Analytical Placement of Mixed-size Circuits for Better Detailed-Routability

Shuai Li, Cheng-Kok Koh
ECE, Purdue University
{li263, chengkok}@purdue.edu

Outline

- Routability-driven placement
- Pin density oriented formulation
- Scaled smoothing
- Experimental results
- Conclusion

Routability-driven Placement

- Routability has become a critical issue because of
  - high number of metal layers, complex design rules, etc.
- Placers without considering routability
  - analytical placers

    formulated into an optimization problem minimizing HPWL
  - pack cells together to reduce HPWL, leading to poor routability
- Routability-driven placers
  - great improvement with the promotion of ISPD11, DAC12, ICCAD12 contests
  - most resort to placement refinement

(1) initial placement generation;

(2) congestion estimation mainly based on global routing;

(3) refinement techniques like white space allocation, cell bloating

Motivation

- Congestion estimation with global routing
  - The chip is partitioned into non-overlapping uniform *gcells*;
    gedges connect neighboring gcells;
    Overflows on gedges provide congestion estimation information.
  - Local nets connecting pins in the same gcell are ignored;
    Good *global routability* may not mean good *detailed routability*
- Use pin density as a compensate
  - High pin density indicates high routing demand, and possible routing congestion;
  - Add a weighted pin density factor in congestion estimation;
    Incorporate a pin denstiy term in the analytical placement formulation
- Pin density oriented formulation for placement

Motivation (cont'd)

- Macro blocks on mixed-size circuits
  - obstacles in placement, preventing cell movement;
  - blockages in routing, forcing wires to "detour" or "climb" to high layers.
- Another negative effect of fixed blocks
  - In analytical placement results, many cells may end up being placed on top of large macro blocks;
  - In legalization, moving these cells perturbs placement quality;
  - After legalization, many cells may be placed around macro blocks, blockages in the routing stage
- Scaled smoothing technique

Outline

- Routability-driven placement
- Pin density oriented formulation
- Scaled smoothing
- Experimental results
- Conclusion

Pin density oriented formulation

- The chip is partitioned into uniform non-overlapping bins.

$$\min \quad \text{HPWL}$$

$$\text{s.t.} \quad \sum_{c \in \mathcal{C}} r_c P_{bc}(x_c, y_c) \leqslant avg_{pd} S_b \quad \forall b \in \mathcal{B}. \qquad \textit{pin density constraint}$$

pin upperbound

- Pin upperbound

  avg_{pd} average pin density of all cells, S_b available area in bin b

$$avg_{pd} = \sum_{c \in \mathcal{C}} d_c \Big/ \sum_{c \in \mathcal{C}} (w_c h_c), \quad S_b = t_{den}(w_b h_b - F_b)$$

w_c, h_c: width, height of cell c;

d_c: number of pins on cell c;

w_b, h_b: width, height of bin b;

F_b: area occupied by fixed blocks;

t_{den}: target density (user-set).

Pin density oriented formulation

- The chip is partitioned into uniform non-overlapping bins.

$$\min \quad \text{HPWL}$$

$$\text{s.t.} \quad \sum_{c \in \mathcal{C}} r_c P_{bc}(x_c, y_c) \leqslant avg_{pd} S_b \quad \forall b \in \mathcal{B}. \qquad \textit{pin density constraint}$$

pin potential

- Pin potential
  - P_{bc}, overlapping portion of cell c in bin b

    pins on cell c are distributed to bins proportionally based on P_{bc}
  - r_c, normalization factor to guarantee cell c contributes exactly d_c pins to all bins

$$r_c = d_c \Big/ \sum_b P_{bc} \Rightarrow \sum_b r_c P_{bc} = d_c$$

Pin density oriented formulation

- The chip is partitioned into uniform non-overlapping bins.

$$\min \quad \text{HPWL}$$

$$\text{s.t.} \quad \sum_{c \in \mathcal{C}} r_c P_{bc}(x_c, y_c) \leqslant avg_{pd} S_b \quad \forall b \in \mathcal{B}. \qquad \textit{pin density constraint}$$

