A Unified Optimization Framework for Simultaneous Gate Sizing and Placement under Density Constraints

Jason Cong
Computer Science Department
University of California, Los Angeles
California NanoSystems Institute
Los Angeles, CA, United States
cong@cs.ucla.edu

John Lee
Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA, United States
lee@ee.ucla.edu

Guojie Luo
Computer Science Department
University of California, Los Angeles
Los Angeles, CA, United States
gluo@cs.ucla.edu

***Abstract*—A unified optimization framework is presented for simultaneous gate sizing and placement. These processes are unified using Lagrangian multipliers, which synchronize the efforts of the gate sizing and placement subproblems. As far as we know, this is the first work that formulates and solves the simultaneous gate sizing and placement under area density constraints, which are handled by the quadratic penalty method. We show that this rigorous framework results in an algorithm that is faster than separate iterations of gate sizing and placement steps, and leads to more robust results for a set of benchmarks.**

I. INTRODUCTION

The success of Moore's Law in the past four decades has resulted in designs with many millions of transistors, if not billions. This increase in the density of the transistors has also resulted in a tremendous increase in the power of each design. While the number of transistors increases by a factor of 2x every 18 months, the power that each transistor consumes decreases at a much slower pace.

From a design standpoint, this presents a problem. The increased power drives the need for increasingly effective algorithms, while the increased size makes these problems more complex. Nonetheless, there has been good progress in physical design over the past decade, with multilevel paradigms in circuit placement [3] and with Lagrangian relaxation methods in circuit sizing [5]. These algorithms scale well with the size of the problem and are algorithms that facilitate design at a large-scale.

This leads to a natural question – can circuit sizing and circuit placement be unified? There are heuristic techniques that can be used, but the problem may be too difficult to obtain an exact solution. It is well-known that the general placement problem is NP-hard [12]. In addition, the gate-level delay function, with placement and gate sizes as variables, is nonconvex [6]. These difficulties create challenges for simultaneous sizing and placement.

Delay minimization using simultaneous sizing and placement has been shown to be effective. In [6] the authors begin with the simultaneous optimization problem, and reduce the problem into smaller iterations of placement and sizing for the worst paths in the circuit. They show that this results in an improvement of 15% over sizing alone for a 0.35-µm process. In [7] the authors formulate an exact algorithm for the delay optimal mapping of trees, and extend the results to general circuit topologies using Lagrangian multipliers. Their results show that this performs better than an ad hoc net-weighting methodology, but overlap or area density constraints are not considered.

The authors of [8] consider the power minimization problem. The algorithm first uses iterations of placement and sizing to improve the slack vs. power tradeoff. The slacks are then used by a V_{th} assignment algorithm to minimize the leakage power of the design. They show significant improvement for a 65nm process.

In this paper we present a new unified optimization method for simultaneous sizing and placement. This algorithm is interesting from both an algorithmic and mathematical point of view for several reasons: Lagrangian multipliers are used to join the placement and sizing problems, acting as both net weights in the placement, and delay weightings in the sizing. The overlap constraints are handled using a single penalty factor. In practice, our algorithm is not slower than a single round of sizing and placement, thus allowing the solution of large-scale problems.

Our algorithm is distinct from previously referenced algorithms because it features all of the following:

- Area density constraints
- Power minimization
- Unified framework for sizing and placement

The contributions of this paper are:

- Area-density-aware simultaneous placement and sizing
- Scalable methodology for handling area density, timing, and power
- Results that compare separate iterations of sizing and Lagrangian multiplier-based, timing-driven placement with our unified optimization method

The remainder of this paper is organized as follows. In Section II we provide an example of the unified sizing and placement optimization, which motivates us to design a different approach. In Section III we describe the notations and mathematical formulation of the simultaneous optimization problem. In Section IV we present our unified optimization framework, which uses Lagrangian multipliers to provide a consistent objective for the placement and sizing subproblems. In Section V experimental results are presented to show the runtime and solution quality of our algorithm.

