A Faster Implementation of APlace *

Andrew B. Kahng and Qinke Wang

UCSD CSE Department, La Jolla, CA 92093-0404 USA

{abk,qiwang}@cs.ucsd.edu

ABSTRACT

APlace is a high quality, scalable analytical placer. This paper describes our recent efforts to improve APlace for speed and scalability. We explore various wirelength and density approximation functions. We speed up the placer using a hybrid usage of wirelength and density approximations during the course of multi-level placement, and obtain 2-2.5 times speedup of global placement on the IBM ISPD04 and ISPD05 benchmarks. Recent applications of the APlace framework to supply voltage degradation-aware placement and lens aberration-aware timing-driven placement are also briefly described.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: INTEGRATED CIRCUITS—*Design Aids*; J.6 [**Computer Applications**]: COMPUTER-AIDED ENGINEERING

General Terms

Algorithms, Design, Performance

Keywords

Analytical Placement, Scalability, Supply Voltage Degradation, Lens Aberration

1. INTRODUCTION

APlace is a general analytical placement engine developed at UCSD, originally implemented to study novel ideas of Naylor et al. [8]. This led to a flexible research framework, APlace1; our previous work explored the adaptability of APlace1 to a variety of contexts with good quality of results. E.g., the framework was extended to traditional wirelength-driven standard-cell placement in [2], and achieved good results with respect to both placed HPWL and routed final wirelength. We also extended the framework to perform congestion-directed placement, mixed-size placement, timing-driven placement and I/O-core co-placement [2]. Subsequent work dramatically improved the placer on a number of fronts for the ISPD-2005 placement contest. The resulting program, APlace2, achieved high solution quality and won the contest [3, 4]. The recent paper [4] shows that APlace2 results are better than the best published results till that time by 3%, 12%, 14%, and 6% for the IBM ISPD'04, IBM-PLACE 2.0, ICCAD'04, and ISPD'05 benchmark sets respectively.

However, APlace remains slower than recent academic placement tools, such as FastPlace [10] and mPL [1]. In this work, we continue to modify the implementation of APlace for speed and scalability. We explore various wirelength and density approximation functions, and speed up the placer with a hybrid usage of wirelength and density approximations during the multi-level placement algorithm. As of this writing, the global placement engine of APlace3 is 2-2.5 times faster than APlace2 on IBM ISPD04 and ISPD05 benchmarks.

In other recent work, we have successfully applied the APlace framework to perform supply voltage degradation-aware placement [5] and lens aberration-aware timing-driven placement [6]. We briefly describe each of these adaptations in this paper.

2. THE APLACE FRAMEWORK

APlace applies a top-down multi-level algorithm to provide better global optimization and scalability. Like most modern placement tools, APlace builds up a hierarchy of clusters before global placement to provide multiple levels of reduction (i.e., "coarsening") of the flat netlist. APlace also applies multiple levels of grids to facilitate cell spreading and provide different levels of relaxations of cell density constraints. For each level of clusters, a proper grid is adaptively determined according to the average cluster size; for the flat level, multiple levels of grids are applied from coarser grid to finer grid.

For each cluster and grid level, APlace solves a relaxation of the original *Global Placement* problem, with a reduced netlist defined by the clusters and relaxed density constraints defined by the grid; the solution of the previous relaxation is used as the initial solution of the next-level relaxation with more placable objects and a finer grid. We believe that such a hierarchy of optimizations gives our placer tremendous advantages in terms of runtime, scalability and placement quality.

The relaxed Global Placement problem is formulated as the following *constrained minimization problem*:

$$\begin{array}{ll} min & HPWL(\mathbf{x},\mathbf{y}) \\ s.t. & D_g(\mathbf{x},\mathbf{y}) = D_g \quad \text{for each bin} g \end{array} \tag{1}$$

where $(\mathbf{x},\mathbf{y})$ gives the center coordinates of placeable objects, $HPWL(\mathbf{x},\mathbf{y})$ is the total HPWL of the current placement, $D_g(\mathbf{x},\mathbf{y})$ is a density function that equals the total cell area in bin g, and D_g is the expected total area in bin g, which is usually a constant denoting the average cell area.

