Research on Global Placement and Routability Analysis

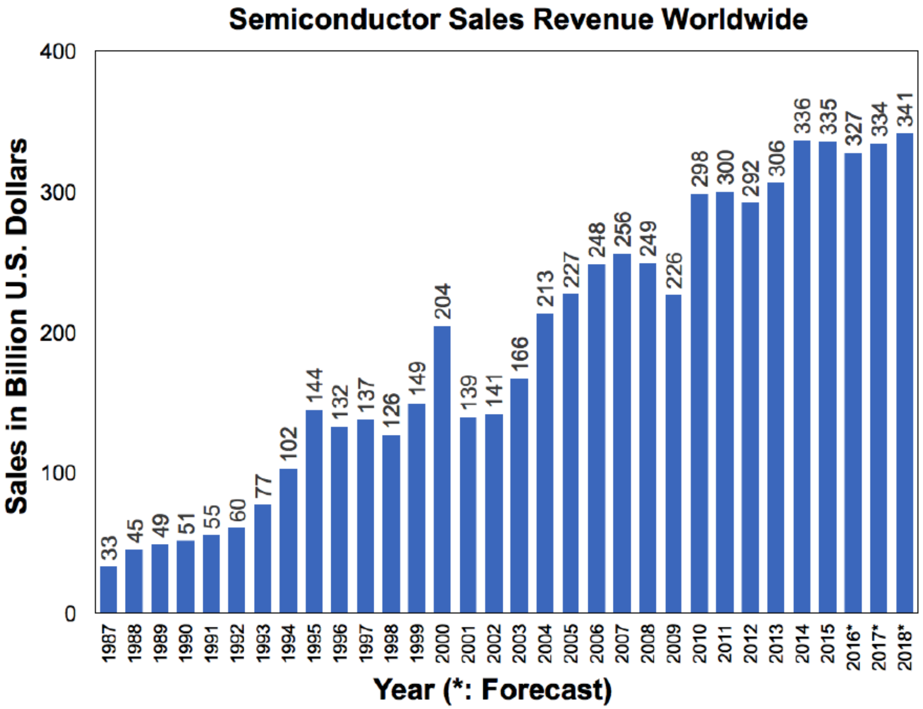
Chung-Kuan Cheng
CSE department
UC San Diego

OUTLINE

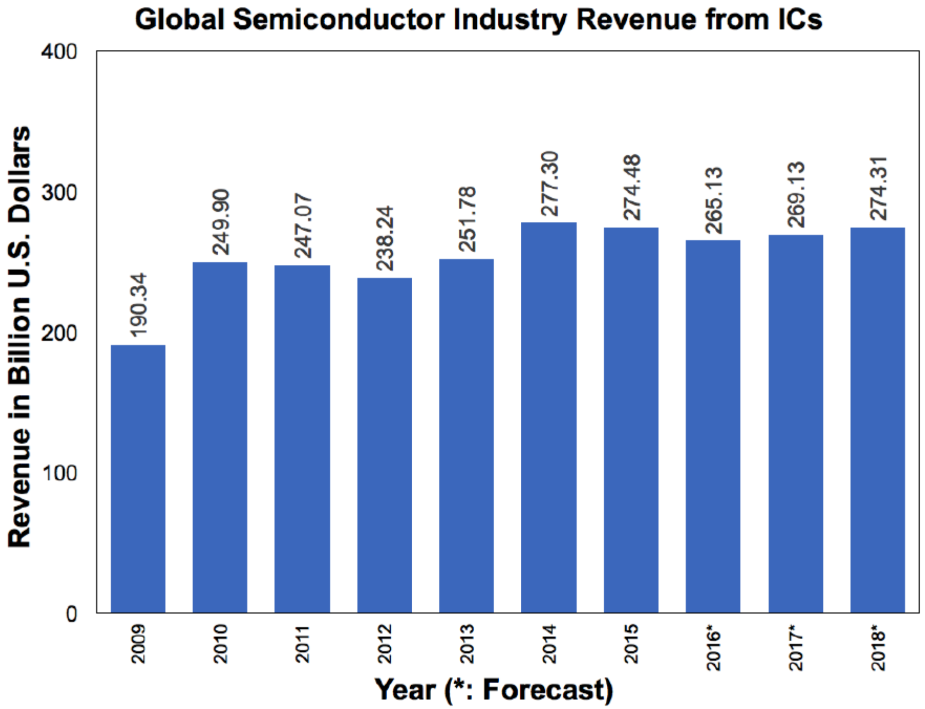
- Introduction
- Global Placement
    - Shadow Price for Cell Density Constraints
    - Mata Parameter Tuning
- Routability Analysis
    - Conditional Design Rules
    - ILP and SAT formulation
    - Routability Diagnosis
- Conclusion

INTEGRATED CIRCUIT (IC) INDUSTRY

- Contributed to the modern society and human welfare
- Physical designers: Design automation and optimization
  - Innovations and advancements
- Growing market size



(a) Semiconductor sales revenue worldwide from 1987 to 2018 (in billion U.S. dollars) [1].

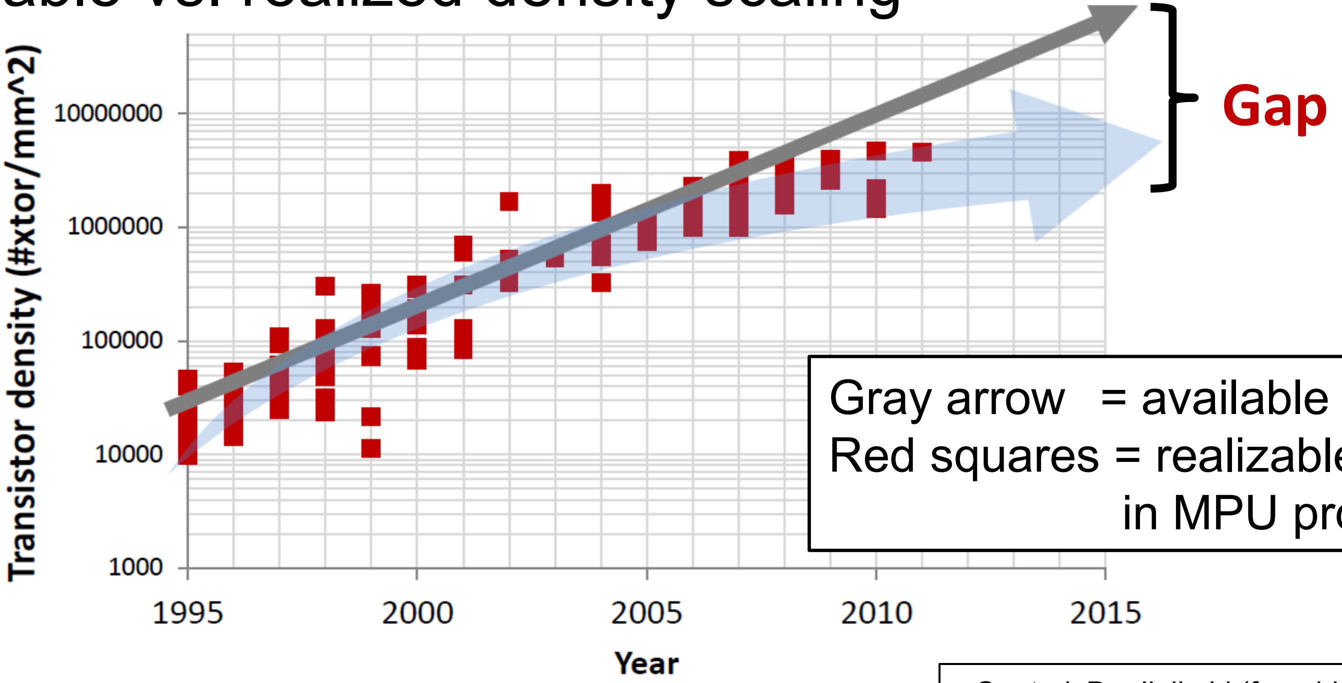


(b) Global semiconductor industry revenue from ICs from 2009 to 2018 (in billion U.S. dollars) [2].

Figures (a) and (b): [1] Semiconductor Sales Revenue Worldwide from 1987 to 2018, http://www.statista.com/statistics/266973/.
[2] Global Semiconductor Industry Revenue from ICs 2009-2018, http://www.statista.com/statistics/519456/.

CHALLENGES IN IC DESIGN

- “*Moore’s Law*” is slowing down…
  - “Design capability Gap” (2013 ITRS report) between available vs. realized density scaling



Gray arrow = available scaling
Red squares = realizable scaling in MPU products

Quoted, Dr. Jiajia Li (from his dissertation)

Compensate for design capability gap

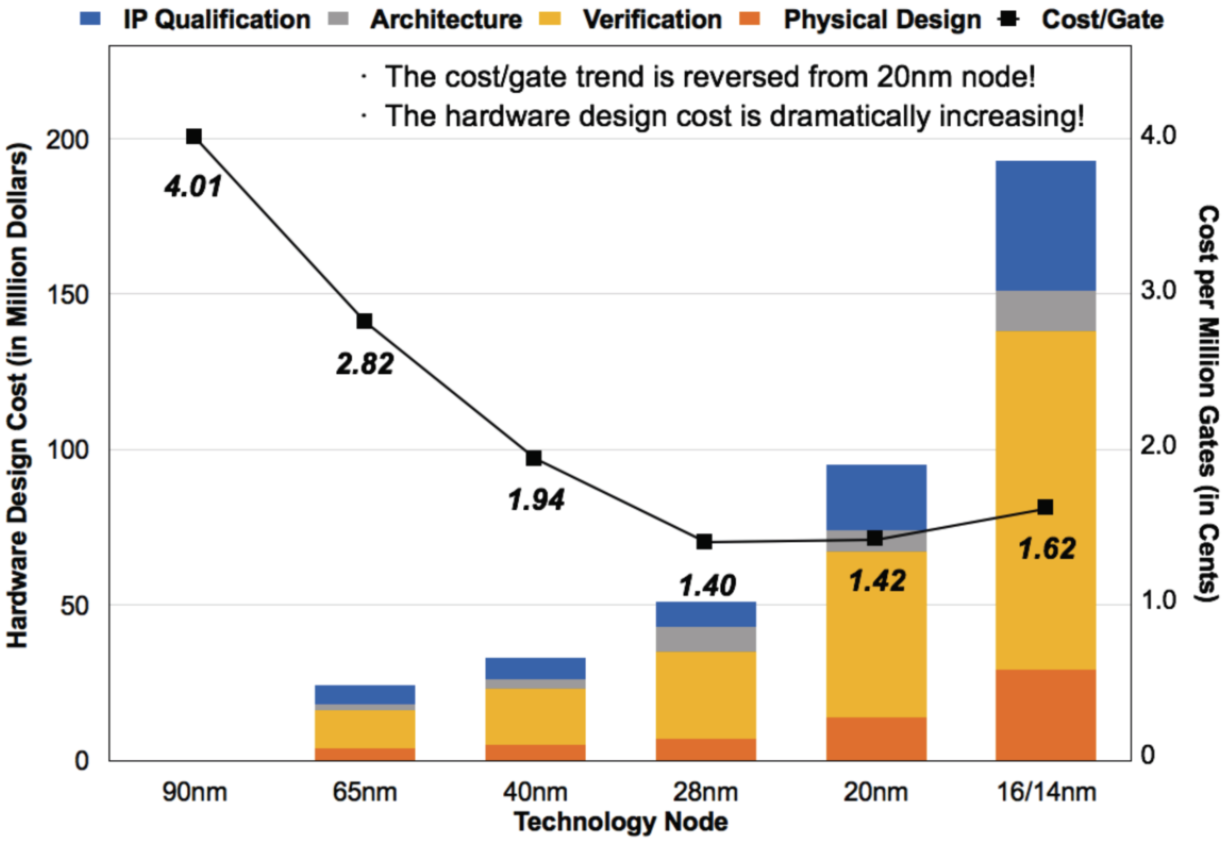
⇒ Design-based equivalent scaling = Better IC design and design methodology

(Design-based equivalent scaling: Design technologies that improve PPAC tradeoffs to rescue Moore’s-Law scaling of value)

Figure: [3] A. B. Kahng, H. Lee and J. Li, "Measuring Progress and Value of IC Implementation Technology", *Proc. ICCAD*, 2016, pp. 27:1-27:8.

CHALLENGES IN IC DESIGN

- Hardware design cost is rapidly increasing…
  - The cost/gate trend is reversed from 20nm node.
  - The hardware design cost is increasing.



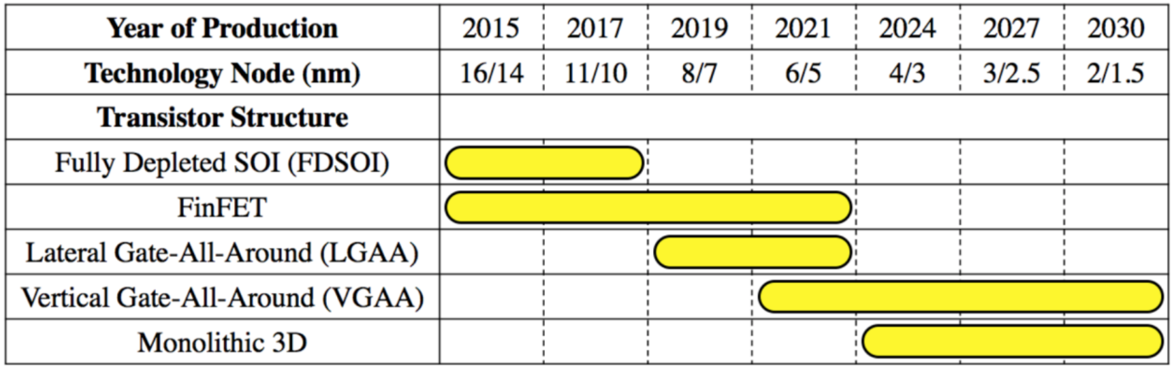
Bar chart: Distribution of hardware design cost per technology node

Line chart: Design cost per gate trend

Figure: [4] Hardware Design Cost: Faster, Cooler, Simpler, could FD-SOI be Cheaper too?, https://www.semiwiki.com/forum/content/2991-faster-cooler-simpler-could-fd-soi-cheaper-too.html.

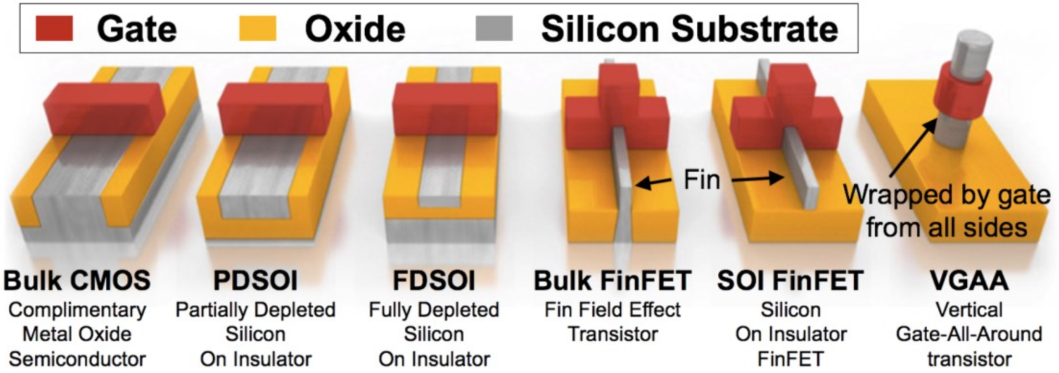
NEW OPPORTUNITIES

- Continuously shrinking logic device, but …
  - According to the latest ITRS [5] and the recent report from ASML [6]



Year of Production	2015	2017	2019	2021	2024	2027	2030
Technology Node (nm)	16/14	11/10	8/7	6/5	4/3	3/2.5	2/1.5
Transistor Structure							
Fully Depleted SOI (FDSOI)	🟨	🟨					
FinFET	🟨	🟨	🟨	🟨			
Lateral Gate-All-Around (LGAA)			🟨	🟨			
Vertical Gate-All-Around (VGAA)				🟨	🟨	🟨	🟨
Monolithic 3D					🟨	🟨	🟨

ITRS 2015 report


Gate
Oxide
Silicon Substrate
Fin
Wrapped by gate from all sides
Bulk CMOS
Complimentary Metal Oxide Semiconductor
PDSOI
Partially Depleted Silicon On Insulator
FDSOI
Fully Depleted Silicon On Insulator
Bulk FinFET
Fin Field Effect Transistor
SOI FinFET
Silicon On Insulator FinFET
VGAA
Vertical Gate-All-Around transistor

ASML report

We (i.e., IC designers and researchers) must prepare the future design methodology !!

Figures: [5] ITRS Report 2015 Edition, http://www.semiconductors.org/main/2015_international_technology_roadmap_for_semiconductors_itrs/.
[6] Many Ways to Shrink: The Right Moves to 10 Nanometer and Beyond, https://staticwww.asml.com/doclib/investor/asml_3_Investor_Day-Many_ways_to_shrink_MvdBrink1.pdf.

GLOBAL PLACEMENT OVERVIEW

- ***Global Placement***
  - Determines the locations of standard cells and/or logic elements while addressing optimization objectives
  - Highly important physical design step in integrated circuit (IC) design flow
    - Directly impacts on timing closure, die utilization, routability, design turnaround time (TAT) → operating frequency, yield, power consumption, cost
- **Placement instances**
  - *Hypergraph* $G = (V, E)$
    - V:= a set of vertices, i.e., standard cells and macros
    - E:= a set of *hyperedges*, i.e., nets
- **Placement solution** $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y})$, X- and Y-coordinates of all placeable vertices
  - *legal solution*?