978-1-4244-9472-9/11/$26.00 ©2011 IEEE

II. A MOTIVATING EXAMPLE

A typical physical synthesis flow includes iterations of separate placement and gate sizing steps. Based on a wirelength-driven placement, iterations of density-aware gate sizing and timing-driven placement are performed to meet the timing constraints and minimize power consumption.

In Figure 1 we show the optimization results of iterations of gate sizing and placement on the circuit c1355 from the ISCAS-85 benchmarks [13]. The sum of wire and gate capacitances is used as an indirect measure of power consumption. The actual power value can be viewed as a weighed sum of the capacitances, and does not affect the following analysis in terms of optimization. In Figure 1 we can see that the total capacitance continues to decrease after several iterations. However, the iterations of full sizing and placement are time-consuming, and furthermore, each step of the separate sizing and placement ignores the objective of the other and slows down the convergence. Therefore, we develop a unified optimization for the simultaneous sizing and placement problem, which is faster and has a stronger mathematical foundation.

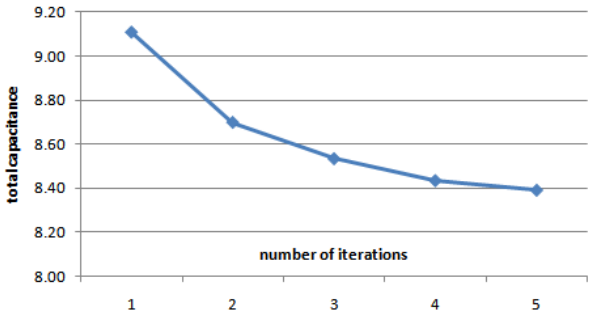

Figure 1 – Plots of power minimization results for c1355

III. MATHEMATICAL FORMULATION

The power-driven simultaneous gate sizing and placement optimization problem can be written as

$$\begin{aligned} \text{minimize} \quad & c_{wire}(x) + c_{gate}(s) \\ \text{subject to} \quad & A(x,s) \le C \\ & D(x,s) \le T \end{aligned} \tag{1}$$

where x is the vector of placement variables, and s is the vector of gate sizing variables. The box constraints, i.e., the fixed-outline placement constraints on x and upper and lower bounds on s, are omitted for simplicity.

The objective $c_{wire}(x) = \sum_{i=1}^{m} \alpha_i l_i(x)$ and $c_{gate}(s) = \sum_{i=1}^{n} \beta_i s_i$ are the weighted capacitances of wires and gates. This is a good proxy for the total power, as total capacitance is related to the power dissipation of the design. In the above, α_i is set as the per unit-length capacitance of net e_i, and set β_i as the capacitance of gate v_i for a minimum-sized gate. Improved models can also be considered by adding information about the switching activity and the leakage power. In this paper we shall assume that these parameters are given and focus on solving problem (1). The non-overlapping constraints are usually formulated as area density constraints $A(x,s) \le C$ in analytical placers [9].

The constraint $D(x,s) \le T$ is a shorthand notation for the delay constraints. Given a circuit where the set of primary inputs is PI, the set of primary outputs is PO, and the set of fan-ins of node j is $FI(j)$, the delay constraints can be written using the dynamic-programming formulation:

$$\begin{cases} t_i + D_{ij}(x,s) \le t_j & i \in FI(j) \\ t_i \equiv AT & i \in PI \\ t_j \le T & j \in PO \end{cases} \tag{2}$$

The minimum values of t that satisfy the conditions in (2) are the arrival times at each node.

Posynomial models are used to model the size vs. delay relations, and the Elmore delays are used for wire delay [11]. In other words, we assume that the gate delay is given as a function:

$$D_{ij}(x,s) = \left(a_i x_i^{\alpha_i} + \rho_r l_j(x)\right)\left(\sum_{k \in FO(j)} c_k x_k^{\beta_k} + \rho_c l_j(x)\right) + \rho_r \rho_c l_j^2(x) \tag{3}$$

where the coefficients α_i, β_k are fitted from the data. For simplicity, we use the half-perimeter wirelength for $l(x)$:

$$l_i(x) = \frac{1}{2}\max_{k,l \in \text{net}_i}\{x_j - x_k\} + \frac{1}{2}\max_{k,l \in \text{net}_i}\{y_j - y_k\} \tag{4}$$

The problem (1) does not have a convex formulation – it is not jointly convex in x and s. However, the delay function is convex in s for a fixed x and is convex in x for a fixed s.