- Non-differentiable functions, $HPWL$, P_{bc}, are smoothed with existing techniques
- Optimized by solving a series of unconstrained optimization problem with λ being doubled gradually.

$$\min \quad \text{HPWL} + \lambda \sum_{b \in \mathcal{B}} [max(avg_{pd} S_b - \sum_{c \in \mathcal{C}} (r_c P_{bc}), 0)]^2$$

Each is solved by Limited-memory Broyden–Fletcher–Goldfarb-Shanno (L-BFGS)

Cell density oriented formulation

- Used in placement algorithms without considering routability
- Instead of pin density constraints

$$\sum_{c \in \mathcal{C}} r_c P_{bc}(x_c, y_c) \leqslant avg_{pd} S_b \quad \forall b \in \mathcal{B}. \qquad r_c = d_c \Big/ \sum_b P_{bc}$$

apply *cell density constraint* to achieve even cell distribution

$$\sum_{c \in C} k_c P_{bc} \leqslant S_b \quad \forall b \in \mathcal{B}. \qquad k_c = w_c h_c \Big/ \sum_b P_{bc}$$

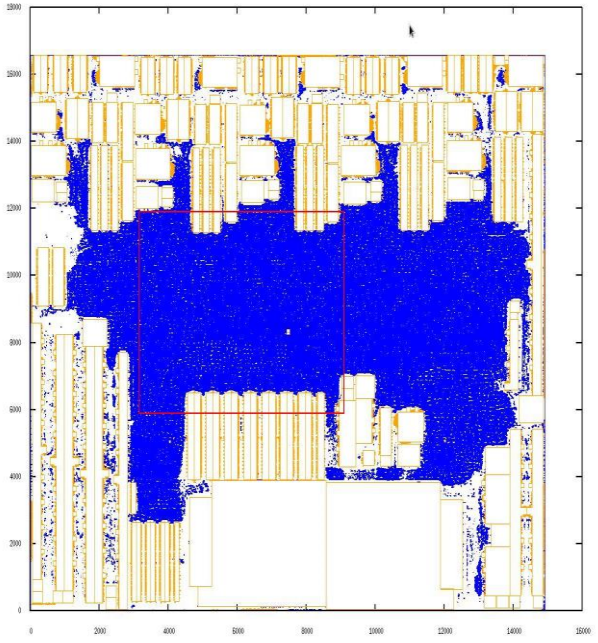
k_c: normalization factor such that each cell c contributes a total potential equal to its area

- When all cells have the same pin density avg_{pd} , the two are equivalent
- In reality, pin density on cells vary a lot

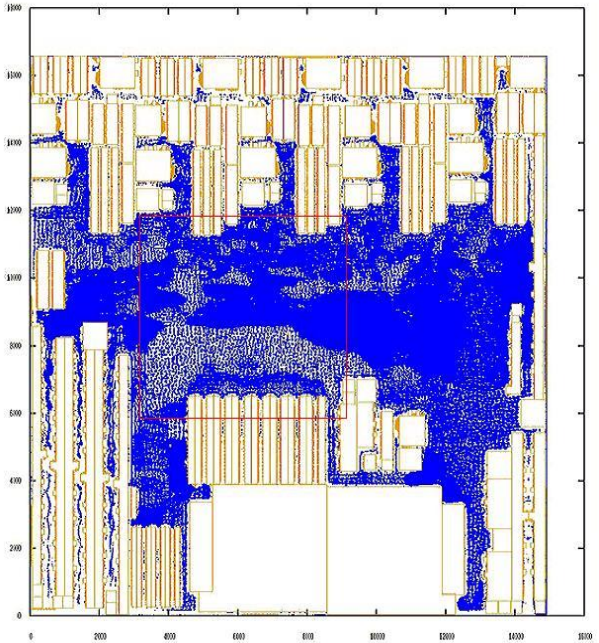
e.g. superblue4 has $avg_{pd} = 0.0299$, whereas pin density varies from 0.0024 to 0.1111