  1. Every instance should be settled in the placement region.
  2. Every standard cell should be spaced within predefined rows.
  3. No overlap is allowed between instances including both standard cells and macros.

GLOBAL PLACEMENT FORMULATION

Global Placement

- $W(x,y) = \sum_n \max_{i\in n} x_i - min_{i\in n} x_i + \ max_{i\in n} y_i \ - min_{i\in n}\ y_i$
- Constraint: $Demand\ b < Capacity\ b\ \forall\ \mathrm{bin}\ b$.

ePlace

- *Function:* $W(x.y) + \lambda D(x,y)$,
  - $D(x,y)\ total\ charge\ potential$

Shadow Price of Primal Dual Formulation

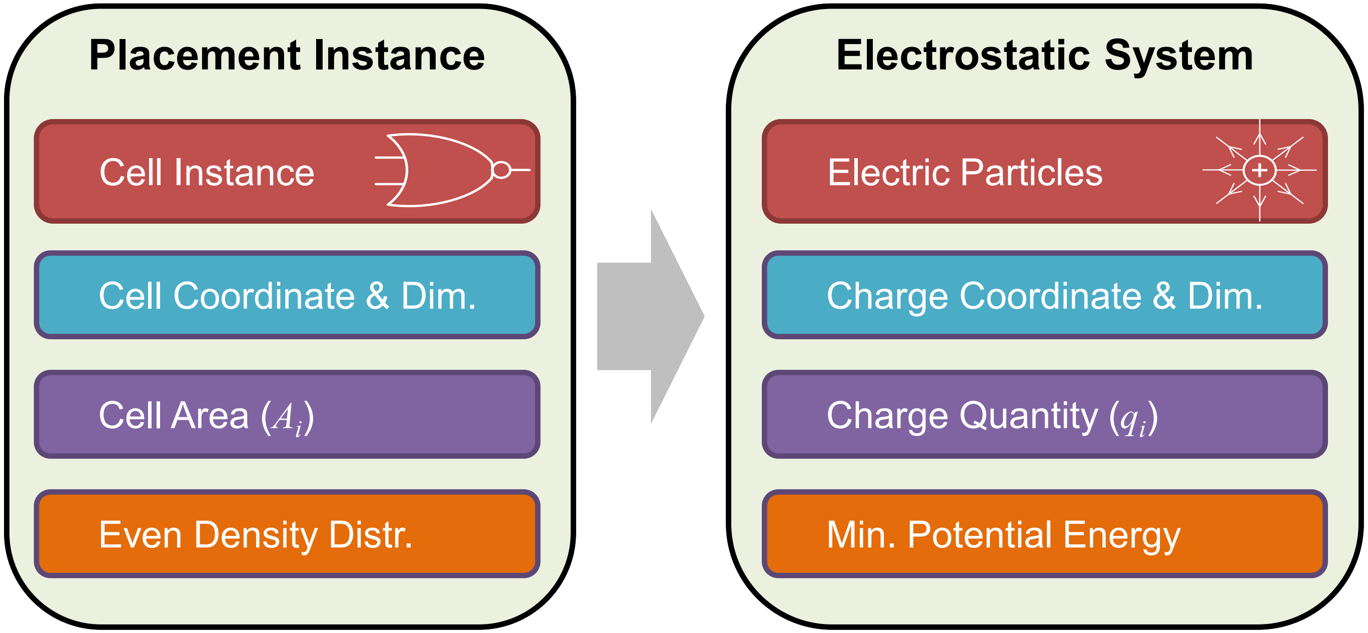
- *Function:* $W(x.y) + \sum_b \mathcal{V}_b\, D_b(x,y)$,
  - $D_b(x,y)\ charge\ potential\ in\ bin\ b$
- The shadow price of bin i is proportional to the demand/capacity constraint violation.

PROBLEM FORMULATION

- Placement Objective Function $f(v)$

$$\min_{\boldsymbol{v}} f(\boldsymbol{x},\boldsymbol{y}) = W(x,\boldsymbol{y}) + \sum_{b} \mathcal{V}_b \, D_b(x,y)$$

- ePlace: Electrostatics-based global-smooth density function



Placement Instance

Cell Instance

Cell Coordinate & Dim.

Cell Area (A_i)

Even Density Distr.

Electrostatic System

Electric Particles

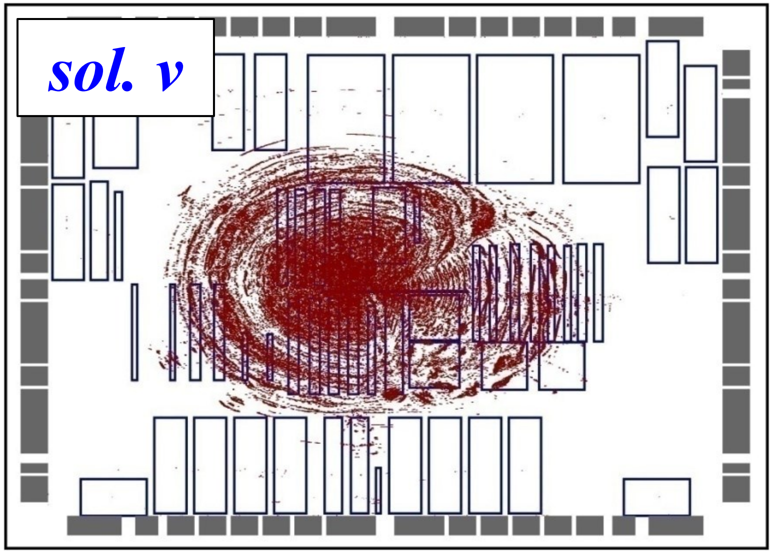
Charge Coordinate & Dim.

Charge Quantity (q_i)

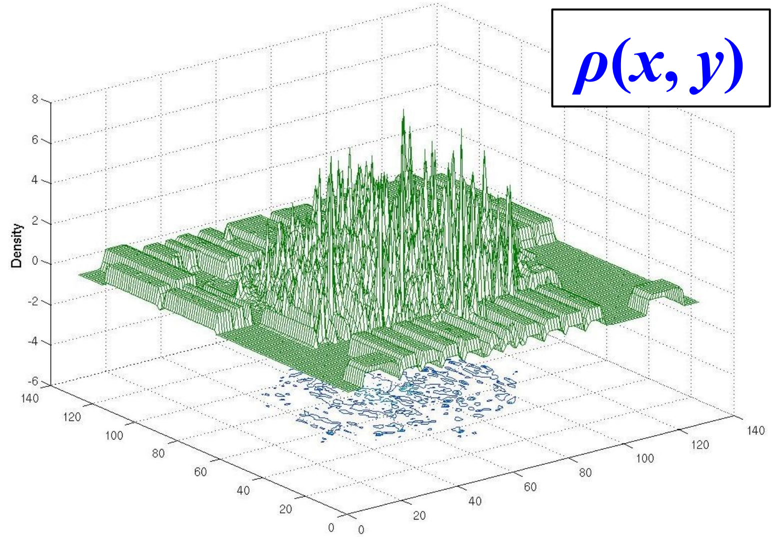
Min. Potential Energy

[13] J. Lu, “Analytic VLSI Placement Using Electrostatic Analogy”, Ph.D. Dissertation, UC San Diego, CA, 2014.

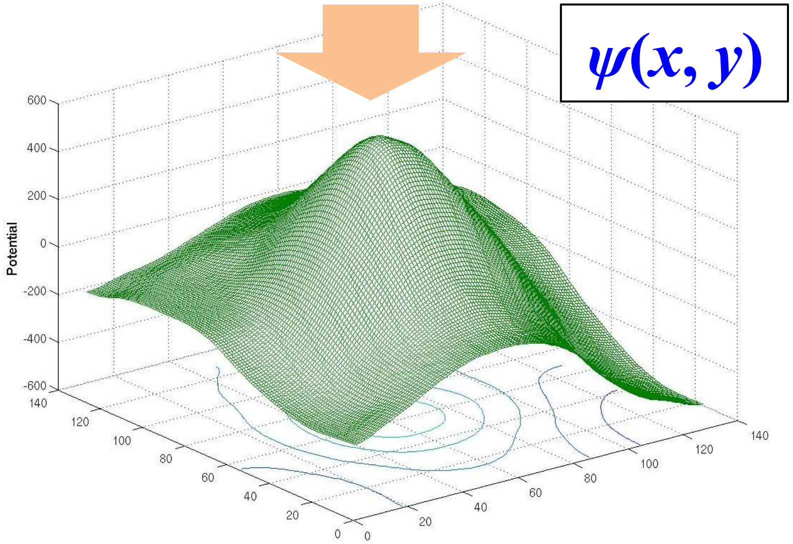
ePlace: In Each Iteration



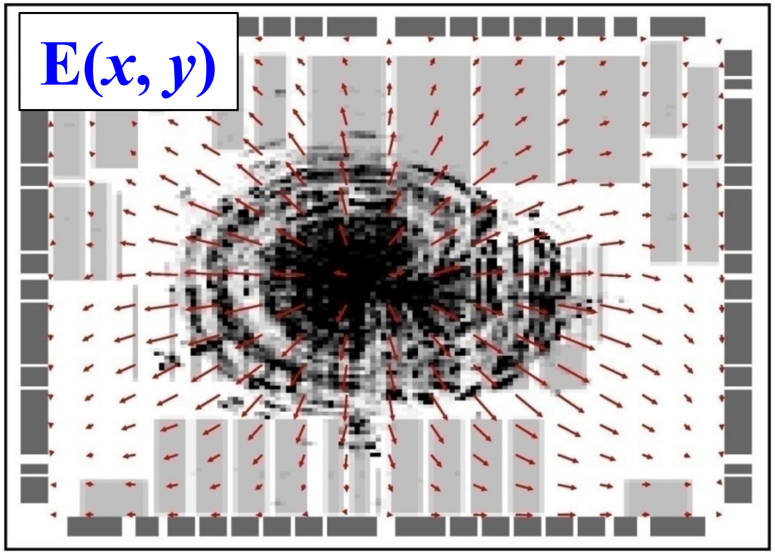
cell & macro distr.



charge density distr.



electric potential distr.



electric field distr.

[13] J. Lu, “Analytic VLSI Placement Using Electrostatic Analogy”, Ph.D. Dissertation, UC San Diego, CA, 2014.

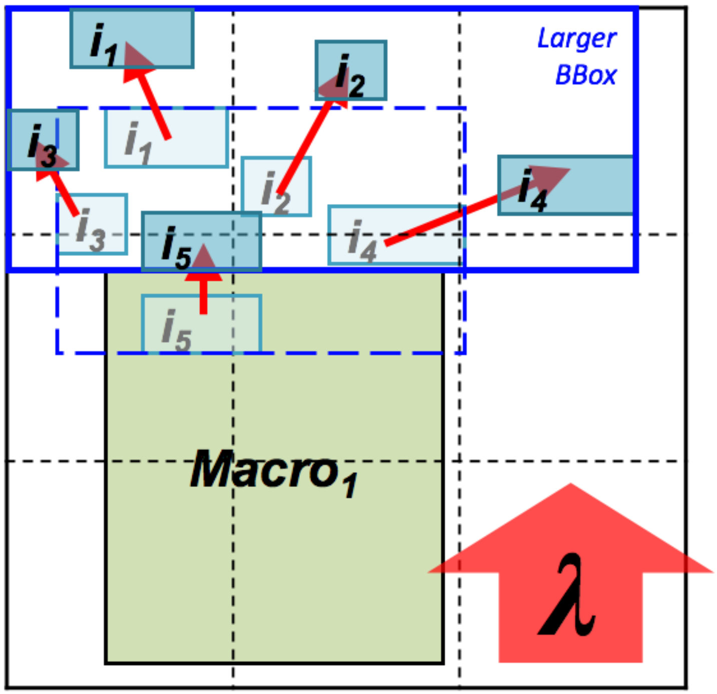
REPLACE

- RePlAce Overview
  - A single placement engine which solves five classes of benchmarks
  - Local Lagrange multiplier
    - Enables local smoothing that comprehend local over-demanded bins
  - Meta parameter tuning
    - Adjust step size of numerical method
  - Routability optimization
    - Simple but effective metal layer-aware superlinear cell inflation technique

- Differences from the previous *ePlace 2.0*
  - We conduct experiments using DAC-2012 and ICCAD-2012 global routability-driven placement benchmark suites with new routability optimization techniques, significantly improving over best known results.
  - Code optimizations have significantly improved runtime for RePlAce (≈4x faster)
  - Misc. improvements contribute to improve the existing placement mechanism
    - E.g., macro legalization using annealing, the amount of rollback after fixing macro, pushing more overflow, bin size determination and local smoothing, taking-off z-dimension computations, etc.

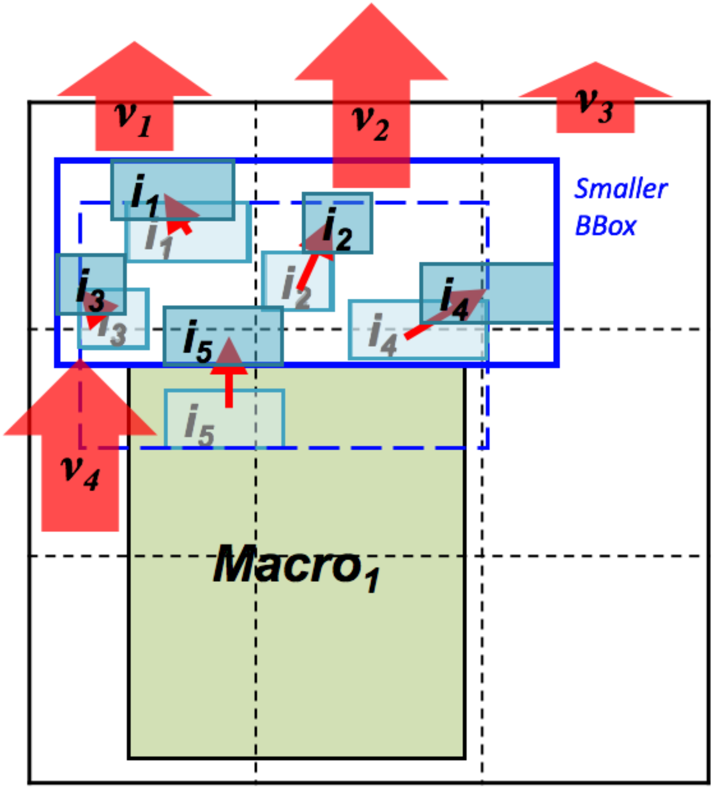
REPLACE: CONSTRAINT-DRIVEN PLACEMENT

- Formulation: Local Lagrangian multiplier
  - $\mathcal{V}_b = exp(\alpha(Demand_b - Capacity_b)/Capacity_b)$



Global λ ↑, and Larger HPWL ↑

[ePlace]

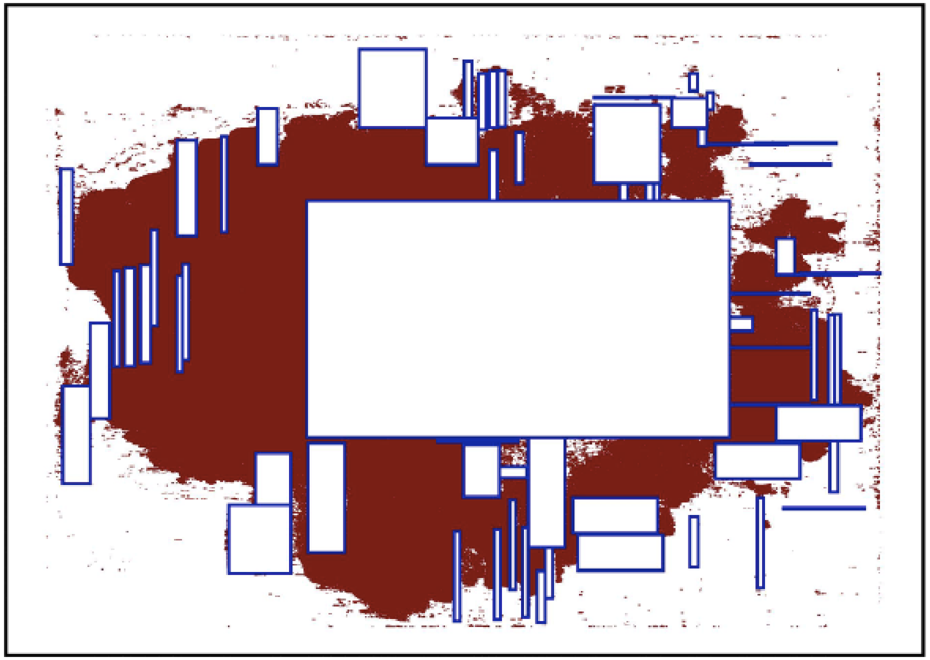


Local ν_i ↑, and Smaller HPWL ↑

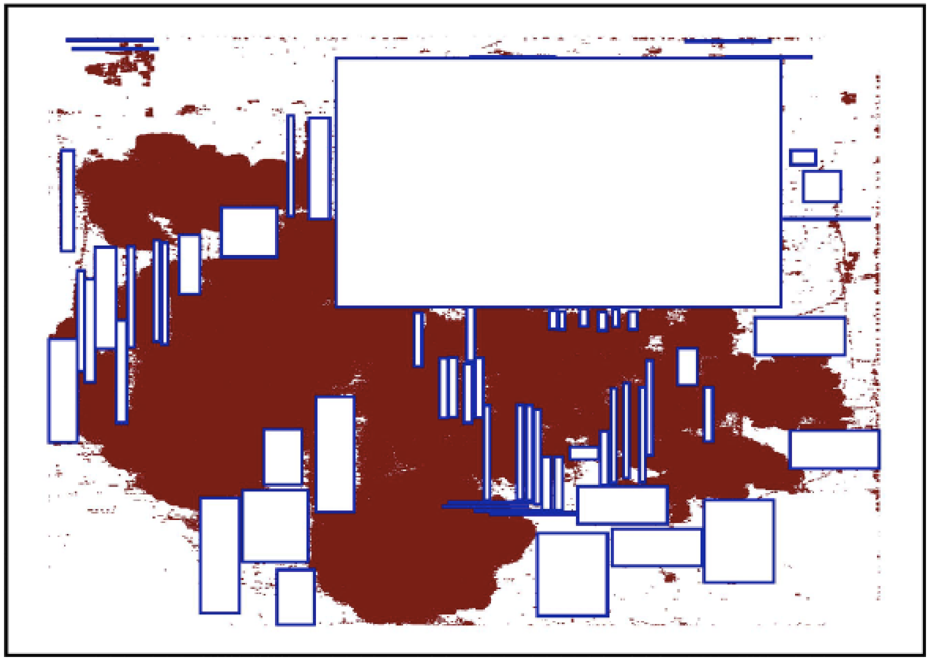
[RePlAce with Constraint-Driven]

RePLAce: Constraint-Driven Placement

- newblue1 (MMS [14])
  - Final placement layout after legalization and detailed placement



[RePlAce without local Lagrangian multiplier]
HPWL = 6.39E+7, #Iter = 708 (580+128), run = 9.2 min
Target density = 100%



[RePlAce with local Lagrangian multiplier]
HPWL = 5.60E+7, #Iter = 762 (623+139), run = 9.9 min
Target density = 100%

In the present *RePlAce*, we significantly improve the solution quality and runtime.

