Progress and Challenges in VLSI Placement Research

Special Session Paper

Igor L. Markov, Jin Hu and Myung-Chul Kim
University of Michigan, Department of EECS, 2260 Hayward St, Ann Arbor, MI 48109-2121
{imarkov, jinhu, mckima}@eecs.umich.edu

ABSTRACT

Given the significance of placement in IC physical design, extensive research studies performed over the last 50 years addressed numerous aspects of global and detailed placement. The objectives and the constraints dominant in placement have been revised many times over, and continue to evolve. Additionally, the increasing scale of placement instances affects the algorithms of choice for high-performance tools. We survey the history of placement research, the progress achieved up to now, and outstanding challenges.

1. INTRODUCTION

Research on VLSI Placement can be traced back to the 1960s, when the first netlist partitioning methods were developed in the industry, and subsequently motivated improvements in graph partitioning heuristics. Analytical placers[1] started appearing in the early 1980s, but were eclipsed by combinatorial techniques when simulated annealing was invented. Annealing-based placers [147] dominated industry use and academic results for a decade, but by the mid 1990s, annealing was no longer scalable for newer and larger designs. Despite the steady improvement rate of analytical placement, partitioning-based methods improved enough to provide leading-edge performance: (*i*) (multilevel) *Fiduccia-Mattheyses (FM)* heuristics produced much better results much faster than previous methods, (*ii*) the use of end-case techniques (optimal partitioning and end-case placement) during top-down layout optimization provided high-quality detailed placement [14], and (*iii*) the use of flat and multilevel FM heuristics was carefully optimized, including cutline selection and hierarchical whitespace allocation [18].

By 2005, several analytical techniques have matured to the point where they reliably outperformed min-cut placement on contemporary large global placement instances. In addition to the innovations in algorithms, this was due to the change in the nature of placement instances. In particular, having 100K-10M movable objects during global placement provided a better justification to modeling each object as a dimensionless dot. Industry methodologies provided global placement with a large number of fixed objects (I/O pads, fixed pins and macro blocks, etc). Due to concerns of routability, physical synthesis, power density, large size of I/O pads, large IP blocks, etc, core placement area now often includes a large amount of unused space [109] which provides analytical algorithms with useful freedom. In terms of algorithms, high-quality detailed placement was developed, resolving a long-standing handicap in analytical engines. These improvements fueled algorithmic developments based on multivariate calculus, numerical analysis, and combinatorial optimization [109]. Resulting reductions in interconnect length enhanced many types of semiconductor designs — from FPGAs and ASICs to CPUs and mixed-signal SoCs. They have surpassed the length reduction typical for a new technology node. Despite such major progress, there is currently no agreement on which algorithms are considered best. Comparisons remain largely empirical [109] and are often affected by the quality and maturity of software implementations, use of parallel processing and high-performance libraries, as well as reporting methodologies.

Recent advances in placement algorithms include (*i*) high-performance wirelength-driven placement, (*ii*) mixed-sized placement (i.e., simultaneous placement of both cells and macros), (*iii*) routability-driven placement, (*iv*) timing- and power-driven placement, as well as (*v*) the integration of global placement into physical synthesis.

2. WIRELENGTH-DRIVEN PLACEMENT

Modern placement is an optimization problem with many objectives and constraints. However, the most common approach is to first develop a *wirelength-driven* global placement engine that solves a straightforward mathematical formulation and is competitive on common benchmarks. This is a prerequisite for strong performance in multiobjective optimization, and the handling of additional objectives and constraints can be implemented as the next step.

We formulate global placement as constrained optimization and explain how the objective is approximated by smooth functions to facilitate more efficient numerical methods. Common types of global placement algorithms are reviewed next, followed by legalization and detailed placement.

2.1 The objectives and the constraints

Global placement [79, Chapter 4] of a netlist $\mathcal{N} = (E, V)$ with nets E and n nodes (cells) V seeks a set of planar node locations $(\vec{x}, \vec{y}) \in [x_{min}, x_{max}]^n \times [y_{min}, y_{max}]^n$ that minimize the weighted Half-Perimeter WireLength (wHPWL). For locations $\vec{x} = \{x_i\}$, $\vec{y} = \{y_i\}$ and net weights $\vec{w} = \{w_i\}$, $\text{wHPWL}_{\mathcal{N}}(\vec{x}, \vec{y}) = \text{wHPWL}_{\mathcal{N}}(\vec{x}) + \text{wHPWL}_{\mathcal{N}}(\vec{y})$, where

$$wHPWL_{\mathcal{N}}(\vec{x}) = \Sigma_{e \in E} w_e [\max_{i \in e} x_i - \min_{i \in e} x_i] \qquad (1)$$

The HPWL function is continuous and convex, but not everywhere differentiable. It is amenable to combinatorial

[1]Analytical placers model interconnect length by differentiable functions and use smooth optimization techniques.

optimization, especially linear programming and network flows. However, these techniques do not combine well with other co-objectives and are less scalable than smooth optimization applied to approximations of HPWL.

