

Sub-Quadratic Objectives in Quadratic Placement

Markus Struzyna

Research Institute for Discrete Mathematics, University of Bonn

Lennéstr. 2, 53113 Bonn, Germany

struzyna@or.uni-bonn.de

Abstract—This paper presents a new flexible quadratic and partitioning-based global placement approach which is able to optimize a wide class of objective functions, including linear, sub-quadratic, and quadratic net lengths as well as positive linear combinations of them. Based on iteratively re-weighted quadratic optimization, our algorithm extends the previous linearization techniques. If l is the length of some connection, most placement algorithms try to optimize l^1 or l^2 . We show that optimizing l^p with $1 < p < 2$ helps to improve even linear connection lengths. With this new objective, our new version of the flow-based partitioning placement tool BonnPlace [25] is able to outperform the state-of-the-art force-directed algorithms SimPL, RQL, ComPLx and closes the gap to MAPLE in terms of (linear) HPWL.

I. INTRODUCTION

In large-scale placement, one looks for overlap-free positions to modules while their interconnect (net) length is minimized. This key step in physical design has been of ongoing interest for more than two decades and is still being subject of research and improvement in both, industry and academia.

Due to typical instance sizes (millions of modules and interconnects) and the separate, later routing step, the traditional objective for large-scale placement is the linear half-perimeter wirelength (HPWL) [14]. Though *linear* wirelength (or its approximations) is of particular interest for *routability* [15], [7], [9], [24] and *power consumption* (assuming fix wire widths and voltage) [1], for timing itself, super-linear functions are of major importance. In fact, the well-known *RC-delay* is a positive linear combination of a quadratic and a linear function of the wire length (assuming constant wire width) [22]. It is hence necessary to be able to optimize combinations of linear and super-linear functions, including the quadratic ones. Here, the linear objective imposes a particular difficulty due to its non-differentiability. Another issue are the negative consequences by loss of the strictly-convex property in the (purely) linear case.

Modern placement tools either optimize quadratic net length instead (and use implicit linearization of force-directed approaches [21], [20]) or partitioning with terminal propagation [16], [4], or circumvent the non-differentiable linear function by approximating the linear interconnect lengths by some other smooth mapping [2], [5], [6].

State-of-the-art placement algorithms [19], [8], [11], [10] use the B2B net model [19] in order to minimize linear HPWL. The B2B net model is motivated by two ideas. Once the leftmost and rightmost (lowest and highest) modules of a net are determined (e.g. by an existing placement or by quadratic clique/star minimization [20]), the *internal* connections which do not contribute to the HPWL are canceled. The second idea is the iterative re-weighting, already proposed in [18], to cover the gap between quadratic and the non-smooth linear net model.

However, it should be mentioned that iterative re-weighting has indeed a much longer history. Actually Weiszfeld studied a particularly easy “placement” problem: he asked for a position of one “module” minimizing the Euclidean length of its “interconnects” to a set of given fixed “modules” [23].

We argue that the iterative linearization has to be seen in a much broader concept of iterative least-square methods. Such a point of view allows then the optimization of a much broader class of functions in a quadratic placement framework. While intuition says that iterative re-weighting allows to optimize linear net length of some net and a quadratic of some other net, it is by far not evident for which class of functions such an iterative re-weighting converges in general. It turns out, that for a wide class of objective functions that are interesting for placement, convergence of this method can be guaranteed under certain circumstances, extending the results of [17].

The non-smoothness of the linear wirelength is not the only matter in placement. Given a connected netlist with at least one preplaced module, recall that the strict convexity of the quadratic function leads to a unique optimal solution for each module, which is no longer true for the linear case. Moreover, the strict convexity tends to spread the modules across the area, but these (still highly-overlapping) placements contain much more *relative* position information than placements obtained from explicit combinatorial (by solving the dual of a minimum-cost-flow-problem) linear length optimization. The latter tends to group on discrete points (cf. [1], p. 334). Hence, for a connection length l we propose to minimize l^p with $1 < p < 2$.

