What's So Hard About (Mixed-Size) Placement?

Patrick H. Madden/SUNY Binghamton
pmadden@binghamton.edu

ISPD 2022 Panel on Traditional Algorithms vs. Machine Learning Approaches

Traditional Algorithms vs. Machine Learning Approaches

10:30 - 12:00 pm PDT. Session 4 - Panel on Traditional Algorithms Versus Machine Learning Approaches

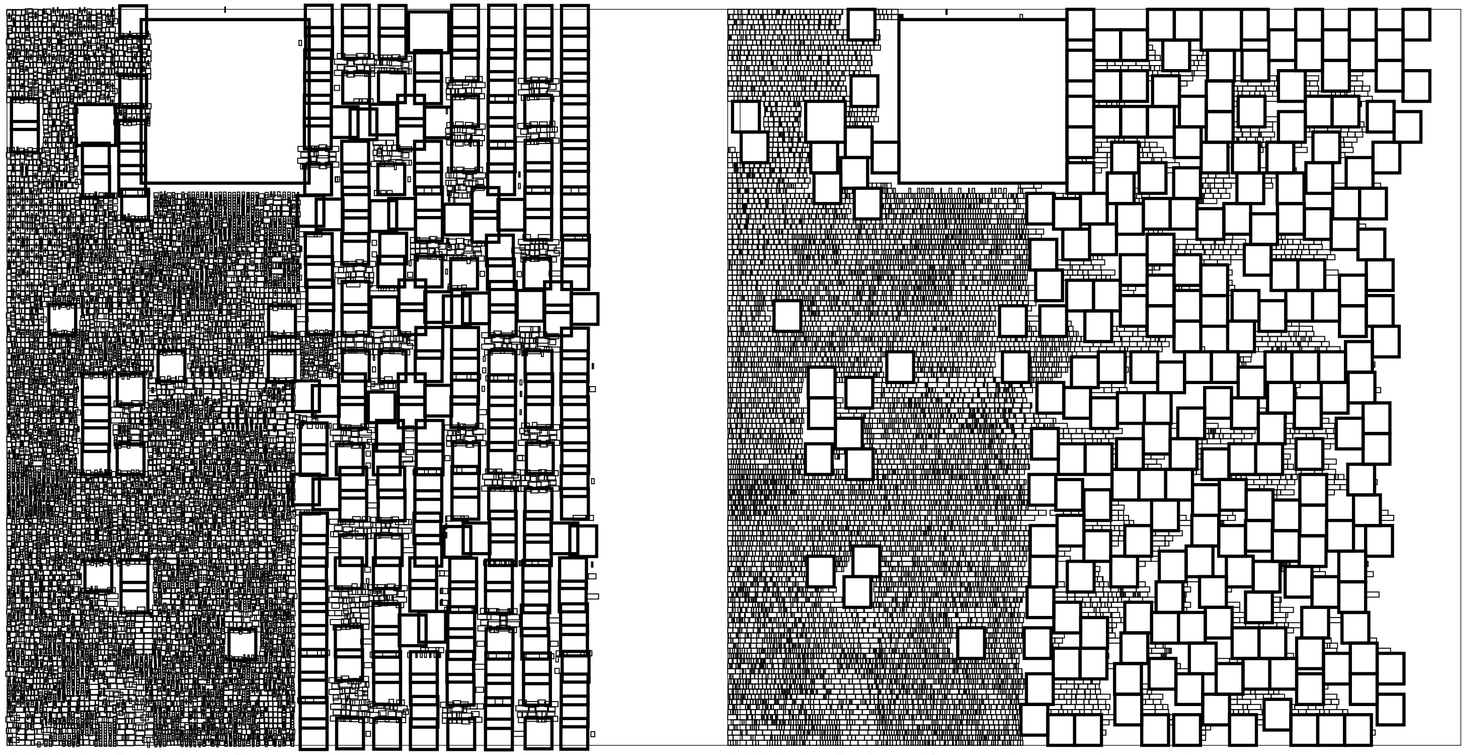
Chair: Patrick Groeneveld (Cerebras Systems)

1. "From Hard-Coded Heuristics to ML-Driven Optimization: New Frontiers for EDA", Patrick Groeneveld (Cerebras Systems)
2. "Embracing Machine Learning in EDA", Mark Ren (NVIDIA)
3. "Multi-level AutoML Optimization for Vertical ML Platforms and How it Echoes in the EDA Design Flow", Igor Markov (University of Michigan and Meta)
4. "What's So Hard About (Mixed-Size) Placement?", Patrick Madden (State University of New York at Binghamton)
5. "Scalability and Generalization of Circuit Training for Chip Floorplanning", Anna Goldie and Azalia Mirhoseini (Google Brain)

I'm firmly on the "**traditional algorithms**" side of things.