APlace applies smooth approximations of placement objectives and density functions and solves the constrained minimization problem using the augmented Lagrangian method and a Conjugate Gradient (CG) solver. I.e., we solve a sequence of unconstrained minimization problems of the form

$$\begin{array}{ll} min & WL(\mathbf{x},\mathbf{y}) + \sum_g \lambda_g (D_g(\mathbf{x},\mathbf{y}) - D_g) \\ & + \frac{1}{2\mu} \sum_g (D_g(\mathbf{x},\mathbf{y}) - D_g)^2 \end{array} \tag{2}$$

*Work partially supported by the MARCO Gigascale Systems Research Center and NSF MIP-9987678. A. B. Kahng is currently with Blaze DFM, Inc., Sunnyvale, CA 94089

ISPD'06, April 9–12, 2006, San Jose, California, USA.
ACM 1-59593-299-2/06/0004.

for a carefully constructed sequence of Lagrange multiplier estimates λ_g's and penalty parameter μ, and use the solution of the previous unconstrained problem as an initial guess for the next one.

3. WIRELENGTH FUNCTIONS

While placement quality is typically evaluated according to HPWL, this "linear wirelength" function can not be efficiently minimized. Smooth linear wirelength objectives have been proposed in many works. In this section, we examine three important wirelength approximation functions, LOG-SUM-EXP [8], GORDIAN-L [9] and L_p-NORM [7], and propose to use them in a hybrid way in order to speed up the placer without losing wirelength quality.

The LOG-SUM-EXP approximation of HPWL was first proposed in [8] and recently applied in mPL [1] and APlace. The horizontal wirelength of a net e is written as

$$WL(e) = \alpha \cdot log(\sum_i e^{x_i/\alpha}) + \alpha \cdot log(\sum_i e^{-x_i/\alpha}) \quad (3)$$

where α is a smoothing parameter: The LOG-SUM-EXP approximation is strictly convex, continuously differentiable and converges to HPWL as α converges to 0. In APlace, we adaptively modify the smoothing parameter α according to the grid spacing W_g, instead of using a small constant value. Thus, long nets (probably connecting cells in different bins) are "chosen to be minimized" and short nets (probably connecting cells in the same bin) are "ignored". Empirical results show that not only speedup of the placer, but also better global optimization and hence better placement quality, can be achieved in this way.

We are now able to compare the other two wirelength functions against the LOG-SUM-EXP approximation within the APlace framework. The method of GORDIAN-L is proposed in [9] to minimize a linear wirelength objective using iterated quadratic minimizations. The horizontal wirelength of a net e is formulated as

$$WL(e) = \sum_{i,j}(x_i - x_j)^2/\gamma_{i,j} \quad (4)$$

where the factor $\gamma_{i,j} = max\{r_0, |x_i - x_j|\}$, which is constantly updated, and r_0 is the minimum value that $\gamma_{i,j}$ can take in order to prevent overflow. However, in our implementation, we use r_0 to choose long nets for more accurate minimization, and we decide the value of r_0 according the grid spacing W_g.

Our empirical studies show that the value of r_0 significantly affects the placer performance. We have implemented GORDIAN-L using the clique net model with edge weight equal to $1/q^2$ for a q-pin net, and obtained results on, e.g., the IBM ISPD04 benchmark set. We observe that when the value of r_0 is small, although the GORDIAN-L function is closer to linear wirelength, this does not necessarily lead to a better result. A proper value of r_0 can dramatically reduce average WL increase from 25.9% when $r_0 = 0.1W_g$ to only 4.1% when $r_0 = 2W_g$. We have also compared the two functions using the more realistic IBM ISPD05 benchmark circuits, and have found that the smallest average wirelength increase is 3.6%.

Last, we have also implemented the L_p-NORM approximation of HPWL that was proposed in [7]. The horizontal wirelength of a net e is expressed as

$$WL(e) = (\sum_{i,j}(x_i - x_j)^p + \beta)^{\frac{1}{p}} \quad (5)$$

where β is the smoothing parameter and the approximation converges to HPWL as β converges to 0 and p converges to ∞. In APlace3, similar to with the other wirelength functions, we adaptively increase p with cluster and grid levels during the placement and decide the value of β according to grid spacing. The average wirelength increase of the L_p-NORM function is 1.5% on the IBM ISPD05 circuits.

