Title:	Circuit Placement
Name:	Andrew A. Kennings[1], Igor I. Markov[2]
Affil./Addr. 1:	Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada
Affil./Addr. 2:	Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA
Keywords:	EDA; Netlist; Layout; Min-cut placement; Min-cost maxflow; Analytical placement; Mathematical programming
SumOriWork:	2000; Caldwell, Kahng, Markov 2002; Kennings, Markov 2006; Kennings, Vorwerk

Circuit Placement

Andrew A. Kennings[1], Igor I. Markov[2]

[1] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada

[2] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

Years and Authors of Summarized Original Work

2000; Caldwell, Kahng, Markov
2002; Kennings, Markov
2006; Kennings, Vorwerk

Keywords

EDA; Netlist; Layout; Min-cut placement; Min-cost maxflow; Analytical placement; Mathematical programming

Problem Definition

This problem is concerned with efficiency determining constrained positions of objects while minimizing a measure of interconnect between the objects, as in a physical layout of integrated circuits, commonly done in 2-dimensions. While most formulations are NP-hard, modern circuits are so large that practical algorithms for placement must have near-linear runtime and memory requirements, but not necessarily produce optimal solutions. While early software for circuit placement was based on Simulated Annealing, research in algorithms identified more scalable techniques which are now being adopted in the Electronic Design Automation industry.

One models a circuit by a hypergraph $G_h(V_h, E_h)$ with (I) vertices $V_h = \{v_1, \ldots, v_n\}$ representing logic gates, standard cells, larger modules, or fixed I/O pads

and (ii) $E_h = \{e_1, \ldots, e_m\}$ representing connections between modules. Every incident pair of a vertex and a hyperedge connect through a pin for a total of P pins in the hypergraph. Each vertex $v_i \in V_h$ has width w_i, height h_i and area A_i. Hyperedges may also be weighted. Given G_h, circuit placement seeks center positions (x_i, y_i) for vertices that optimize a hypergraph-based objective subject to constraints (see below). A placement is captured by $\mathrm{x} = (x_1, \ldots, x_n\}$ and $\mathrm{y} = (y_1, \ldots, y_n\}$.

Objective Let C_k be the index set of the hypergraph vertices incident to hyperedge e_k. The total halfperimeter wirelength (HPWL) of the circuit hypergraph is given by $(\mathrm{HPWL}(G_h) = \sum_{e_k \in E_h} \mathrm{HPWL}(e_k) = \sum_{e_k \in E_h} [\max_{i,j \in C_k} |x_i - x_j| + \max_{i,j \in C_k} |y_i - y_j|]$.

HPWL is piece-wise linear, separable in the x and y directions, convex, but not strictly convex. Among many objectives for circuit placement, it is the simplest and most common.

Constraints

1. **No overlap.** The area occupied by any two vertices cannot overlap; i. e., either $|x_i - x_j| \geq \frac{1}{2}(w_i + w_j)$ or $|y_i - y_j| \geq \frac{1}{2}(h_i + h_j), \forall v_i, v_j \in V_h$.
2. **Fixed outline.** Each vertex $v_i \in V_h$ must be placed entirely within a specified rectangular region bounded by $x_{\min}(y_{\min})$ and $x_{\max}(y_{\max})$ which denote the left (bottom) and right (top) boundaries of the specified region.
3. **Discrete slots.** There is only a finite number of discrete positions, typically on a grid. However, in large-scale circuit layout, slot constraints are often ignored during *global placement*, and enforced only during *legalization* and *detail placement*.

Other constraints may include alignment, minimum and maximum spacing, etc. Many placement techniques temporarily relax overlap constraints into density constraints to avoid vertices clustered in small regions. A $m \times n$ regular bin structureB is superimposed over the fixed outline and vertex area is assigned to bins based on the positions of vertices. Let D_{ij} denote the density of bin $B_{ij} \in B$, defined as the total cell area assigned to bin B_{ij} divided by its capacity. Vertex overlap is limited implicitly by $D_{ij} \leq K; \forall B_{ij} \in B$, for some $K \leq 1$ (density target).