Comparison of the two formulations

- Placement results of benchmark circuit *superblue4*



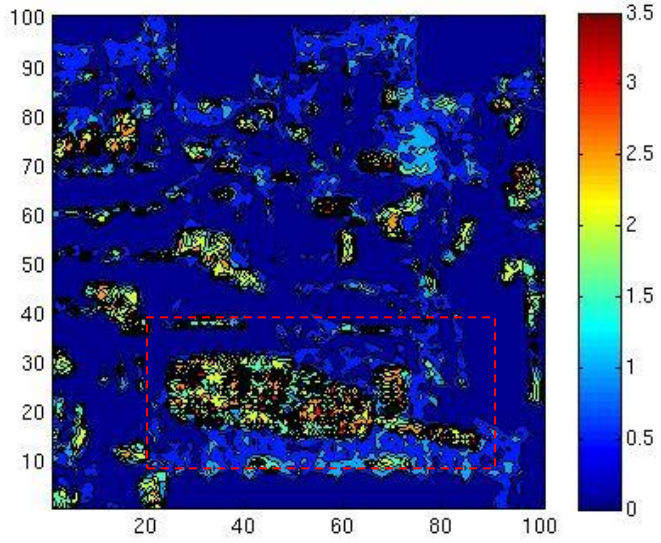
cell density constraints



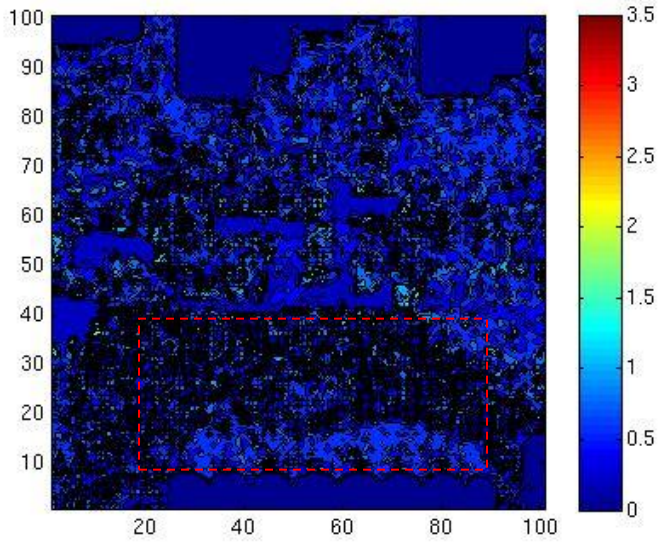
pin density constraints

Comparison of the two formulations (cont'd)

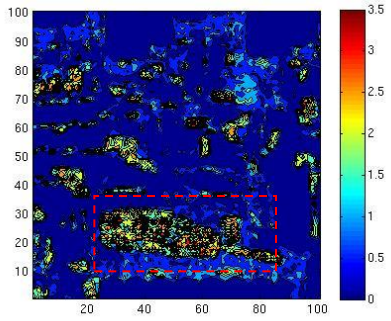
- Pin density contours in the red square box

