Quadratic Timing Objectives for Incremental Timing-Driven Placement Optimization

Mateus Fogaça, Guilherme Flach, Jucemar Monteiro, Marcelo Johann and Ricardo Reis
PGMicro/PPGC - Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
{mpfogaca, gaflach, jucemar.monteiro, johann, reis}@inf.ufrgs.br

***Abstract*—In this work, two quadratic formulations for incremental timing-driven placement are proposed. We include the delay model formulation into the quadratic function objective. Our timing-driven quadratic techniques perform path smoothing by optimizing the distance of neighbor critical pins and balance wire load capacitance in the critical nets by reducing their total quadratic length. In our incremental quadratic techniques, the previous placement solution is kept unchanged by using the linear system neutralization operation before solving it. Our incremental quadratic timing-driven placement flow outperforms the state-of-art results by 9.4% and 7.6%, on average, regarding WNS and TNS improvement for late timing violation, respectively.**

Keywords*—*Timing-Driven Placement, Timing Optimization, Physical Synthesis, Electronic Design Automation.

I. Introduction

The feature sizes of integrated circuits have experienced an impressive and steadily reduction since the first chips were released. However, from a speed point of view, scaling tremendously benefits cells timing propagation and significantly worsen interconnections delay. As interconnections shrink, their resistance increase due to reduced cross section and their capacitance also increase due to reduced spacing and decreased thickness [1]. From 130nm and below technology nodes, interconnections overcame gates internal timing propagation and became the dominant factor in the circuit delay.

Since placement is the primary step defining the interconnections length, it is now also an essential step to achieving timing closure. The classical placement objective – total wirelength minimization – continues to be the core goal of placement as minimizing the total wirelength helps to achieve other goals as routability and timing [2]. However, timing cannot be ignored during placement and only aiming to optimize timing indirectly via wirelength minimization is insufficient. Therefore, timing optimization techniques need to be incorporated within placement flow, creating the algorithms categorized as timing-driven placement. Timing-driven techniques may be divided into Global and Incremental ones [3].

Global techniques perform the placement of the entire circuit, not necessarily respecting previous solutions. They usually apply net-weighting techniques [4], [5] to prioritize nets with high violations or assign a maximum wirelength or delay value for them. These techniques can deal with several timing violations at the same time, keeping a global view of the problem. However, while these nets are optimized, other violations may show up and, thereby, new constraints need to be generated. In the end, the problem may be over-constrained, and the solution may be a local minimum. Over-constrained solutions also may lead to congestion and affect routability. Incremental techniques aim to improve an existing solution by moving just a reduced number of cells, commonly through path-based techniques. The key idea is to reduce the critical paths lengths by straightening it through applying local search heuristics [6], [7] or linear programing [8]. The drawback of these techniques is the loss of the global view of the problem since linear programming runtime may be prohibitive for a large number of cells. On the other hand, these techniques are more robust to oscillation when compared to global techniques.

We address quadratic placement techniques, which are commonly applied to solve the placement problem [9]. In these techniques, circuit netlist is modeled as a graph $G = (V, E)$, where the cells, macros and primary inputs and outputs nodes are the set of vertices V, and the nets are decomposed into a set of binary edges E. Placement is thus solved as a spring system, where nets represent attraction forces between circuit nodes and additional spreading forces are added to the system to avoid congestion. This same approach can be applied to timing-driven placement as seen in [3].

In this work, we proposed an incremental quadratic approach, which relies on an operation called *neutralization*. Given the current solution and the circuit netlist, the traditional spring system is constructed. However, the neutralization finds for each cell, an additional force such that the equilibrium point of that cell in the spring system is the given current position. This approach allows improving a given placement incrementally using the quadratic formulation. It also considers all cells as movable and the attraction forces between them, keeping the global view of the problem.

The contributions of this work may be summarized in the following items:

- Two approaches to integrate timing optimization into an incremental quadratic placement modeling – One for path smoothing and other for load balancing;
- A flow integrating analytical formulation and [7] local search algorithms flow is proposed;

This paper is organized as follows: Section II presents the related works; Section III explores the adopted methodology to apply quadratic placement incrementally; The net weighting formulations for quadratic placement are presented in Section IV and the integration in the proposed flow is shown in Section V; Section VI discusses the experimental results and in Section VII are made the final remarks.