This should make for a fun panel!

What’s so hard about (mixed-size) placement?



IBM 01 before legalization

IBM 01 after legalization

Exponentially large solution space

A mix of NP-Complete problems
Facility location, bin packing, ….
Plus the down-stream effects on routing, timing, power, ….

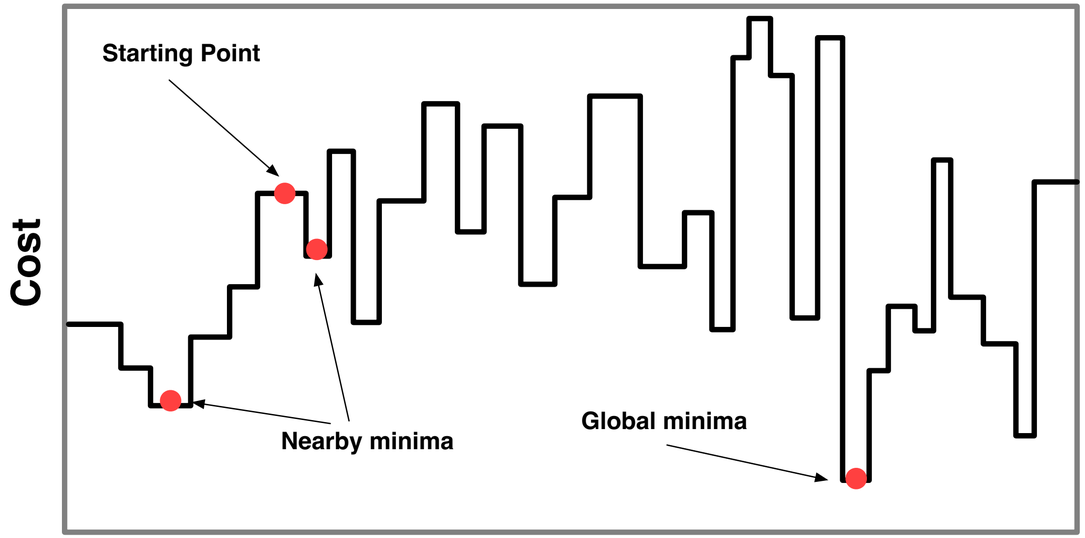
Big macro blocks cause a puzzle-fit sort of challenge

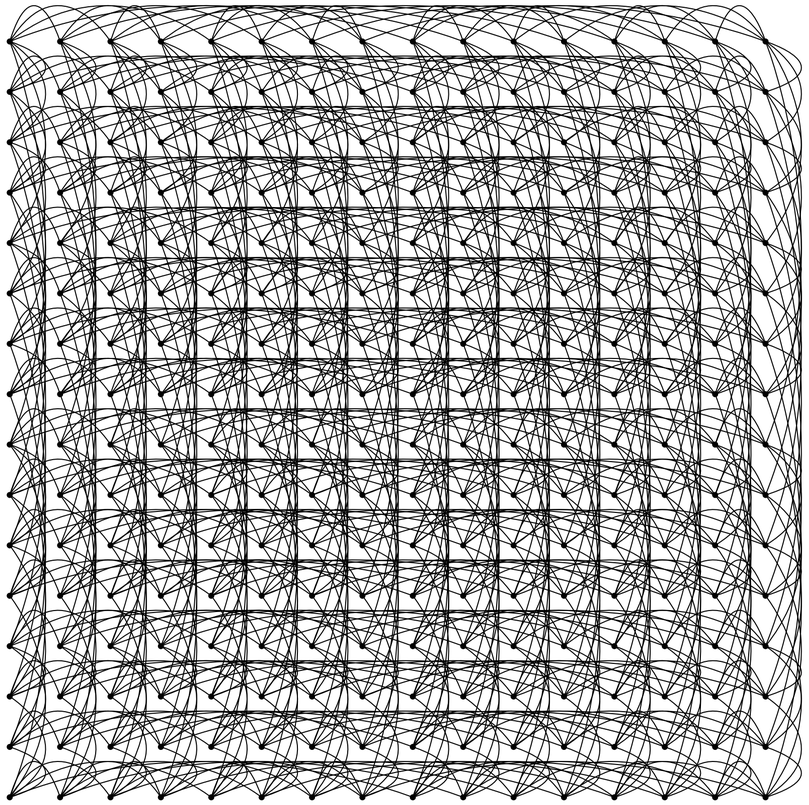
The size differential between macro blocks and standard cells is probably the trickiest thing to deal with