IV. OPTIMIZATION FRAMEWORK

There are three levels to the optimization process. At the highest level, the density penalty μ is adjusted to find a density-feasible solution. The next level down updates Lagrangian multipliers λ to find a power-minimized solution that meets the delay constraints. At the lowest level, placement and sizing subproblems are solved efficiently for a given set of Lagrangian multipliers λ.

A. *Quadratic Penalty for Density Constraints*

Density and overflow are major considerations in problem (1) and have been an active research topic in the past decade [9]. In our work the density and overflow constraints are handled using an analytical framework [3] with an area penalty.

The first step is to convert the area density inequality constraints in problem (1) to equality constraints by adding filler cells [4]:

$$\begin{aligned} \text{minimize} \quad & c_{wire}(x) + c_{gate}(s) \\ \text{subject to} \quad & A(x,s) = C \\ & D(x,s) \le T \end{aligned} \tag{5}$$

The area density equality constraint is achieved by penalizing the objective function with a quadratic penalty term [10], where the penalized objective function is defined as

$$Q(\mu) = \min_{\{x,s|\ D(x,s)\le T\}} \left\{ c_{wire}(x) + c_{gate}(s) + \frac{\mu}{2}\left(A(x,s) - C\right)^2 \right\} \quad (6)$$

$Q(\mu)$ is computed for different values of μ. In theory, the area equality constraint will be satisfied as μ increases. This process is outlined in Algorithm 1, which solves problem (5). It is required that $\mu_{k+1} > \mu_k$, and we set the penalty update factor γ to 1.1 in the implementation.

Given $\mu^{(0)} > 0$, and a starting point $(x^{(0)}, s^{(0)})$;
For ($k = 0$; Area$(x^{(k)}, s^{(k)}) \neq C$; $k++$)
  Find a minimizer $(x^{(k+1)}, s^{(k+1)})$ to obtain $Q(\mu^{(k)})$, starting at $(x^{(k)}, s^{(k)})$; (using Algorithm 2)
  Choose new penalty factor $\mu^{(k+1)} = \gamma \cdot \mu^{(k)}$;
End For
Return $(x^{(k)}, s^{(k)})$;

Algorithm 1 – Quadratic penalty method for density constraints

B. *Lagrangian Method for Delay Constraints*

Algorithm 1 requires the computation of $Q(\mu)$, which solves the following delay constrained problem:

$$\begin{aligned} &\text{minimize} \quad c_{wire}(x) + c_{gate}(s) + \frac{\mu}{2}\left(A(x,s) - C\right)^2 \\ &\text{subject to} \quad t_i + D_{ij}(x,s) \le t_j \quad \forall (p_i, p_j) \in E_T \end{aligned} \quad (7)$$

We can handle the delay constraints by using the Lagrangian method [2], where the Lagrangian function with dual multipliers λ_{ij} is defined as:

$$\begin{aligned} L(x,s,t;\lambda) = {} & c_{wire}(x) + c_{gate}(s) + \frac{\mu}{2}\left(A(x,s) - C\right)^2 \\ & + \sum_{\forall j, i \in \text{fi}(j)} \lambda_{ij}\left(t_i + D_{ij}(x,s) - t_j\right) \end{aligned} \quad (8)$$

It was pointed out in [5] that λ should satisfy the flow (dual feasible) condition as in equation (9); otherwise the function $L(x,s,t;\lambda)$ is unbounded below.

$$\lambda^* \in \Omega \equiv \left\{ \lambda \;\middle|\; \sum_{i \in \text{fi}(j)} \lambda_{ij} = \sum_{k \in \text{fo}(j)} \lambda_{jk} \right\} \quad (9)$$

