SRC 1091.001 Deliverable Report for May 2006: Highly Scalable Multilevel Placement Algorithm for Mixed-size with Complex Constraints

Tony F. Chan*, Jason Cong†, Guojie Luo†, Joseph R. Shinnerl†, Kenton Sze*, Min Xie†
*UCLA Mathematics Dept.
†UCLA Computer Science Dept.

May 31, 2006

1 Name, ID, Title

Deliverable Name: Highly Scalable Multilevel Placement Algorithm for Mixed-size with Complex Constraints

Task ID: 1091.001

Task Title: Highly Scalable Placement by Multilevel Optimization

2 Summary

This report summarizes recent work on robust techniques for multilevel mixed-size placement with complex constraints: routability and thermal. The mPL package has been augmented by a fast, Poisson-based numerical PDE (partial-differential-equation) solver in order to support global refinement of a given placement consistent with accurate, non-convex models of constraints and objectives. Numerical results show that our latest placement tool–mPL6 produces the best solutions comparing to available academic placers. It has produced the best quality of solutions in the ISPD'06 placement contest [8] where the main objective of the contest addresses the routability capability of placers. On the IBM version 2 examples [9], mPL6 produces results that can be successfully routed using an industrial router with 5 - 13% shorter wirelength than other academic placers. For thermal driven placement, a wirelength-driven placement is used for post-processing. mPL6 can reduce the maximum temperature by 11% with only 3% wirelength increase.

3 Technical Results

3.1 Wirelength-Driven Placement

The multilevel placement package mPL6 combines improved implementation of the global placer mPL5 [3] and the XDP legalizer and detailed placers [4]. It consistently produces robust, high-quality solutions to difficult instances of mixed-size placement in a fast and scalable runtime. Best-choice clustering [1] is used to construct a hierarchy of the problem formulations. Generalized force-directed placement guides global placement at each level of the cluster hierarchy. During the declustering pass from coarsest to finest level, large movable objects are gradually fixed in positions without overlapping with one another.

Comparisons on ISPD'05 benchmark [7], a set of benchmark derived from real industrial structure ASIC designs, are shown in Table 1 and Table 2. Our latest placement tool mPL6 has achieve over 14% wirelength

Circuit	mPL6 HPWL	APlace2 Rel. WL	Capo10 Rel. WL	mPL5 Rel. WL
adaptec1	7.79E+07	1.01	1.14	1.12
adaptec2	9.20E+07	1.04	1.12	1.14
adaptec3	2.14E+08	1.02	1.09	1.24
adaptec4	1.94E+08	1.08	1.06	1.18
bigblue1	9.68E+07	1.03	1.12	1.16
bigblue2	1.52E+08	1.01	1.05	1.32
bigblue3	3.44E+08	1.20	1.17	1.26
bigblue4	8.29E+08	1.05	1.16	1.15
average	1.00	1.05	1.11	1.20

Table 1: Comparisons of HPWL between APlace2, Capo10, mPL5, and mPL6 on ISPD05 benchmark.

Circuit	mPL6 Time (s)	APlace2 Rel. time	Capo10 Rel. time	mPL5 Rel. time
adaptec1	2894	3.02	2.20	1.83
adaptec2	2995	4.22	2.60	1.90
adaptec3	9353	3.27	1.94	0.75
adaptec4	8812	3.90	2.19	0.73
bigblue1	3636	3.16	2.58	0.95
bigblue2	10207	2.67	2.19	0.70
bigblue3	13564	3.90	5.04	2.42
bigblue4	30540	5.26	4.37	1.19
average	1.00	3.67	2.89	1.31

Table 2: Comparisons of runtime between APlace2, Capo10, mPL5, and mPL6 on ISPD05 benchmark.

reduction over mPL5 (an older version developed a year ago) and 30% runtime speed-up. Comparing to APlace2 [5] – a high-quality analytical placer producing the best wirelength in ISPD'05 placement contest, mPL6 has around 5% shorter wirelength and is 3.7X faster. Comparing to Capo10 [2] – a stable and well established min-cut based placer, mPL6 has around 10% shorter wirelength and is 2.9X faster.

3.2 Routability-Driven Placement

Based on the Generalized Force Directed method, we propose a congestion-driven multilevel global placement method that enhances the routability during global placement by re-placing cells to avoid congested regions. We also propose a congestion-driven white space allocation method that after global placement stage, allocates white space to provide appropriate routing resources to congested regions. A fast LZ router is used to evaluate the routing density associated with the intermediate placement solutions. Goto-based moves are used to re-place cells. Nets are subsequently weighted by the average routing congestion of the bins they cross. Before detailed placement, a slicing tree is recursively constructed based on current placement. Cutlines are shifted so that the relative amount of white space in neighboring regions are proportional to routing congestion estimation. Detailed description of the algorithm can be found in [6]. Table 3 and Table 4 give detailed routing results on IBM-Dragon version 2 circuits using mantle from Magma Design Automation Inc. Column "WS" gives the percentage of white space in each circuit. Column "RWL" gives the routed wirelength. Column "Via", "Vlts" and "O.C" give the number of vias, routing violations and the percentage of over congested global routing bins. Column "RRT(m)" gives the routing time. We can see that combining the proposed routability-driven global placement and white space allocation methods, we achieve placements with the best routability among the publicly available placement tools, with all IBM-Dragon version 2 easy and hard benchmark circuits successfully routed.