EDA Optimization Problems

minimize *f(x1, x2, x3, ….)*

In EDA, the variables are typically discrete, leaving a jagged cost surface with many local minima. The underlying decision problems are frequently NP-Complete. There are no easy ways to decompose into tractable subproblems, or place tight bounds on solution quality.





Solution space for eight binary variables (256 possible configurations).

Yes, it's a mess, even for n=8.

For circuit designs, we have n in the millions (or billions)

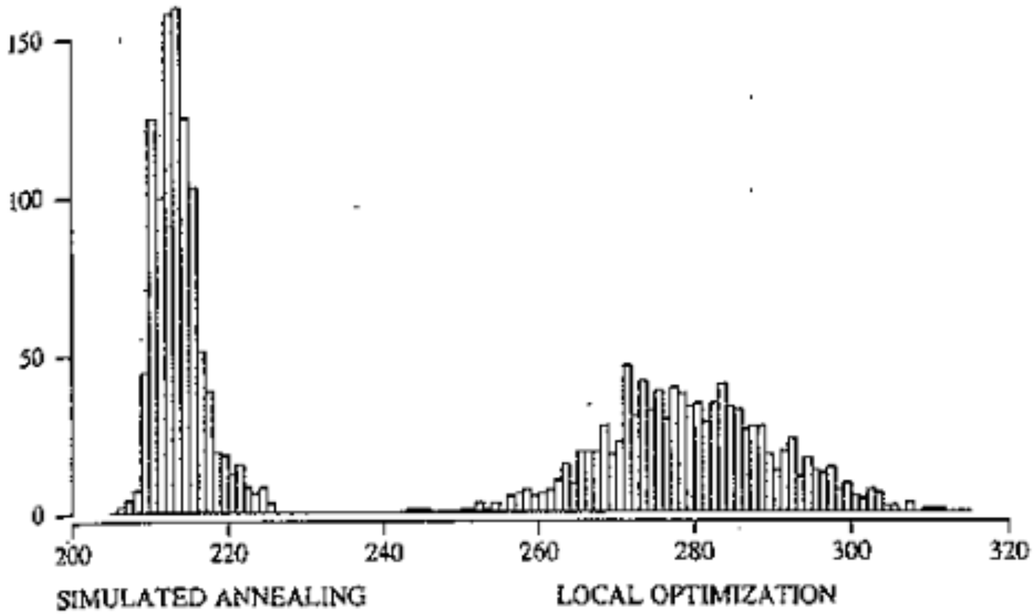
And it's frequently not just a binary decision. Often, the solution space is O(n!), or worse.

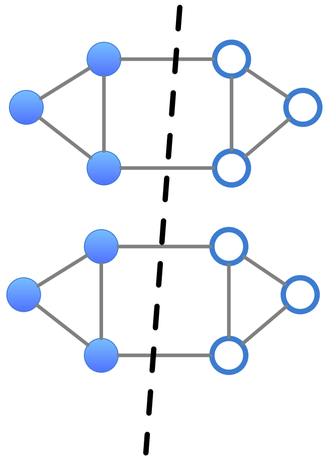
Graph Partitioning

A classic NP-Complete problem, with an interesting optimization variant

Given a graph, split it into two roughly-equally sized parts, minimizing the number of cut edges

Lots (and lots and lots) of local minima. Local optimization runs out of gas.