[ePlace 2.0 without local Lagrangian multiplier]

- HPWL = 6.38E+7
- #Iter = 1089 (615 + 474)
- runT = 43 min

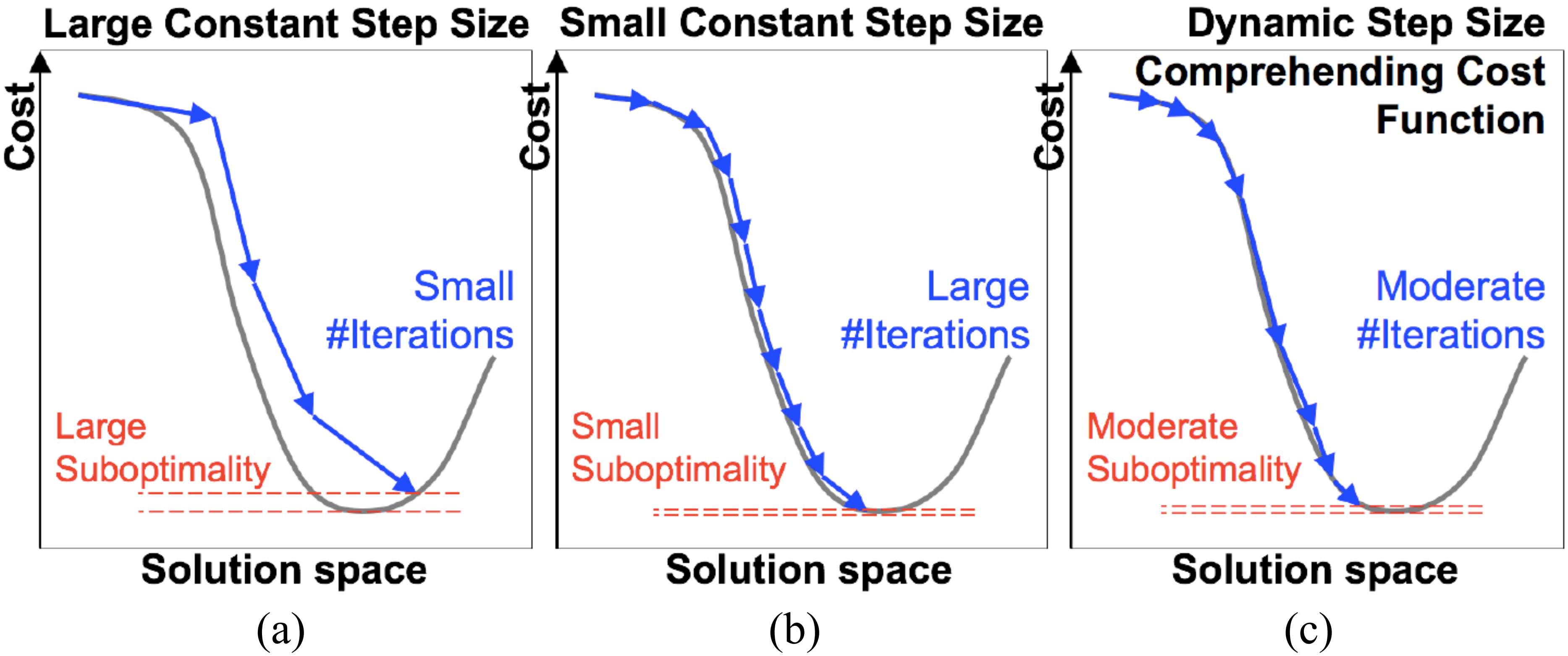
[ePlace 2.0 with local Lagrangian multiplier]

- HPWL = 5.71E+7
- #Iter = 1078 (609 + 469)
- runT = 42 min

[14] J. Z. Yan, N. Viswanathan and C. Chu, "Handling Complexities in Modern Large-Scale Mixed-Size Placement", *Proc. DAC*, 2009, pp. 436-441.

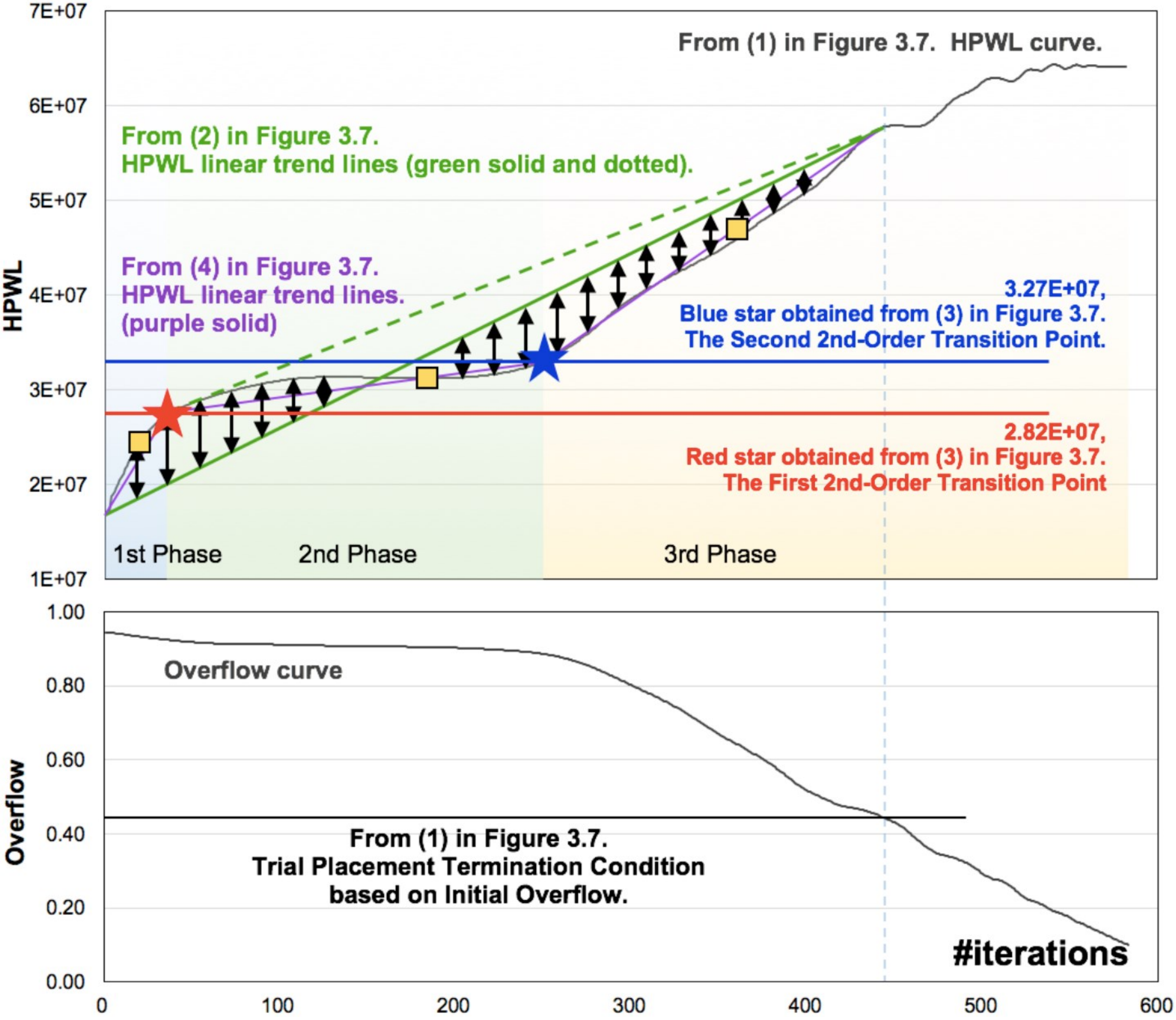
REPLACE: META PARAMETER TUNING

- General Idea of Dynamic Step Size



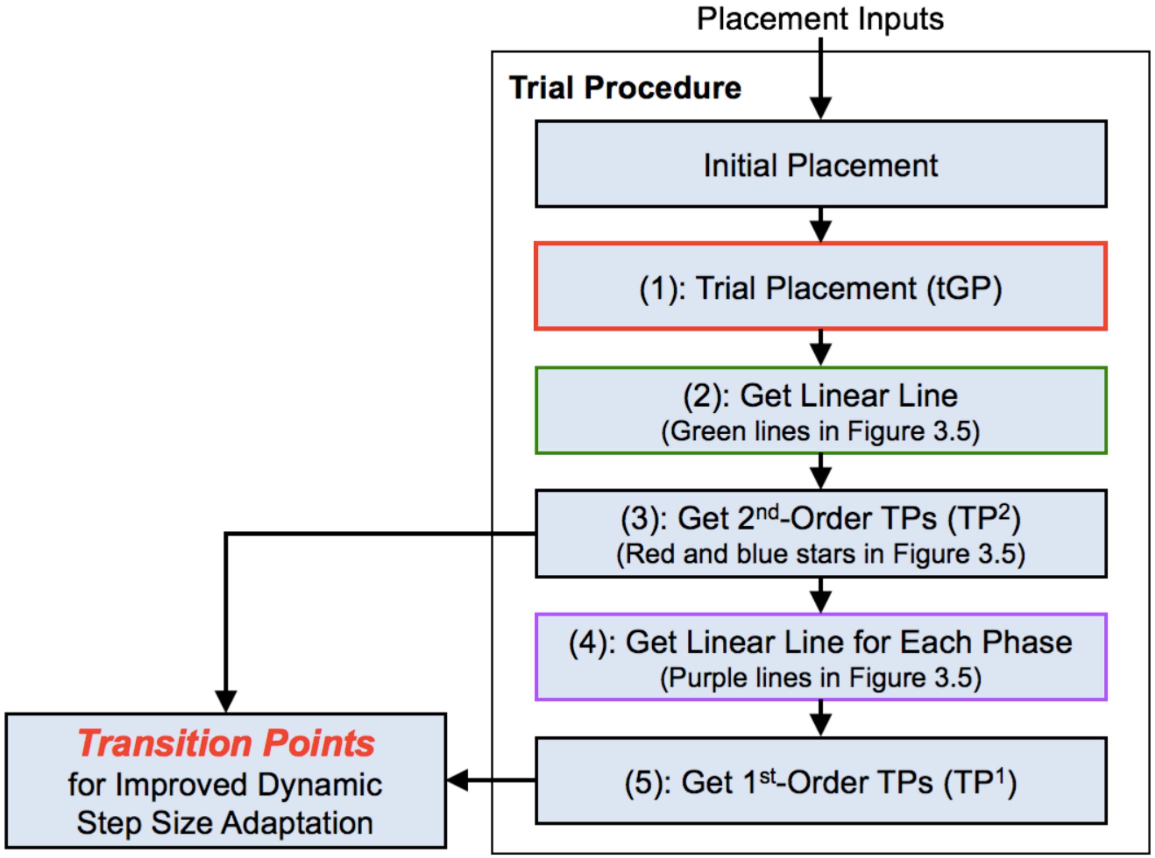
REPLACE: IMPROVED DYNAMIC STEP SIZE

ADAPTEC1 trial placement procedure and estimated transition points



REPLACE: IMPROVED DYNAMIC STEP SIZE

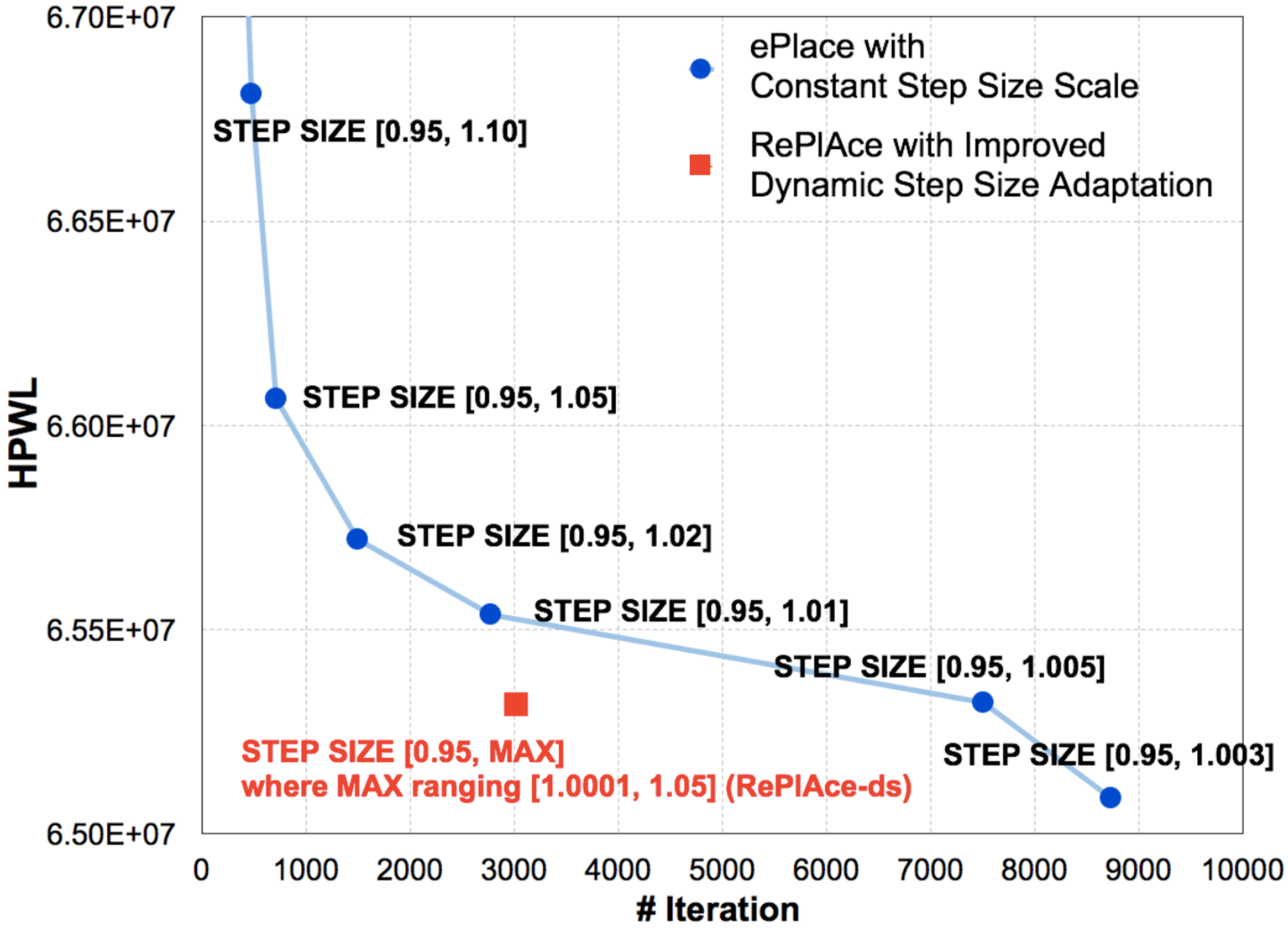
- Trial procedure to capture transition points on HPWL curve.


Placement Inputs
Trial Procedure
Initial Placement
(1): Trial Placement (tGP)
(2): Get Linear Line
(Green lines in Figure 3.5)
(3): Get 2nd-Order TPs (TP²)
(Red and blue stars in Figure 3.5)
(4): Get Linear Line for Each Phase
(Purple lines in Figure 3.5)
(5): Get 1st-Order TPs (TP¹)
Transition Points
for Improved Dynamic
Step Size Adaptation

▲ Flowchart of our trial placement procedure. The red rectangle indicates nonlinear optimization using Nesterov's method. The actual placement procedure follows this tGP procedure.

