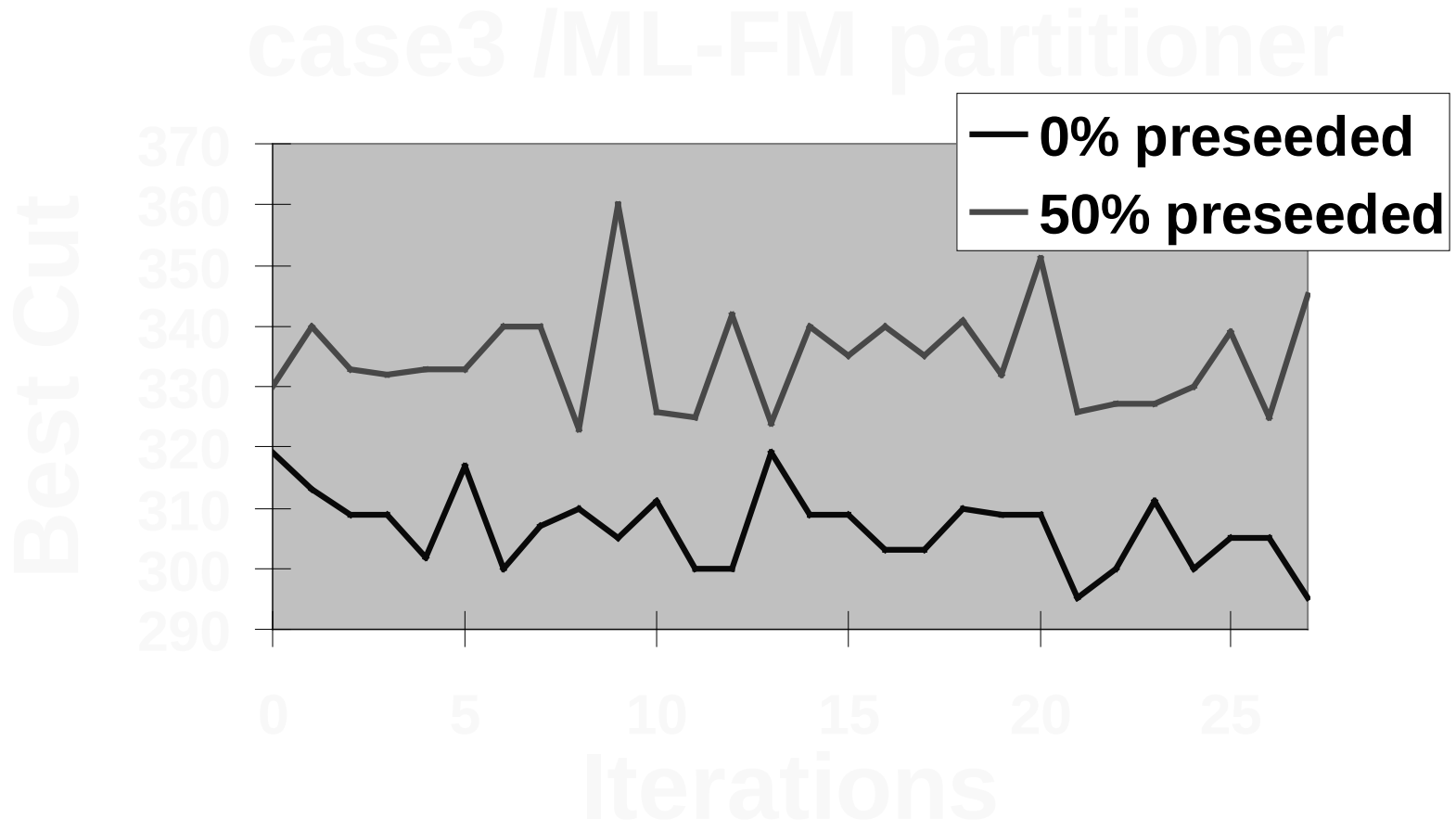
Solver-Partitioner Experiments

- Compute:
  - Average, worst and best cut from a number of *FM* and *ML-FM* runs
  - Hamming distance to the best partition in each case

Solver-Partitioner Experiments



Solver-Partitioner Experiments



Results

- No advantage from further solver iterates
- *FM* benefits from *AP*-derived seeding
- *ML-FM* does not
- Partitions are structurally similar (Hamming distance studies)

Interpretations

- Linear solvers achieve most in first few iterations
- Modern partitioners do not benefit from linear solvers
- Hierarchical partitioning-based placement using modern partitioners only ?
- … but improving each step might not improve the final result ?!

Outline of Talk

- Motivations and background
- Improved numerical solvers
- Analyses of solver-partitioner interaction
  - Order convergence criteria
  - Do modern partitioners need solvers?
- Futures and conclusions

Futures

- Weaknesses of quadratic placers :
  - Modeling of performance constraints
  - Requirement of pad pre-placement
  - Relevance to smaller, constraint-dominated instances
- Future alternatives will address these issues

Conclusions

- Modern placers can benefit from better linear solvers, preconditioners
- *Order convergence* versus *tolerance-based convergence* criteria
- Top-down placement possible without embedded linear solvers ?
- Alternatives ?