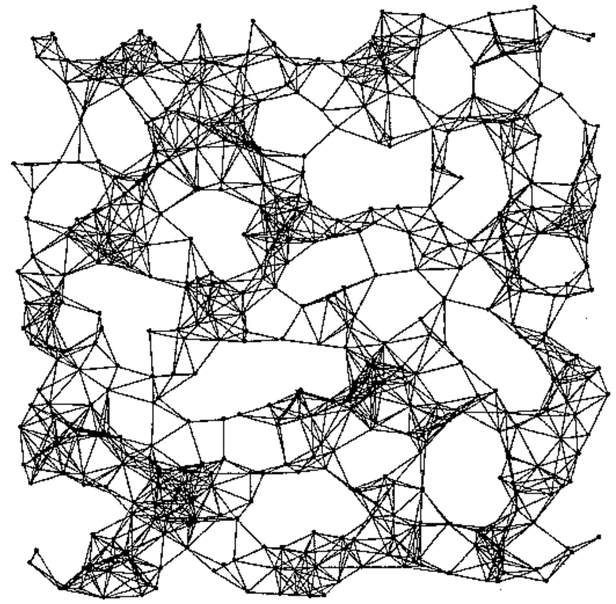


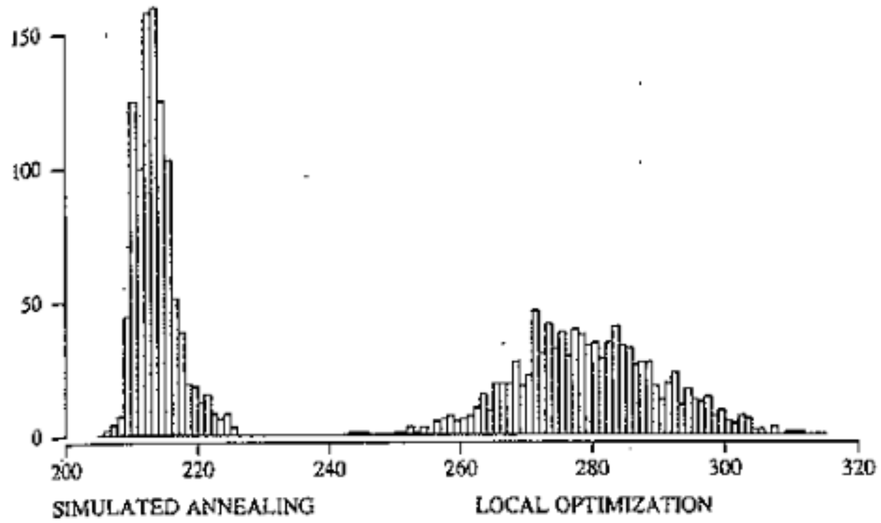
Figure 8. A geometric graph with $n = 500$ and $n\pi d^2 = 10$.

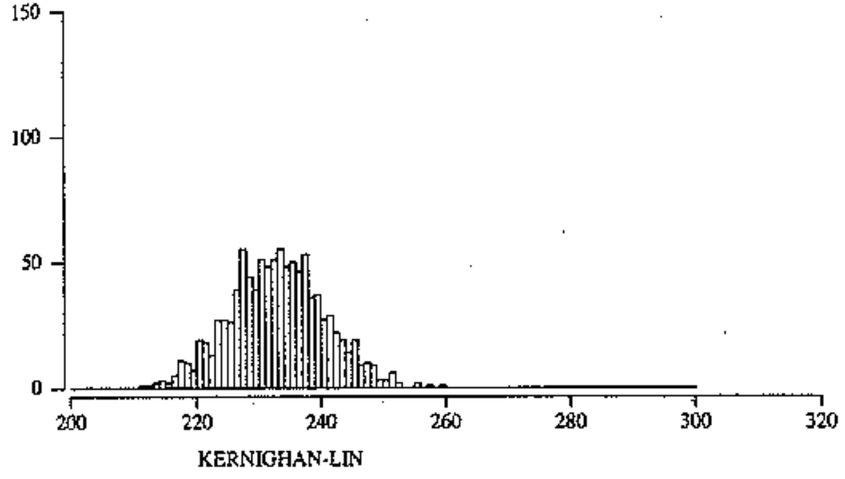
D. S. Johnson, C. R. Aragon, L. A. McGeoch, C. Schevon “Optimization by Simulated Annealing: An Experimental Evaluation”
Operations Research, Vol. 37, No. 6, Nov-Dec 1989

