Partitioning With Fixed Terminals: A “New” Problem and New Benchmarks

Charles J. Alpert, Andrew E. Caldwell, Andrew B. Kahng
and Igor L. Markov

UCLA VLSI CAD Lab ABKGroup

{caldwell,abk,imarkov}@cs.ucla.edu

alpert@austin.ibm.com

April 13, 1998

Supported by Cadence

Outline

- Introduction and motivation
- A look at partitioning with fixed terminals
- Need for new benchmarks
- Proposed file formats
- New partitioning benchmarks with terminals

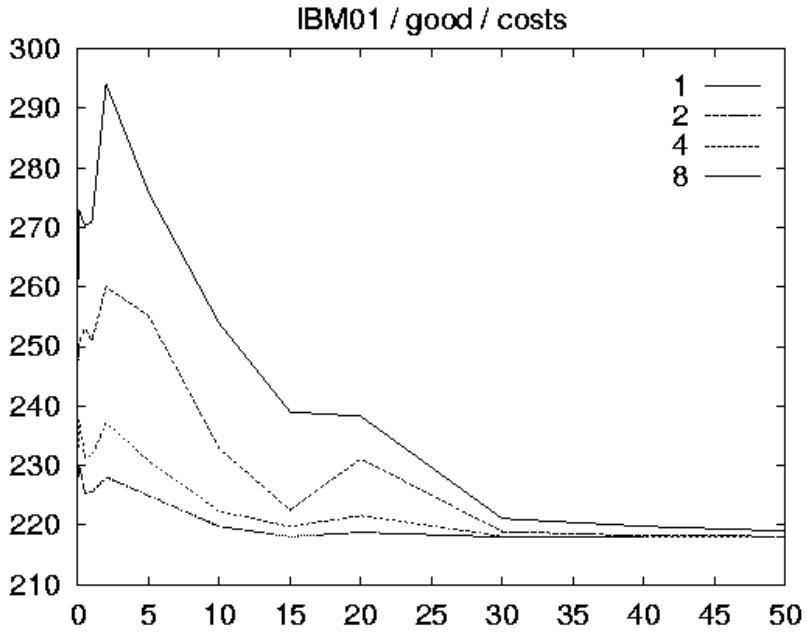
Motivation

- Hypergraph partitioning research has large, active, 30-year history
- Driving application: top-down VLSI placement
- In top-down VLSI placement, all partitioning instances have fixed terminals (I/O pads, terminal propagation, etc.)
  - Rent parameter analysis: % of vertices fixed is quite large!
- In VLSI partitioning research, all literature and benchmarks are for free hypergraphs (no terminals)

Motivating Experiment

- Run multi-start MLFM on hypergraph w/o fixed cell
- Record *average best* cutsize out of 1, 2, 4 and 8 starts
- Fix (assign) arbitrary 1% of cells, run again
- Fix (assign) more cells, run again
- Plot the average bests versus % of cells fixed
- Assignments:
  - random
  - as in a good (lowest-cost known) solution