RePLAce: Improved Dynamic Step Size

- Solution quality in terms of the final HPWL
  - ePlace vs. RePlAce-ds (ADAPTEC1)
    - RePlAce-ds achieves a dominating runtime and solution quality (red square)
    - Below [0.95, 1.003], e.g., [0.95, 1.002], [0.95, 1.001], etc., results are not converged



REPLACE: ROUTABILITY-DRIVEN PLACEMENT

- Simple but effective routability optimization
  - A layer-aware cell inflation technique
  - Integrate the official global router NCTU-GR [17] of the DAC-2012 [18] and ICCAD-2012 [19] benchmark suites for congestion estimation.
  - Superlinear cell inflation technique to mitigate global routing congestion during global placement.
  - We further include a post-placement optimization by [20]
    - Following the strategy of recent leading works [21] [22]

[17] NCTU-GR, http://people.cs.nctu.edu.tw/˜whliu/NCTU-GR.htm

[18] N. Viswanathan, C. J. Alpert, C. N. Sze, Z. Li and Y. Wei, "The DAC 2012 Routability-driven Placement Contest and Benchmark Suite", *Proc. DAC*, 2012, pp. 774-782.

[19] N. Viswanathan, C. J. Alpert, C. N. Sze, Z. Li and Y. Wei, "ICCAD-2012 CAD Contest in Design Hierarchy Aware Routability-Driven Placement and Benchmark Suite", *Proc. ICCAD*, 2012, pp. 345-348.

[20] W.-H. Liu, C.-K. Koh and Y.-L. Li, "Optimization of Placement Solutions for Routability", *Proc. DAC*, 2013, pp. 1-9.

[21] X. He, T. Huang, L. Xiao, H. Tian and E. F. Y. Young, "Ripple: A Robust and Effective Routability-Driven Placer", *IEEE Trans. on CAD* 32(10) (2013), pp. 1546-1556.

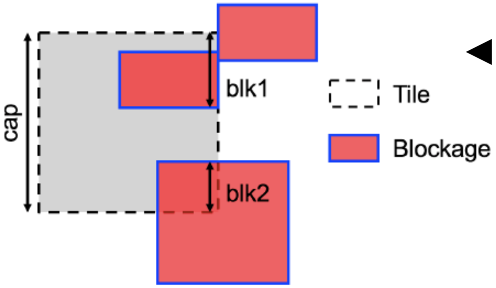
[22] X. He, Y. Wang, Y. Guo and E. F. Y. Young, "Ripple 2.0: Improved Movement of Cells in Routability-Driven Placement", *ACM Trans. on DAES* 22(1) (2016), pp. 10:1-10:26.

RePLAce: Routability-Driven Placement

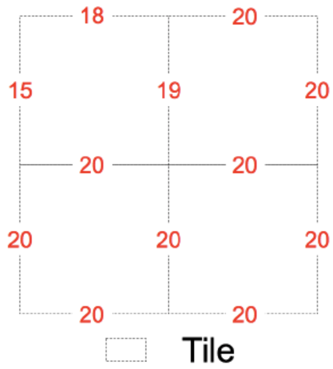
- Metal layer-aware superlinear cell inflation

$$infl\_ratio = \max_{all\ e,ml}\left(\left(\frac{demand_{e,ml}+blk_{e,ml}}{cap_{e,ml}}\right)^{\gamma_{super}}, 2.5\right)$$

  - e = one of the four edges of a given global routing tile
  - ml = a specific metal layer
  - γ_{super} = 2.33 (empirically determined)



◀ Blockage calculation. For the vertical edge on the right, $blk = blk1 + blk2$. Note the union of blocked capacity for the upper two blockages.

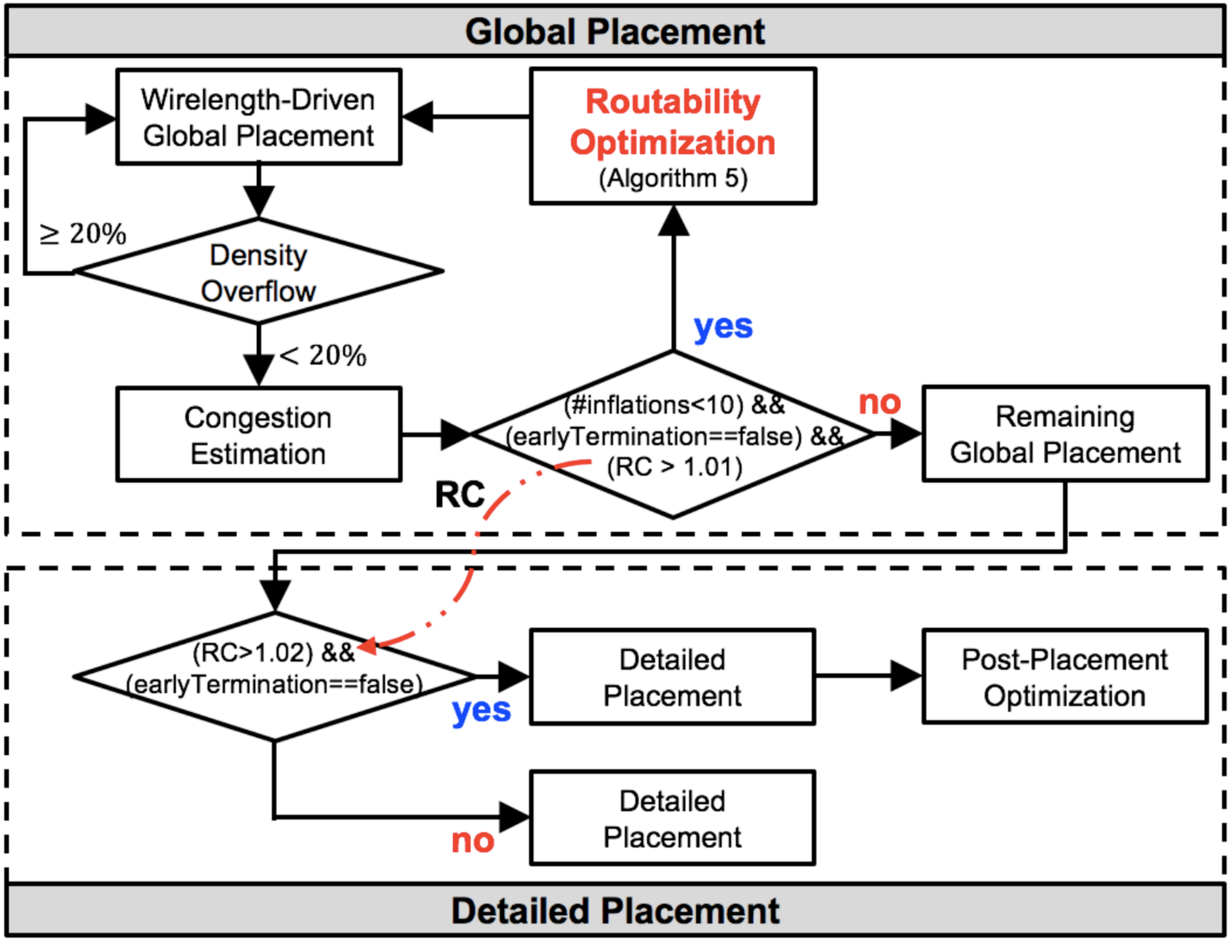


◀ Routing demand calculation: the upper-left tile has a horizontal routing demand of $\max(15,19)$ = 19

- Considers the total available whitespace
  - Starting from 90% die utilization,
  - We limit ‘the maximum cell-inflated area’ ≤ 10%
  - If exceed, then perform ‘inflation ratio adjustment’
    - Divides the inflation ratio for each tile by the inflation ratio of the least-congested tile that has a ratio greater than one

RePLAce: Routability-Driven Placement

- Our routability optimization flow

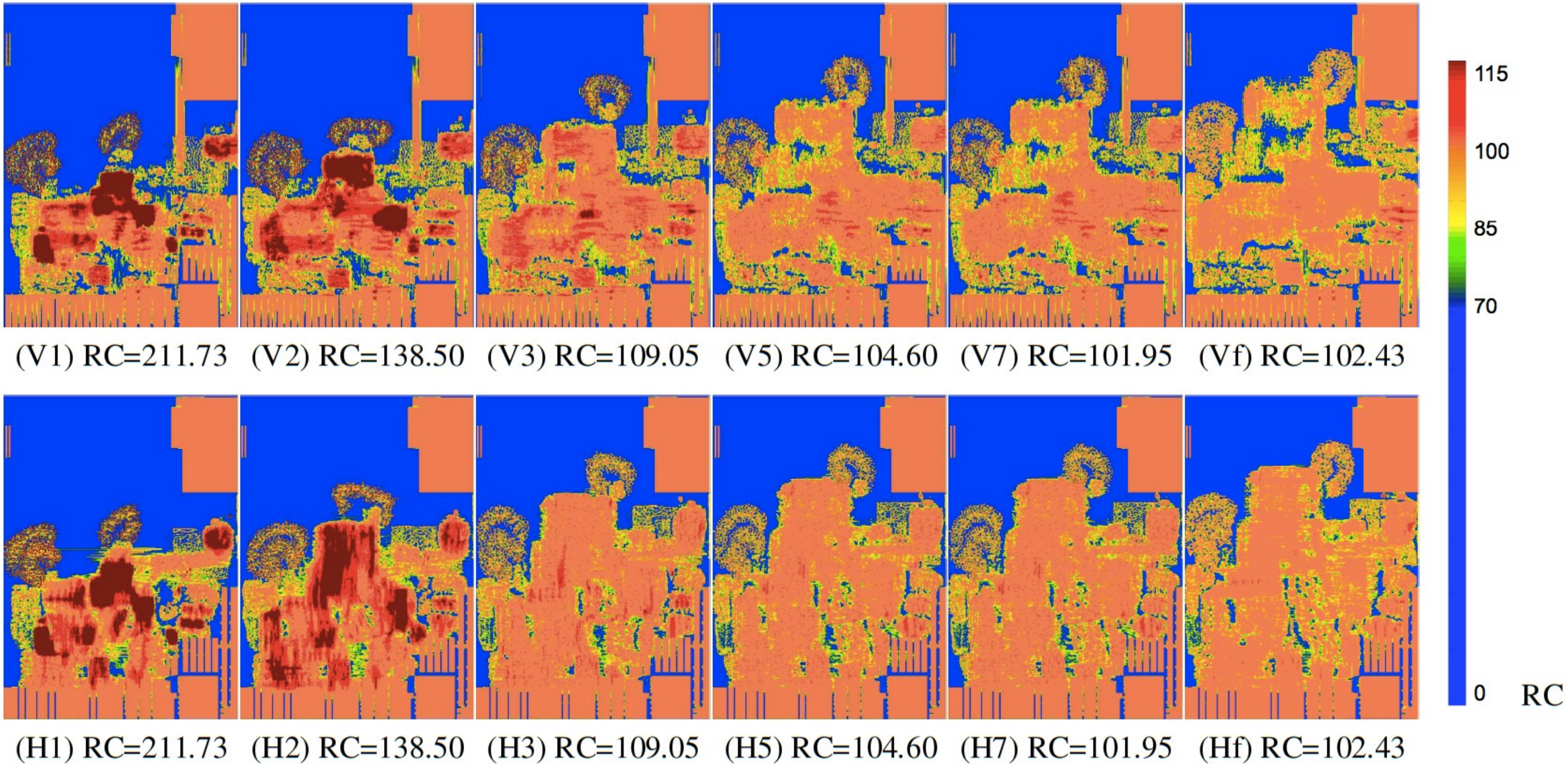

Global Placement
Wirelength-Driven Global Placement
Routability Optimization (Algorithm 5)
≥ 20%
Density Overflow
< 20%
yes
Congestion Estimation
(#inflations<10) && (earlyTermination==false) && (RC > 1.01)
no
Remaining Global Placement
RC
(RC>1.02) && (earlyTermination==false)
yes
Detailed Placement
Post-Placement Optimization
no
Detailed Placement
Detailed Placement

- Our detailed placer: *NTUplace3* [23]

[23] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen and Y.-W. Chang, "NTUplace3: An Analytical Placer for Large-Scale Mixed-Size Designs with Preplaced Blocks and Density Constraints", *IEEE Trans. on CAD* 27(7) (2008), pp. 1228-1240.

REPLACE: ROUTABILITY-DRIVEN PLACEMENT

- Global routing overflow (SUPERBLUE12) during routability-driven global placement procedure



(V1) RC=211.73 (V2) RC=138.50 (V3) RC=109.05 (V5) RC=104.60 (V7) RC=101.95 (Vf) RC=102.43

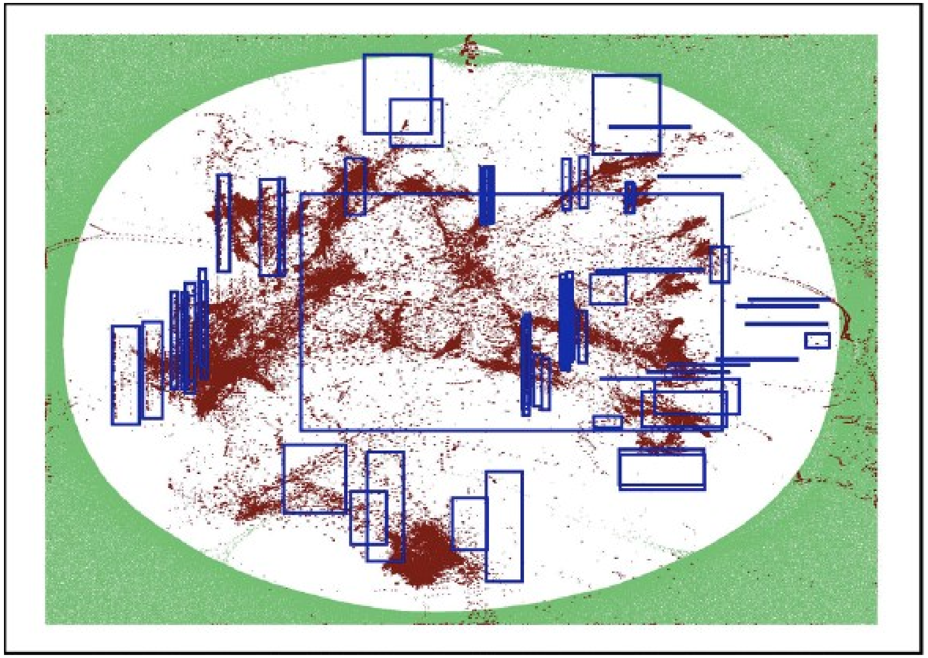
(H1) RC=211.73 (H2) RC=138.50 (H3) RC=109.05 (H5) RC=104.60 (H7) RC=101.95 (Hf) RC=102.43

CONCLUSION

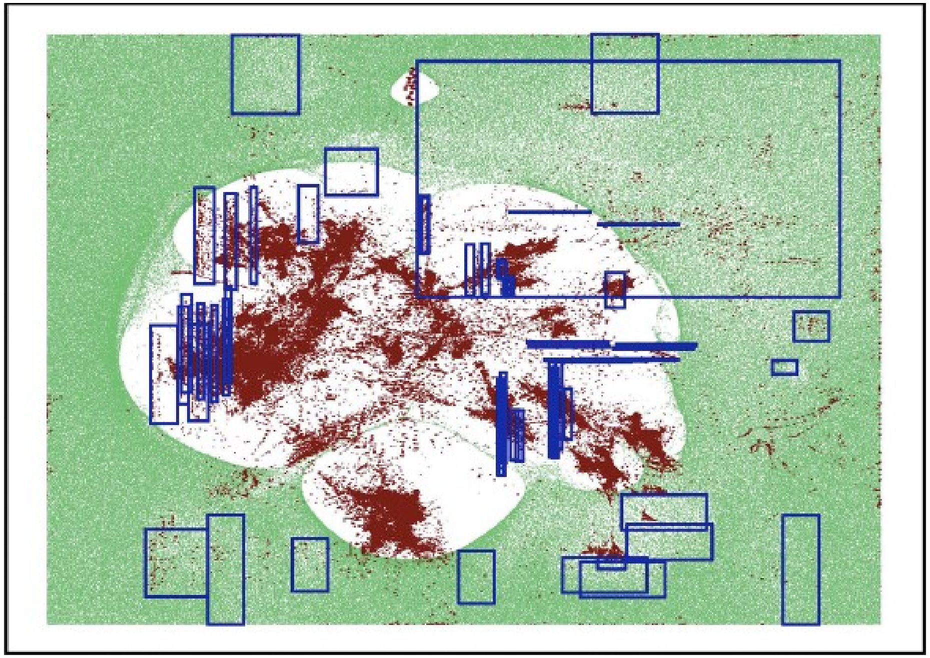
- RePlAce: Advancing solution quality and routability validation in global placement
  - Local density function
    - Enables local smoothing that comprehend local over-demanded bins
  - Improved dynamic step size adaptation
    - To further improve the solution quality by compromising runtime
  - Routability optimization
    - Simple but effective metal layer-aware superlinear cell inflation technique
- Results: Superior solution qualities
  - Standard cell placement
    - An average HPWL reduction of 2.00% over the best known ISPD benchmark results
  - Mixed-size placement
    - An average HPWL reduction of 2.73% over the best known MMS benchmark results
  - Routability-driven placement
    - Achieves on average 8.50% to 9.59% scaled HPWL reduction over previous leading academic placers for the DAC-2012 and ICCAD-2012 benchmark suites
  - vs. Industrial tool
    - Achieves 2.4% reduction of routed wirelength on average with similar number (<100) of DRC violations, and consumes less than 2× runtime

RePLAce: Constraint-Driven Placement

- newblue1 (MMS)
  - Iteration 400 (macro and standard cell placement)



[ePlace]