Annealing is Good….
but Kernighan-Lin is good too

Table I
Comparison of Annealing and Kernighan-Lin on G_{500}

k	Anneal (Best of k)	K-L (Best of k)	K-L (Best of $100k$)
1	213.32	232.29	214.33
2	211.66	227.92	213.19
5	210.27	223.30	212.03
10	209.53	220.49	211.38
25	208.76	217.51	210.81
50	208.20	215.75	210.50
100	207.59	214.33	210.00





Experiments from D. S. Johnson Operations Research paper, implementing KL.

B. Kernighan, S. Lin, “an effective heuristic procedure for partitioning graphs”
Bell Systems Technical Journal, pp. 291-308, 1970

Partitioners Kept Getting Better

B. Kernighan and S. Lin, “An effective heuristic procedure for partitioning graphs”
Bell System Technical Journal, 1970

C. M. Fiduccia and R. M Mattheyses, “A Linear-Time Heuristic for Improving Network Partitions”
Proc. Design Automation Conference, 1982

J. Cong and M. Smith, “Parallel bottom-up clustering algorithm with applications to circuit partitioning in VLSI design”
Proc. Design Automation Conference, 1993

G. Karypis, V. Kumar, “Multilevel Hypergraph Partitioning”
Proc. Design Automation Conference, 1999

BM	Module	Nets	FM Min	FM Avg	CLIPMin	Clip Avg	hMetis Min	hMetis Avg	FM/hMetis	FM/hMetis	CLIP/H min	CLIP/H avg
ibm01	12752	14111	191	466	181	390	181	236	1.06	1.97	1.00	1.65
ibm02	19501	19584	266	506	265	545	262	312	1.02	1.62	1.01	1.75
ibm03	23136	27401	1150	2131	1068	1593	959	1068	1.20	2.00	1.11	1.49
ibm04	27507	31970	603	1105	563	1030	542	588	1.11	1.88	1.04	1.75
ibm05	29374	28446	1874	3063	2146	3016	1740	1838	1.08	1.67	1.23	1.64
ibm06	32498	34826	973	1384	977	1520	885	1023	1.10	1.35	1.10	1.49
ibm07	45926	48117	1037	2036	929	1987	848	930	1.22	2.19	1.10	2.14
ibm08	51309	50513	1285	2757	1261	2137	1159	1194	1.11	2.31	1.09	1.79
ibm09	53395	60902	912	2547	674	1770	624	685	1.46	3.72	1.08	2.58
ibm10	69429	75196	1490	2260	1420	2745	1265	1573	1.18	1.44	1.12	1.75
ibm11	70558	81454	1459	4173	1063	2657	963	1146	1.52	3.64	1.10	2.32
ibm12	71706	77240	2256	3791	2387	3770	1899	2133	1.19	1.78	1.26	1.77
ibm13	84199	99666	1181	2249	913	1955	841	979	1.40	2.30	1.09	2.00
ibm14	147605	152772	2963	6824	2356	4176	1928	2126	1.54	3.21	1.22	1.96
ibm15	161570	186608	5106	7770	3571	5689	2750	3218	1.86	2.41	1.30	1.77
ibm16	183484	190048	2363	5668	2638	5974	1758	2339	1.34	2.42	1.50	2.55
ibm17	185495	189581	3052	7212	2803	6998	2341	2430	1.30	2.97	1.20	2.88
ibm18	210613	201920	1706	3686	2268	5227	1528	1669	1.12	2.21	1.48	3.13
								Ratio:	1.27	2.28	1.17	2.02

Table taken from C. Alpert, “The ISPD98 Circuit Benchmark Suite,” Proc. ISPD 1998