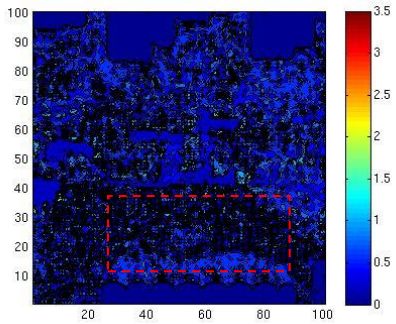


cell density constraints

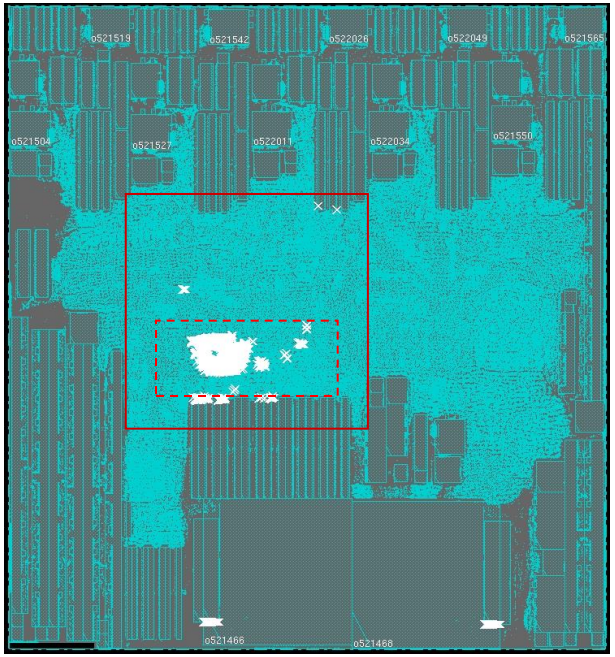


pin density constraints

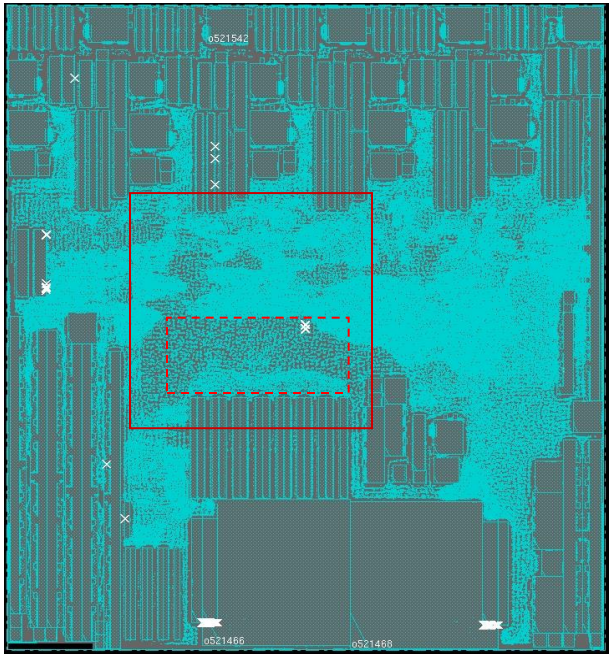




- Detailed routing results with white crosses denoting routing violations.



cell density constraints

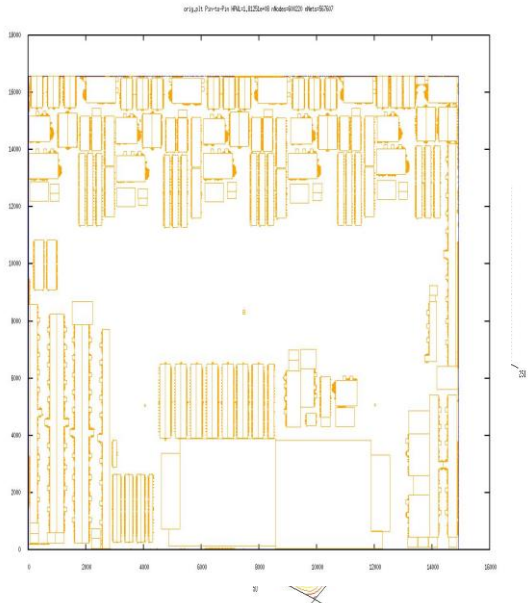


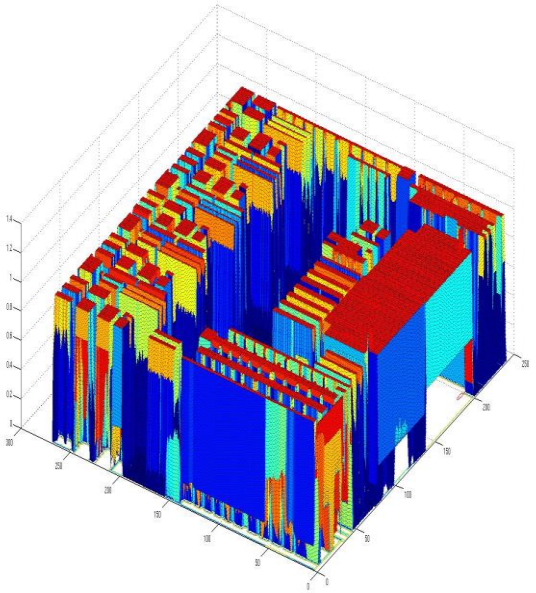
pin density constraints

Outline

- Routability-driven placement
- Pin density oriented formulation
- Scaled smoothing
- Experimental results
- Conclusion