Problem 1 (Circuit Placement)

INPUT*: Circuit hypergraph $G_h(V_h, E_h)$ and a fixed outline for the placement area.*
OUTPUT*: Positions for each vertex $v_i \in V_h$ such that (1) wirelength is minimized and (2) the area-density constraints $D_{ij} \leq K$ are satisfied for all $B_{ij} \in B$.*

Key Results

An unconstrained optimal position of a single placeable vertex connected to fixed vertices can be found in linear time as the median of adjacent positions [8]. Unconstrained HPWL minimization for multiple placeable vertices can be formulated as a linear program [7; 10]. For each $e_k \in E_h$, upper and lower bound variables U_k and L_k are added. The cost of e_k (x-direction only) is the difference between U_k and L_k. Each $U_k(L_k)$ comes with p_k inequality constraints that restricts its value to be larger (smaller) than the position of every vertex $i \in C_k$. A hypergraph with n vertices and m hyperedges is represented by a linear program with $n + 2m$ variables and $2P$ constraints.

Linear programming has poor scalability, and integrating constraint-tracking into optimization is difficult. Other approaches include non-linear optimization and partitioning-based methods.

Combinatorial Techniques for Wirelength Minimization

The no-overlap constraints are not convex and cannot be directly added to the linear program for HPWL minimization. Such a program is first solved directly or by casting its dual as an instance of the min-costmax-flow problem [12]. Vertices often cluster in small regions of high density. One can lower-bound the distance between closely-placed vertices with a single linear constraint that depends on the relative placement of these vertices [10]. The resulting optimization problem is incrementally re-solved, and the process repeats until the desired density is achieved.

The *min-cut placement* technique is based on balanced min-cut partitioning of hypergraphs and is more focused on density constraints [11]. Vertices of the initial hypergraph are first partitioned in two similar-sized groups. One of them is assigned to the left half of the placement region, and the other one to the right half. Partitioning is performed by the Multi-level Fiduccia-Mattheyses (MLFM) heuristic [9] to minimize connections between the two groups of vertices (the net-cut objective). Each half is partitioned again, but takes into account the connections to the other half [11]. At the large scale, ensuring the similar sizes of bi-partitions corresponds to density constraints and cut minimization corresponds to HPWL minimization. When regions become small and contain < 10 vertices, optimal positions can be found with respect to discrete slot constraints by branch-and-bound [2]. Balanced hypergaph partitioning is NP-hard [4], but the MLFM heuristic takes $O((V + E) \log V)$ time. The entire min-cut placement procedure takes $O((V + E)(\log V)^2)$ time and can process hypergraphs with millions of vertices in several hours.

A special case of interest is that of one-dimensional placement. When all vertices have identical width and none of them are fixed, one obtains the NP-hard Minimum Linear Arrangement problem [4] which can be approximated in polynomial time within $O(\log V)$ and solved exactly for trees in $O(V^3)$ time as shown by Yannakakis. The min-cut technique described above also works well for the related NP-hard Minimum-Cut Linear Arrangement problem [4].

Nonlinear Optimization

Quadratic and generic non-linear optimization may be faster than linear programming, while reasonably approximating the original formulation. The hypergraph is represented by a weighted graph where w_{ij} represents the weight on the 2-pin edge connecting vertices v_i and v_j in the weighted graph. When an edge is absent, $w_{ij} = 0$, and in general $w_{ii} = -\sum_{i \neq j} w_{ij}$.

Quadratic Placement A quadratic placement (x-direction only) is given by

$$\Phi(x) = \sum_{i,j} w_{ij} \left[(x_i - x_j)^2\right] = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} + \text{const.} \tag{1}$$

The global minimum of $\Phi(x)$ is found by solving $\mathbf{Q}\mathbf{x} + \mathbf{c} = 0$ which is a sparse, symmetric, positive-definite system of linear equations (assuming ≥ 1 fixed vertex), efficiently solved to sufficient accuracy using any number of iterative solvers. Quadratic placement may have different optima depending on the model (clique or star) used to represent hyperedges. However, for a k-pin hyperedge, if the weight on the 2-pin edges introduced is set to W_c in the clique mode and kW_c in the star model, then the models are equivalent in quadratic placement [10].

Linearized Quadratic Placement Quadratic placement can produce lower quality placements. To approximate the linear objective, one can iteratively solve Eq. (1) with $w_{ij} = 1/|x_i - x_j|$ computed at every iteration. Alternatively, one can solve a single β-regularized optimization problem given by $\Phi^{\beta}(x) = \min_x \sum_{i,j} w_{ij}\sqrt{x_i - x_j)^2 + \beta}$. $\beta > 0$, e.g., using a Primal-Dual Newton method with quadratic convergence [1].