Run times improved

KL is $O(n^2 \log n)$, while FM is $O(n)$

Clustering, and in particular multilevel clustering, is a game changer

Modern multi-level partitioners are astonishingly good

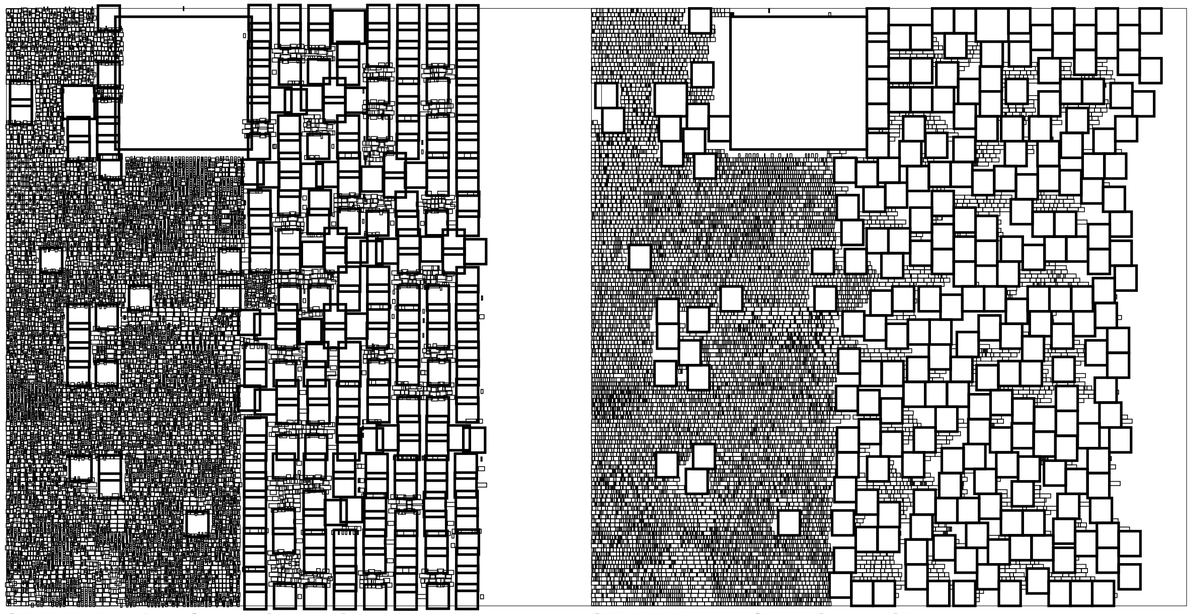
The core ideas are applicable to a wide range of problems. It’s not just partitioning.

Mixed Size Placement circa 2004

Armed with the hMetis partitioner, we built a remarkably simple recursive bisection based placer, ***feng shui***

“Fractional cut” bisection, tracking only the circuit area, followed by “tetris” placement legalization.

It’s about as simple as you can get.



IBM 01 before legalization

IBM 01 after legalization

A. Khatkhate, C. Li, A. R. Agnihotri, M. C. Yildiz, C.-K. Koh, P. H. Madden, “Recursive Bisection Based Mixed Block Placement” Proc. ISPD 2004