Gaussian smoothing





- steep ”mountains” in the contour of F_b, distribution of macro blocks

  hard for cells to “climb”;

  cells on flat top may be “trapped”

- Gaussian smoothing is effective to solve the first problem

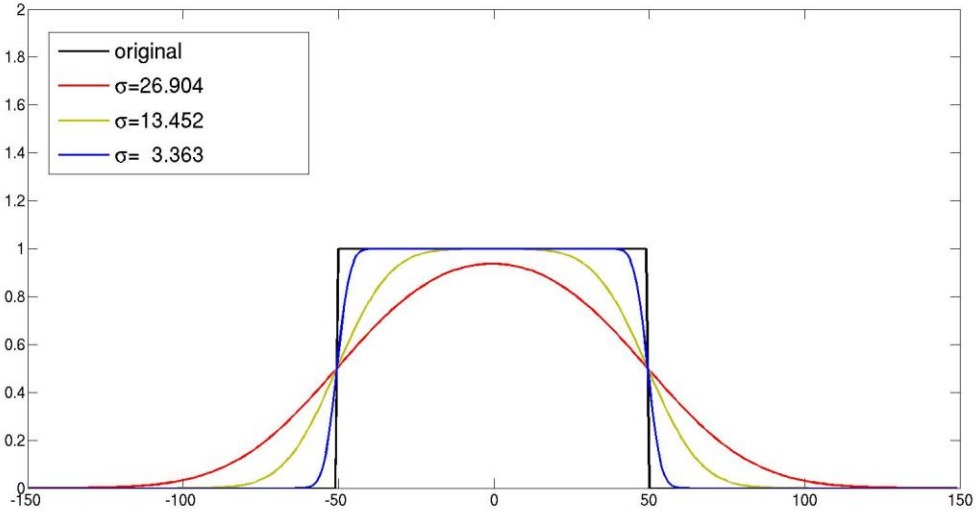
  T.-C. Chen et al. NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints. IEEE TCAD, 27(7):1228–1240, July 2008.

Gaussian smoothing (cont'd)

- Replace F_b with normalized F_b' , calculated by Gaussian smoothing

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \sum_b F_b' = \sum_b F_b$$

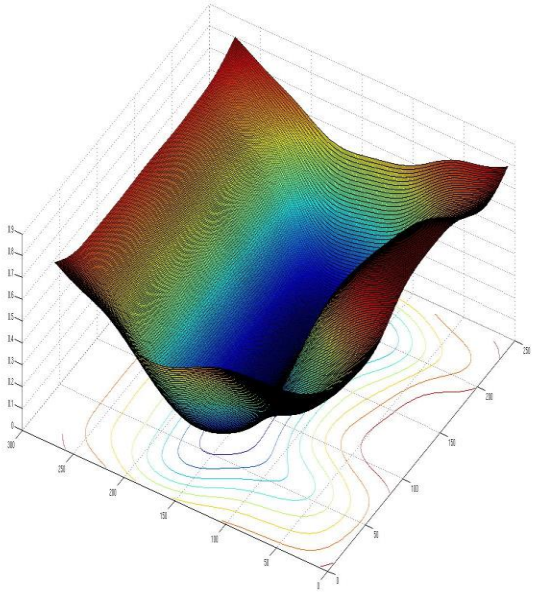
- Start with smoother F_b' with larger σ ;
  Iteratively, F_b' are recalculated with halved σ, until F_b' is close to F_b.
- Illustration of a 1-D example with single block