Half-Perimeter Wirelength Placement HPWL can be provably approximated by strictly convex and differentiable functions. For 2-pin hyperedges, β-regularization can be used [1]. For an m-pin hyperedge ($m \geq 3$), one can rewrite HPWL as the maximum (l_∞-norm) of all $m(m-1)/2$ pairwise distances $|x_i - x_j|$ and approximate the l_∞-norm by the l_p-norm (p-th root of the sum of p-th powers). This removes all non-differentiabilities except at 0 which is then removed with β-regularization. The resulting HPWL approximation is given by

$$\text{HPWL}_{p-\beta-\text{reg}}(G_h) = \sum_{e_k \in E_h} \left(\sum_{i,j \in C_k} |x_i - x_j <^p + \beta \right)^{1/p} \tag{2}$$

which overestimates HPWL with arbitrarily small relative error as $p \to \infty$ and $\beta \to 0$ [10]. Alternatively, HPWL can be approximated via the log-sum-exp formula given by

$$\text{HPWL}_{\text{log-sum-exp}}(G_h) = \alpha \sum_{e_k \in E_h} \left[\ln \left(\sum_{i \in C_k} \exp\left(\frac{x_i}{\alpha}\right) \right) + \ln \left(\sum_{v_i \in C_k} \exp\left(\frac{-x_i}{\alpha}\right) \right) \right] \tag{3}$$

where $\alpha > 0$ is a smoothing parameter [6]. Both approximations can be optimized using conjugate gradient methods.

Analytic Techniques for Target Density Constraints

The target density constraints are non-differentiable and are typically handled by approximation.

Force-Based Spreading The key idea is to add constant forces f that pull vertices always from overlaps, and recompute the forces over multiple iterations to reflect changes in vertex distribution. For quadratic placement, the new optimality conditions are $\mathbf{Qx} + \mathbf{c} + \mathbf{f} = \mathbf{0}$ [8]. The constant force can perturb a placement in any number of ways to satisfy the target density constraints. The force $\mathbf{f}$ is computed using a discrete version of Poisson's equation.

Fixed-Point Spreading A fixed point f is a pseudovertex with zero area, fixed at (x_f, y_f), and connected to one vertex $H(f)$ in the hypergraph through the use of a pseudo-edge with weight $w_{f,H(f)}$. Quadratic placement with fixed points is given by $\Phi(x) = \sum_{i,j} wi, j(x_i - x_j)^2 + \sum_f w_{f,H(f)}(x_{H(f)} - x_f)^2$. Each each fixed point Ff introduces a quadratic term $w_{f,H(f)}(x_{H(f)} - x_f)^2$. By manipulating the positions of fixed points, one can perturb a placement to satisfy the target density constraints. Compared to constant forces, fixed points improve the controllability and stability of placement iterations [5].

Generalized Force-Directed Spreading The Helmholtz equation models a diffusion process and makes it ideal for spreading vertices [3]. The Helmholtz equation is given by

$$\frac{\partial^2\phi(x,y)}{\partial x^2}\frac{\partial^2\phi(x,y)}{\partial y^2} - \epsilon\phi(x,y) = D(x,y), \tag{4}$$
$$(x,y) \in R\frac{\partial\phi}{\partial v} = 0, (x,y) \text{ on the boundary of } R$$

where $\epsilon > 0, v$ is an outer unit normal, R represents the fixed outline, and $D(x,y)$ represents the continuous density function. The boundary conditions, $\partial\Phi/\partial v = 0$, specify that forces pointing outside of the fixed outline be set to zero – this is a key difference with the Poisson method which assumes that forces become zero at infinity. The value Φ_{ij} at the center of each bin B_{ij} is found by discretization of Eq. (4) using finite differences. The density constraints are replaced by $\Phi_{ij} = \hat{K}, \forall b_{ij} \in B$ where $\hat{K}$ is a scaled representative of the density target K. Wirelength minimization subject to the smoothed density constraints can be solved via Uzawa's algorithm. For quadratic wirelength, this algorithm is a generalization of force-based spreading.

Potential Function Spreading Target density constraints can also be satisfied via a penalty function. The area assigned to bin B_{ij} by vertex v_i is represented by Potential(v_i, B_{ij}) which is a bell-shaped function. The use of piecewise quadratic functions make the potential function non-convex, but smooth and differentiable [6]. The penalty term given by

$$\text{Penalty} = \sum_{B_{ij}\in B}\left(\sum_{v_i\in V_h}\text{Potential}(v_i, B_{ij}) - K\right)^2 \tag{5}$$

can be combined with a wirelength approximation to arrive at an unconstrained optimization problem which is solved using an efficient conjugate gradient method [6].