Moreau's theorem from convex analysis [141, Proposition IV.1.8] implies that the gradient of such a convex function is defined almost everywhere and can be approximated by the so-called Yosida approximation of the function whose accuracy is controlled by the positive parameter $\alpha \to 0$. While this specific approximation has not been used in placement applications, a number of specialized approximations of HPWL have been proposed 2.2.

Minimizing a convex objective is not difficult in itself. The main source of complexity in global placement is the non-convex constraints. Despite the variety of constraints imposed in practice, two types are considered first. The *fixed-position constraints* enforce given locations of certain cells or macros. They are convex, can be substituted directly into the objective function, and do not require dedicated computations. The *density constraints* (*i*) ensure *porosity*, which is critical for routability and timing-optimization transformations, and (*ii*) prevent movable objects from concentrating in small regions and ensure that legalization can find non-overlapping positions for all objects with minimum disturbance from global placement. When analytical techniques for global placement are used, fixed-position constraints play the important role of spreading movable nodes between fixed positions as a side effect of interconnect optimization *before* density constraints are seriously considered. A second use, now commonly seen in quadratic placers is to temporarily extend the netlist with fake nets (psuedonets) and fake cells (anchors) fixed at carefully chosen locations. Again, the purpose is to decrease peak density as a side effect of interconnect optimization.

2.2 Smooth approximations of HPWL

Quadratic approximations used in many placers are obtained by pricing every edge by its squared length, with a certain weight. Their general form is

$$\Phi_Q(\vec{x},\vec{y}) = \vec{x}^T Q_x \vec{x}/2 + \vec{f}_x\vec{x} + \vec{y}^T Q_y \vec{y}/2 + \vec{f}_y\vec{y} \qquad (2)$$

with matrices Q_x, Q_y derived from the netlist and vectors $\vec{f}_x, \vec{f}_y$ that reflect connections to fixed objects. When sufficiently many nodes in a connected netlist are fixed, Φ_Q is strictly convex. The x and y components can be optimized separately and quickly by solving sparse systems of linear equations $Q_x\vec{x} = -\vec{f}_x$ and $Q_y\vec{y} = -\vec{f}_y$.

To make such quadratic functions more appropriate for HPWL modeling, they are *linearized* [142] by adjusting the approximations at every global placement iteration. In particular, single-edge terms of the form $w_{ij}(x_i - x_j)^2$ are changed to $\frac{w_{ij}(x_i-x_j)^2}{|x'_i-x'_j|+\varepsilon}$ where the primed values are constants based on *the result of the last iteration*. These adjustments modify nonzero elements of Q_x and Q_y, but do not change the sparsity pattern.

To approximate HPWL by a quadratic objective, the netlist $\mathcal{N}$ is first transformed into graphs $\mathcal{G}_x$ and $\mathcal{G}_y$ that preserve the node set V and represent each two-pin net by a single edge with weight $1/length$. Larger nets are decomposed depending on the relative placement of vertices using the Bound2Bound (B2B) model [145] — for each p-pin net, the *extreme* nodes (min and max) are connected to each other and to each *internal* node by edges, with the following weight

$$w^{B2B}_{x,ij} = \frac{1}{(p-1)\big(|x_i - x_j| + \varepsilon\big)} \qquad (3)$$

The Bound2Bound model captures HPWL exactly, but only at the point of instantiation. As x,y locations change, the decomposition is recalculated, the sparsity patterns of matrices Q_x and Q_y change, but their numbers of nonzeros do not change.

Non-quadratic approximations of HPWL, such as the log-sum-exp technique [135] which approximates the bracketed term in Formula 1 for $\gamma \to 0$ by the convex function

$$\gamma\big(\log\sum_{k\in e} e^{x_k/\gamma} + \log\sum_{k\in e} e^{-x_k/\gamma}\big) \to [\max_{i\in e} x_i - \min_{i\in e} x_i]$$

Other such techniques are surveyed and compared in [20, 63, 94]. As with quadratic approximations, gradients are computed in closed form, but their numerical values must be updated as the placement changes. Numerical minimization requires more than solving two sparse linear systems.