Recently, the latest version of mPL participated in the second ISPD placement contest, where area density is used to approximate routing density. According to the guideline of the contest, each circuit is also given

a target area density that the placer needed to achieve. Placements that violate the area density constraint in any placement bins will be penalized. Detailed description of the overflow penalty can be found in [8]. We give our results on the ISPD06 circuits in Table 5. According to the evaluation by IBM, our final results ranked 1st among a total of 9 academic teams.

3.3 Thermal-Driven Placement

The purpose of thermal-aware placement is to predict and remove hotspots in the placement stage. It is well-known that temperature variation has negative effects on leakage and electromigration, and brings in difficulties to designers. Thus it makes sense to enhance the placement with thermal-awareness.

In the current version, we implement the thermal-aware placement as a post process of a given placement. Similar to [mPL], we formulate the problem in a nonlinear programming form,

$$\begin{aligned} \text{minimize} \quad & \mathrm{WL}(x, y) \\ \text{subject to} \quad & T_{ij} = T_{des} \\ & \psi_{ij} = K_\epsilon \end{aligned}$$

We use evenly distributed temperature (T_{ij}) and cell area (ψ_{ij}) constraints to achieve a thermal-aware and non-overlapping placement. In the mean while minimizing the total wirelength is also the objective.

The solver is implemented with Uzawa algorithm by solving the following equations,

$$\begin{cases} \nabla \mathrm{WL}(x^{k+1}, y^{k+1}) + \sum \delta_{ij} \nabla T_{ij} + \sum \lambda_{ij} \nabla \psi_{ij} = 0 \\ \delta_{ij}^{k+1} = \delta_{ij}^{k} + \alpha_t (T_{ij} - T_{des}) \\ \lambda_{ij}^{k+1} = \lambda_{ij}^{k} + \alpha (\psi_{ij} - \bar{K}_\epsilon) \end{cases}$$

We generate power $P = \alpha C V^2$ for each cell, where $C \propto \text{area} \cdot \sum \#pin$, proportional to cell area and the total number of pins in adjacent nets, and the activity factor α is a random number between 0 and 1. We simplified the compact thermal model [Cell-Level] to be a 2D resistive network to evaluate the temperature.

Experiments are done on IBM-PLACE 2.0 benchmark suits. The results are compared with a lower bound of the optimal temperature, which is generated by evenly distributing the same amount of power on the resistive network. The term gap represents the difference between the highest temperature on the chip of a placement and the lower bound.

The results are listed in Table 6. The initial placement is generated by mPL 6.0. And the results after thermal-aware post-processing and detailed placement are compared with the initial placement. Column "T_{low}" is the lower bound of the optimal temperature. Columns "init. T_{max}" and "init. WL" are initial temperature and wirelength. And Columns "T_{max}(gap%)" and "WL(%)" are temperature and wirelength generated by our method, with the percentage of quality improvement or degradation. It shows that our method can reduce the gap by more than 45% with about 3% wirelength degradation in average.

References

[1] C. Alpert, A.B. Kahng, G. Nam, S. Reda, and P. Villarrubia. A semi-persistent clustering technique for vlsi circuit placement. In *Proceedings of the International Symposium on Physical Design*, pages 200–207, Apr 2005.

[2] A.E. Caldwell, A.B.Kahng, and I.L. Markov. Improved algorithms for hypergraph partitioning. In *Proceedings of the Asia South Pacific Design Automation Conference*, 2000.

[3] T. Chan, J. Cong, and K. Sze. Multilevel generalized force-directed method for circuit placement. In *Proceedings of the International Symposium on Physical Design*, pages 185–192, Apr 2005.

[4] J. Cong and M. Xie. A robust detailed placement for mixed-size ic designs. In *Proceedings of the Asia South Pacific Design Automation Conference*, pages 188–194, Jan 2006.

[5] A. B. Kahng and Q. Wang. Implementation and extensibility of an analytic placer. In *Proceedings of the International Symposium on Physical Design*, pages 18–25, 2004.

[6] C. Li, M. Xie, C.K. Koh, J. Cong, and P. Madden. Routability-driven placement and white space allocation. In *Proceedings of the International Conference on Computer Aided Design*, pages 394–401, 2004.

