

mPL6: Enhanced Multilevel Mixed-Size Placement

Tony F. Chan,[†] Jason Cong, Joseph R. Shinnerl, Kenton Sze,[†] and Min Xie

UCLA Computer Science Department {cong,romesis,shinnerl,xie}@cs.ucla.edu

[†] UCLA Mathematics Department {chan,nksze}@math.ucla.edu

ABSTRACT

The multilevel placement package mPL6 combines improved implementations of the global placer mPL5 (ISPD05) and the XDP legalizer and detailed placer (ASPDAC06). It consistently produces robust, high-quality solutions to difficult instances of mixed-size placement in fast and scalable run time. Best-choice clustering (ISPD05) is used to construct a hierarchy of problem formulations. Generalized force-directed placement guides global placement at each level of the cluster hierarchy. During the declustering pass from coarsest to finest level, large movable objects are gradually fixed in positions without overlapping with one another. This progressive legalization of large objects during continuous optimization supports determination of a completely overlap-free configuration as close as possible to the continuous solution. Various discrete heuristics are applied to this legalized placement in order to improve the final wirelength.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids—*placement and routing*; G.4 [Mathematical Software]: Algorithm Design and Analysis; J.6 [Computer-Aided Engineering]: Computer-Aided Design

General Terms

Algorithms, Design

Keywords

Mixed-Size Placement, Legalization, Multilevel Optimization, Force-Directed Placement, Helmholtz Equation

1. INTRODUCTION

mPL6 consists of three basic ingredients: global placement by multilevel nonlinear programming [8], discrete graph-based macro legalization and greedy standard-cell legalization [10], and detailed placement [10]. It is designed for speed and scalability, low wirelength results, adaptability to complex constraints, and robustness under low white space. Compared to the 2005 implementation [7], the main improvements to mPL6 are as follows.

- (i) improved clustering by the “best-choice” heuristic [1];

- (ii) 2× reduction in the number of levels of clusters;
- (iii) more aggressive weighting of wirelength relative to overlap removal during optimization at each level;
- (iv) a faster single-V-cycle iteration flow;
- (v) gradual determination of the locations of large objects earlier in the multilevel flow;
- (vi) density-sensitive legalization and detailed placement;
- (vii) improved handling of unconnected filler cells supporting convergence to nonuniform module-area distributions.

Given a weighted hypergraph-netlist circuit representation $H = (V, E)$, mPL6 formulates constrained placement as a nonlinear programming problem of weighted wirelength minimization subject to generalized density constraints. For computational modeling, the placement region R is divided into a regular bin grid. Let \bar{x} denote an arbitrary location in R , let x denote the vector holding cells’ current locations in R , and let x_i denote the location of the i th cell (x_i is also a vector, with n_d components, n_d equal to the number of spatial dimensions — 2 or 3). Constraint values d_i are scalar fields over R that need be determined only to the resolution appropriate for a given level of hierarchy. Symbolically, we express placement as follows.

$$\begin{aligned} \min_x \quad & f_w(x) = \sum_{\text{nets } e} w(e)\ell(e) \\ & = \sum_{\text{nets } e} w(e) \left\{ \sum_{k=1}^{n_d} \max_{\text{cells } i,j \in e} |(x_i)_k - (x_j)_k| \right\}. \end{aligned} \quad (1)$$

$$\begin{aligned} \text{subject to} \quad & \max_{\bar{x} \in R} d_i(x) \leq u_i \\ & \text{for } i \in \{\text{overlap, routability, temperature, } \dots\} \end{aligned}$$

The net weights $w(e)$ can be chosen dynamically to capture a wide variety of objectives, including timing performance. Function $\ell(e)$ is an estimate of the wirelength of net e . The u_i are fixed upper bounds.

The functions w , ℓ , and d_i are continuous but not necessarily smooth. Some facts from partial differential equations (PDE) are used to obtain smooth reformulations of the constraints. The Poisson equation associates a smooth, scalar potential function $\phi(\bar{x})$ with a given density function $d(\bar{x})$:

$$\Delta\phi(\bar{x}) \equiv \sum_k \frac{\partial^2 \phi}{\partial \bar{x}_k^2}(\bar{x}) = d(\bar{x}), \quad \bar{x} \in R. \quad (2)$$

As observed by Eisenmann and Johannes [11], the Poisson equation applies generally to a wide variety of “supply-and-demand” formulations of placement constraints. Although

density gradients ∇d_i often do not exist, density-balancing forces $\nabla\phi$ always exist and can be calculated in fast linear time by good numerical solvers [15].