2.3 Modern global placement algorithms

Quadratic placement is the foundation of BonnPlace [13], DPlace2.0 [103], mFAR [66], Kraftwerk [145], FastPlace [158] and RQL [158], as well as SimPL [85,87], MAPLE [89] and ComPLx [88]. The netlist is modeled by a graph, for which the quadratic objective is formulated. The placer alternates between quadratic optimization and steps that involve density estimation (including fixed obstacles), in some cases along with dedicated spreading of movable objects. To decrease peak density, placers employ a variety of combinatorial and numerical techniques based on (*i*) network flows (BonnPlace), (*ii*) self-contained estimation of density gradients (FastPlace and RQL), as well as (*iii*) "full spreading" (SimPL) and derived algorithms. In all cases, the objective in a previously solved quadratic program is modified, and the program is re-solved.

Force-directed placement models interconnects by coil springs, describes those springs using Hooke's law, formulates equations for force equilibrium, and solves these equations using quadratic programming. We illustrate the principles of force-directed placement by Kraftwerk [145], where *three* forces are involved in force-equilibrium equation — *the spring force, the hold force* and *the density-based* (spreading) force. The spring force corresponds to a quadratic approximation of the HPWL objective outlined in Section 2.2, the hold force is its "opposite action" in the spirit of Newton's Third Law of motion, and the density force seeks to even out cell density. If we denote cell density at (x, y) by $\rho(x, y)$, the density force is, loosely speaking, weight-averaged $-\nabla\rho$ (Formula 5). The *hold* and *density* forces acting on a cell are cumulatively represented by a pseudonet connected to a carefully placed fake fixed cell (anchor).

Density-based force computations were pioneered in Kraftwerk [145]. Admitting that the density function $\rho(x, y)$ may not be smooth, Kraftwerk looks for a twice-differentiable potential function $u(x, y)$, whose gradient $\vec{F} = \nabla u$ represents a *conservative* force pointing away from dense regions.[2] The latter condition is formalized in terms of *flux* over an arbitrary closed contour C that bounds region R:

[2] A *conservative* force $\vec{F}$ satisfies the following equivalent conditions (*i*) $\exists\, u:\ \vec{F} = \nabla u$ (a potential function exists), (*ii*) $\forall$ closed contour $C:\ \oint_C \vec{F}\cdot d\vec{c} = 0$ (path-independence), and (*iii*) $\nabla \times \vec{F} = 0$ (curl-free force).

$\oint_C (\vec{F} \cdot n_C)\ \mathrm{d}\vec{c} = \int\int_R \rho\ \mathrm{d}x\mathrm{d}y$. Then, using the divergence theorem and $\vec{F} = \nabla u$, we have

$$\oint_C (\vec{F} \cdot n_C)\ \mathrm{d}\vec{c} = \int\int_R \Delta u\ \mathrm{d}x\mathrm{d}y = \int\int_R \rho\ \mathrm{d}x\mathrm{d}y \qquad (4)$$

This condition is satisfied by solutions of the Poisson's equation $\Delta u = \rho$, which can be interpreted as finding the electrostatic potential u for the curl-free field ∇u generated by a spatial charge distribution ρ. A word of caution: *the Poisson's equation only gives a way to satisfy properties postulated for* $\vec{F}$. The developers of mPL6 [21] noted that the Poisson's equation can be ill-defined and added a new term, producing the nonhomogenous Helmholtz (screened Poisson's) equation $\Delta u - \varepsilon u = \rho$, both of which are linear second-order elliptic PDEs. Figures 2 and 3 in [111] illustrate solutions u of Poisson's equation, which look like smoothened versions of ρ. This smoothing effect is formalized in [42] for both Poisson's and Helmholtz PDEs by representing solutions as convolutions (over the entire placement region R) of ρ with certain Green's functions $G(x, s)$ dependent on the boundary conditions:

$$u(\xi) = \int_R G(\xi, \nu)\rho(\nu)\mathrm{d}\nu \qquad (5)$$

where ξ and ν represent (x, y) pairs in R. This observation leads to a particularly efficient computation of ∇u. For example, for the Poisson's equation (i) without boundary conditions, and (ii) with zero gradients at infinity,

$$(i)\ G(\xi, \nu) = \frac{1}{||\xi - \nu||} \qquad (ii)\ G(\xi, \nu) = \frac{\ln ||\xi - \nu||}{2\pi} \qquad (6)$$

Non-convex placers such as APlace [83], mPL6 [21], NTUPlace3 [31], and NTUPlace4 [64] typically approximate HPWL by non-quadratic objectives functions for which gradients (and Hessians) can be computed analytically (Section 2.2). The contribution of fixed obstacles (e.g., macros) is modeled by two-dimensional step functions, and then approximated by bell-shaped [31, 83] functions, but often post-processed by Gaussian smoothing and leveling [31]. The combined optimization function is non-convex, in stark contrast to quadratic and force-directed placement. Nonlinear optimization employed by these algorithms is time-consuming and is often accelerated using multilevel clustering.