The key contributions in this paper are:

- We show that iterative re-weighting of quadratic functions minimizes linear, super-linear and sub-quadratic objective functions. We embed these into the concept of iterative least-squares methods which generalizes the approaches previously used in placement.
- We extend this result to a wide class of functions, including the positive linear combination of quadratic, sub-quadratic, super-linear and linear functions.
- We show that focusing on a *super-linear* objective, *linear* HPWL of legal placements can be improved in a partitioning-based placement tool.
- We evaluate our partitioning-based placement algorithm on the well-known benchmarks and demonstrate its effectiveness.
- In particular, we show that a sub-quadratic objective combined with a reliable partitioning is able to close the gap to state-of-the-art force-directed algorithms. We are able to produce several best results in terms of HPWL.

This paper is organized as follows. In Section II we introduce the notation and formalize the objective functions. Section III provides some theoretical background on iterative re-weighted approaches. Section IV focuses on re-weighting in placement and while V describes our placement scheme. In Section VI we present the experimental results and finally we conclude by the summary and the outlook in Section VII.

II. NOTATION AND OBJECTIVES

Given a netlist $\mathcal{N} = (V, E)$ with movable modules V and interconnecting nets E , the placement objective is to find positions (x_i, y_i) for each module $i \in V$ (some of the module positions may have already been fixed),

in such a way that the weighted interconnect lengths are minimized. We use the notation w_e to denote a non-negative net weight, and w_{ij} for a non-negative module pair interconnection weight. The functions that are considered here are separable (the horizontal and vertical coordinates can be optimized independently), so we will focus on the horizontal part only. Let $x = (x_i)_{i \in V}$.

Then, the weighted HPWL is:

$$HPWL_{\mathcal{N}}(x) = \sum_{e \in E} w_e (\max_{i \in e} x_i - \min_{j \in e} x_j). \quad (1)$$

Now, let $1 \leq p$. Then, the p -clique reads:

$$CL_{\mathcal{N}}^p(x) := \sum_{e \in E} CL_e^p(x) = \sum_{e \in E} \sum_{i, j \in e} w_{ij} |x_i - x_j|^p. \quad (2)$$

In particular, in this notation the 2-clique is the traditional clique which can be replaced by a sparse model, the star.

Several placement algorithms use the B2B net model, which requires an initial placement $x^{(0)}$. Given such a placement, let us call the modules which do not contribute to the HPWL *inner modules*. Then the B2B model of a connection $e \in \mathcal{N}$ in the k -th iteration $k = 0, 1, \dots$ uses the placement $x^{(k)}$ and is defined as

$$B2B_e(x)^{(k)} = \sum_{i, j \in e} w_{ij}^{B2B, (k)} (x_i - x_j)^2, \quad (3)$$

and $B2B_{\mathcal{N}}(x) := \sum_{e \in E} B2B_e(x)$. The weights are

$$w_{ij}^{B2B, (k)} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ are inner modules in } x^{(k)} \\ \frac{w_{ij}}{|x_i^{(k)} - x_j^{(k)}|} & \text{else.} \end{cases} \quad (4)$$

Then, the next placement iteration is computed via $x^{(k+1)} = \operatorname{argmin}_x B2B_{\mathcal{N}}(x)^{(k)}$ as long as $\|x^{(k+1)} - x^{(k)}\|_{\infty}$ is significant. Finally, $B2B_e := B2B_e^{(k)}$ is set for some sufficient large $k \in \mathbb{N}$. However, no convergence proof is possible as the denominator may vanish or a cyclic behavior can occur even for the simplest case of one movable module [26].

To simplify the considerations, we restrict ourselves to two-terminal nets: then the iteratively re-weighted clique and the B2B model coincide. It should be noted that this is not a real restriction at all: the choice of the net model (clique, star or B2B) itself corresponds to a mapping of multi-terminal into two-terminal nets. We consider a re-weighting of two-terminal connections which does apply to both models.