2. GLOBAL PLACEMENT

Global placement in mPL6 is based on multilevel optimization (Figure 1): recursive aggregation followed by interleaved optimization and disaggregation at every level [3, 9]. The multilevel hierarchy is built by best-choice clustering [1]; the target ratio of the number of nodes at each level to its adjacent coarser level is set to $4\times$. Intralevel optimization, known as *relaxation*, is by generalized force-directed placement [8], described below. Disaggregation is called *interpolation* and is based on ideas from Algebraic Multigrid [5, 6]. Multilevel optimization strongly supports (i) scalability and parallelizability; (ii) correct handling of complex constraints, including timing, routability, heat dissipation, noise, etc.; (iii) the incorporation of multiple, diverse, and complementary optimization heuristics; (iv) adaptability to rapidly changing formulations of multiple objectives and constraints.

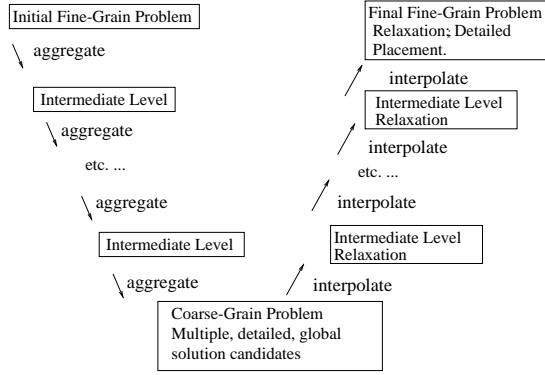


Figure 1: Multilevel Optimization V-Cycle and Iterated Multilevel Flow

mPL6’s approach to placement generalizes the force-directed framework of Eisenmann and Johannes in two ways [8]. First, mPL6 incorporates force-directed placement within a multilevel-placement engine as intralevel relaxation. This approach leads to improvement in both scalability and solution quality. Second, mPL6 reformulates force-directed placement within a systematic nonlinear-programming model. This reformulation gives a systematic means of scaling density-balancing forces before combining them with the wirelength gradients and removes the need for extensive ad-hoc tuning. A brief overview of the reformulation is given below. For further details, see the paper on mPL5 [8].

At each level of the cluster hierarchy, placement objectives and constraints are approximated by smooth functions. A bounding-box weighted half-perimeter wirelength objective is approximated by the log-sum-exp model [16, 13]

$$W(\mathbf{x}, \mathbf{y}) = \gamma \sum_{\text{nets } e \in E} \left(\log \sum_{\text{nodes } v_k \in e} \exp(x_k/\gamma) + \log \sum_{v_k \in e} \exp(-x_k/\gamma) + \log \sum_{v_k \in e} \exp(y_k/\gamma) + \log \sum_{v_k \in e} \exp(-y_k/\gamma) \right),$$

where \mathbf{x} and \mathbf{y} denote vectors of cell x - and y -coordinates. The smaller the parameter γ , the more accurate the approximation. Area-density constraints are imposed separately in each rectangular bin of a uniform grid laid over the placement region. Let D_{ij} denote the cell-area density of bin B_{ij} and let K denote total cell area divided by the total placement area. The area-density constraints are initially expressed simply as $D_{ij} = K$ over all bins B_{ij} . Viewing the D_{ij} as a discretization of a continuous density function $d(x, y)$ defined at points $(x, y) \in R$, these constraints are smoothed by approximating d by the solution ψ to the Helmholtz equation

$$\begin{cases} \Delta\psi(x, y) - \epsilon\psi(x, y) = d(x, y), & (x, y) \in R \\ \frac{\partial\psi}{\partial\nu} = 0, & (x, y) \in \partial R \end{cases} \quad (3)$$