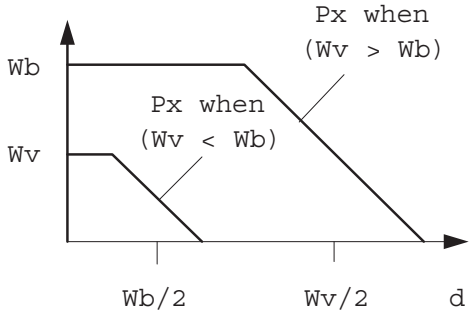

Figure 1: Overlap functions $P_x(g, v)$ when $W_v < W_b$ and $W_v > W_b$.

The very similar wirelength quality of the three approximation functions suggests that we may alternate them to achieve specific advantages. For a faster implementation of APlace, we may choose to apply GORDIAN-L with star net model during cluster-level placement to speed up the placer, but then apply the LOG-SUM-EXP approximation during flat-level placement to maintain the placement quality.

We also find out that most of the runtime of APlace is consumed by the line search algorithm of the CG solver. The wirelength function needs to be executed several times per CG iteration to decide the step length in the conjugate direction. Since the LOG-SUM-EXP approximation is very close to HPWL with fine grids and small smoothing parameters during flat-level placement, we only apply the LOG-SUM-EXP approximation to compute the wirelength derivatives for each iteration, but apply the actual HPWL function during line search, in order to reduce runtime.

4. DENSITY FUNCTIONS

Density function $D_g(\mathbf{x}, \mathbf{y})$ in Equation (1), which is the total cell area in bin g, can be expressed in the form

$$D_g(\mathbf{x}, \mathbf{y}) = \sum_v P_x(g, v) \cdot P_y(g, v) \quad (6)$$

where functions $P_x(g, v)$ and $P_y(g, v)$ denote the overlap between the grid bin g and cell v along the x and y directions, respectively. Figure 1 shows the trapezoidal-shaped $P_x(g, v)$ when the cell width W_v is larger than the bin width W_b and when $W_v < W_b$, as a function of the horizontal distance between cell and bin.

Naylor et al. [8] proposed to use a bell-shaped function $p_x(d)$ in quadratic form to approximate the overlap function as

$$p_x(d) = \begin{cases} W_v(1 - 2d^2/W_b^2) & (0 \le d \le W_b/2) \\ 2W_v(d - W_b)^2/W_b^2 & (W_b/2 \le d \le W_b) \end{cases} \quad (7)$$

for standard-cell placement. APlace2 follows this idea, but extends the function to handle large blocks for the purpose of mixed-size placement.

In our present work, we compare the quadratic function with two other approximations with Gaussian function and ERFC function, as captured respectively by

$$p_x(d) = min\{W_b, W_v\} \cdot e^{-4d^2/(max\{W_b, W_v\})^2} \quad (8)$$

and

$$p_x(d) = \begin{array}{l}(d-a)\cdot erfc((d-a)/\theta)/2- \\ (d-b)\cdot erfc((d-b)/\theta)/2\end{array} \quad (9)$$

where $0 \leq d \leq W_b$, $a = |W_v - W_b|/2$ and $b = (W_v + W_b)/2$. The second function provides a more flexible approximation of the step function in Figure 1, with smoothness controlled by the factor θ: The approximation converges to the step function when θ converges to 0. In our implementation, the value of θ is set to grid spacing W_g, which gives a smoother approximation for standard cells but a more accurate approximation for large macro blocks.

Benchmarking with IBM ISPD04 circuits shows roughly equal (within 1% error) wirelength quality for standard-cell placement using each of the three approximation functions. However, the Gaussian approximation leads to an average of 7.4% wirelength increase on IBM ISPD05 circuits, compared to APlace2, while the ERFC function slightly reduces the wirelength results.

For a faster implementation, we apply the ERFC approximation for flat-level placement following the same idea as in Section 3. I.e., we only use the ERFC approximation to compute the partial derivatives related to density terms for each CG iteration, but apply the actual cell density function during line search, in order to speed up the placer.

5. RECENT APPLICATIONS

In this section, we briefly describe recent applications of the APlace framework to supply voltage degradation-aware placement and lens aberration-aware timing-driven placement.

Supply voltage degradation aware placement. Increasingly significant supply voltage degradation in nanometer VLSI designs leads to system performance degradation and even malfunction. Existing techniques focus on design and optimization of power supply networks. However, supply voltage degradation occurs not only due to a poor power network design, but also due to unexpected large supply currents at specific time steps and specific locations, which suggests application of circuit design and placement techniques for reduction of such supply voltage degradation.