Let M be an module-net incidence matrix, i.e.:

$$M = (m_{ei})_{e \in E, i \in V} \text{ with } m_{ei} = \begin{cases} -1 & \text{if } e \text{ enters } i \\ 1 & \text{if } e \text{ leaves } i \\ 0 & \text{else.} \end{cases}$$

Now, given non-negative net weights in a diagonal matrix form $W = \text{diag}(w_e)$, minimizing $CL_{\mathcal{N}}^1(x)$ turns out to be:

$$\min_x CL_{\mathcal{N}}^1(x) = \min_x \|WMx - b\|_1 \quad (5)$$

for some vector b encoding connection offsets. If M has full rank and $w_e > 0 \forall e \in E$ (what we assume in the following), then the placement problem to minimize the 2-clique can be interpreted as a least-square solution to the Euclidean version of (5): $\min_x CL_{\mathcal{N}}^2(x) = \min_x \|WMx - b\|_2$. This problem is easy to solve as:

$$x^* = \text{argmin}_x CL_{\mathcal{N}}^2(x) \text{ iff } M^T WMx^* = M^T b \quad (6)$$

and $M^T WM$ is positive definite.

III. ITERATIVE RE-WEIGHTING IN THEORY

Here, we recall briefly the theory of iterative re-weighted least squares and extend the results of [17]. The two cases of $CL_{\mathcal{N}}^1$ and $CL_{\mathcal{N}}^2$ can be seen as special cases of a much wider class of functions. Let $\{\varphi_e\}_{e \in E}$ be a set of differentiable functions of its arguments and φ'_e their derivates. Given a placement x , let $r = Mx - b$ be the residuum and r_e its e -th component. Then (5) and (6) are special cases of $\min_x F(x)$ with

$$F(x) := \sum_{e \in E} \varphi_e(|r_e|) \quad (7)$$

where for some length value l the summands consist of functions $\varphi_e(l) = w_e l$ in the linear and $\varphi_e(l) = w_e l^2$ in the quadratic case. One option to minimize F is to use a sequence of least-squares, where given a placement $x^{(k)}$ we set $r^{(k)} := Mx^{(k)} - b$ and

$$F^{(k)}(x) = \sum_{e \in E} \frac{\varphi_e(|r_e^{(k)}|)}{|r_e^{(k)}|^2} r_e^2, \quad (8)$$

$$x^{(k+1)} = \text{argmin}_x F^{(k)}(x). \quad (9)$$

It is easy to see that fix points of (9) are indeed minimizers of (7) for

$$\varphi_e(l) = w_e l^p \text{ with } 1 < p. \quad (10)$$

Under certain circumstances it is possible to show that this method converges [17]. To guarantee convergence for the general case (including the linear case $p = 1$) one has to postulate strong assumptions like *boundedness*

from below which basically guarantees that no denominator of (8) vanishes for $k = 1, 2, \dots$ [17]. The authors of [17] also show that iterative re-weighted least squares work also for another class of $\{\varphi_e(l)\}_{e \in E}$, namely if $\{\varphi'_e(l)/l\}_{e \in E}$ are positive for $l \geq 0$ and non-increasing (cf. [17], pp. 252ff). The reader should note that the original proof in [17] made the (unnecessary) assumption that $\varphi_e \equiv \varphi$ for $e \in E$. The proof in [17] can be adapted to a much broader class of functions, in particular to "individual" $\{\varphi_e\}, e \in E$. Iterative re-weighting of these $\{\varphi_e\}_{e \in E}$ makes use of strict convex quadratic functions $G^{(k)}$ with

$$G^{(k)}(x) = \sum_{e \in E} \frac{\varphi'_e(|r_e^{(k)}|)}{|r_e^{(k)}|} r_e^2 + \text{const} \quad (11)$$

which can also be seen as iterative least-squares with weights

$$\varphi'_e(|r_e^{(k)}|)/|r_e^{(k)}| \text{ for } e \in E. \quad (12)$$

Although the convergence is slow in general, there are methods to accelerate it by applying - for example - Newton-like methods [13]. It can be shown that the convergence speed is highly dependent on the exponent p . The more p tends towards 1, the slower the convergence process [13], [17]. Thus, using $p > 1$ does not only have positive effects like strict convexity and hence uniqueness of the solution, but also helps to reduce the number of iterations.