978-1-5090-6113-6/16/$31.00 ©2016 IEEE

II. Related Work

A two-phase timing-driven placement algorithm is presented in RITUAL [10]. The first phase is a continuous optimization of the weighted quadratic wirelength through quadratic placement, and the later is a discrete optimization using recursive partitioning and weighted assignment. In both phases, the weight is obtained by modeling timing constraints in a Lagrangian Relaxation formulation. The delays are estimated through Bakoglu [11] timing model and star net topology.

SPEED [12] proposes a net weighting approach applied to quadratic placement. Net delays are estimated using Elmore delay model and a star net topology. Initially, all weights of the circuit are set as the number of pins in the net. The weights are updated dynamically. As mentioned before, when a critical net is optimized, other may become critical. The algorithm keeps track of the nets criticality in the last two iterations to avoid this kind of oscillation. When the net is evaluated critical, its weight is increased. Otherwise, if the net stays non-critical for two iterations, its weight is reduced by half.

ITOP [3] proposes various techniques to achieve timing closure. The first one is a netlist transformation in which virtual 2-pin nets are created linking cells in critical paths to raise attraction between them in global placement. Furthermore, an incremental path smoothing algorithm locally moves critical modules trying to achieve local improvements. Unlike most algorithms, after changing the solution, small movements are performed to mitigate congestion and ensure routability. Finally, the authors combine other techniques, like buffering and repowering, to further improve the solution quality.

Finally, in [7] we proposed a flow of techniques to address both early and late timing violations. To mitigate early violations useful clock skew, iterative cell spreading, register swaps and register-to-register path fixing techniques are presented. Late violations are addressed through clustered movements (based on [6]) and single cell movements aiming to reduce the load capacitance in critical nets and balance load based on cells drive strength. The techniques were integrated into a flow that produces the best-known results for ICCAD 2015 CAD contest [13] on incremental timing-driven placement.

III. Incremental Quadratic Timing Objective Placement Technique

A good global placement solution carries on several important properties as reduced wirelength, overlap, congestion and timing. The incremental placement tries to improve a given solution while keeping its properties. To make use of an initial solution in the quadratic placement formulation constant forces or anchors may be added to the linear system to hold movable elements in their current position [14]. We call this procedure neutralization as the linear system is adjusted in such a way that its solution provides the current element positions and all forces acting on the system cancel each other. After the neutralization, the linear system is perturbed again to optimize timing violation. Therefore, a new placement solution is generated.

In this work we neutralize the linear system using anchors as they can be viewed as a generalization of constant forces and are typically more stable [14]. We also assume that the linear system presented in Equation (1) describe a quadratic placement formulation. No assumption on net modeling (e.g. clique, star, etc.) is made.

$$A\vec{x} = \vec{b} \tag{1}$$

A movable element is connected to a fixed element whenever the diagonal value respective to that element is greater than the sum of the absolute values of the off-diagonal elements. Therefore, to add an anchor to every cell, one needs to add a positive value to the diagonal of the connectivity matrix as shown in Equation (2) where $w_i > 0$ is the weight associated to the ith anchor. Multiple fixed points are automatically merged into one. So preexistent movable-to-fixed connections are seamlessly handled.

$$\dot{A} = A + diag(w_1, w_2, \cdots, w_n) \tag{2}$$

To finally neutralized the system, the anchor positions need to be defined. This is accomplished by setting the right-hand-side as in Equation (3) where $\vec{\mathbf{x_0}}$ is the current element positions.

$$\dot{\vec{b}} = \dot{A}\vec{x_0} \tag{3}$$

IV. Net Weighting Techniques for Timing Critical Path Smoothing

The Elmore delay for a wire is given by Eq. 4, which is a second-degree function whose quadratic term is the length of the interconnection (L_{ij}) multiplied by the resistance and capacitance per unit length of interconnections R_w and C_w, respectively. The linear term is the interconnection length multiplying the resistance per unit length and the capacitance being load by the wire (C_l). Eq. 5 represents the objective function of the quadratic placement, which is the summation of the quadratic distance between connected nodes (L_{ij}^2) weighed by a factor g_{ij}. If we assume that each 2-pin connection between cells in quadratic placement is a wire and $g_{ij} = RC$, then minimizing quadratic placement function (Eq. 5) also should minimize the wire delay. Since RC is a constant multiplying every term in the circuit, removing them from the formulation do not change the results.

$$D_w(i,j) = R_w C_w (L_{ij})^2 + C_l R_w (L_{ij}) \tag{4}$$

$$W(\vec{x}, \vec{y}) = \sum_{i}^{N} \sum_{j}^{N} g_{ij} (L_{ij})^2 \tag{5}$$

However, minimizing the wires alone does not produce good results. The ideal for a cell, between its driver and sinks, is a function of the driver and cell strengths. One way to cope with this issue in quadratic placement is to multiply the quadratic distance between a driver and its sink by the driver resistance.