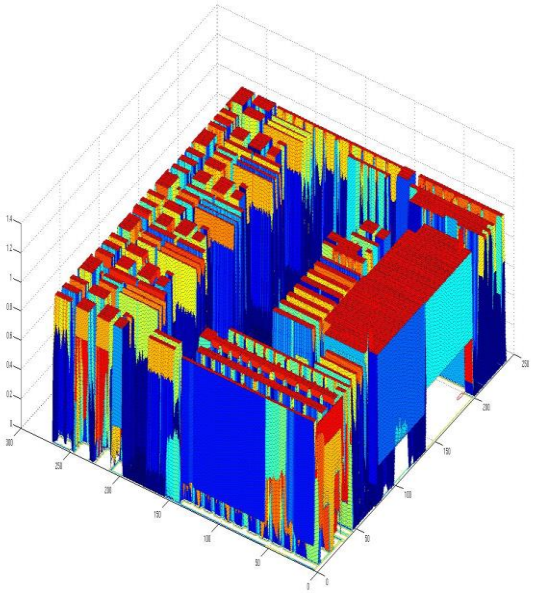


- Temporary empty space over blocks, when is larger σ
- Cells moved to these empty space are trapped after σ is decreased

Placement results illustration

- Gaussian smoothing

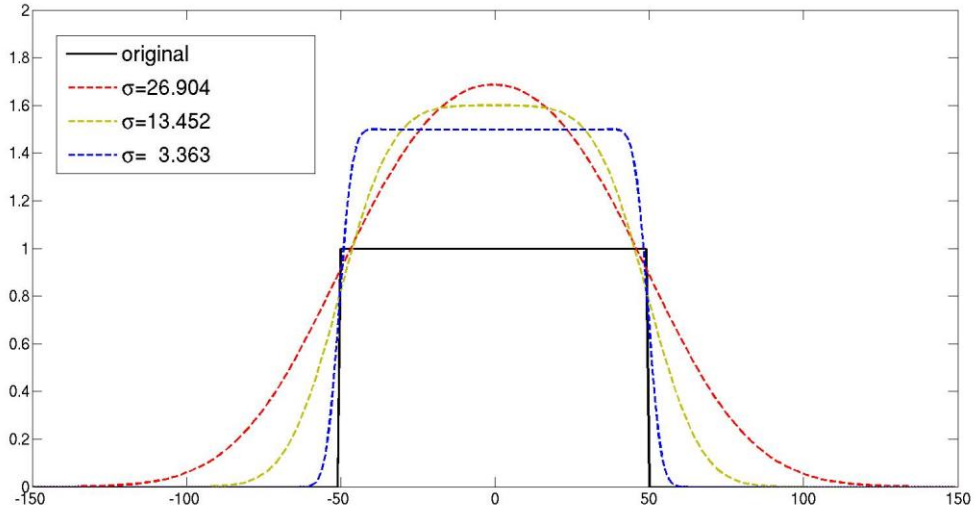




7.71% cells overlapping with blocks;

average displacement of cells in legalization is 45.6

Scaled smoothing



- Scale up F_b' properly, $F_b'' = \alpha F_b' \quad \forall b \in \mathcal{B}$
  little influence on the placement in area far from macro blocks;
  keeping cells away from “dangerous” area around macro blocks.
- Set scale-up factor such that the number of bins with $F_b'' \geq w_b h_b$ equals that of bins with $F_b = w_b h_b$

Scaled smoothing (cont'd)

- Negative effect on cell movement

  2-stage work in our implementation
- *Stage 1*: *Spreading cells*

  Gaussian smoothing with a large σ = a quarter of the chip width;

  Optimize placement by solving the constrained optimization problem.
- *Stage 2*: *Relocating cells overlapping with macro blocks*

  Initialize σ so that no "flat" top exist on any blocks;

  Scaled Gaussian smoothing;

  Optimize placement by solving the constrained optimization problem;

  If $\sigma > 3w_b$, $\sigma = \sigma / 2$ and repeat the last two steps; Otherwise, stop.
- To get the initial σ in stage 2