where $\epsilon > 0$, ν is the outer unit normal, ∂R is the boundary of the placement region R , $d(x, y)$ is the continuous density function at a point $(x, y) \in R$, and Δ is the Laplacian operator $\Delta \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. The smoothing operator $\Delta_\epsilon^{-1}d(x, y)$ defined by solving (3) is well defined, because (3) has a unique solution for any $\epsilon > 0$. As the solution of (3) has two more derivatives than $d(x, y)$ (e.g., [12]), ψ is a smoothed version of d . Discretized versions of (3) can be solved rapidly by fast numerical multilevel methods. Recasting the density constraints as a discretization of ψ gives the nonlinear programming problem

$$\min W(\mathbf{x}, \mathbf{y}) \quad \text{s.t.} \quad \psi_{ij} = -K/\epsilon, \quad 1 \leq i \leq m, 1 \leq j \leq n,$$

where the ψ_{ij} are obtained by solving (3) with the discretization defined by the given bin grid. Interpolation from the adjacent coarser level defines a starting point. This nonlinear-programming problem is solved by the Uzawa iterative algorithm [2], which does not require second derivatives or large linear-system solves.

When available white space exceeds 2% of the area of placement region, unconnected artificial “filler” cells are added to underutilized regions in order to allow the given, interconnected cells to assume non-uniform configurations. The formulation of both the initial placement and subsequent handling of these filler cells is still under investigation.

3. LEGALIZATION

mPL6 combines top-down hierarchical macro legalization and flat, post-global-placement legalization of standard cells. Following iterative improvement at each cluster level, all overlap among movable macros larger than approximately 25 times the average cluster area is removed, and these large macros are subsequently held fixed during iterative refinement at finer levels. However, all movable macros are allowed to move at subsequent levels during macro legalization at those levels. On test cases with low white space (e.g., [18]), this gradual macro legalization is observed to reduce final wirelength by 10%–15%.

The method of removing overlap among macros is based on a constraint-graph formulation [10] similar to those used in floorplanning. Two directed acyclic graphs are constructed, G_h representing horizontal adjacency (x -direction) and G_v representing vertical adjacency (y -direction). Vertices in the graphs are macros in a given global placement; edges represent relative order along a coordinate direction. Edge e_{ij} in G_h indicates that macro v_i lies to the left of v_j ; pairs of nearby macros are constrained not to overlap by either

an edge in G_h or an edge in G_v but not both. As in timing analysis, the edges are weighted to facilitate calculation of critical paths, i.e., sequences of adjacent macros whose combined total length is maximal. Iterations in the macro legalization proceed by moving edges between G_h and G_v in a way that reduces critical path lengths until all macros fit in the given placement region. E.g., moving an edge from G_h to G_v allows adjacent macros to overlap in the horizontal direction but forces them not to overlap in the vertical direction, thus shortening the longest path in G_h and possibly increasing some path in G_v . Edges to move are selected as cut sets from min-cut partitioning on the zero-slack subgraphs of G_h and G_v . These subgraphs consist of all modules and edges occurring in paths exceeding the given core-region width or length, respectively.

Once legal, non-overlapping relative x and y orderings of all large macros at a given cluster level has been determined, the actual locations of these macros are determined by displacement-minimizing linear programming which preserves the given x and y orderings.

Legalization of standard cells and small macros proceeds only on the finest level, after global placement, and after all large-macro locations have been determined. To legalize these remaining movable small objects (collectively called “cells”), a greedy technique similar to that in [14] is applied, but with both front-end and back-end contours of placed objects maintained. Cell displacement costs are a combination of scaled half-perimeter wirelength increase and spatial displacement from their positions in the global placement.¹ Depending on cells’ relative locations and the order in which they are legalized, some subregions between fixed objects may not have sufficient available space for the cells assigned to them during this step. In this case, network-flow-based cell redistribution [17, 4] is applied to even out the area density between different chip subregions, with dynamic programming used to select cells for movement between subregions.

4. DETAILED PLACEMENT

Detailed placement begins only once a strictly legal configuration of macros and cells is obtained. Its objective is the reduction of scaled half-perimeter wirelength over both macros and cells. We apply window-based cell swapping to further reduce wirelength. All the cell permutations within the window are examined, and the one giving the shortest scaled wirelength is accepted. The window will be slid by half of its width after the swapping within it is done.

5. SUMMARY

Compared to the 2005 implementation of mPL6 on the ISPD 2005 suite, the current, 2006 implementation produces gains of approximately 10–30% when both placers are run in default mode. The 2005 contest results for mPL6 were obtained only after instance-specific parameter tuning. The results are shown in Table 1 below.

