APlace: A General Analytic Placement Framework *

Andrew B. Kahng, Sherief Reda and Qinke Wang
UCSD CSE Department, La Jolla, CA 92093-0114 USA
{abk,sreda,qiwang}@cs.ucsd.edu

ABSTRACT

We streamline and extend APlace, the general analytic placement engine based on ideas of Naylor et al. [7] and described in [3, 4, 5]. Previous work explored the adaptability of APlace to multiple contexts with good quality of results. For example, the framework was extended to traditional wirelength-driven standard-cell placement in [3, 5], achieving good results in placed HPWL and routed final wirelength. The framework was also extended to top-down multilevel placement, congestion-directed placement, mixed-size placement, timing-driven placement, I/O-core co-placement and constraint handling for mixed-signal contexts [3, 4, 5]. In this work, we have modified the implementation of APlace for speed and scalability. Improvements have been made in clustering, legalization and detailed placement strategies, as well as via a distributable solution framework for both global and detailed placement phases.

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, Multi-Level, Congestion, Mixed Size

1. INTRODUCTION

New analytical placement methods that simultaneously spread cells and optimize wirelength have recently received much attention from both academia and industry. In such methods, forces based on the current cell distribution are often applied to iteratively reduce cell overlaps. A novel and simple objective function for spreading cells over the placement area was proposed in the recent patent of Naylor et al. [7]. Combined with a wirelength objective function, it allows efficient simultaneous cell spreading and wirelength optimization using nonlinear optimization techniques. We previously developed APlace, an analytic placement framework, according to these ideas; the framework can be easily extended to cope with different contexts throughout the chip design process. Notably, the engine appears adaptable to multiple contexts with good quality of results.

Our previous work applied the general framework to traditional wire-length-driven standard-cell placement [3, 5], achieving good results with respect to (placed) half-perimeter wirelength and (routed) final wirelength. We also extended the basic framework to perform top-down multilevel placement, congestion-directed placement, timing-driven placement, mixed-size placement, I/O-core co-placement and constraint handling for mixed-signal contexts [3, 4, 5]. These works empirically demonstrated that the APlace analytic framework is a general and extensible platform for placement tasks across many aspects of physical implementation.

In this paper, we describe the theory and implementation details that underlie APlace. Beyond this basic framework of APlace, a competitive implementation addresses issues of speed and scalability via clustering, legalization and detailed placement, as well as a distributable solution framework in both global and detailed placement phases.

2. FORMULATIONS

A basic goal of placement is to minimize wirelength subject to the constraint that modules do not overlap. Therefore, the objective function for analytic placement historically includes two terms: a *density objective* to spread modules, and a *wirelength objective* to minimize wirelength.

2.1 Module Spreading

Standard-Cell Placement. To distribute cells evenly over the placement area, a generic strategy is to divide the placement region into grids and then attempt to equalize the total cell area in every grid. The straightforward "squared deviation" penalty for uneven cell distribution

$$Penalty = \sum_{Grid\ g} (TotalCellArea(g) - AverageCellArea)^2 \quad (1)$$

is neither smooth nor differentiable, and is hence difficult

*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'05, April 3–6, 2005, San Francisco, California, USA.
ACM 1-59593-021-3/05/0004.

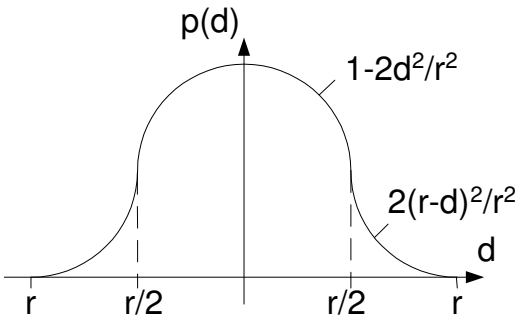

Figure 1: The bell-shaped function.

to optimize. APlace employs the technique proposed by Naylor et al. [7], who smooth this penalty function with a "bell-shaped" cell potential function. The placement area is divided into grids, each cell has a potential or influence with respect to nearby grids, and the placer seeks to equalize the total cell potential at each grid. For a cell c with center at $(CellX, CellY)$ and area A, the potential with respect to grid point $g = (GridX, GridY)$ is given by

$$\begin{aligned} &Potential(c, g) = \\ &C \cdot p(|CellX - GridX|) \cdot p(|CellY - GridY|) \end{aligned} \tag{2}$$

where

$$p(d) = \begin{cases} 1 - 2d^2/r^2 & (0 \le d \le r/2) \\ 2(d-r)^2/r^2 & (r/2 \le d \le r) \end{cases} \tag{3}$$

Here, $p(d)$ defines the bell-shaped function, which is illustrated in Figure 1; r controls the *radius* of any given cell's potential (range of interaction); and C is a normalization factor so that $\sum_g Potential(c, g) = A$, i.e., each cell has a total potential equal to its area. Then, the penalty function in Equation (1) is transformed to:

$$Penalty = \sum_{Grid\ g} \Big(\sum_{Cell\ c} Potential(c, g) - ExpPotential(g) \Big)^2 \tag{4}$$

where $ExpPotential(g) = TotalCellArea/NumGrids$ is the expected total potential at the grid point g.