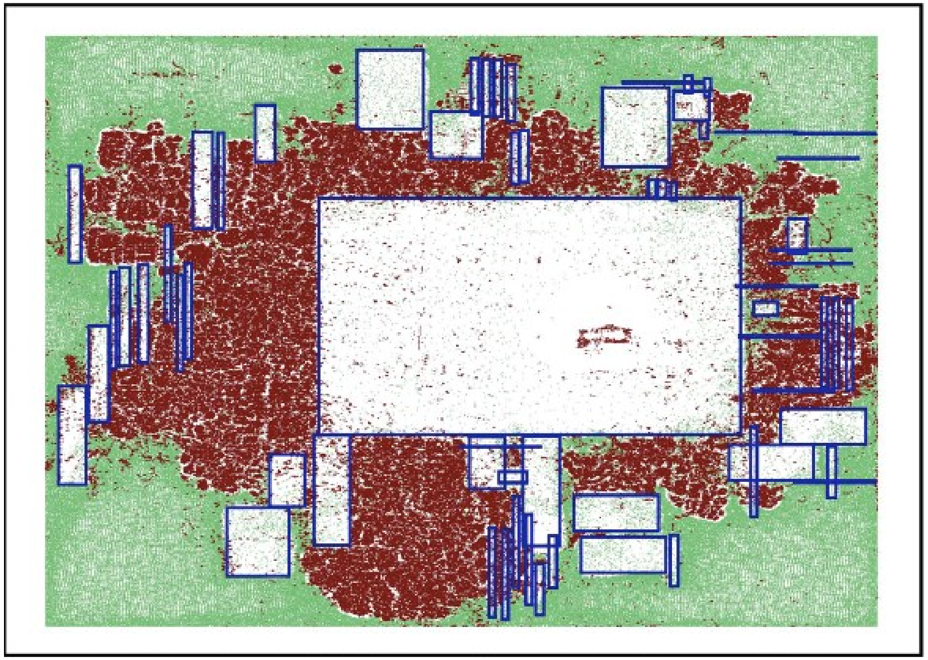
[RePlAce with Constraint-Driven]

- HPWL = 4.51E+7

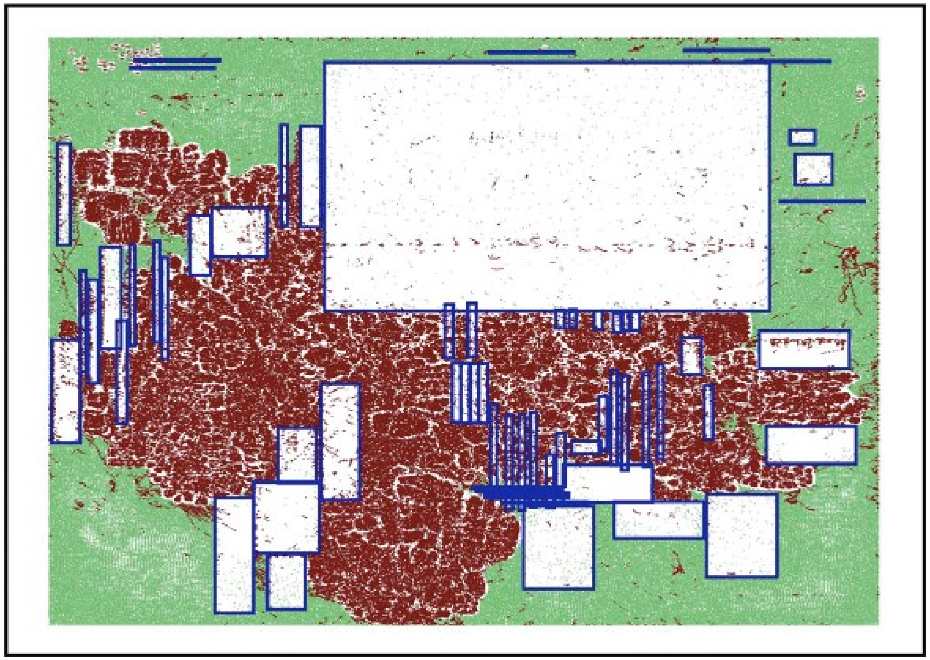
- HPWL = 4.32E+7

RePLAce: Constraint-Driven Placement

- newblue1 (MMS)
  - Iteration 600 (macro and standard cell placement)



[ePlace]



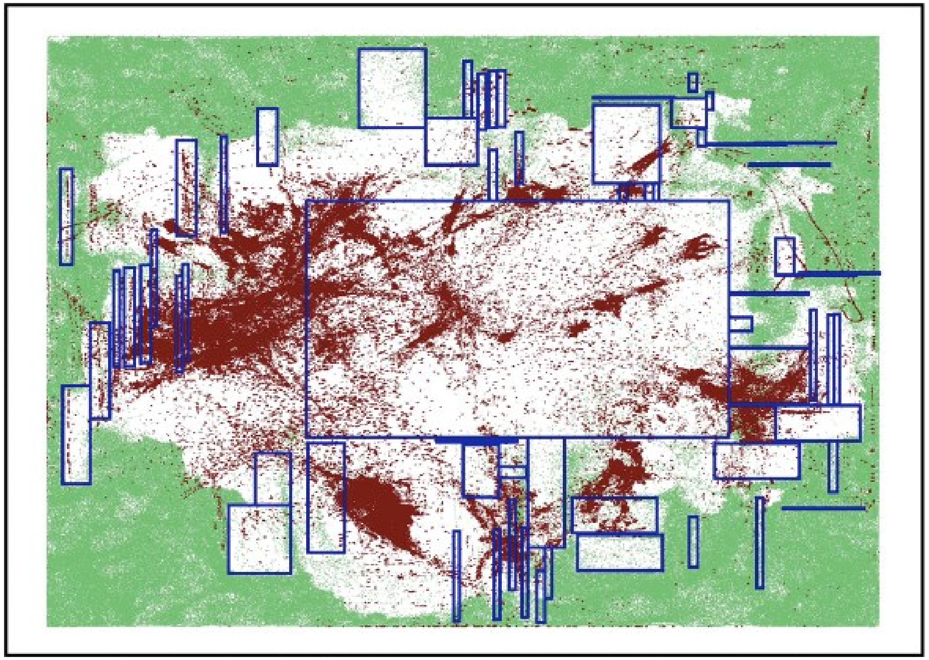
[RePlAce with Constraint-Driven]

- HPWL = 6.55E+7

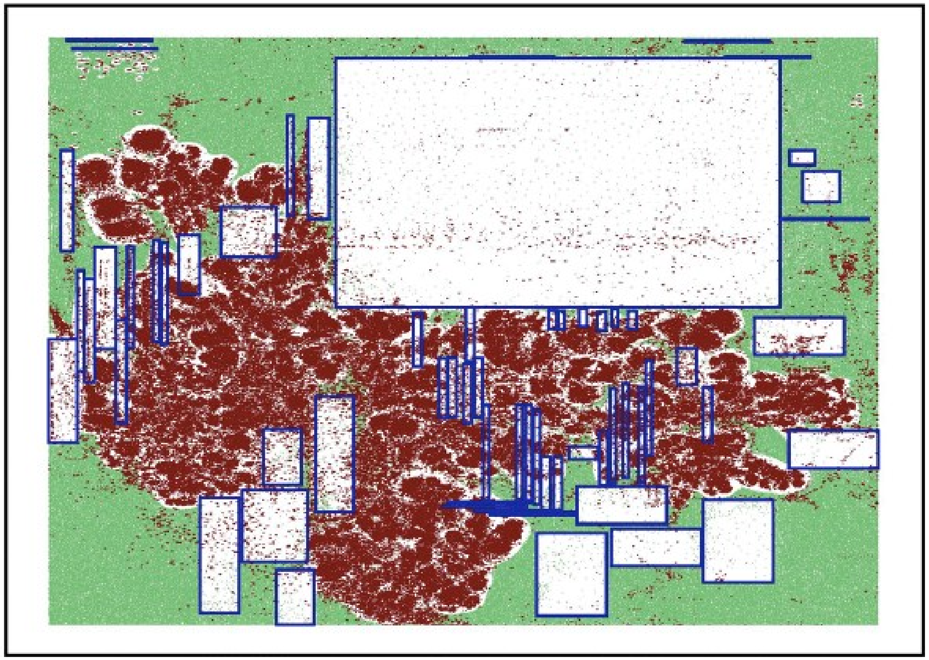
- HPWL = 5.32E+7

RePlAce: Constraint-Driven Placement

- newblue1 (MMS)
  - Iteration 30 (standard cell placement after fixing macro)



[ePlace]



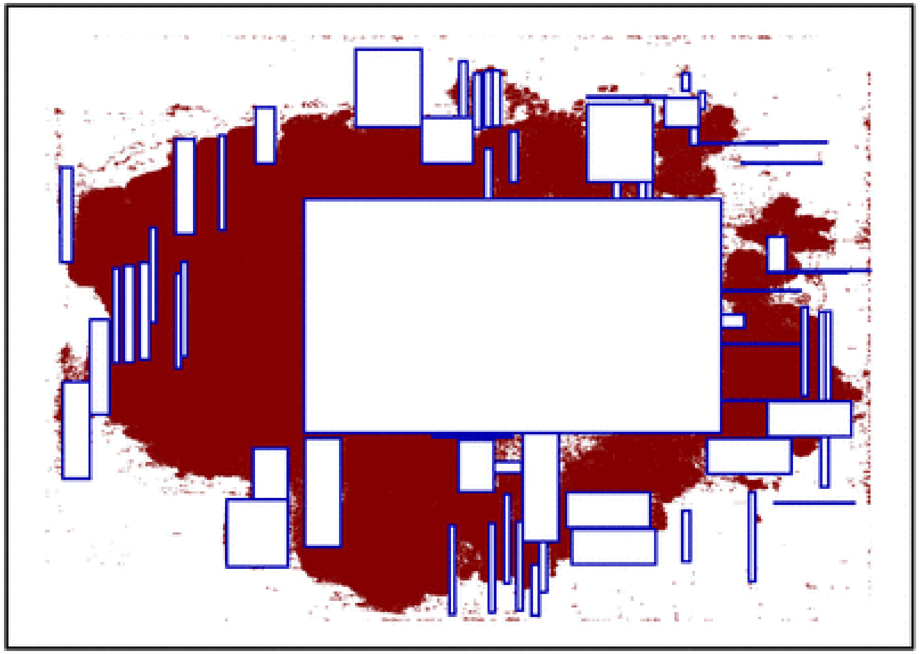
[RePlAce with Constraint-Driven]

- HPWL = 5.18E+7

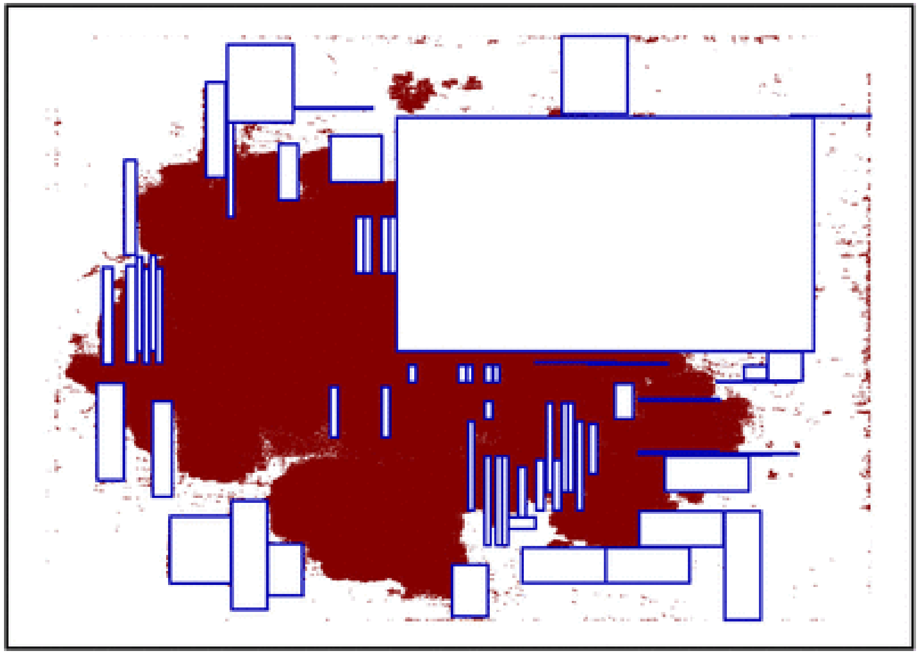
- HPWL = 5.96E+7

ePlace 2.0: Constraint-Driven Placement

- **Animation, ePlace vs. ePlace 2.0, *NEWBLUE1***
  - Cell movement/location per each iteration



[ePlace]



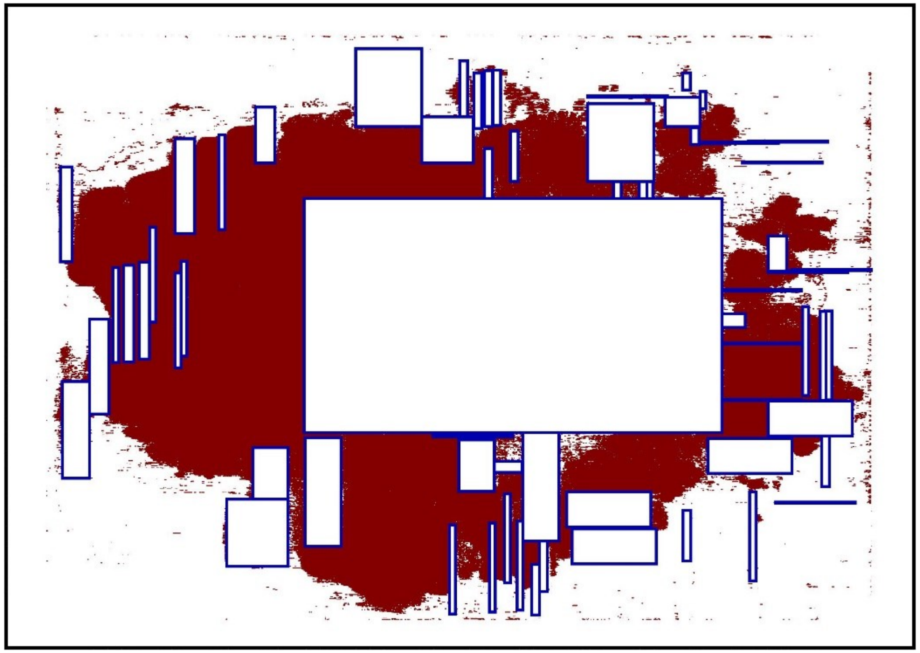
[ePlace 2.0 with Constraint-Driven]

- HPWL = 6.38E+7 (After Detailed Placement)
- #Iter = 1089 (615 + 474)
- runT = 43 min

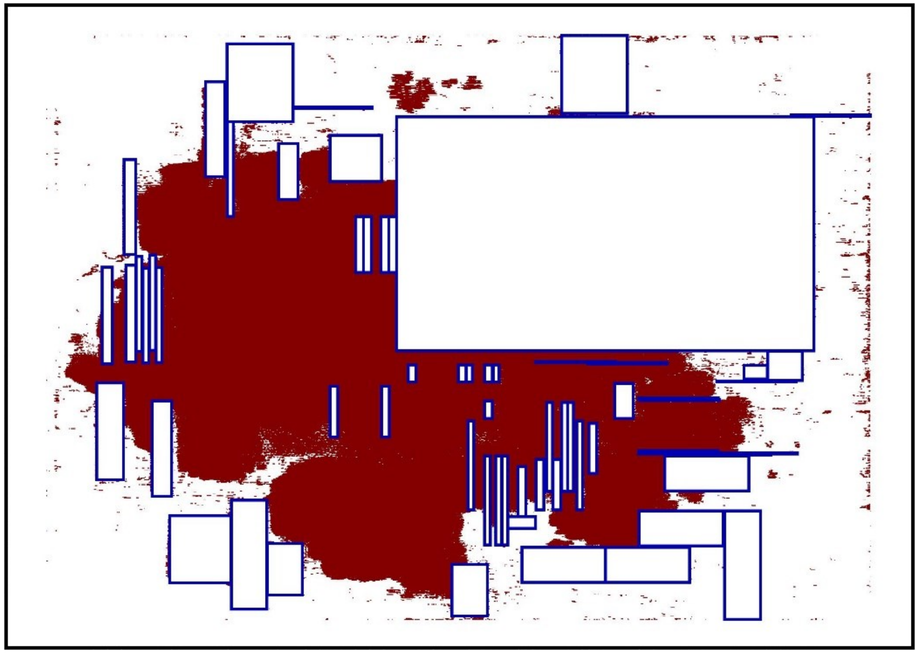
- HPWL = 5.71E+7 (After Detailed Placement)
- #Iter = 1078 (609 + 469)
- runT = 42 min

ePLACE 2.0: CONSTRAINT-DRIVEN PLACEMENT

- **Animation, ePlace vs. ePlace 2.0, *NEWBLUE1***
  - Final placement result after detailed placement



[ePlace]



[ePlace 2.0 with Constraint-Driven]

- HPWL = 6.38E+7 (After Detailed Placement)
- #Iter = 1089 (615 + 474)
- runT = 43 min

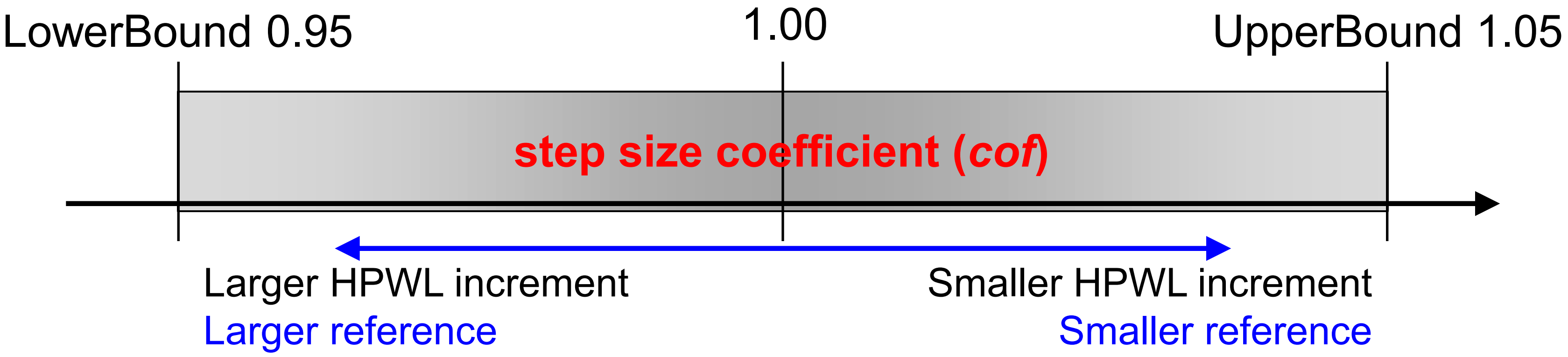
- HPWL = 5.71E+7 (After Detailed Placement)
- #Iter = 1078 (609 + 469)
- runT = 42 min

ePlace 2.0: Dynamic Step Size

- Dynamic Step Size in the Previous ePlace
  - Accounts for *diff*(HPWL).
  - For each iteration, get 'reference'.