[7] G.-J. Nam, C.J. Alpert, P. Villarubbia, B. Winter, and M. Yildiz. The ispd2005 placement contest and benchmark suite. In *Proceedings of the International Symposium on Physical Design*, pages 216–219, 2006.

[8] Gi-Joon Nam. Ispd 2006 placement contest: Benchmark suite and results. In *Proceedings of the International Symposium on Physical Design*, pages 167–167, 2006.

[9] X. Yang, B.K. Choi, and M. Sarrafzadeh. Routability-driven white space allocation for fixed-die standard-cell placement. *TCAD*, 22(4):410–419, April 2003.

Circuit	WS	Algorithm	RWL(m)	Vias	Vlts	O.C%	RRT(m)
ibm01-easy	14.88%	Dragon 3.01	0.863	98132.8	0.0	9.800	3.1
		Capo 10.0	0.760	90747	0.0	6.2	2.8
		Fengshui 5.1	0.87	98103	0.0	9.8	3.6
		mPL-R+WSA	0.774	95062	0.0	5.7	2.6
ibm02-easy	9.58%	Dragon 3.01	2.093	203691.2	0.0	3.040	7.4
		Capo 10.0	1.962	193437	8.0	3.7	9.6
		Fengshui 5.1	2.222	205964	8.0	5.5	13.7
		mPL-R+WSA	1.827	196025	0.0	1.3	5.5
ibm07-easy	10.05%	Dragon 3.01	4.260	394051	13.6	1.700	19.7
		Capo 10.0	4.112	366074	2547.0	2.2	76.8
		Fengshui 5.1	4.145	387089	0.0	1.20	14.1
		mPL-R+WSA	3.802	385653	0.0	0.3	10.2
ibm08-easy	9.97%	Dragon 3.01	4.668	468620.4	0.8	0.760	14.1
		Capo 10.0	4.560	446484	114.0	1.9	27.3
		Fengshui 5.1	5.013	477123	0.0	1.20	16.7
		mPL-R+WSA	4.257	466140	0.0	0.000	11.8
ibm09-easy	9.76%	Dragon 3.01	3.713	411053.2	0.0	0.000	9.2
		Capo 10.0	3.414	377715	2.0	0.0	9.0
		Fengshui 5.1	3.738	403770	0.0	0.0	9.3
		mPL-R+WSA	3.309	410310	0.0	0.0	8.9
ibm10-easy	9.78%	Dragon 3.01	7.140	624391	0.0	0.1	16.5
		Capo 10.0	6.669	590823	0.0	0.3	18.4
		Fengshui 5.1	7.043	613396	0.0	0.0	15.3
		mPL-R+WSA	6.363	618877	0.0	0.0	14.6
ibm11-easy	9.89%	Dragon 3.01	5.390	521756	0.4	0.0	12.8
		Capo 10.0	4.931	487142	0.4	0.2	12.8
		Fengshui 5.1	5.599	516250	0.0	0.2	13.9
		mPL-R+WSA	4.983	524287	0.0	0.0	12.2
ibm12-easy	14.78%	Dragon 3.01	10.855	783041.8	698.6	2.180	57.8
		Capo 10.0	9.804	717937	138.0	1.2	51.5
		Fengshui 5.1	10.321	744567	0.0	1.00	29.0
		mPL-R+WSA	9.341	733525	0.0	0.1	21.4
Overall		Dragon 3.01	1.121	1.021	89.2	2.143	1.448
		Capo 10.0	1.041	0.954	351.2	2.100	2.299
		Fengshui 5.1	1.134	1.011	1.0	2.305	1.408
		mPL-R+WSA	1x	1x	0.0	1x	1x

Table 3: Routability results on IBM version 2 easy examples.