A. Driver-sink additional forces

This net weighting approach aims to perform critical path smoothing. The first step is to traverse all nets in the circuit. For each net it is verified if the driver is critical, i.e. there is at least one critical path passing through it. Consider the example illustrated in Fig 1 – There is a net with a critical driver o and 3 sinks; 2 of them, $i1$ and $i2$ are also critical. Two extra edges are added to the system to align both paths passing through this net, one between o and $i1$ (dashed red) and other between o and $i2$ (dotted green). The weight of those connections is also enhanced by a factor α and the critically of the sink. In this work, the criticality is a metric to estimate the importance of a pin or a cell w.r.t the circuit TNS. The criticality of a pin is the negative slack of the pin divided by the worst negative slack found in the circuit while the criticality of a cell is equal to the maximum criticality in the pins that belong to the cell. Therefore, criticality is a real value in the range $[0, 1]$. The final weight for the new edges are:

$$g_{ij} = \alpha R_{drive}(1 + critically(j)) \tag{6}$$

Where R_{drive} is the driver's strength and $critically(j)$ is the sink criticality. By following these approach, we address the worst path passing through every pair driver-sink in the circuit.

B. Drive-strength-aware clique model

Another may mitigate timing violations by properly balance the capacitance in the nets composing critical paths. Critical nets were decomposed into cliques, as shown in Fig. 1(b), to address capacitance balancing. The weight of each edge is given by:

$$g_{ij} = \frac{\alpha R_{drive}(1 + criticality(i))}{(k-1)} \tag{7}$$

Where R_{drive} is the driver resistance; k is the number of pins in the net and $criticality(i)$ is the driver criticality. Following this approach, the total quadratic length of the net will be optimized, pondered by its drive strength and criticality. Similarly as done in wirelength-driven placement, we avoid overweighting in large-fanout multi-pin nets dividing the weight by a factor $k - 1$.

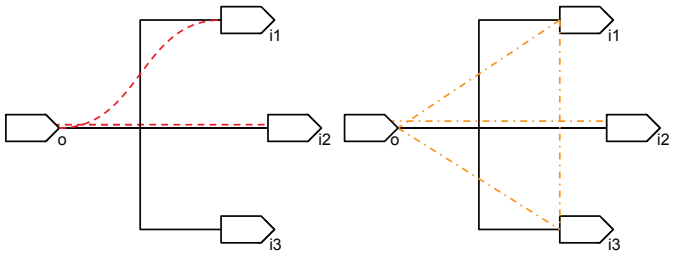

Fig. 1: Addition of attraction forces related to timing violation using driver-sink methodology (a) and drive-strength-aware clique model (b).

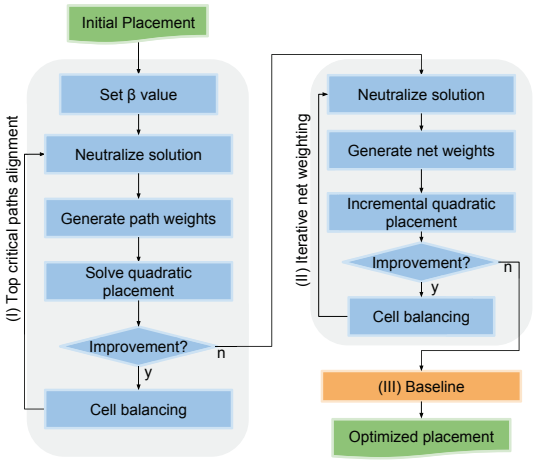

Fig. 2: Proposed incremental timing-driven placement flow using quadratic timing objective formulation.

V. Proposed Incremental Timing-driven placement Flow Using Quadratic Timing Objectives

The proposed analytical formulations were integrated into the flow depicted in Fig. 2. It relies on three major stages: (I) Top critical paths alignment, (II) iterative net weighting and (III) the baseline flow. The baseline flow is an implementation of the flow composed of local search techniques previously proposed by us [7] and presented in Section II.

Top critical path alignment consists of iterating through the circuit netlist finding the nets whose driver has a criticality higher than a threshold β. In these nets, the edges connecting the driver cell and their criticals sinks are strengthened following the methodology driver-sink presented in Section IV-A. In the experiments performed, the value of β was set automatically for each benchmark as the criticality of the third most critical endpoint. This threshold estimation methodology proved to be efficient for tested benchmarks. The path alignment formulation is repeated until the quality of result stops improving.