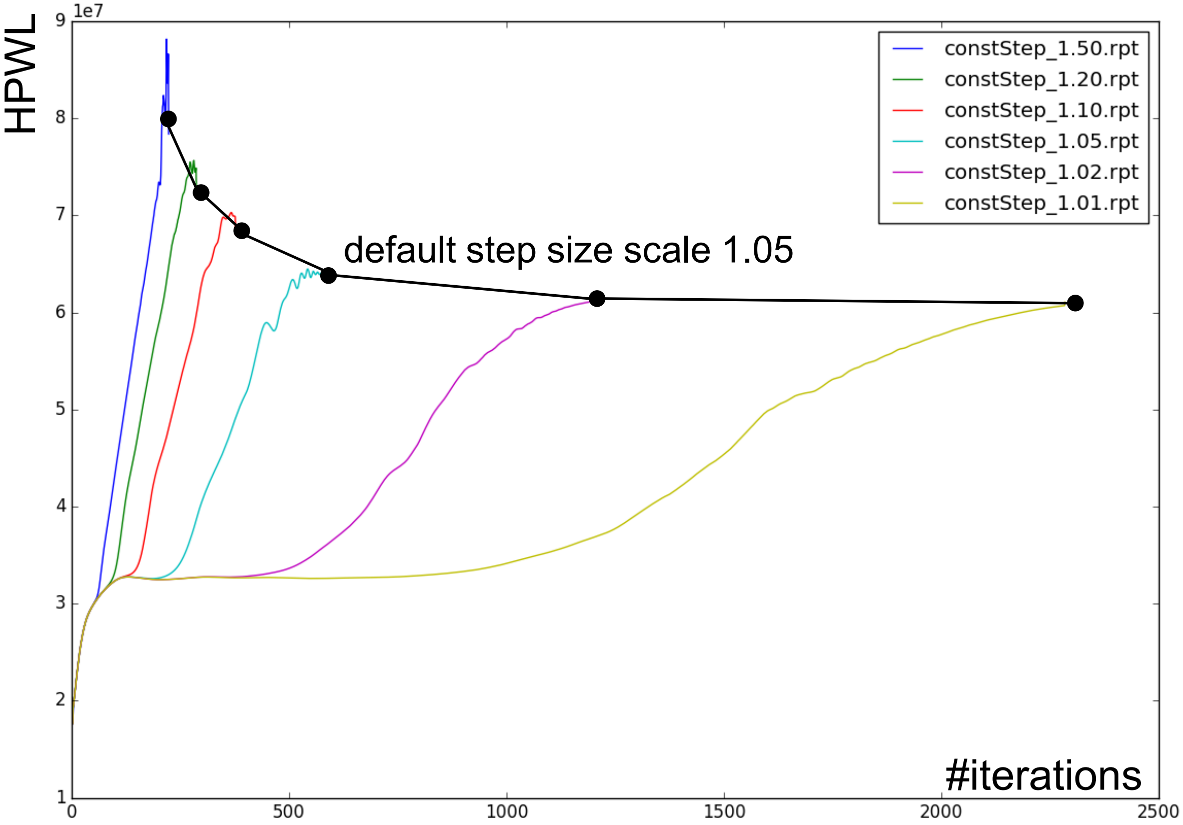
$$(HPWL_{iter} - HPWL_{iter-1})/reference\_\delta\_wirelength$$

  - If reference < 0, then step size coefficient is the upper bound value.
  - If reference > 0, then



ePlace 2.0: Improved Dynamic Step Size

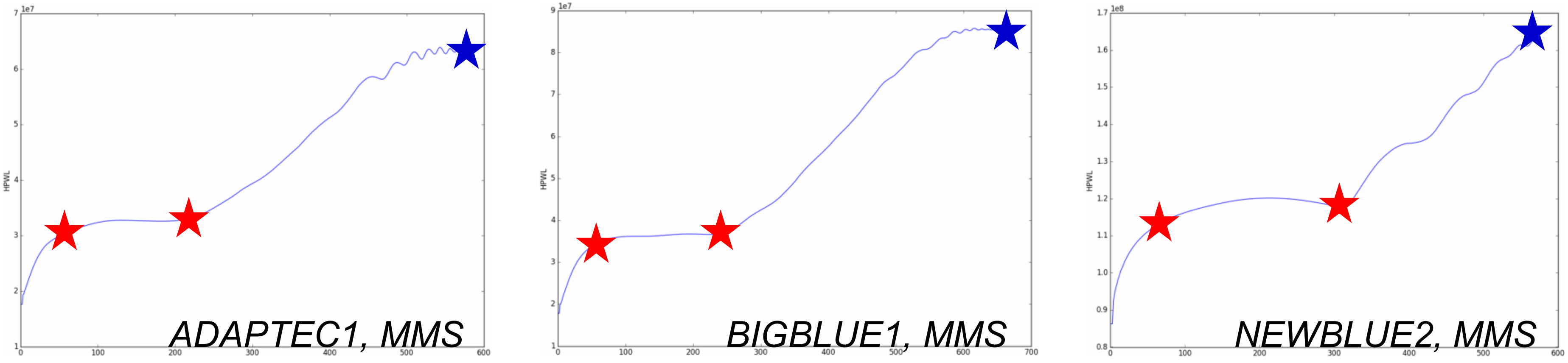
- Motivation
  - Trade-off: Solution quality vs. runtime
  - Obtain better solution quality by investing runtime efficiently.
- **Improved Dynamic Step Size Adaptation:**
  - Dynamic Step Size + **Control upper-bound** + Apply **reference WL per case**



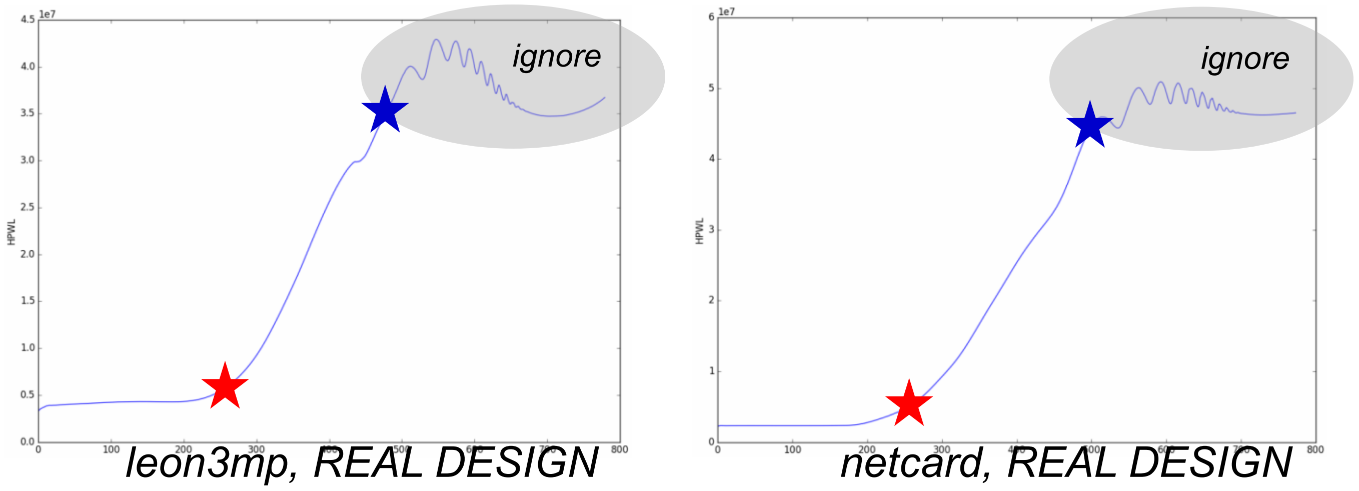
HPWL vs. #iterations across constant step sizes. Testcase: *ADAPTEC1*

ePlace 2.0: Improved Dynamic Step Size

- Background: HPWL curves of testcases
  - Transitions points are depicted by red stars on HPWL curves.



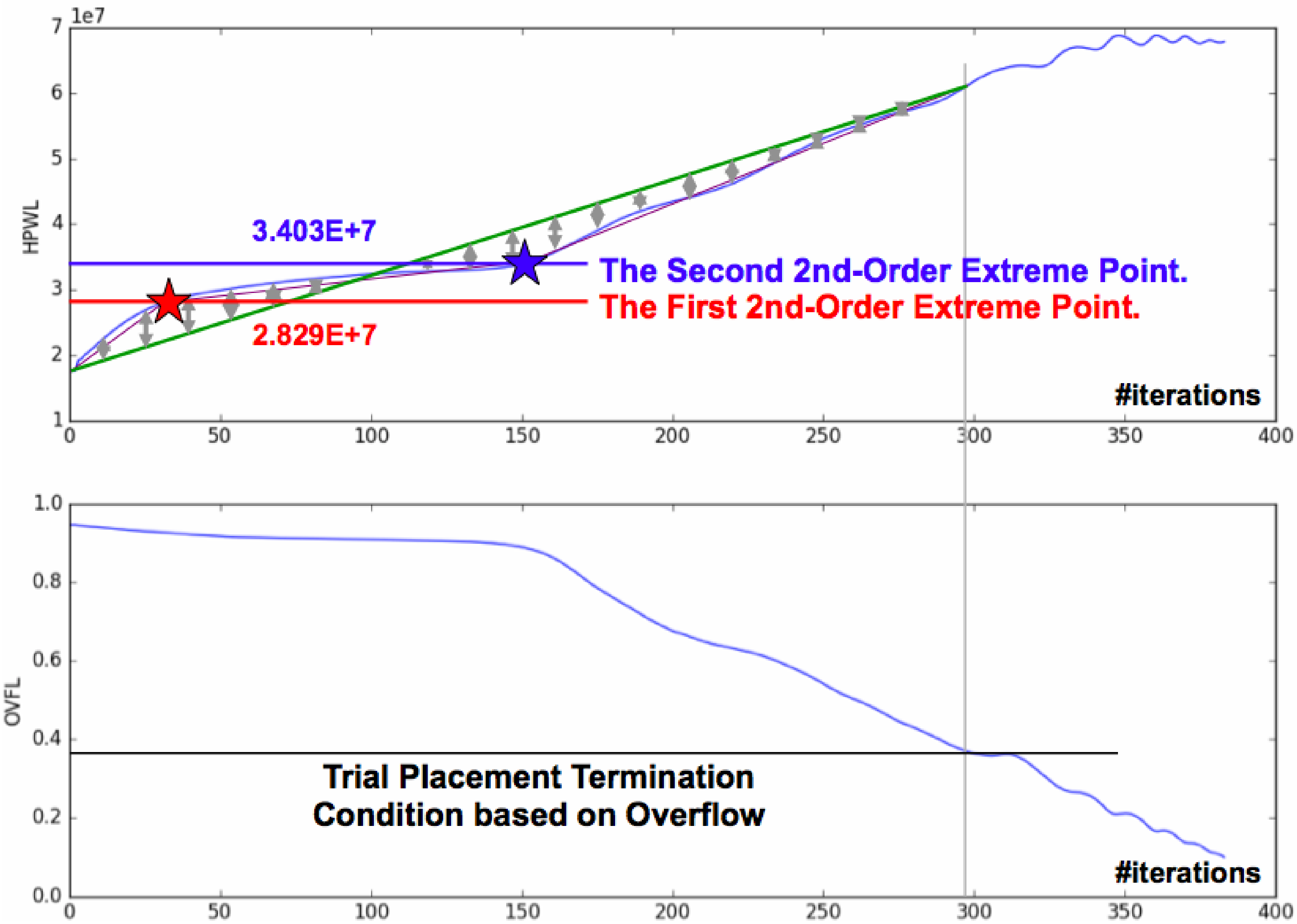
Two transition points on HPWL curve if macros exist



One transition points on HPWL curve if macros do not exist

ePlace 2.0: Improved Dynamic Step Size

- Our Approach (Example: *ADAPTEC1*)
  - Trial procedure to capture transition points on HPWL curve.



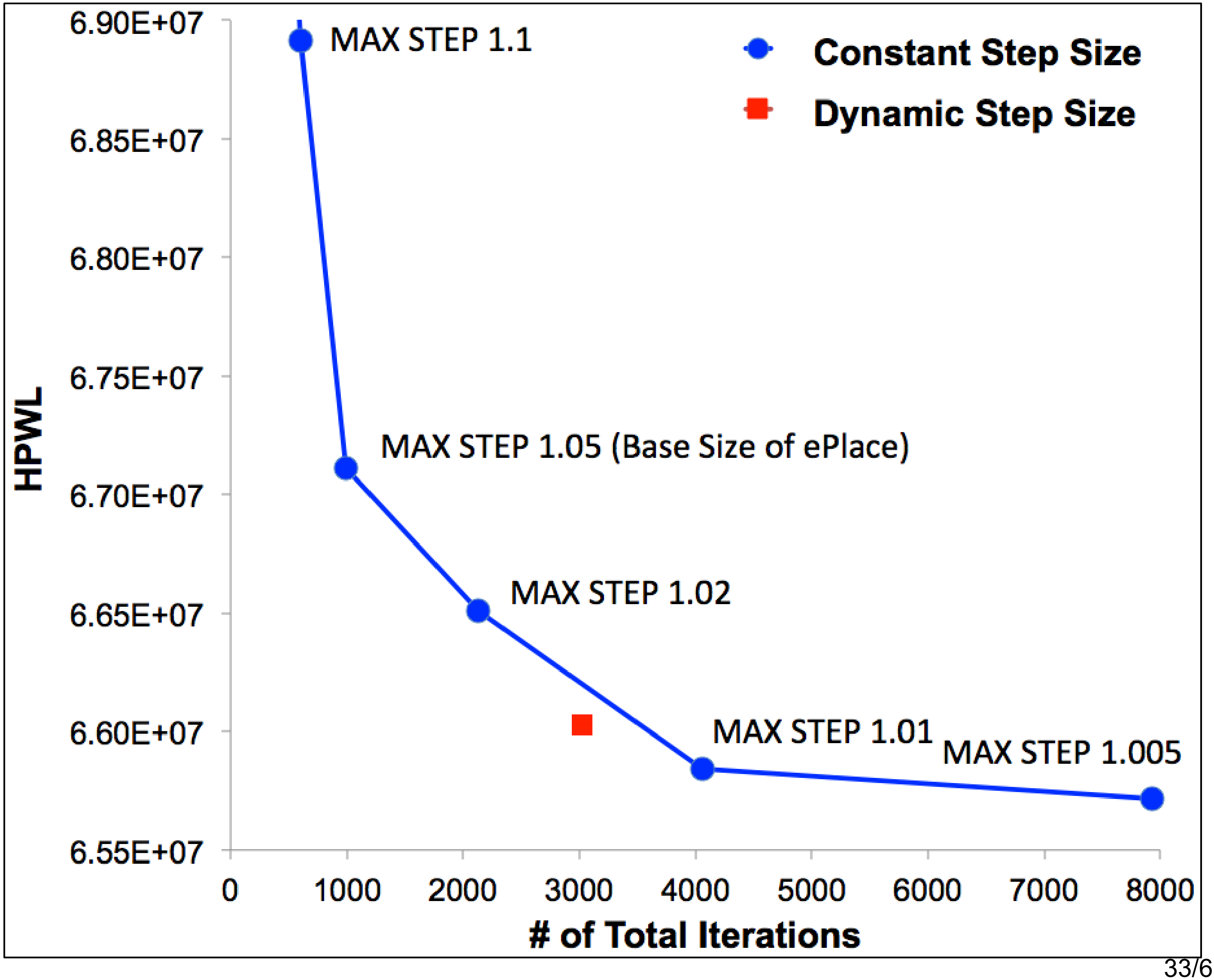
ePlace 2.0: Improved Dynamic Step Size

- Extreme Points (i.e., Transition Points)
  - Trial Procedure (Constant Scale 1.05), *ADAPTEC1*
    - EP: Extreme Point

Class of Extreme Points	HPWL	Upper Bound of Step Size
The initial point	1.762E+7	1.001
The first 1st-order EP	2.461E+7	1.01
The first 2nd-order EP	2.829E+7	1.005
The second 1st-order EP	3.287E+7	1.05
The second 2nd-order EP	3.402E+7	1.01
The third 1st-order EP	4.680E+7	1.05
The last point	6.007E+7	1.01