The auxiliary variables for delay constraints are eliminated when $\lambda \in \Omega$. The Lagrangian function is simplified as in equation (10), and the Lagrangian method is listed in Algorithm 2. We use a sequence of step sizes $\{\delta^{(k)}\}$ for the dual multiplier update that satisfies $\lim_{k\to\infty} \delta^{(k)} = 0$ and $\sum_{k=0}^{\infty} \delta^{(k)} = \infty$. We set $\delta_k = 1/(k+1)$ in implementation.

$$\begin{aligned} & L(x,s;\lambda) \equiv L(x,s,t;\lambda)\big|_{\lambda \in \Omega} \\ & = c_{wire}(x) + c_{gate}(s) + \frac{\mu}{2}\left(A(x,s) - C\right)^2 \\ & + \sum_{\forall (i,j), i \in \text{fi}(j)} \lambda_{ij} \cdot D_{ij}(x,s) + \sum_{p_i \in PI} \lambda_{ij} \cdot AT_i - \sum_{p_j \in PO} \lambda_{ij} \cdot RAT_j \end{aligned} \quad (10)$$

Given $\lambda^{(0)} \ge 0$, and a starting point $(x^{(0)}, s^{(0)})$;
For($k = 0$; $D(x^{(k)}, s^{(k)}) > T$; $k++$)
  Find a minimizer $(x^{(k+1)}, s^{(k+1)})$ of $L(x,s;\lambda^{(k)})$, starting at $(x^{(k)}, s^{(k)})$; (using Algorithm 3)
  Compute $\tilde{\lambda}_{ij}^{(k+1)} = \lambda_{ij}^{(k)} + \delta^{(k)}(-\text{slack}_{ij}(x,s))$;
  Update $\lambda^{(k+1)} = \text{Projection}_{\Omega \cap \{\lambda \ge 0\}}\left(\tilde{\lambda}^{(k+1)}\right)$;
End For
Return $(x^{(k)}, s^{(k)})$;

Algorithm 2 – Lagrangian method for delay constraints

Algorithm 2 requires the minimization of $L(x,s;\lambda)$ over x and s for a given dual multiplier λ. We use the block coordinate descent method [2] for this purpose, as shown in Algorithm 3. Since all the constraints are converted to either the quadratic penalty terms or the Lagrangian terms, the unconstrained placement subproblem $\min_x L(x,s^{(k)};\lambda)$ and the unconstrained sizing subproblem $\min_s L(x^{(k+1)},s;\lambda)$ can be solved efficiently. Algorithm 3 stops when the reduction of the Lagrangian function is smaller than ε, which is a user-defined stopping criterion. In the implementation we set ε to 0.001.

Given a starting point $(x^{(0)}, s^{(0)})$;
For ($k = 0$; $\left|L^{(k)} - L^{(k+1)}\right| > \varepsilon \left|L^{(k)}\right|$; $k++$)
  $x^{(k+1)} \leftarrow \min_x L(x, s^{(k)}; \lambda)$;
  $s^{(k+1)} \leftarrow \min_s L(x^{(k+1)}, s; \lambda)$;
End For
Return $(x^{(k)}, s^{(k)})$;

Algorithm 3 – Block descent method to minimize the Lagrangian function

V. EXPERIMENTAL RESULTS

Our unified optimization method was tested on the ISCAS-85 benchmarks [13]. They are synthesized to a 45 nm library [1], and gate delay and power for the continuous sizes are modeled in a least-square fit. The placement region is set to be a square with 20% whitespace, and the timing constraint is set to be 0.90X of the delay after one iteration of wirelength-driven placement and a subsequent gate sizing for delay minimization. The sizes of the benchmarks range from hundreds of gates to thousands of gates, and are run on a Quad Core Xeon 2GHz machine with 2GB memory. The statistics of the synthesized netlist are shown in the first three columns of Table 1.

For comparison, the separate Lagrangian multiplier-based gate sizing and timing-driven placement algorithms are implemented in a way that is similar to the unified optimization algorithm, i.e., by solving only the sizing problem (Problem (1) with x fixed), or the placement problem (Problem (1) with s fixed). Separate iterations of the gate-sizing algorithm, followed by timing-driven placement, were performed. In other words, a sizing is run to minimize power with timing constraints, and then a placement is run to minimize power with timing constraints. The power is

measured by the total gate capacitances and wire capacitances. This is used to compare the benefits of using a unified formulation to those of a separate formulation.