IV. NON-LINEAR OBJECTIVE IN PLACEMENT

Translating these results back to placement yields several consequences:

- To minimize $CL_{\mathcal{N}}^p(x)$ for $1 \leq p$ one should minimize the sequence of re-weighted functions

$$F^{(k)}(x) = \sum_{e \in E} \sum_{i, j \in e} \frac{w_{ij}}{(|x_i^{(k)} - x_j^{(k)}|)^{(2-p)}} (x_i - x_j)^2. \quad (13)$$

as already proposed by [12]. In particular $p = 1$ and $p = 2$ cover the currently most popular objectives.

- For a net e and $1 \leq q \leq p \leq 2$ and weights $\alpha_e, \beta_e \in \mathbb{R}_+$ (12) allows the minimization of $\alpha_e CL_e^p(x) + \beta_e CL_e^q(x)$. In particular, for the interesting case of RC-delay minimization [22] with $p = 2$ and $q = 1$, (12) yields the sequence

$$F^{(k)}(x) = \sum_{e \in E} \sum_{i, j \in e} w_{ij} \left(2\alpha_e + \beta_e \frac{1}{|x_i^{(k)} - x_j^{(k)}|} \right) (x_i - x_j)^2 \quad (14)$$

of easily computable quadratic programs.

- For different nets, different objectives can be applied. In particular it is possible to optimize linear length of one connection, and a super-linear length of some other simultaneously. This is interesting in order to address congestion (more linear objective) for most nets while being able to focus on timing criticality/optimal repeater chain placement (more quadratic objective) on others.

V. PARTITIONING-BASED PLACEMENT

We have implemented iterative re-weighting in our quadratic flow-based partitioning global placement scheme [25], which used to minimize quadratic cliques and stars. In each step in which a quadratic net length minimization of the clique/star model was computed before, we now compute (i) a quadratic clique followed by (ii) up to 3 iterations of a modified B2B model: instead of using linear iterative re-weighting as in (4) in the k -th iteration, $k = 1, 2, \dots$, we rather use the weights:

$$w_{ij}^{B2B,(k),p} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ are inner modules in } x^{(k)} \\ \frac{w_{ij}}{|x_i^{(k)} - x_j^{(k)}|^{2-p}} & \text{else.} \end{cases} \quad (15)$$

This corresponds to minimizing CL_e^p for each net $e \in E$ without connections between inner modules of e , which is nothing else than a super-linear version of the *B2B* for two-terminal nets. Unlike *Kraftwerk2* [19], *ComPLx* [11], *SimPL* [8], in each step $k = 1, \dots$ the ordering, thus the external/internal modules of a net are computed from scratch. Once a global placement is computed, we use the legalization algorithm of [3] to compute the final positions.

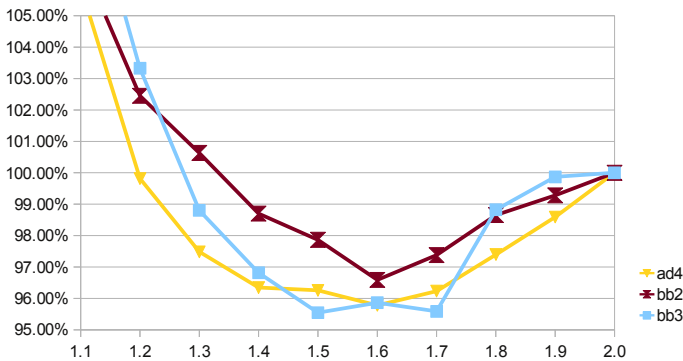


Figure 1. Relative (linear) HPWL of legal placements (y-axis) for different objective functions l^p (x-axis), $p = 1.1, 1.2, \dots, 2$ relative to l^2 for selected chips (adaptec4, bigblue2, bigblue3)