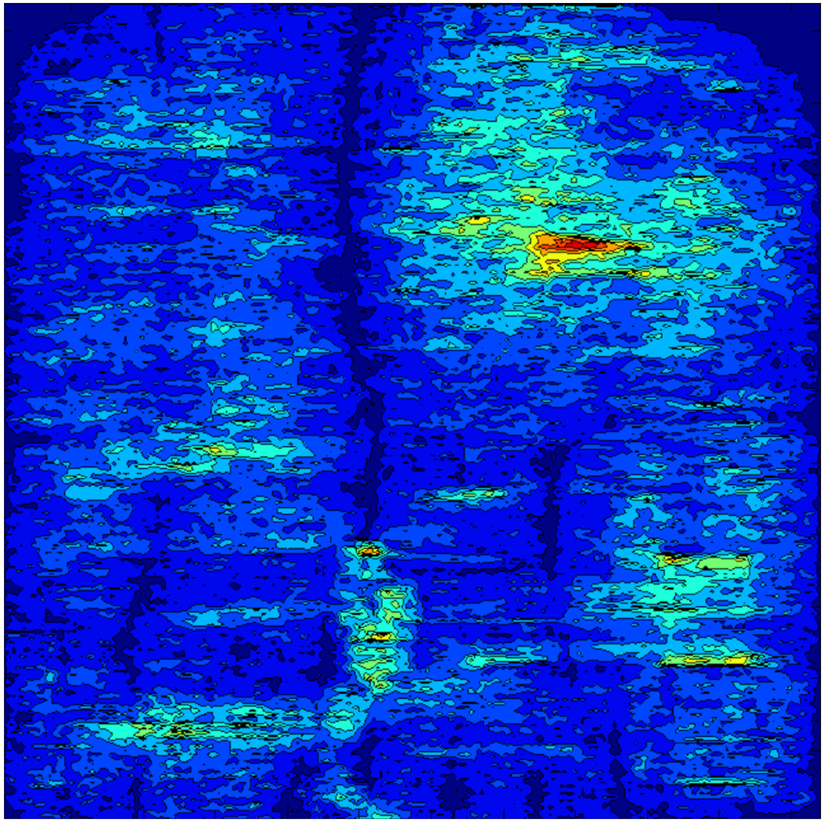
ePlace 2.0: Improved Dynamic Step Size

- Solution Quality in terms of the final HPWL
  - *ADAPTEC1*
  - *CPUtime*
    - *30min* (1.05)
    - 60min (1.02)
    - ***90min* (DS)**
    - 120min (1.01)

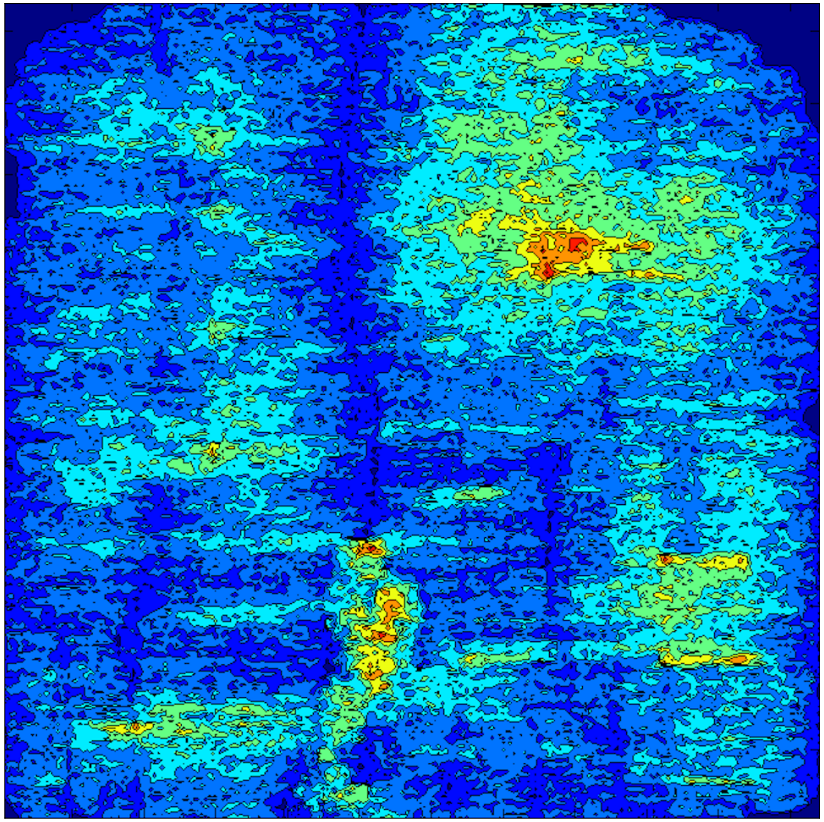


EPLACE 2.0: ROUTABILITY-DRIVEN PLACEMENT

- Comparison of congestion estimation by our model vs. commercial tool
  - Figures generated by MATLAB
- *leon3mp*: realistic test case.
  - #STD cells: 437465, #I/O ports: 332, #nets: 437718, #pins: 1388822



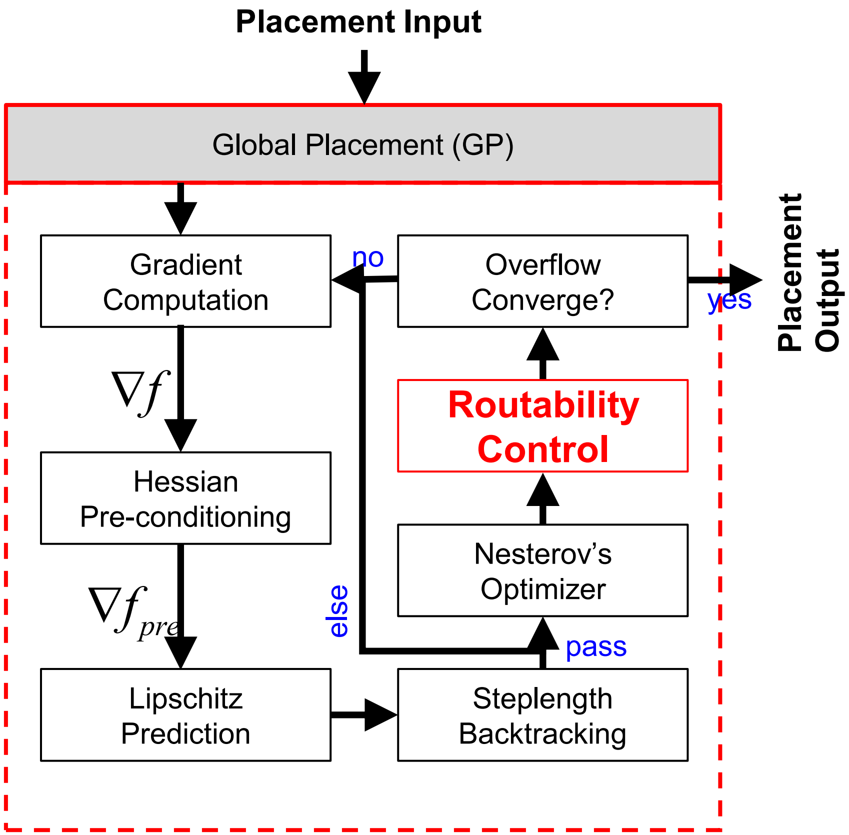
Horizontal Congestion by ePlace 2.0



Horizontal Congestion by commercial tool

ePlace 2.0: Routability-Driven Placement

- Cell Inflation
  - Each cell inflation procedure: (6-10) times of inflation phases
  - (4-6) times of inflation procedures
  - Horizontal(vertical) congestion → inflate in V(H) direction
  - $routeOVFL_h(g_i) = \frac{TrackDemand_v(g_i)}{TrackSupply_v(g_i)}$
- Implementation
  - Alternate H and V inflation
  - Continuous inflation phase
    - 6-10 times before restoring the cell size
  - Inflation Bound
    - total inflation area ≤ white space

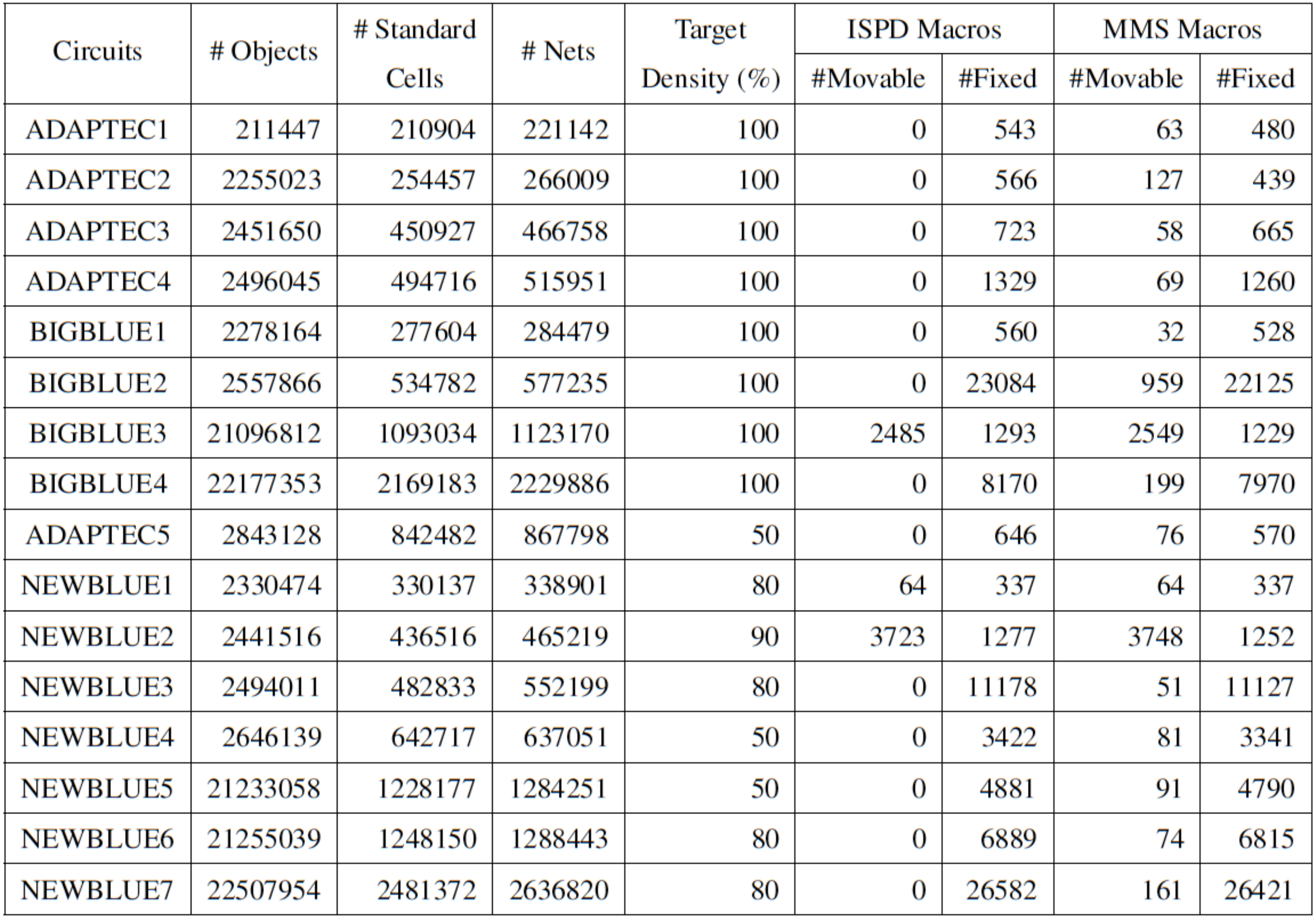


[6] X. He, T. Huang, L. Xiao, H. Tian and E. F. Y. Young, "Ripple: A Robust and Effective Routability-Driven Placer", IEEE Trans. CAD, 32(10) (2013), pp. 1546-1556.

[7] J. Lu, P. Chen, C.-C. Chang, L. Sha, D. J.-H. Huang, C.-C. Teng and C.-K. Cheng, "ePlace: Electrostatics Based Placement Using Nesterov's Method", Proc. DAC, 2014, pp. 1-6.

REPLACE: ISPD-2005, ISPD-2006, MMS

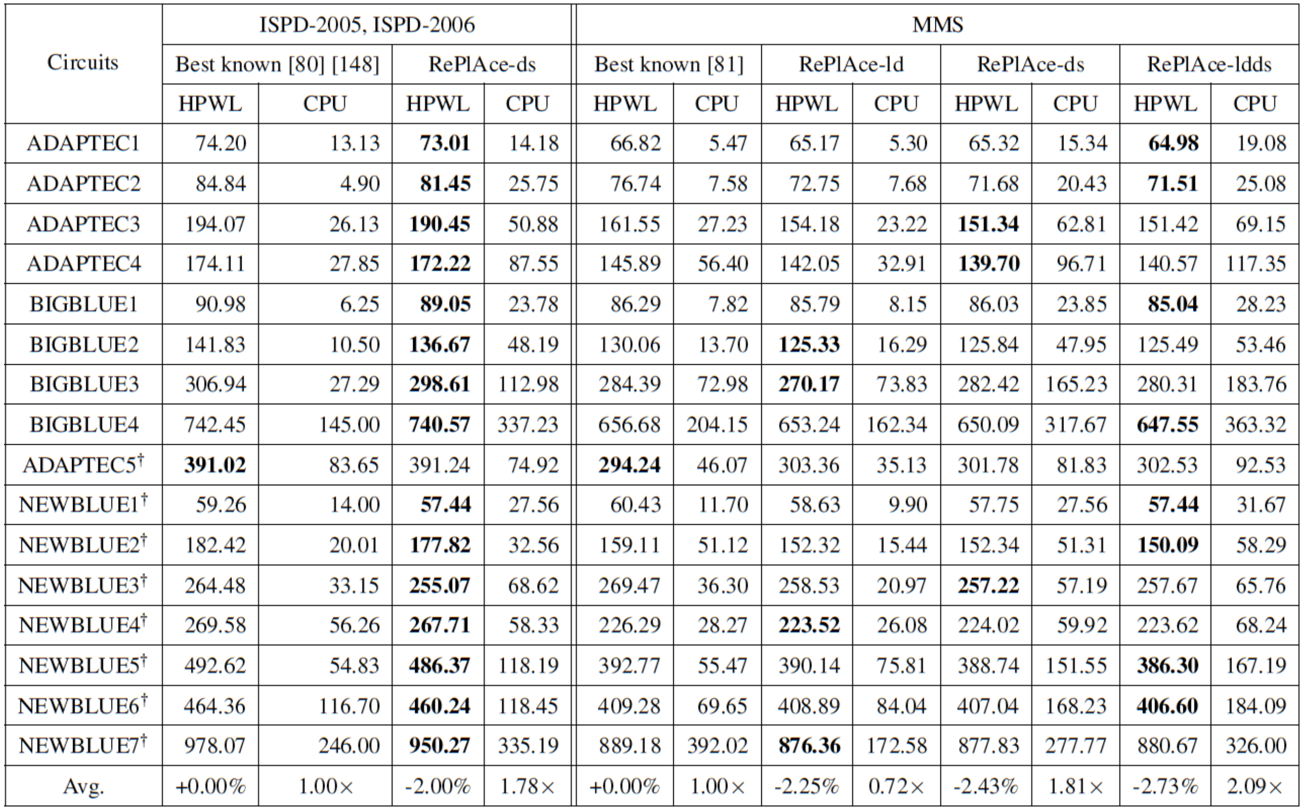
- Benchmark Statistics



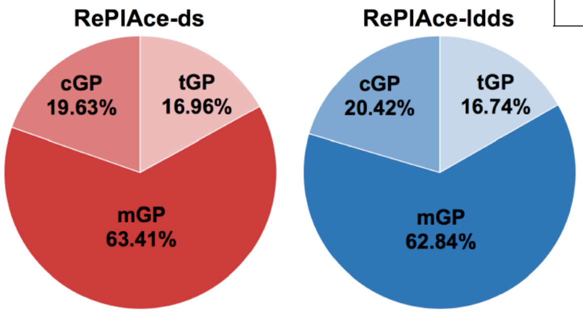
Circuits	# Objects	# Standard Cells	# Nets	Target Density (%)	ISPD Macros		MMS Macros	
					#Movable	#Fixed	#Movable	#Fixed
ADAPTEC1	211447	210904	221142	100	0	543	63	480
ADAPTEC2	2255023	254457	266009	100	0	566	127	439
ADAPTEC3	2451650	450927	466758	100	0	723	58	665
ADAPTEC4	2496045	494716	515951	100	0	1329	69	1260
BIGBLUE1	2278164	277604	284479	100	0	560	32	528
BIGBLUE2	2557866	534782	577235	100	0	23084	959	22125
BIGBLUE3	21096812	1093034	1123170	100	2485	1293	2549	1229
BIGBLUE4	22177353	2169183	2229886	100	0	8170	199	7970
ADAPTEC5	2843128	842482	867798	50	0	646	76	570
NEWBLUE1	2330474	330137	338901	80	64	337	64	337
NEWBLUE2	2441516	436516	465219	90	3723	1277	3748	1252
NEWBLUE3	2494011	482833	552199	80	0	11178	51	11127
NEWBLUE4	2646139	642717	637051	50	0	3422	81	3341
NEWBLUE5	21233058	1228177	1284251	50	0	4881	91	4790
NEWBLUE6	21255039	1248150	1288443	80	0	6889	74	6815
NEWBLUE7	22507954	2481372	2636820	80	0	26582	161	26421

RePlAce: ISPD-2005 [15], ISPD-2006 [16], MMS [14]