Iterative net weighting improves the current solution quality by assigning higher weights to the cliques proportionally to their driver criticality following the technique proposed in Section IV-B. At this stage, the goal is to reduce the wire load capacitance on the critical nets and to align the critical paths. As also pointed in other timing-driven analytical works, the net weighting technique may present oscillations before converging to the final solution.

Every time a new solution of the quadratic placement is accepted in stages (I) and (II), it is applied one iteration of the cell balancing technique from the baseline flow. Experiments showed that combining this load balancing technique helped to improve the gains of the quadratic placement. The loop verifications are done before cell balancing execution to assure that the gains come from quadratic placement, avoiding local minima – This conjecture was also confirmed empirically. The last stage is the baseline flow.

VI. Experimental Results

The timing-driven algorithms were implemented in C++11 language and evaluated on an Intel Core i7-4790K CPU @ 4.00GHz × 8 CPU with 32GB running Ubuntu 14.04 LTS (64-bit). The experiments were validated using the infrastructure of 2015 ICCAD CAD Contest in Incremental Timing-Driven Placement [13]. The infrastructure provides eight mixed-size benchmark circuits ranging from 700k to 2M components; A metric to evaluate the quality of the solution based on weighted arithmetic mean to measure early and late slack improvement and ABU change; Two cell displacement constraints (short and long) and an evaluation script to check legality and measure quality of the solution. The proposed quadratic techniques are unaware of cell displacement constraint. Therefore, experimental results are presented only for long cell displacement constraint.

A comparison among solutions of the initial placement, ICCAD 2015 contest winner, the baseline ([7]) and the proposed flow is presented in Table I. Our proposed incremental quadratic timing-driven placement flow outperforms baseline and ICCAD 2015 contest winner by 16.18% and 97.56%, on average, in the quality of the solution, respectively. However, the proposed flow is unable to achieve timing violation improvement compared to the baseline flow for the superblue10 circuit. The timing violation gains on path smoothing for this benchmark were lost in legalization due to the high-density usage areas around the macro blocks. The proposed flow keeps the gains in the early timing violation improvement achieved by the baseline flow.

TABLE I: Experimental results of our incremental timing-driven placement flow on ICCAD 2015 contest benchmarks.

Benchmark Cells / Macros Max Disp.	Solution	Metric					
		ABU	StWL (μm) $\times 10^7$	Late (ps) WNS $\times 10^3$	Late (ps) TNS $\times 10^5$	Run-time (min)	Quality Score
superblue1	Initial	0.05	9.59	-4.98	-4.60	-	-
1.21M	Contest winner	0.06	9.61	-4.57	-3.51	3.20	346.64
3787	Baseline	0.01	9.88	-4.46	-3.40	2.33	512.57
400um	Proposed	0.01	9.91	-4.21	-3.26	5.89	568.64
superblue3	Initial	0.03	11.43	-10.15	-15.03	-	-
1.21M	Contest winner	0.03	11.46	-8.71	-11.60	2.70	551.74
2074	Baseline	0.01	11.59	-8.30	-9.68	2.66	735.72
400um	Proposed	0.01	11.61	-6.87	-8.10	9.78	915.24
superblue4	Initial	0.04	7.15	-6.22	-34.77	-	-
796k	Contest winner	0.05	7.16	-5.76	-24.65	1.86	507.31
3471	Baseline	0.04	7.53	-5.51	-23.61	2.96	680.91
400um	Proposed	0.04	7.55	-4.64	-22.93	5.25	770.34
superblue5	Initial	0.02	10.75	-25.70	-69.65	-	-
1.09M	Contest winner	0.02	10.78	-24.29	-58.42	2.53	179.53
1872	Baseline	0.00	10.92	-24.42	-59.75	1.97	476.82
400um	Proposed	0.00	11.02	-20.92	-53.71	6.65	634.89
superblue7	Initial	0.03	14.01	-15.22	-18.57	-	-
1.93M	Contest winner	0.03	14.03	-15.22	-15.11	5.31	200.72
4910	Baseline	0.01	14.22	-15.22	-13.61	3.13	275.64
500um	Proposed	0.01	14.24	-15.22	-11.55	8.63	389.19
superblue10	Initial	0.04	20.53	-16.49	-331.53	-	-
1.88M	Contest winner	0.04	20.55	-16.08	-315.18	3.74	181.33
1696	Baseline	0.01	21.08	-15.54	-279.71	4.96	499.38
500um	Proposed	0.01	21.08	-15.53	-280.03	7.35	498.80
superblue16	Initial	0.03	9.33	-4.58	-7.76	-	-
982k	Contest winner	0.04	9.37	-3.85	-2.66	2.24	894.76
419	Baseline	0.00	9.50	-3.48	-2.03	1.78	1196.97
400um	Proposed	0.00	9.54	-3.11	-1.72	5.15	1279.93
superblue18	Initial	0.04	5.77	-4.55	-10.35	-	-
768k	Contest winner	0.05	5.78	-3.82	-7.76	1.59	613.07
653	Baseline	0.01	5.90	-3.74	-6.19	1.50	815.37
400um	Proposed	0.01	5.90	-3.69	-6.38	2.82	801.98
	Initial	-73.08	2.68	-21.39	-37.72	-	-
Avg Change (%)	Contest winner	-75.05	2.47	-12.78	-17.53	133.34	97.56
	Baseline	2.85	0.25	-9.47	-7.60	156.26	16.18