¹For the ISPD 2006 placement contest entry, the wirelength scaling incorporates runtime and the 60% bin-density-utilization target.

circuit	mPL6-2006		mPL6-2005 WL	
	DP HPWL	cpu	Default	Tuned
adaptecl	7.87E+07	3066	1.14	1.07
adaptecl2	9.65E+07	3518	1.04	1.03
adaptecl3	2.21E+08	11700	1.28	1.14
adaptecl4	2.00E+08	13643	1.03	1.02
bigblue1	9.87E+07	4100	1.05	1.02
bigblue2	1.54E+08	11857	1.30	1.17
bigblue3	3.54E+08	19845	1.24	1.09
bigblue4	8.46E+08	40901	1.16	1.09

Table 1: Improvement in mPL6-2006 (default mode) vs. mPL6-2005 (default and hand-tuned). The run time also has been reduced over 3×.

6. ACKNOWLEDGMENTS

Financial support from SRC Contract 2003-TJ-1091, ONR Contract N00014-03-1-0888, and NSF Awards ACI-0072112, CCF-0430077, and CCF-0528583 is gratefully acknowledged.

7. REFERENCES

- [1] C. Alpert, A. B. K. G.-J. Nam, S. Reda, and P. Villarrubia. A semi-persistent clustering technique for vlsi circuit placement. In *Proc. Int’l Symp. on Phys. Design*, pages 200–207, Apr 2005.
- [2] K. Arrow, L. Huriwicz, and H. Uzawa. *Studies in Nonlinear Programming*. Stanford University Press, Stanford, CA, 1958.
- [3] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comp.*, 19:23–56, 1986.
- [4] U. Brenner, A. Pauli, and J. Vygen. Almost Optimum Placement Legalization by Minimum Cost Flow and Dynamic Programming. In *Proc. Int’l Symp. on Phys. Design*, pages 2–8, 2004.
- [5] W. Briggs, S. McCormick, and V. Henson. *A Multigrid Tutorial*. SIAM, Philadelphia, second edition, 2000.
- [6] T. Chan, J. Cong, T. Kong, J. Shinnerl, and K. Sze. An enhanced multilevel algorithm for circuit placement. In *Proc. Int’l Conf. on Comp.-Aided Design*, San Jose, CA, Nov 2003.
- [7] T. Chan, J. Cong, M. Romesis, J. Shinnerl, K. Sze, and M. Xie. mPL6: a robust multilevel mixed-size placement engine. In *Proc. Int’l Symp. on Phys. Design*, pages 227–229, Apr 2005.
- [8] T. Chan, J. Cong, and K. Sze. Multilevel generalized force-directed method for circuit placement. In *Proc. Int’l Symp. on Phys. Design*, 2005.
- [9] J. Cong and J. Shinnerl, editors. *Multilevel Optimization in VLSICAD*. Kluwer Academic Publishers, Boston, 2003.
- [10] J. Cong and M. Xie. A robust detailed placement for mixed-size IC designs. In *Proc. Asia South Pacific Design Automation Conf.*, pages 188–194, 2006.
- [11] H. Eisenmann and F. Johannes. Generic global placement and floorplanning. In *Proc. 35th ACM/IEEE Design Automation Conference*, pages 269–274, 1998.
- [12] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2002.
- [13] A. Kahng and Q. Wang. Implementation and extensibility of an analytic placer. In *Proc. Int’l Symp. on Phys. Design*, pages 18–25, 2004.
- [14] A. Khatkhate, C. Li, A. R. Agnihotri, S. Ono, M. C. Yildiz, C.-K. Koh, and P. H. Madden. Recursive Bisection Based Mixed Block Placement. In *Proc. Int’l Symp. on Phys. Design*, 2004.
- [15] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 1994.
- [16] W. Naylor et al. Nonlinear optimization system and method for wire length and delay optimization for an automatic electric circuit placer, Oct. 2001.
- [17] J. Vygen. Algorithms for large-scale flat placement. In *Proc. 34th ACM/IEEE Design Automation Conference*, pages 746–751, 1997.
- [18] <http://www.faraday-tech.com/structuredasic/download.html>.