- Standard cell placement
  - An average HPWL reduction of 2.00% over the best known ISPD benchmark results
- Mixed-size placement
  - An average HPWL reduction of 2.73% over the best known MMS benchmark results



Circuits	ISPD-2005, ISPD-2006				MMS							
	Best known [80] [148]		RePlAce-ds		Best known [81]		RePlAce-ld		RePlAce-ds		RePlAce-ldds	
	HPWL	CPU	HPWL	CPU	HPWL	CPU	HPWL	CPU	HPWL	CPU	HPWL	CPU
ADAPTEC1	74.20	13.13	**73.01**	14.18	66.82	5.47	65.17	5.30	65.32	15.34	**64.98**	19.08
ADAPTEC2	84.84	4.90	**81.45**	25.75	76.74	7.58	72.75	7.68	71.68	20.43	**71.51**	25.08
ADAPTEC3	194.07	26.13	**190.45**	50.88	161.55	27.23	154.18	23.22	**151.34**	62.81	151.42	69.15
ADAPTEC4	174.11	27.85	**172.22**	87.55	145.89	56.40	142.05	32.91	**139.70**	96.71	140.57	117.35
BIGBLUE1	90.98	6.25	**89.05**	23.78	86.29	7.82	85.79	8.15	86.03	23.85	**85.04**	28.23
BIGBLUE2	141.83	10.50	**136.67**	48.19	130.06	13.70	**125.33**	16.29	125.84	47.95	125.49	53.46
BIGBLUE3	306.94	27.29	**298.61**	112.98	284.39	72.98	**270.17**	73.83	282.42	165.23	280.31	183.76
BIGBLUE4	742.45	145.00	**740.57**	337.23	656.68	204.15	653.24	162.34	650.09	317.67	**647.55**	363.32
ADAPTEC5†	**391.02**	83.65	391.24	74.92	**294.24**	46.07	303.36	35.13	301.78	81.83	302.53	92.53
NEWBLUE1†	59.26	14.00	**57.44**	27.56	60.43	11.70	58.63	9.90	57.75	27.56	**57.44**	31.67
NEWBLUE2†	182.42	20.01	**177.82**	32.56	159.11	51.12	152.32	15.44	152.34	51.31	**150.09**	58.29
NEWBLUE3†	264.48	33.15	**255.07**	68.62	269.47	36.30	258.53	20.97	**257.22**	57.19	257.67	65.76
NEWBLUE4†	269.58	56.26	**267.71**	58.33	226.29	28.27	**223.52**	26.08	224.02	59.92	223.62	68.24
NEWBLUE5†	492.62	54.83	**486.37**	118.19	392.77	55.47	390.14	75.81	388.74	151.55	**386.30**	167.19
NEWBLUE6†	464.36	116.70	**460.24**	118.45	409.28	69.65	408.89	84.04	407.04	168.23	**406.60**	184.09
NEWBLUE7†	978.07	246.00	**950.27**	335.19	889.18	392.02	**876.36**	172.58	877.83	277.77	880.67	326.00
Avg.	+0.00%	1.00×	-2.00%	1.78×	+0.00%	1.00×	-2.25%	0.72×	-2.43%	1.81×	-2.73%	2.09×

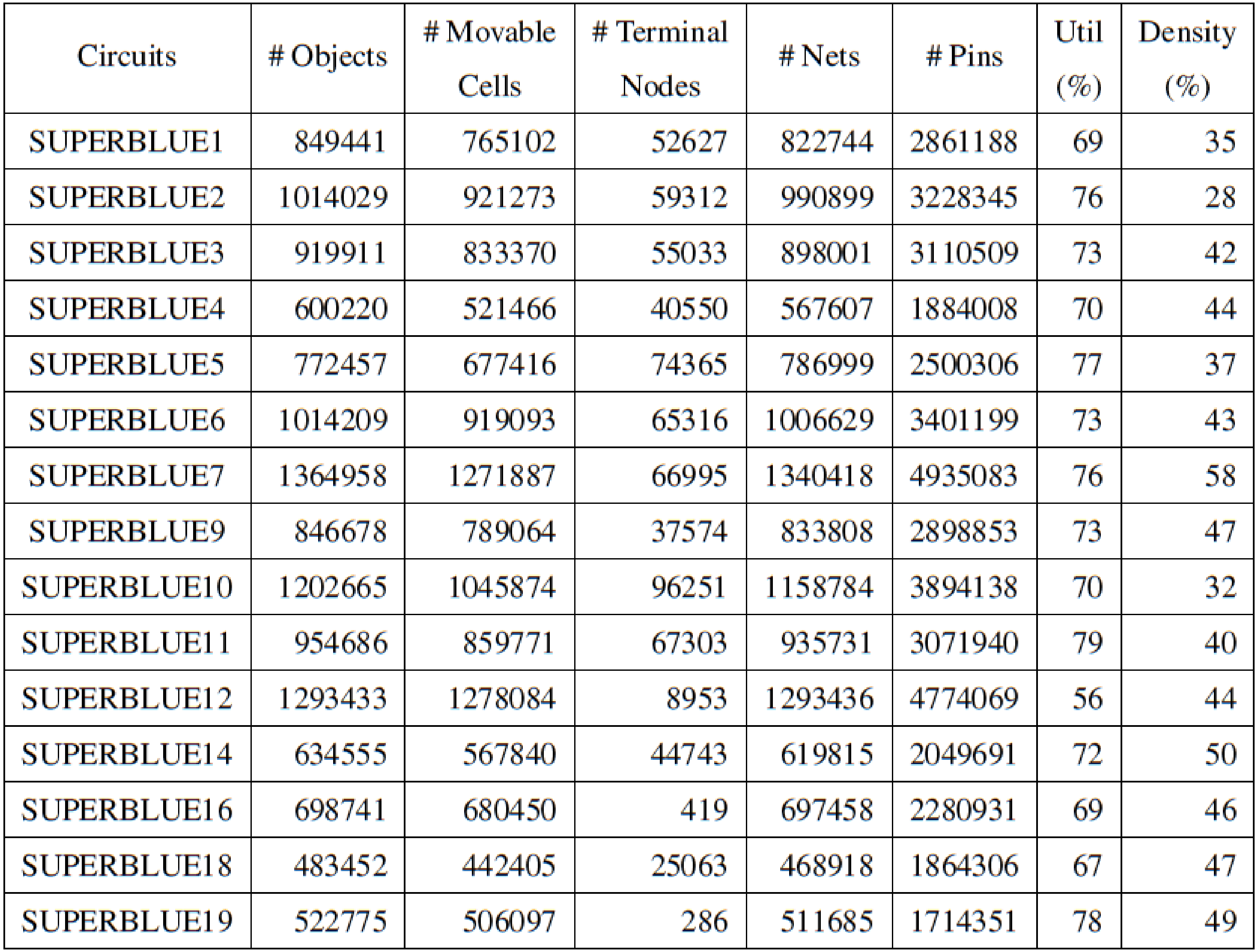


◄ Runtime breakdown (#iterations) aggregated over all reported testcases for RePlAce-ds (left) and RePlAce-ldds (right). tGP, mGP, and cGP respectively denote trial placement, macro and standard cell placement, and standard cell-only placement.

[14] J. Z. Yan, N. Viswanathan and C. Chu, "Handling Complexities in Modern Large-Scale Mixed-Size Placement", *Proc. DAC*, 2009, pp. 436-441.
[15] G.-J. Nam, C. J. Alpert, P. Villarrubia, B. Winter and M. Yildiz, "The ISPD 2005 Placement Contest and Benchmark Suite", *Proc. ISPD*, 2005, pp. 216-220.
[16] G.-J. Nam, "ISPD 2006 Placement Contest: Benchmark Suite and Results", *Proc. ISPD*, 2006, pp. 167.

RePlAce: DAC-2012 and ICCAD-2012

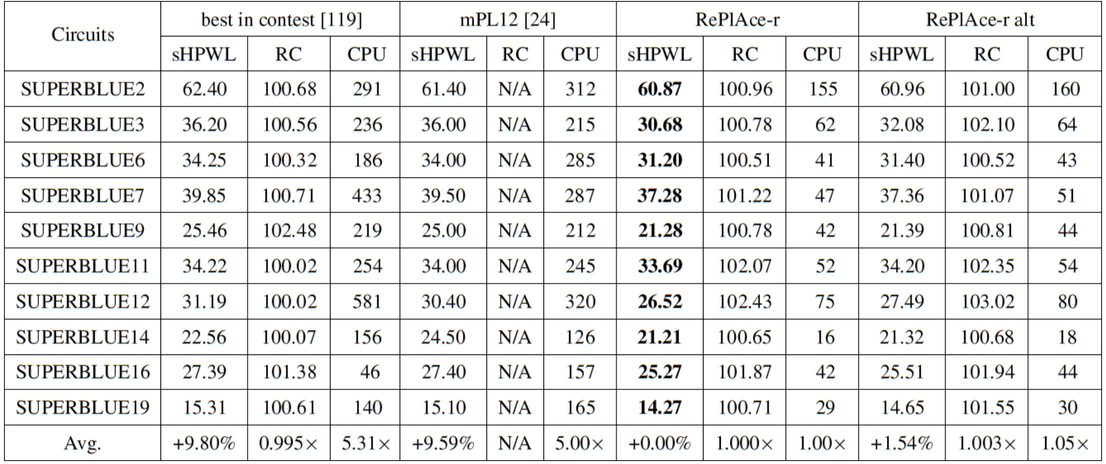
- Benchmark Statistics



Circuits	# Objects	# Movable Cells	# Terminal Nodes	# Nets	# Pins	Util (%)	Density (%)
SUPERBLUE1	849441	765102	52627	822744	2861188	69	35
SUPERBLUE2	1014029	921273	59312	990899	3228345	76	28
SUPERBLUE3	919911	833370	55033	898001	3110509	73	42
SUPERBLUE4	600220	521466	40550	567607	1884008	70	44
SUPERBLUE5	772457	677416	74365	786999	2500306	77	37
SUPERBLUE6	1014209	919093	65316	1006629	3401199	73	43
SUPERBLUE7	1364958	1271887	66995	1340418	4935083	76	58
SUPERBLUE9	846678	789064	37574	833808	2898853	73	47
SUPERBLUE10	1202665	1045874	96251	1158784	3894138	70	32
SUPERBLUE11	954686	859771	67303	935731	3071940	79	40
SUPERBLUE12	1293433	1278084	8953	1293436	4774069	56	44
SUPERBLUE14	634555	567840	44743	619815	2049691	72	50
SUPERBLUE16	698741	680450	419	697458	2280931	69	46
SUPERBLUE18	483452	442405	25063	468918	1864306	67	47
SUPERBLUE19	522775	506097	286	511685	1714351	78	49

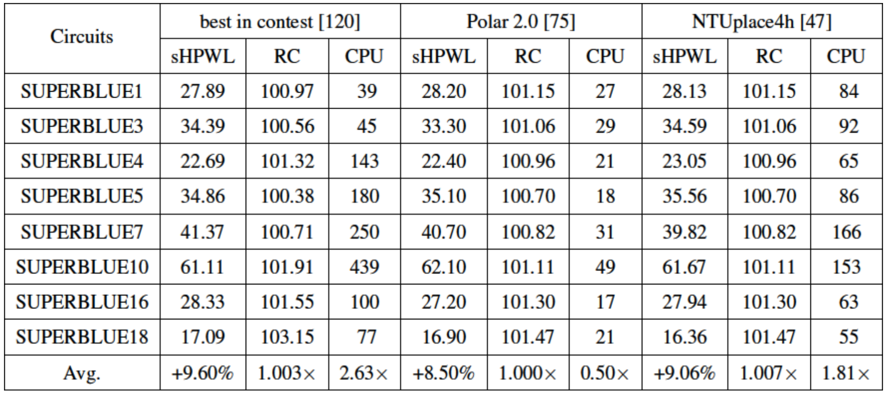
REPLACE: DAC-2012 AND ICCAD-2012

- Results
  - Achieves on average 8.50% to 9.59% scaled HPWL reduction over previous leading academic placers for the DAC-2012 and ICCAD-2012 benchmark suites
  - *RePlAce-r alt* uses *NTUplace4h* [24] as its detailed placer

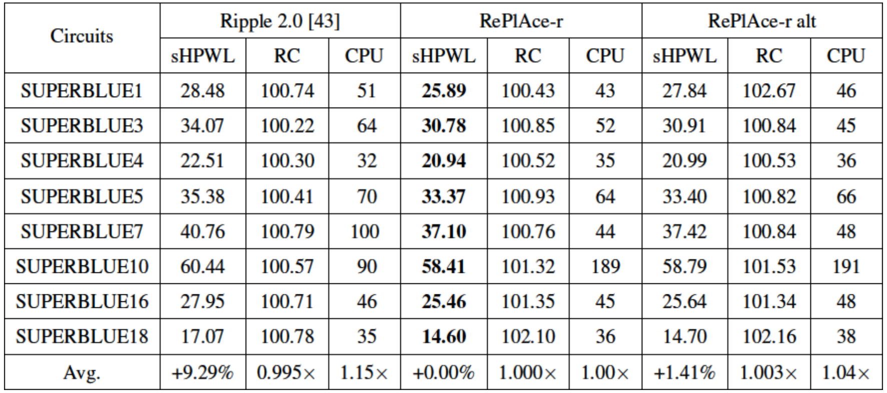


Circuits	best in contest [119]			mPL12 [24]			RePlAce-r			RePlAce-r alt		
	sHPWL	RC	CPU	sHPWL	RC	CPU	sHPWL	RC	CPU	sHPWL	RC	CPU
SUPERBLUE2	62.40	100.68	291	61.40	N/A	312	**60.87**	100.96	155	60.96	101.00	160
SUPERBLUE3	36.20	100.56	236	36.00	N/A	215	**30.68**	100.78	62	32.08	102.10	64
SUPERBLUE6	34.25	100.32	186	34.00	N/A	285	**31.20**	100.51	41	31.40	100.52	43
SUPERBLUE7	39.85	100.71	433	39.50	N/A	287	**37.28**	101.22	47	37.36	101.07	51
SUPERBLUE9	25.46	102.48	219	25.00	N/A	212	**21.28**	100.78	42	21.39	100.81	44
SUPERBLUE11	34.22	100.02	254	34.00	N/A	245	**33.69**	102.07	52	34.20	102.35	54
SUPERBLUE12	31.19	100.02	581	30.40	N/A	320	**26.52**	102.43	75	27.49	103.02	80
SUPERBLUE14	22.56	100.07	156	24.50	N/A	126	**21.21**	100.65	16	21.32	100.68	18
SUPERBLUE16	27.39	101.38	46	27.40	N/A	157	**25.27**	101.87	42	25.51	101.94	44
SUPERBLUE19	15.31	100.61	140	15.10	N/A	165	**14.27**	100.71	29	14.65	101.55	30
Avg.	+9.80%	0.995×	5.31×	+9.59%	N/A	5.00×	+0.00%	1.000×	1.00×	+1.54%	1.003×	1.05×

▲ DAC-2012



Circuits	best in contest [120]			Polar 2.0 [75]			NTUplace4h [47]		
	sHPWL	RC	CPU	sHPWL	RC	CPU	sHPWL	RC	CPU
SUPERBLUE1	27.89	100.97	39	28.20	101.15	27	28.13	101.15	84
SUPERBLUE3	34.39	100.56	45	33.30	101.06	29	34.59	101.06	92
SUPERBLUE4	22.69	101.32	143	22.40	100.96	21	23.05	100.96	65
SUPERBLUE5	34.86	100.38	180	35.10	100.70	18	35.56	100.70	86
SUPERBLUE7	41.37	100.71	250	40.70	100.82	31	39.82	100.82	166
SUPERBLUE10	61.11	101.91	439	62.10	101.11	49	61.67	101.11	153
SUPERBLUE16	28.33	101.55	100	27.20	101.30	17	27.94	101.30	63
SUPERBLUE18	17.09	103.15	77	16.90	101.47	21	16.36	101.47	55
Avg.	+9.60%	1.003×	2.63×	+8.50%	1.000×	0.50×	+9.06%	1.007×	1.81×



Circuits	Ripple 2.0 [43]			RePlAce-r			RePlAce-r alt		
	sHPWL	RC	CPU	sHPWL	RC	CPU	sHPWL	RC	CPU
SUPERBLUE1	28.48	100.74	51	**25.89**	100.43	43	27.84	102.67	46
SUPERBLUE3	34.07	100.22	64	**30.78**	100.85	52	30.91	100.84	45
SUPERBLUE4	22.51	100.30	32	**20.94**	100.52	35	20.99	100.53	36
SUPERBLUE5	35.38	100.41	70	**33.37**	100.93	64	33.40	100.82	66
SUPERBLUE7	40.76	100.79	100	**37.10**	100.76	44	37.42	100.84	48
SUPERBLUE10	60.44	100.57	90	**58.41**	101.32	189	58.79	101.53	191
SUPERBLUE16	27.95	100.71	46	**25.46**	101.35	45	25.64	101.34	48
SUPERBLUE18	17.07	100.78	35	**14.60**	102.10	36	14.70	102.16	38
Avg.	+9.29%	0.995×	1.15×	+0.00%	1.000×	1.00×	+1.41%	1.003×	1.04×

◄ ICCAD-2012

[24] M.-K. Hsu, Y.-F. Chen, C.-C. Huang, S. Chou, T.-H. Lin, T.-C. Chen and Y.-W. Chang, “NTUplace4h: A Novel Routability-Driven Placement Algorithm for Hierarchical Mixed-Size Circuit Designs”, *IEEE Trans. on CAD* 33(12) (2014), pp. 1914-1927.