Experimental results are shown in Table 2 for the unified optimization and iterations of separate optimizations. The total capacitance, which is the sum of the gate capacitance and wire capacitance, is listed as well as the final slack. The runtime comparison of the unified optimization and the separate iterations is shown in Table 1.

Our unified optimization method can meet almost all timing constraints (9 out of 10) with 2% lower power than the separate iterations. Moreover, the runtime of the unified optimization is shorter than a single iteration of the sizing and placement (40% shorter on average), which demonstrates a clear advantage of using the unified optimization method. The runtime benefit comes from the shared Lagrangian multipliers that enable a global view of the joint sizing and placement problem, such that the sizing and the placement subproblems are optimizing a consistent objective function during the timing satisfaction process.

VI. CONCLUSIONS AND FUTURE WORK

We presented an algorithm that joins the gate sizing and cell placement methodologies into a unified framework. This method uses a penalty to manage the density and overlap constraints, and uses Lagrangian multipliers to manage the timing constraints. The algorithm performs well, with large runtime benefits.

Future work will include applying the unified optimization methodology to the simultaneous detailed placement and discrete sizing problem. Studies will be done to apply hierarchical methods to update the multipliers for large-scale designs.

ACKNOWLEDGMENT

This research is partially supported by Semiconductor Research Corporation (SRC) under task 1460.001. The authors would like to thank Prof. Lieven Vanderberghe for participating in the discussions.

REFERENCES

[1] Nangate Open Cell Library v1.2. Available from http://www.si2.org/openeda.si2.org/projects/nangatelib

[2] D. Bertsekas, "Constrained Optimization and Lagrange Multiplier Methods," Athena Scientific, Belmont, MA, 1996.

[3] T.F. Chan, J. Cong, and K. Sze, "Multilevel Generalized Force-directed Method for Circuit Placement," Proceedings of the 2005 International Symposium on Physical Design, pp.185-192, 2005.

[4] T.F. Chan, J. Cong, J.R. Shinnerl, K. Sze, and M. Xie, "mPL6: Enhancement Multilevel Mixed-Size Placement with Congestion Control," in Modern Circuit Placement, ed. G.-J. Nam and J. Cong, Springer Publishers, 2007.

[5] C.-P. Chen, C.C.N. Chu, and D.F. Wong, "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 7, pp. 1014-1025, 1999.

[6] W. Chen, C.-T. Hsieh, and M. Pedram, "Simultaneous Gate Sizing and Placement," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 19, no. 2, pp. 206-214, 2000.

[7] Y. Liu, R.S. Shelar, and J. Hu, "Delay-Optimal Simultaneous Technology Mapping and Placement with Applications to Timing Optimization," Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, pp.101-106, 2008.

[8] T. Luo, D. Newmark, and D.Z. Pan, "Total Power Optimization Combining Placement, Sizing and Multi-Vt through Slack Distribution Management," Proceedings of the 2008 Conference on Asia and South Pacific Design Automation, pp.352-357, 2008.

[9] G.-J. Nam and J. Cong, "Modern Circuit Placement : Best Practices and Results," Springer, New York, 2007.

[10] J. Nocedal and S.J. Wright, "Numerical Optimization 2nd ed.," Springer, 2006.

[11] T. Sakurai, "Approximation of Wiring Delay in MOSFET LSI," IEEE Journal of Solid-State Circuits, vol. 18, no. 4, pp. 418-426, 1983.

[12] N.A. Sherwani, "Algorithms for VLSI Physical Design Automation," Kluwer Academic Publishers, 1995.