VI. EXPERIMENTAL RESULTS

We use the ISPD2005 benchmarks [14] for our tests. In the first experiment we compare the HPWL of legal placements depending on the exponent p of our objective. Figure 1 shows a typical behavior of our partitioning-based placement tool, exemplary for adaptec4, bigblue2 and bigblue3: the best net lengths are achieved for $p = 1.5$ or $p = 1.6$. While the net length increases for $p \rightarrow 2$ is expected due to higher focus on quadratic length than a linear one, there is a much more dramatic impact for $p \rightarrow 1$. This has two reasons. The first one, as already mentioned, is the fact that linear net length minimization leads to much higher overlaps. The high amount of overlaps amplifies the impact once more due to our partitioning-based algorithm itself. Partitioning decisions are made based on module positions. The more the modules overlap before a partitioning step is computed, the more arbitrariness is added to the partitioning decision. We use $p = 1.6$ in the following experiments.

Although we did not focus on runtime (yet) in our implementation and there is a lot of space for further acceleration, our algorithm already has reasonable runtime. We ran mPL6 [5] and our tool on an Intel Xeon X5690 with 3.46 GHz and show the runtimes in Table I. We did not have access to MAPLE, but an indirect comparison with mPL6 shows comparable runtimes to MAPLE [10], too. However, it should be noted that MAPLE used clustering which reduced the instance size by the factor of 2 with *BestChoice*, our results were obtained without clustering. Further acceleration of our tool is not only possible by a more efficient matrix construction which currently dominates the runtime, but more sophisticated methods to iterative re-weighted least squares minimization can also be applied [13].

Finally we compare our tool to mPL6 [5], SimPL [8], ComPLx [11], RQL [21] and MAPLE [10] and present the (linear) HPWL of legal placements. As we did not have access to RQL, MAPLE nor to SimPL/ComPLx, we cite the results for RQL and MAPLE from [10] and take the numbers from [8] for SimPL and from [11] for ComPLx. All these results are summarized in Table II. The results show that the flow-based partitioning based tool BonnPlace is able to outperform the currently best published results on adaptec2, adaptec3 and bigblue3. Moreover, the net lengths of our tool are shorter on average than the HPWL reported for any other tool. We have an advantage of more than 2.1% and 2.5% on SimPL and RQL and more than 1.7% on ComPLx and finally 5.3% on mPL6. Comparing to the currently

best tool w.r.t. the linear HPWL, MAPLE, we produce slightly better net lengths on average.

Wall clock runtimes in minutes				
	our		mPL6	
adaptec1	18.28	100.00%	20.20	90.51%
adaptec2	26.40	100.00%	21.88	120.64%
adaptec3	48.50	100.00%	63.68	76.16%
adaptec4	41.12	100.00%	58.17	70.69%
bigblue1	26.22	100.00%	26.28	99.75%
bigblue2	43.13	100.00%	78.85	54.70%
bigblue3	114.68	100.00%	99.03	115.80%
bigblue4	258.68	100.00%	203.48	127.13%
	577.01	100.00%	571.58	99.06%

TABLE I
WALL CLOCK RUNTIMES ON ISPD2005 BENCHMARKS

VII. SUMMARY AND OUTLOOK

In this paper we have generalized the iterative re-weighting for linearization purposes in quadratic placement to a much broader class of functions. Combined with the flow-based partitioning placement tool BonnPlace [25] and the B2B model we are able to close the gap to modern force-directed approaches. We argue that instead of focusing on extremes such as linear net length optimization (good for routing but highly-overlapping and slow in convergence) or a quadratic one (good for timing, unique solution, improved spreading and fast computation), choosing an intermediate exponent balancing the pros and cons of both objectives should be taken into consideration. Our results show that in a partitioning-based placement scheme for a connection length l optimizing $l^{1.5}$ leads to significantly better results than optimizing l^1 or l^2 . It would be interesting to know whether such effects can be confirmed with partitioning-free (e.g. force-directed) tools using the B2B model.