VII. Conclusions

This paper proposes two quadratic objectives to integrate timing information in the quadratic placement formulation. The quadratic objectives are smooth critical paths and balance load capacitance in the critical nets. Both techniques were applied incrementally through an operation called neutralization. The advantage of using neutralization is to optimize an existing solution through a quadratic formulation without knowing the technique that generated it. The techniques were integrated into a flow whose baseline is the techniques proposed by [7]. Experimental results show that the use of the quadratic formulation with the global view of the problem produced effective results alone, outperforming both the ICCAD winner and the baseline by more than 10% in WNS, on average. The proposed flow also outperforms the winner and [7] by 7.60% and 17.53% in TNS, respectively.

Acknowledgment

This work is partially supported by Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES) and by the National Council for Scientific and Technological Development (CNPq).

References

[1] M. V. Rossum, "MOS Device and Interconnects Scaling Physics," in *Advanced Nanoscale ULSI Interconnects: Fundamentals and Applications*. Springer New York, 2009, ch. 2.

[2] X. Yang, B.-K. Choi, and M. Sarrafzadeh, "A Standard-Cell Placement Tool for Designs with High Row Utilization." in *ICCD*. IEEE Computer Society, 2002.

[3] N. Viswanathan, G.-J. Nam, J. A. Roy, Z. Li, C. J. Alpert, S. Ramji, and C. Chu, "ITOP: Integrating Timing Optimization Within Placement," in *ISPD*. ACM, 2010.

[4] T. Kong, "A novel net weighting algorithm for timing-driven placement," in *ICCAD*, Nov 2002, pp. 172–176.

[5] R.-S. Tsay and J. Koehl, "An analytic net weighting approach for performance optimization in circuit placement," in *DAC*, 1991.

[6] A. Bock, S. Held, N. Kammerling, and U. Schorr, "Local search algorithms for timing-driven placement under arbitrary delay models," in *DAC*, June 2015, pp. 1–6.

[7] G. Flach, M. Fogaça, J. Monteiro, M. Johann, and R. Reis, "Drive strength aware cell movement techniques for timing driven placement," in *ISPD*, 2016.

[8] T. Luo, D. Newmark, and D. Z. Pan, "A new LP based incremental timing driven placement for high performance designs," in *DAC*, 2006, pp. 1115–1120.

[9] N. Viswanathan and C. C. N. Chu, "FastPlace: efficient analytical placement using cell shifting, iterative local refinement,and a hybrid net model," *TCAD*, vol. 24, 2005.

[10] A. Srinivasan, K. Chaudhary, and E. S. Kuh, "RITUAL: a performance driven placement algorithm for small cell ICs," in *ICCAD*, 1991.

[11] H. B. Bakoglu, *Circuits, interconnections, and packaging for VLSI*. Reading, Mass: Addison-Wesley Pub. Co., 1990.

[12] B. M. Riess and G. G. Ettelt, "SPEED: fast and efficient timing driven placement," in *ISCAS*, vol. 1, 1995.

[13] M.-C. Kim, J. Hu, J. Li, and N. Viswanathan, "ICCAD-2015 CAD contest in incremental timing-driven placement and benchmark suite," in *ICCAD*, Nov 2015, pp. 921–926.

[14] B. Hu and M. Marek-Sadowska, "Multilevel fixed-point-addition-based vlsi placement." *TCAD*, 2005.