[13] http://www.eecs.umich.edu/~jhayes/iscas/

Circuit	#inst	#net	Unified runtime (s)	Separate (5 iter) runtime (s)	Separate/Iter runtime (s)
Circuit432	118	158	35.35	147.42	29.48
c880	346	406	42.87	318.62	63.72
c1355	572	613	42.61	367.07	73.41
c499	586	627	40.63	495.75	99.15
c1908	696	729	48.03	602.97	120.59
c2670g	1041	1365	54.81	266.88	53.38
c3540g	1615	1665	131.05	1024.78	204.96
c5315g	2019	2214	127.23	1894.46	378.89
c7552g	2895	3151	192.45	3184.56	636.91
Circuit6288	3089	3121	309.42	1425.43	285.09
geomean			**76.49**	**643.21**	**128.64**

Table 1 – Runtime comparison between the unified optimization and separate iterations

Circuit	Unified				Separate (1st iter)				Separate (2nd iter)				Separate (3th iter)				Separate (4th iter)				Separate (5th iter)			
	Total (pF)	Gate (pF)	Wire (pF)	Slack (ns)	Total (pF)	Gate (pF)	Wire (pF)	Slack (ns)	Total (pF)	Gate (pF)	Wire (pF)	Slack (ns)	Total (pF)	Gate (pF)	Wire (pF)	Slack (ns)	Total (pF)	Gate (pF)	Wire (pF)	Slack (ns)	Total (pF)	Gate (pF)	Wire (pF)	Slack (ns)
Circuit432	1.61	0.98	0.63	-0.07	1.63	1.00	0.63	-0.08	1.65	1.01	0.64	-0.08	1.64	1.01	0.63	-0.08	1.62	1.00	0.62	-0.08	1.61	1.00	0.61	-0.08
c880	4.79	2.52	2.27	0.00	5.15	2.75	2.40	-0.33	4.93	2.63	2.30	0.00	4.89	2.57	2.32	-0.04	4.92	2.54	2.38	-0.22	4.91	2.52	2.38	-0.27
c1355	8.09	5.21	2.88	0.00	9.11	6.25	2.86	0.00	8.70	5.89	2.81	0.00	8.53	5.74	2.80	0.00	8.43	5.64	2.79	0.00	8.39	5.60	2.79	0.00
c499	7.93	5.28	2.64	0.00	8.73	6.15	2.58	-0.06	8.92	6.33	2.59	-0.03	8.72	6.15	2.57	-0.06	8.83	6.25	2.58	-0.05	8.80	6.19	2.61	0.00
c1908	6.03	3.07	2.96	0.02	6.51	3.61	2.90	-0.02	6.26	3.40	2.87	-0.06	6.25	3.38	2.87	0.00	6.22	3.35	2.87	0.00	6.24	3.37	2.87	0.01
c2670g	8.91	3.09	5.82	0.04	8.99	3.17	5.82	0.03	8.92	3.14	5.78	0.01	8.91	3.13	5.78	0.03	8.91	3.13	5.78	0.03	8.95	3.12	5.83	0.03
c3540g	11.03	4.31	6.72	0.00	12.54	4.50	8.04	-0.12	11.50	4.04	7.46	-0.04	11.05	3.93	7.12	-0.05	12.29	4.05	8.24	-0.40	12.32	4.08	8.25	-0.45
c5315g	22.57	7.69	14.88	0.02	23.45	8.83	14.62	0.00	22.70	8.08	14.61	-0.03	22.58	7.97	14.61	0.00	22.57	7.96	14.61	-0.03	22.61	7.99	14.62	-0.05
c7552g	26.95	9.23	17.72	0.01	27.89	10.50	17.40	-0.09	27.46	10.09	17.37	-0.10	27.38	10.02	17.36	-0.07	27.31	9.97	17.34	-0.05	27.30	9.96	17.34	-0.07
Circuit6288	23.32	13.39	9.93	0.01	23.07	10.61	12.46	-0.96	23.18	9.99	13.19	-1.12	23.53	10.03	13.50	-1.08	23.78	10.06	13.73	-1.04	23.80	10.04	13.76	-1.06
geomean	**9.16**	4.39	4.49	-	**9.70**	4.71	4.66	-	**9.48**	4.51	4.62	-	**9.39**	4.45	4.60	-	**9.49**	4.45	4.68	-	**9.48**	4.44	4.68	-

Table 2 – Experimental results of the unified optimization and separate iterations