Mixed Size Placement circa 2004

Bench mark	*Capo I*[2]		*Capo II*[3]		*Capo III*[3]		*mPG*[8]		*Feng Shui 2.4*				
	HPWL	CPU (min)	HPWL	CPU (min)	HPWL	CPU (min)	HPWL	CPU (min)	HPWL	%Better (Capo)	%Better (MPG)	CPU (min)	Legal. **(sec)**
ibm01	3.96	18	3.36	13	3.05	20	3.01	18	2.41	26.56	24.90	3	1
ibm02	8.37	31	8.23	240	6.83	11	7.42	32	5.34	27.90	38.95	5	< 1
ibm03	12.16	42	11.53	22	10.38	59	11.20	32	7.51	38.22	49.13	6	2
ibm04	13.48	47	11.93	25	10.11	15	10.50	42	7.96	27.01	31.91	7	< 1
ibm05	11.51	8	11.20	5	11.10	5	10.90	36	10.10	9.90	7.92	8	1
ibm06	10.25	56	9.63	19	9.94	18	9.21	45	6.82	41.20	35.04	10	2
ibm07	15.75	58	15.80	39	15.25	25	13.70	68	11.71	30.23	16.99	13	1
ibm08	21.18	94	18.85	111	17.91	29	16.40	82	13.60	31.69	20.59	16	1
ibm09	19.59	66	17.52	178	19.88	29	18.60	84	13.83	26.68	34.49	15	1
ibm10	60.72	229	53.58	490	45.46	116	43.60	172	37.48	21.29	16.33	22	17
ibm11	28.49	106	26.47	69	29.40	45	26.50	112	19.96	32.62	32.77	21	2
ibm12	51.74	675	55.12	119	55.79	25	44.30	153	35.57	45.46	24.54	23	3
ibm13	39.39	151	33.56	88	37.73	53	37.70	151	24.95	34.51	51.10	26	2
ibm14	56.19	286	52.67	333	50.26	155	43.50	276	38.48	30.61	13.05	52	5
ibm15	70.48	237	64.69	264	65.00	195	65.50	385	52.14	24.07	25.62	87	10
ibm16	N/A	N/A	83.14	580	90.01	162	72.40	436	61.33	35.56	18.05	93	15
ibm17	92.38	443	91.50	249	89.17	188	78.50	606	70.60	26.30	11.19	104	16
ibm18	54.90	378	54.11	397	51.84	127	50.70	437	45.05	15.07	12.54	114	18
Avg										29.16	25.84		

Close to 30% better on average, compared to other work from the era.

A. Khatkhate, C. Li, A. R. Agnihotri, M. C. Yildiz, C.-K. Koh, P. H. Madden, “Recursive Bisection Based Mixed Block Placement” Proc. ISPD 2004

Mixed Size Placement circa 2004-2006

Benchmark	Capo	mPG-MS	feng shui	Aplace	Uplace
ibm01	0.31	0.30	0.24	0.23	0.25
ibm02	0.68	0.74	0.53	0.50	0.54
ibm03	1.04	1.20	0.75	0.72	0.73
ibm04	1.01	1.05	0.80	0.83	0.82
ibm05	1.11	1.09	1.01	0.98	1.09
ibm06	0.99	0.92	0.68	0.68	0.67
ibm07	1.53	1.37	1.17	1.05	1.18
ibm08	1.79	1.64	1.36	1.46	1.34
ibm09	1.99	1.86	1.38	1.38	1.48
ibm10	4.55	4.36	3.75	3.00	3.52

Benchmark	APlace2	feng shui	SCAMPI
040	20.70	20.60	18.80
098	22.60	24.00	30.70
336	2.20	7.60	3.30
353	4.60	31.50	6.30
523	27.50	348.70	37.10
542	0.70	x	0.80
566	46.90	493.60	69.30
583	20.60	x	25.10
588	4.80	x	6.90
643	3.00	15.30	3.70
DCT	33.10	184.70	37.20

Ouch! Placement legalization disaster!

Results from

A. B. Kahng and Q. Wang, “An Analytic Placer for Mixed-Size Placement and Timing-Driven Placement,” ICCAD 2004
B. Yao, H. Chen, C.-K. Cheng, N.-C. Chou, T.-L. Liu, P. Suaris, “Univied Quadratic Programming Approach for Mixed Mode Placement,” ISPD 2005
A. Ng, I. L. Markov, R. Aggarwal, V. Ramachandran, “Solving Hard Instances of Floorplacement,” ISPD 2006

Why is macro block placement still manual?

There are lots of good attempts at mixed size placement, but they have not caught on.

Why? Circuit design is an iterative process. The designers have an initial net list, and then edit/tweak the RTL until they get what they want.

A software developer doesn’t just run a compiler once. Neither does a circuit designer.

Large fixed macro blocks give stability and predictablity. In run after run of the tools, the flow converges to the desired design. Analytic placers work well with fixed macro blocks, fixed macro blocks work well with analytic placers, and the methodology that is in place is the one that is working well.

Traditional Algorithms vs. Machine Learning

IMO, if designers want automated mixed size placement, fractional cut recursive bisection is the way to go. Especially if there's a fix for the legalization problem.

The reinforcement learning approach is interesting, but I would really like to see results on standard benchmarks.

And to hit this point one more time:

Modern multi-level partitioners are amazing
There's a lot more here than one might expect!