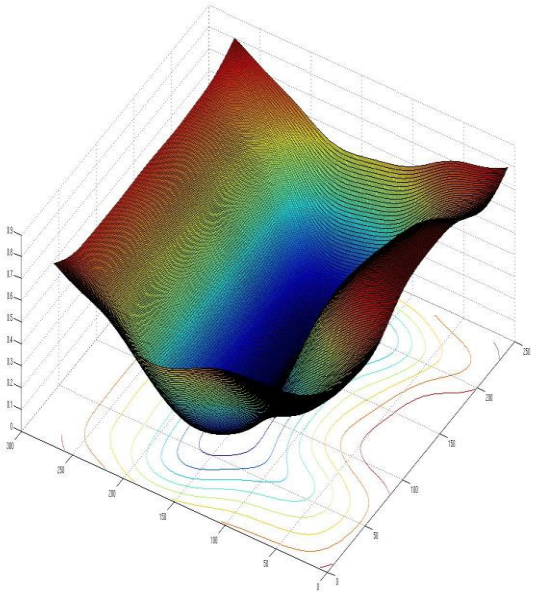
$$\frac{G(w_m/2, h_m/2)}{G(0,0)} = tol$$

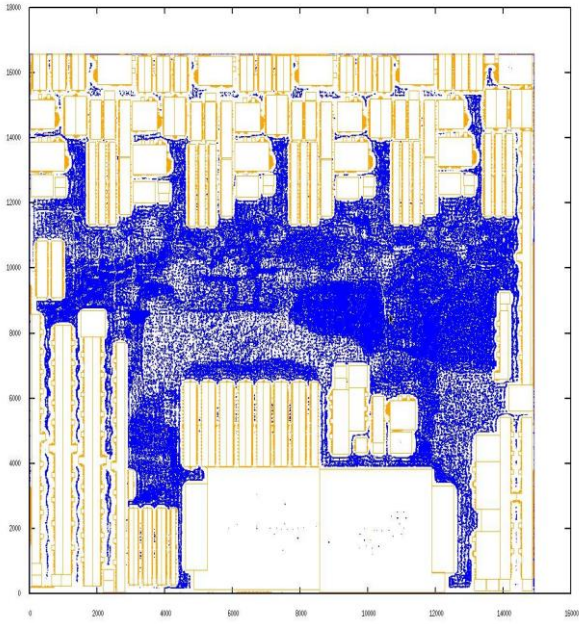
w_m, h_m: width and height of the largest block

tol: user-set tolerance value, e.g., 0.001

Placement results illustration

- Scaled Gaussian smoothing





0.08% cells overlapping with blocks;

average displacement of cells in legalization is 22.9

Placement results illustration (cont'd)

- Placement results of DAC12 benchmark circuits

  GS: Gaussian Smoothing; GSS: Scaled Gaussian Smoothing

		s2	s3	s6	s7	s9	s11	s12	s14	s16	s19
GS	%	11.32	11.28	11.56	1.66	6.94	8.69	0.73	8.58	5.41	3.20
	dsp	28.4	53.2	62.0	10.42	47.3	30.4	43.4	33.2	19.7	22.9
GSS	%	0.21	0.22	2.43	0.65	0.09	1.04	0.05	0.34	0.19	0.05
	dsp	8.5	16.6	20.5	10.9	15.6	12.0	6.4	21.3	9.4	10.5

%: percentage of cells overlapping with blocks;

dsp: average displacement of cells in legalization

Outline

- Routability-driven placement
- Pin density oriented formulation
- Scaled smoothing
- Experimental results
- Conclusion

Experimental results

- Detailed-routability evaluated by commercial router
  - *Wroute* in *Encounter*
  - Existing translator from Bookshelf files to LEF/DEF files

    W.-H. Liu *et al*. Case study for placement solutions in ISPD11 and DAC12 routability-driven placement contests. In Proc. ISPD, pages 114–119, 2013.
  - Design rule violations

    Many violations usually occur in the initial detailed routing solution, and as many as violations are repaired
  - Routing runtime