Circuit	WS	Algorithm	RWL(m)	Vias	Vlts	O.C %	RRT(m)
ibm01-hard	12.00%	Dragon 3.01	0.868	98348	0	11.4	3.4
		Capo 10.0	0.57	90817	0.4	8.5	3.1
		Fengshui 5.1	0.848	98416	0	10.9	3.5
		mPL-R+WSA	0.745	94005	0	5.8	2.5
ibm02-hard	4.72%	Dragon 3.01	2.021	202735	26.2	4.1	10.4
		Capo 10.0	1.990	196867	1102.0	5.1	53.2
		Fengshui 5.1	2.174	208578	340	6.3	30.5
		mPL-R+WSA	1.819	198801	0	3.5	7.3
ibm07-hard	4.70%	Dragon 3.01	4.199	398999	1018.4	3.4	57.8
		Capo 10.0	4.161	376970	12750.0	3.2	151.5
		Fengshui 5.1	4.202	397682	1155	3.5	72.1
		mPL-R+WSA	3.761	394559	0	1.6	15.7
ibm08-hard	4.84%	Dragon 3.01	4.553	474512	0.4	1.4	16.7
		Capo 10.0	4.568	453848	38.0	2.6	25.5
		Fengshui 5.1	4.815	485442	54	2.9	24.5
		mPL-R+WSA	4.178	474279	0	0.6	13.2
ibm09-hard	4.88%	Dragon 3.01	3.619	408867	0	0.0	9.7
		Capo 10.0	3.386	382505	0.4	0.0	9.3
		Fengshui 5.1	3.546	408356	0	0.0	9.1
		mPL-R+WSA	3.185	411713	0	0.0	9.1
ibm10-hard	4.92%	Dragon 3.01	6.943	632480	0	0.3	18.1
		Capo 10.0	6.890	601615	134.0	0.5	35.5
		Fengshui 5.1	6.976	625658	32	0.4	18.7
		mPL-R+WSA	6.262	627723	0	0.0	15.8
ibm11-hard	4.67%	Dragon 3.01	5.221	526241	0	0.4	15.6
		Capo 10.0	4.853	487113	0.0	0.3	13.3
		Fengshui 5.1	5.471	524574	0	0.6	17.4
		mPL-R+WSA	4.769	528374	0	0.0	14.0
ibm12-hard	9.94%	Dragon 3.01	10.118	769982	545.8	2.4	53.8
		Capo 10.0	9.630	723261	5028.0	2.9	158.6
		Fengshui 5.1	10.425	774132	3602	3.6	163.2
		mPL-R+WSA	9.237	750918	0	0.5	26.4
Overall		Dragon 3.01	1.115	1.012	198.85	1.820	1.638
		Capo 10.0	1.067	0.955	2381.6	1.900	3.792
		Fengshui 5.1	1.138	1.017	647.88	2.186	2.704
		mPL-R+WSA	1x	1x	0.0	1x	1x

Table 4: Routability result on IBM version 2 hard examples.

Circuit	Target	#cell	WL	SWL	RT(s)
adaptec5	0.5	843224	4.26E8	4.30E7	16619
newblue1	0.8	330137	6.72E7	7.32E7	3871
newblue2	0.9	441586	1.99E8	2.02E8	10073
newblue3	0.8	494123	2.82E8	2.84E8	15505
newblue4	0.5	646219	2.96E8	3.01E8	13175
newblue5	0.5	1233154	5.29E8	5.37E8	22204
newblue6	0.8	1255135	5.15E8	5.23E8	19152
newblue7	0.8	2508002	1.07E9	1.08E9	53510

Table 5: Results on ISPD06 examples.

circuit	T_{low}	init. T_{max}	init. WL	T_{max} (gap%)	WL (%)
ibm01	72.08	85.06	1.65E+006	80.31 (63.42%)	1.67E+006 (101.41%)
ibm02	130.1	151.34	3.59E+006	142.6 (58.86%)	3.70E+006 (102.95%)
ibm03	76.48	90.55	4.82E+006	86.95 (74.43%)	4.97E+006 (103.09%)
ibm04	57.76	82.1	5.91E+006	72.34 (59.91%)	6.03E+006 (101.98%)
ibm05	84.68	100.83	9.39E+006	92.58 (48.92%)	9.74E+006 (103.63%)
ibm06	68.45	72.47	4.90E+006	73.93 (136.39%)	5.10E+006 (104.06%)
ibm07	66.28	81.33	8.17E+006	72.67 (42.45%)	8.43E+006 (103.09%)
ibm08	100.78	127.02	9.08E+006	116.65 (60.48%)	9.29E+006 (102.39%)
ibm09	74.47	97.86	9.38E+006	82.08 (32.51%)	9.53E+006 (101.68%)
ibm10	76.83	96.52	1.78E+007	89.17 (62.70%)	1.81E+007 (101.78%)
ibm11	64.06	76.48	1.41E+007	69.74 (45.72%)	1.43E+007 (101.33%)
ibm12	83.39	99.6	2.22E+007	92.73 (57.63%)	2.27E+007 (102.11%)
ibm13	85.94	101.02	1.70E+007	91.85 (39.20%)	1.74E+007 (102.06%)
ibm14	62.1	101.97	3.24E+007	73.47 (28.51%)	3.38E+007 (104.34%)
ibm15	86.95	108.85	3.87E+007	98.71 (53.70%)	3.98E+007 (102.90%)
ibm16	75.52	104.33	4.36E+007	84.59 (31.47%)	4.43E+007 (101.69%)
ibm17	94.08	126.51	6.03E+007	109.62 (47.91%)	6.20E+007 (102.67%)
ibm18	75.24	127.3	4.16E+007	87.87 (24.26%)	4.74E+007 (114.10%)
average				53.80%	103.18%

Table 6: Thermal Placement Results