Results With Fixed Terminals

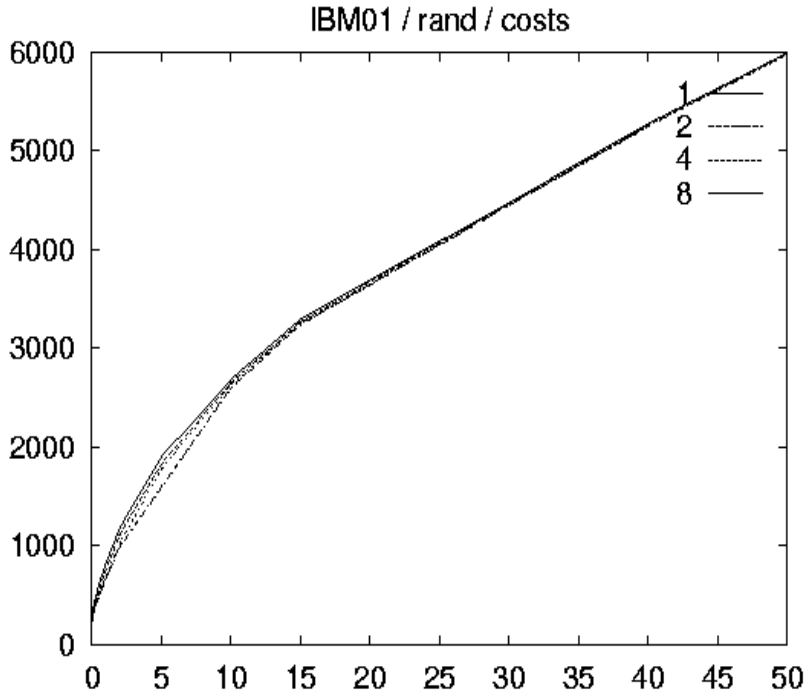


Terminals fixed in ‘good’ locations

- ‘Good’ assignments can ***increase*** achievable costs: consistent spike observed with 2-10% fixed in good locations
  - good solutions possible (and are found with <5% or >15% fixed)

  → failing of the partitioner: new heuristics are needed
- With more terminals, fewer starts are necessary

Results With Fixed Terminals



Terminals fixed in ‘random’ locations

- Only few % of nodes randomly assigned: cost doubles
- With more terminals, fewer starts are necessary
- Heuristics should exploit the terminals to gain speed
- New benchmarks needed for research

New Benchmarks With Terminals

- Based on ISPD '98 partitioning benchmarks
- Produced using final cell locations, and include terminal assignments
- Instances reflect regions in top-down placement
- http://vlsicad.cs.ucla.edu/benchmarks/ispd99/
- hMetis-1.5.3 partitioning results also available

New Benchmark Statistics

Circuit	Cells	Pads	ExtNets	Nets	Pins	Max%
IBM01A_L0	12506	246	246	14111	50566	6.37
IBM01B_L1_V0	6388	1392	761	7384	27236	9.34
IBM01C_L1_V1	6121	1377	763	7370	26951	10.03
IBM01D_L3_C11-33	6739	2155	1227	7330	28661	9.54
IBM06A_L0	32332	166	166	34826	128182	13.56
IBM06B_L1_V0	13245	4360	1867	14786	58809	17.28
IBM06C_L1_V1	19094	4086	1851	21824	81997	16.11
IBM06D_L3_C11-33	10314	7553	3482	12438	55640	18.88
IBM11A_L0	70152	406	406	81454	280786	4.48
IBM11B_L1_V0	33506	4970	3323	40623	142760	6.4
IBM11C_L1_V1	36646	5036	3291	43935	152703	6.29
IBM11D_L3_C11-33	30971	9968	6068	36273	129724	6.92
IBM16A_L0	182980	504	504	190048	778823	1.89
IBM16B_L1_V0	99102	9083	5200	103816	432680	2.88
IBM16C_L1_V1	83884	10491	5211	91190	375500	2.81
IBM16D_L3_C11-33	65041	18326	8581	70899	300966	2.82
IBM17A_L0	184752	743	743	189581	860036	0.94
IBM17B_L1_V0	100763	10834	6684	106877	485785	1.58
IBM17C_L1_V1	84015	12290	6716	89052	409926	1.19
IBM17D_L3_C11-33	51714	23033	11444	58988	263516	2.1

Proposed file formats

- Features
  - ASCII, human-readable/writeable
  - compatible with .netD and .are files
    - .fix contains terminal propagations (fixed assignments)
    - .blk contains target partition capacities/tolerance & geometries
  - partitioning tolerances: absolute (cell area units) or relative (%)
  - embedding information
    - fixed cell assignments and locations
    - multiple partition capacities and geometries
  - can model terminal propagation in top-down placement
- Applications
  - benchmarking, creation of artificial and synthetic testcases
  - standalone debugging and profiling of partitioners
  - benchmark instances generated automatically by top-down placer

Conclusions and Future Research

- Fixing cells simplifies partitioning problems
  - very non-linear effect
  - artifact of move-based heuristics?
- New file formats with extended syntax
  - supports partition geometries and fixed terminals
  - multi-balance and multi-way partitioning
  - plans for distribution of parser
- Open research questions
  - detailed analysis of fixed terminals' effect on iterative methods
  - heuristics that exploit fixed terminals (e.g., flow-based)
  - how to quantify effect of fixed cells (#fixed cells insufficient)