Mixed-Size Placement. For standard-cell placement, the grid usually has a length greater than the average cell width, and the radius of cell's potential, r, is set to be a constant during placement. However, for mixed-size placement, the size range between large and small objects can be as large as a factor of 10,000, and the radius of influence of a cell's potential will need to change according to the cell's dimension. In particular, a larger block will have potential with respect to more grids.

After investigation of several possibilities, we have chosen to address the potential function for large macros in the following way. Suppose a macro block b has same width and height w. The radius or scope of this block's influence is $w/2 + r$, i.e., every grid within the distance of $w/2 + r$ from the block's center has a non-zero potential from this block. Moreover, the total potential of the block over all grids is equal to the block's area. Therefore, the function $p(d)$ in Equation (2) becomes

$$p(d) = \begin{cases} 1 - a * d^2 & (0 \le d \le w/2 + r/2) \\ b * (d - r)^2 & (w/2 + r/2 \le d \le w/2 + r) \end{cases} \tag{5}$$

where

$$\begin{aligned} a &= 4(w + r^2)/((w + 2r^2)(w + r)^2) \\ b &= 4/(w + 2r^2) \end{aligned} \tag{6}$$

so that the function $p(d)$ is continuous when $d = w/2 + r/2$. Also, the method can be easily extended for macros with different width and height.

Congestion-Directed Placement. For real-world contexts, it is necessary for the placer to deliver a 100% autoroutable solution. Thus, APlace also integrates congestion information into the objective, via the Kahng-Xu bend-based congestion estimation method [6]. If a particular grid is determined to be congested (resp. uncongested), the expected total cell potential of the grid in Equation (4) is reduced (resp. increased) accordingly. The sum of expected area potential over all grids is kept constant, and equal to the total cell area. Specifically, expected cell potential is adjusted as follows:

$$ExpPotential(g) \propto 1 + \gamma \left(1 - 2 \frac{Congestion(g)}{\max_g \{Congestion(g)\}} \right) \tag{7}$$

where γ is the congestion adjustment factor and decides the relative importance of congestion-directed placement.

2.2 Wirelength Minimization

While wirelength and overall placement quality is typically evaluated according to half-perimeter wirelength, this "linear wirelength" function can not be efficiently minimized. The approach of Naylor et al. [7], which is implemented in APlace, uses a log-sum-exp method to capture the linear half-perimeter wirelength while simultaneously obtaining the desirable characteristic of continuous differentiability. The log-sum-exp formula picks the most dominant terms among pin coordinates. For a net t with pin coordinates $\{(x_1, y_1),\ (x_2, y_2),\ \ldots\ (x_n, y_n)\}$, the wirelength objective is

$$\begin{aligned} WL(t) = \quad & \alpha \cdot (ln(\textstyle\sum e^{x_i/\alpha}) + ln(\sum e^{-x_i/\alpha})) + \\ & \alpha \cdot (ln(\textstyle\sum e^{y_i/\alpha}) + ln(\sum e^{-y_i/\alpha})) \end{aligned} \tag{8}$$

where α is a smoothing parameter. $WL(t)$ is strictly convex, continuously differentiable and converges to $HPWL(t)$ as α converges to 0.

The APlace analytic placer combines the two objectives and optimizes the function

$$WLWeight * TotalWL + DensityWeight * DensityPenalty \tag{9}$$

Here, the density term drives cell spreading and changes with the current cell distribution. The wirelength term draws connected components back toward each other.

3. IMPLEMENTATION

We now describe several basic implementation details of APlace.

3.1 Conjugate Gradient Optimizer

APlace uses the conjugate gradient method to search the non-constrained minimum of the high-dimensional objective function of Equation (9). In general, the conjugate gradient method finds the minimum by executing a series of *line minimizations*. A line minimization corresponds to one-dimensional function minimization along some search direction. Every iteration, a conjugate direction is first computed

based on the current gradients and the gradients from the last iteration, so that when the function is quadratic and the line search finds the minimum along the direction exactly. And then, a Golden Section search method is used to find the step length. Finally, the result of one line minimization is used as the starting point for the next line minimization.

3.2 Control Factors

Important control parameters within APlace are weights of the wirelength and density objectives. Intuitively, a larger wirelength weight will draw cells together and prevent them from spreading out, while a larger density penalty weight will spread the cells out (without attention to wirelength). These controls are managed by keeping the density weight fixed at some constant, and setting the wirelength weight to be large at the outset, but then decreasing this weight (by a factor of two, or a smaller factor near the final placement) whenever the conjugate gradient optimizer slows down and a stable solution emerges. After every weight change, the conjugate gradient optimizer is used to compute a new stable state wherein cells are distributed more evenly but wirelength is larger.