Applications

Practical applications involve more sophisticated interconnect objectives, such as circuit delay, routing congestion, power dissipation, power density, and maximum thermal gradient. The above techniques are adapted to handle multi-objective optimization. Many such extensions are based on heuristic assignment of net weights that encourage the shortening of some (e.g., timing-critical and frequently-switching) connections at the expense of other connections. To moderate routing congestion, predictive congestion maps are used to decrease the maximal density constraint for placement in congested regions. Another application is in physical synthesis, where incremental placement is used to evaluate changes in circuit topology.

Open Problems

None is reported.

Experimental Results

Circuit placement has been actively studied for the past 30 years and a wealth of experimental results are reported throughout the literature. A 2003 result demonstrated that placement tools could produce results as much as $1:41\times$ to $2:09\times$ known optimal

wirelengths on average (advances have been made since this study). A 2005 placement contest found that a set of tools produced placements with wirelengths that differed by as much as $1:84\times$ on average. A 2006 placement contest found that a set of tools produced placements that differed by as much as $1:39\times$ on average when the objective was the simultaneous minimization of wirelength, routability and run time. Placement run times range from minutes for smaller instances to hours for larger instances, with several millions of variables.

URLs to Code and Data Sets

Benchmarks include the ICCAD '04 suite (`http://vlsicad.eecs.umich.edu/BK/ICCAD04bench/`), the ISPD '05 suite (`http://www.sigda.org/ispd2005/contest.htm`) and the ISPD '06 suite (`http://www.sigda.org/ispd2006/contest.htm`). Instances in these benchmark suites contain between 10K to 2.5M placeable objects. Other common suites can be found, including large-scale placement instances problems with known optimal solutions (`http://cadlab.cs.ucla.edu/~pubbench`).

Cross-References

Performance-Driven Clustering
Energy Minimization in VLSI Circuits

Recommended Reading

1. Alpert CJ, Chan T, Kahng AB, Markov IL, Mulet P (1998) Faster minimization of linear wirelength for global placement. IEEE Trans. CAD 17(1), 3-13
2. Caldwell AE, Kahng AB, Markov IL (2000) Optimal partitioners and end-case placers for standard-cell layout. IEEE Trans. CAD 19(11), 1304-1314
3. Chan T, Cong J, Sze K (2005) Multilevel generalized force-directed method for circuit placement. Proc. Intl. Symp. Physical Design. ACM Press, San Francisco, 3-5 Apr 2005. pp. 185-192
4. Ausiello G, Crescenzi P, Gambosi G, Kann V, Marchetti-Spaccamela A, Protasi M (1998) Complexity and Approximation: Combinatorial optimization problems and their approximability properties. Springer
5. Hu B, Marek-Sadowska M (2005) Multilevel fixed-point-additionbased VLSI placement. IEEE Trans. CAD 24(8), 1188-1203
6. Kahng AB, Wang Q (2005) Implementation and extensibility of an analytic placer. IEEE Trans. CAD 24(5), 734-747
7. Kennings A, Markov IL (2002) Smoothingmax-terms and analytical minimization of half-perimeter wirelength. VLSI Design 14(3), 229-237
8. Kennings A, Vorwerk K (2006) Force-directed methods for generic placement. IEEE Trans. CAD 25(10), 2076-2087
9. Papa DA, Markov IL (2007) Hypergraph partitioning and clustering. In: Gonzalez, T. (ed.) Handbook of algorithms. Taylor & Francis Group, Boca Raton, CRC Press, pp. 61-1
10. Reda S, Chowdhary A (2006) Effective linear programming based placement methods. In: ACM Press, San Jose, 9-12 Apr
11. Roy JA, Adya SN, Papa DA, Markov IL(2006) Min-cut floorplacement. IEEE Trans. CAD 25(7), 1313-1326
12. Tang X, Tian R, Wong MDF (2005) Optimal redistribution of white space for wirelength minimization. In: Tang, T.-A. (ed.) Proc. Asia South Pac. Design Autom. Conf., ACM Press, 18-21 Jan 2005, Shanghai. pp. 412-417