In [5] we present a supply voltage degradation-aware analytical placement method which relocates cells to reduce supply voltage degradation. We represent supply voltage degradation at an observed power supply node as function of supply currents and effective resistances in a DC power network, and integrate supply voltage degradation into the APlace framework. Experimental results show on average 20.9% improvement of worst-case voltage degradation and 11.7% improvement of average voltage degradation with only 4.3% wirelength increase.

Lens aberration-aware timing-driven placement. Aberrations can be described as the departure from ideal imaging induced by an imperfect lens system. Process variations due to lens aberrations are to a large extent systematic, and can be modeled for purposes of analyses and optimizations in the design phase. Because of lens aberrations, a cell placed at different locations will exhibit varying performance characteristics, which motivates a timing-driven placement method with awareness of lens aberration effects.

In [6], we first describe a novel aberration-aware timing analysis flow that integrates: (i) results of lithography simulation to measure critical dimension (CD) across the lens field, (ii) SPICE simulation-based library characterization that captures the impact of aberration-induced CD variation on timing and power, and (iii) placement information. In addition, we propose an aberration-aware timing-driven placement method that accounts for aberration-induced variations during placement. Our approach minimizes the design's clock cycle time under systematic aberration-induced variations. The placer is driven by models that capture the impact of lens position on timing-arc delays in cells, and by weighted-wirelength models. Essentially, we preferentially place cells that are setup-time critical at lens field locations where aberrations cause the cell delay to decrease. On average, the proposed placement technique reduces cycle time by $\sim 5\%$ at the cost of $\sim 2\%$ increase in wirelength.

6. CONCLUSIONS

Our recent efforts toward improved speed and scalability in APlace apply various wirelength and density approximations in a hybrid way during the multi-level placement algorithm. These methods achieve 2-2.5 times speedup of the placer without loss of wirelength quality, and continue the evolution of APlace beyond its original antecedents in the work of Naylor et al. [8]. Our ongoing work seeks to speed up the detailed placer by limiting available whitespace and reducing the number of possible locations within the permutation window for designs with low utilization ratio.

7. REFERENCES

[1] T. F. Chan, J. Cong and K. Sze, "Multilevel Generalized Force-directed Method for Circuit Placement," in *Proc. ACM/IEEE International Symposium on Physical Design*, 2005, pp. 185–192.

[2] A. B. Kahng and Q. Wang, "Implementation and Extensibility of an Analytic Placer," *IEEE Transactions on Computer-Aided Design* 24(5) (2005), pp. 734-747.

[3] A. B. Kahng, S. Reda, and Q. Wang, "APlace: A General Analytic Placement Framework," in *Proc. ACM/IEEE International Symposium on Physical Design*, 2005, pp. 233-235.

[4] A. B. Kahng, S. Reda, and Q. Wang, "Architecture and Details of a High Quality, Large-Scale Analytical Placer," in *Proc. Int. Conf. Computer Aided Design*, 2005, pp. 891-898.

[5] A. B. Kahng, B. Liu, and Q. Wang, "Supply Voltage Degradation Aware Analytical Placement," in *Proc. ACM/IEEE Intl. Conf. on Computer-Aided Design*, 2005, pp. 437-443.

[6] A. B. Kahng, C.-H. Park, P. Sharma and Q. Wang, "Lens Aberration-Aware Timing-Driven Placement," in *Proc. Design Automation and Testing in Europe*, 2006, to appear.

[7] A. A. Kennings and I. L. Markov, "Analytical Minimization of Half-Perimeter Wirelength", *Proc. IEEE/ACM Asia and South Pacific Design Automation Conf.*, Jan. 2000, pp. 179-184.

[8] W. Naylor, "Non-Linear Optimization System and Method for Wire Length and Delay Optimization for an Automatic Electric Circuit Placer," *US Patent* 6301693, 2001.

[9] G. Sigl, K. Doll and F. M. Johannes, "Analytical Placement: A Linear or a Quadratic Objective Function?, " in *Proc. ACM/IEEE Design Automation Conf.*, 1991, pp. 427-431.

[10] N. Viswanathan and C. Chu, "FastPlace: Efficient Analytical Placement Using Cell Shifting, Iterative Local Refinement and a Hybrid Net Model," in *Proc. ACM/IEEE International Symposium on Physical Design*, 2004, pp. 26–33.