Our experiments suggest that even if focusing on the linear HPWL as the overall placement objective, the purely linear objective in quadratic programs is significantly worse even than the purely quadratic one. It is also an interesting open question whether such effects are dominated by the higher amount of overlaps in the linear case or result from the higher arbitrariness in position-based partitioning decisions.

Another vast field of research opens when applying the generalized re-weighting approaches in entire timing and congestion-driven flows. To decide, whether one still should focus on quadratic net length optimization for

some (e.g. timing-critical) nets, while for others (e.g. for routability reasons) a more linear objective might be meaningful is an open task for the future. Iterative re-weighting combined with an quadratic placement provides the flexibility to do both in a common, single and easy to optimize, quadratic objective function. In our strong belief, such individual objectives will play a crucial role in the future.

REFERENCES

- [1] Alpert C.J., Mentha D.P., Sapatnekar S.S. (eds.) Handbook of algorithms for physical design automation CRC Press (2009).
- [2] Alpert, C.J., Chan, T.F., Kahng, A.B., Markov, I.L. and Mulet, P., Faster minimization of linear wirelength for global placement. IEEE TCAD 17 (1998), 3–13.
- [3] Brenner U., VLSI legalization with minimum perturbation by iterative augmentation. Proc of DATE (2012), 1385–1390.
- [4] Brenner, U., Struzyna, M., and Vygen, J., BonnPlace: Placement of leading-edge chips by advanced combinatorial algorithms. IEEE TCAD 27 (2008), 1607–1620.
- [5] Chan, T.F., Cong, J., Shinnerl, J.R., Sze, K., and Xie, M., mPL6: enhanced multilevel mixed-size placement. Proc. of the ISPD (2006), 212–214.
- [6] Chen, T.C., Jiang, Z.W., Hsu, T.C, Chen, H.C. and Chang, Y.W., NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints. IEEE TCAD 27 (2008), 1228–1240.
- [7] Hsu M.K. Chou S. Lin T.H., Chang Y.-W. Routability-driven analytical placement for mixed-size circuit designs Proc. of the ICCAD (2011), 80–84.
- [8] Kim M.-C., Lee D., Markov I.L. SimPL: An Effective Placement Algorithm. IEEE TCAD 31 (2012), 50–60.
- [9] Kim M.-C., Hu J., Lee D., Markov I.L.: A SimPLR method for routability-driven placement. Proc. of the ICCAD (2011), 67–73.
- [10] Kim M.-C., Viswanathan N., Alpert C.J., Markov I.L., Ramji S. MAPLE: multilevel adaptive placement for mixed-size designs. Proc. of ISPD (2012), 193–200.
- [11] Kim M.-C., Markov I.L. ComPLx: A Competitive Primal-dual Lagrange Optimization for Global Placement Proc. of the DAC (2012), 747–752.
- [12] Li J, Lillis, J., Liu L.-T.; Cheng C.-K. New spectral linear placement and clustering approach Proc. of the DAC (1996), 88–93.
- [13] Li Y. A Globally Convergent Method for l_p Problems SIAM J. Optim. 3 (1993), 609–629.
- [14] Nam, G.-J., Alpert, C.J., Villarubbia, P., Winter, B. and Yildiz, M., The ISPD 2005 Placement Contest and Benchmark Suite. <http://www.sigda.org/ispd2005/contest.htm>
- [15] Viswanathan N., Alpert C.J., Sze C., Li Z. Nam G.-J., Roy J.A. The ISPD-2011 routability-driven placement contest and benchmark Proc. of the ISPD (2011), 141–146. http://www.ispd.cc/contests/11/ispd2011_contest.htm
- [16] Nam, G.-J., Reda, S. Alpert, C.J., Villarrubia, P.G. and Kahng, A.B., A fast hierarchical quadratic placement algorithm. IEEE TCAD 25 (2006), 678–691.
- [17] Osborne, M.R. Finite Algorithms in Optimization and Data Analysis Wiley (1985), London.
- [18] Sigl, G., Doll K. Johannes, F.M. Analytical Placement: A Linear or a Quadratic Objective Function, Proc. of the DAC (1991), 427–432.