The number of grid nodes is also an important control knob for APlace. The number of grid nodes increase during the whole placement process. Coarser grids at the beginning spread out the cells faster, while finer grids at the final stages help to reach a more even distribution.

3.3 Top-Down Multi-Level Algorithm

Another aspect of the APlace tool is its integration of top-down multi-level processing according to a clustering hierarchy. For each level in the cell/cluster hierarchy, a coarse grid is determined by the average cluster size. To compute the density penalty, a cluster is essentially regarded as a macro cell with area equal to its total cell area; to compute the wirelength estimate, every cell is assumed to be located at the center of its cluster. The multilevel technique affords additional scalability.

3.4 Fixed Blocks and Obstacles

Fixed blocks and obstacles provide a pre-defined potential landscape for APlace, which prevents modules from passing over "mountains" to the whole placement area. To solve this problem, we apply the technique of landscape smoothing proposed in [1]. Suppose the average value of the normalized potential landscape is denoted as $\overline{p}$. A smooth landscape is defined by a specified smoothing factor δ as follows:

$$p'(x,y) = \begin{cases} \overline{p} + (p(x,y) - \overline{p})^{\delta} & if\ p(x,y) \geq \overline{p} \\ \overline{p} - (\overline{p} - p(x,y))^{\delta} & if\ p(x,y) \leq \overline{p} \end{cases} \quad (10)$$

where $\delta \geq 1$. When δ is decreased step by step from a large number, say 10, to 1, a series of smoothed potential landscape is generated. A potential landscape generated from a larger δ exhibits a smoother terrain surface, and a potential landscape generated from a smaller δ exhibits a more rugged terrain surface.

3.5 Legalization

The placement results of APlace have module overlaps and need to be legalized. A simplified Tetris [2] legalization algorithm is implemented in APlace. The Tetris legalization is applied after global placement: cells are sorted according to their vertical coordinates, and then for each cell from left to right the current nearest available position is found. This greedy algorithm is very fast, with negligible running time compared to that of global placement, and increases wirelength by about 4% for IBM-PLACE v2.0 circuits.

For mixed-size placement, cells are sorted based on a combination of vertical coordinate and width, so that larger blocks may be fixed at a position ahead of nearby small cells. We also scale the module positions to the left side by a fixed factor so that (1) modules will not be pushed outside the placement region, and (2) horizontal overlaps among macros can be properly resolved by the legalization.

3.6 I/O-Core Co-Placement

Fixed IO pads are not necessary for APlace. For peripheral I/O pads, APlace updates the positions of I/O pads on the boundary during each iteration so that total wirelength is minimized. For area-array I/Os, APlace can simultaneously place I/Os and core cells: I/Os are spread over the placement area, in the same way and at the same time as core cells. The results in [3] show that I/Os can be distributed fairly evenly over the placement area, without serious impact on total wirelength, core cell distribution, or running time.

4. CONCLUSIONS

Results reported in [3, 4, 5], as well as extensibility to such contexts as area-array IO co-placement, mixed-signal constrained placement, etc. suggest that APlace analytic engine has strong extensibility and can be competitive in a wide variety of contexts. Our recent efforts toward improved speed and scalability span clustering, legalization and detailed placement facets of the tool, as well as the use of distributed computing resources. On the research side, our current efforts include: (1) extension of the placer to power or IR drop aware placement; and (2) extension to 3D and thermal-aware placement.

5. REFERENCES

[1] J. Gu and X. Huang, "Efficient Local Search with Search Space Smoothing: A Case Study of the Traveling Salesman Problem (TSP)", *IEEE Trans. Systems, Man and Cybernetics* 24(5) (1994), pp. 728-735.

[2] D. Hill, "Method and System for High Speed Detailed Placement of Cells within an Integrated Circuit Design", *US Patent* 6370673, April 2002.

[3] A. B. Kahng and Q. Wang, "Implementation and Extensibility of an Analytic Placer", *Proc. Int. Symp. Physical Design*, 2004, pp. 18-25.

[4] A. B. Kahng and Q. Wang, "An Analytic Placer for Mixed-Size Placement and Timing-Driven Placement", *Proc. Int. Conf. Computer Aided Design*, 2004, 565-572.

[5] A. B. Kahng and Q. Wang, "Implementation and Extensibility of an Analytic Placer", *IEEE Trans. Computer Aided Design*, to appear.

[6] A. B. Kahng and X. Xu, "Accurate Pseudo-Constructive Wirelength and Congestion Estimation", *Proc. ACM Int. Workshop on System-Level Interconnect Prediction*, April 2003, pp. 61-68.

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