    Detailed routing takes much more time than global routing

Effectiveness of the proposed techniques

- DAC12 benchmark circuits

	GS&CellDen					SGS&CellDen					SGS&PinDen				
	VIO	WL	VIA	TR	OC	VIO	WL	VIA	TR	OC	VIO	WL	VIA	TR	OC
s2	637984	7.36	14.10	3326	5.43	79434	7.32	13.31	1857	4.97	681	7.00	11.90	312	3.81
s3	24415	4.06	11.81	1337	8.08	418	4.23	11.34	285	7.20	193	4.09	10.67	196	6.12
s6	60951	4.04	12.03	1604	7.63	1380	4.13	11.65	527	6.87	202	4.08	11.31	217	6.61
s7	248	4.76	16.97	317	9.57	102	5.00	16.66	281	8.61	418	4.89	16.45	280	8.85
s9	43468	2.92	9.92	549	8.02	58	2.93	9.59	167	7.00	33	2.94	9.40	156	6.76
s11	431	3.88	10.66	264	3.65	425	3.85	10.43	257	3.18	425	4.00	10.12	214	3.01
s12	11382082	4.66	19.64	3115	18.87	5210636	4.80	19.10	3026	18.51	104	4.34	16.38	286	15.87
s14	139958	2.69	7.69	1007	7.90	7397	2.79	7.41	276	7.22	621	2.77	7.00	178	6.51
s16	118225	2.84	7.92	818	7.26	54	2.86	7.70	144	6.68	36	3.01	7.38	134	7.10
s19	870	1.86	6.20	236	7.69	362	1.87	6.07	150	6.99	106	1.80	5.76	106	6.56
Norm	–	1.005	1.099	6.048	1.181	–	1.023	1.065	3.353	1.085	–	1.000	1.000	1.000	1.000

- GS: Gaussian smoothing
- SGS: scaled Gaussian smoothing
- CellDen: cell density contraints
- PinDen: pin density constraints

- VIO: number of violations
- WL(e7): wirelength in micrometers
- VIA(e6): number of vias
- TR(m): routing runtime in minutes
- OC(%): percentage of over capacity gedges

Evaluation of other placers' results

	NTUPlace4					Ripple					SimPLR				
	VIO	WL	VIA	TR	OC	VIO	WL	VIA	TR	OC	VIO	WL	VIA	TR	OC
s2	1015942	6.81	12.01	2086	3.57	52155	7.32	12.40	1187	3.83	692	7.01	12.08	301	4.02
s3	1099	4.01	10.80	307	6.2	205	4.39	11.39	260	7.17	162	4.58	11.31	225	7.71
s6	485	3.94	11.31	247	6.4	223	4.14	11.62	230	6.98	265	4.17	11.49	220	7.19
s7	181	4.85	16.50	272	8.75	257	5.43	17.38	300	10.24	8181	5.57	17.59	426	11.25
s9	65	2.94	9.41	190	6.61	98	3.30	9.86	169	7.97	48	3.13	9.69	173	7.67
s11	583	3.87	9.95	236	2.94	441	4.04	10.28	229	3.35	808	3.94	10.25	330	3.46
s12	106	4.31	16.73	442	13.03	217	4.82	17.73	481	15.71	166	4.74	17.27	296	16.17
s14	17476	2.69	7.30	632	6.69	28250	2.80	7.45	290	7.45	262930	2.88	7.48	541	7.66
s16	24	2.93	7.58	136	6.55	36	2.93	7.81	144	7.17	56	3.05	7.85	154	7.44
s19	12574	1.82	5.86	404	6.36	105	1.96	6.10	135	7.38	74199	1.88	5.97	449	6.96
Norm	–	0.981	1.010	2.382	0.942	–	1.058	1.053	1.647	1.085	–	1.053	1.043	1.498	1.117
Norm*	2.38	0.980	1.008	1.295	0.929	1.23	1.052	1.053	1.258	1.063	1.52	1.052	1.040	1.162	1.092

- Results of NTUPlace4, Ripple, SimPLR
  - generated in the DAC12 routability-driven placement contest
  - "abnormal" results with over 5000 violations for 2 to 3 circuits
  - *Norm**: normalization excluding s2, s7, s14 and s19

Outline

- Routability-driven placement
- Pin density oriented formulation
- Scaled smoothing
- Experimental results
- Conclusion

Conclusion

- Pin density Oriented Formulation
  - Realize even pin distribution in an analytical way
- Scaled smoothing technique
  - Avoid too many cells overlapping with macro blocks
  - Keep cells away from area around macro blocks, which are short of routing resources
- Both are helpful in generating placement results with better detailed routability

Thank you!