HPWL Results on ISPD 2005 Benchmarks												
	BonnPlace		mPL6[5]		SimPL ² [8]		RQL ¹ [21]		MAPLE ¹ [10]		ComPLx ³ [11]	
ad1	76.87	100.00%	77.93	101.38%	76.87	100.00%	77.82	101.24%	76.36	99.34%	77.75	101.14%
ad2	86.36	100.00%	92.04	106.58%	90.37	104.64%	88.50	102.48%	86.95	100.68%	88.76	102.78%
ad3	202.00	100.00%	214.16	106.02%	206.38	102.17%	210.96	104.44%	209.78	103.85%	206.57	102.26%
ad4	181.53	100.00%	193.89	106.81%	186.00	102.46%	188.86	104.04%	179.91	99.11%	184.07	101.14%
bb1	94.85	100.00%	96.80	102.06%	95.85	101.05%	94.98	100.14%	93.74	98.83%	96.30	100.47%
bb2	144.21	100.00%	152.34	105.64%	143.56	99.55%	150.03	104.04%	144.55	100.24%	145.87	101.15%
bb3	317.71	100.00%	344.10	108.31%	336.19	105.82%	323.09	101.69%	323.05	101.68%	330.74	104.10%
bb4	781.79	100.00%	829.44	106.09%	796.78	101.92%	797.66	102.03%	775.71	99.22%	789.45	100.98%
avg		100.00%		105.34%		102.18%		102.50%		100.36%		101.78%

TABLE II

HPWL OF LEGAL PLACEMENTS ON ISPD2005 BENCHMARKS. BOLD FONT INDICATES THE BEST HPWL PUBLISHED. THE RESULTS MARKED BY ¹ ARE REPORTED FROM [10], THOSE WITH ² ARE FROM [8]. AND ³ FROM [11]. WE USED THE SAME PARAMETERS FOR ALL THE RUNS ABOVE, IN PARTICULAR $p = 1.6$.

- [19] Spindler, P., Schlichtmann, U. and Johannes F.M., Kraftwerk2 - A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model. IEEE TCAD 27 (2008), 1398–1411.
- [20] Viswanathan, N., Pan, M. and Chu, C.C-N., FastPlace 3.0: A Fast Multilevel Quadratic Placement Algorithm with Placement Congestion Control. Proc. of the ASPDAC (2007), 135–140.
- [21] Viswanathan, N., Nam, G., Alpert, C. J., Villarrubia, P., Ren, H., and Chu, C., “RQL: global placement via relaxed quadratic spreading and linearization”. Proc. of the DAC (2007), 453–458.
- [22] Sapatnekar S.S., Timing, Springer-Verlag New York, Inc. (2004).
- [23] Weiszfeld E., Sur le point par lequel la somme des distances de n points donnés est minimum. Tohoku Math. J., 43 (1937), 355–386.
- [24] He X., Huang T., Xiao L., Tian H., Cui G., Young E.F.Y., Ripple: An effective routability-driven placer by iterative cell movement. Proc. of the ICCAD (2011), 74–79.
- [25] Struzyna M., Flow-based partitioning and position constraints in VLSI placement. Proc of DATE (2011), 607–612.
- [26] Rautenbach D., Struzyna M., Szegedy C., and Vygen J., “Weiszfeld’s algorithm revisited once again,” Research Institute for Discrete Mathematics Report #04946-OR, University of Bonn, Germany, (2004).