Locality, force orientation and force modulation. Density gradients in APlace [83] and NTUPlace [31] are based on local information and, e.g., do not account for possible paths around fixed obstacles. This may require numerous global placement iterations. Kraftwerk and mPL6 find density gradients by solving linear elliptic PDEs as explained above, incorporating more global perspective into their density gradients. However, it remains unclear how well this accounts for possible paths around fixed obstacles. Whether gradients are purely local or point in the best possible direction, it often remains unclear how density gradients should be scaled for proper balance with interconnect-based forces. To this end, the work in [158] distinguishes the tasks of *force orientation* and *force modulation*. It demonstrates that force modulation in FastPlace can be improved by reducing the magnitude of 10% strongest forces, as implemented in RQL. These challenges are addressed in a more systematic way in SimPL [85, 87] by *lookahead legalization (LAL)*, which globally identifies a desired location for every cell such that most overlaps are removed.

2.4 Legalization and Detailed Placement

The cell locations from global placement may overlap, and typically do not align with power rails. The global placement must then be *legalized*, where all cell overlap is removed without undermining design objectives. Legalization removes all cell overlap while minimizing total cell displacement, and is necessary not only after global placement, but also after incremental changes, e.g., physical synthesis optimizations [4]. Unlike *cell spreading* during global placement, legalization is typically performed when cells are both (i) well-distributed over the entire region and (ii) have relatively small overlap. A legalized placement can be improved with respect to a given objective by *detailed placement*, e.g., swapping neighboring cells to reduce total wirelength, or sliding cells to one side of the row when unused space is available. Table 1 summarizes methods for legalization and detailed placement.

Stage	Technique
Legalization	greedy moves to free locations [31, 59, 60, 144, 157]
	ripple cell movement [72]
	diffusion PDE [126]
	dynamic programming [3, 80, 144]
	computational geometry [104]
	network flow [9, 12, 34, 47]
	linear programming [44]
	top-down opt. & clustering [60, 93]
Detailed placement	branch-and-bound [14, 18]
	network flow [47]
	simulated annealing [137]
	mixed ILP [19, 97]
	single-row optimization [11, 82]
	cell-to-slot matching [31]
	cell swapping [44, 115]
	clustering [72, 115]
	dynamic programming [70]
	global-placer integration [131]

Table 1: Legalization and detailed placement.

Legalization algorithms can be classified as (i) local approaches, where cells are moved one at a time to a best-available location, and (ii) global, where cells move in accordance with a general strategy. In the former case, legalizers such as Tetris [59] greedily assign each cell to its nearest legal location while respecting row capacity. Abacus [144] also finds the best row the cell belongs to, but uses dynamic programming to re-place the already-placed cells such that the total displacement is minimized. The authors of [60, 93] also explicitly minimize global wirelength during legalization. In the latter case, approaches based on network flows help find global optima. Extensions include incorporating history [34] and modifying path augmentation algorithms [9]. Additional techniques are shown in Table 1.

Detailed placers preserve legality while improving solutions by relocating movable cells.[3] Branch-and-bound placers [18] reorder groups of neighboring cells in a row by a sliding-window technique, where cells are reordered optimally inside each window. However, this approach can handle typically up to eight cells at a time. A more scalable optimization, handling up to 20 cells at a time, splits the cells in a given window into left and right halves, and optimally interleaves the two groups while preserving the relative order

[3] Legality may be temporarily violated.

of cells from each group [70]. The authors of [115] improve wirelength by swapping pairs of non-adjacent cells, and cycling triplets. When unused space is available between cells in a row, these cells can be shifted to either side or to intermediate locations. Wirelength-optimal locations in a given row can be found by a polynomial-time algorithm [82], which is practical in many applications.

Detailed placers bundled with legalizers are illustrated by FastPlace-DP [115], which uses simple but efficient incremental operations that shorten interconnect by several percent. ECO-System [131], integrated in Capo [18, 133], identifies areas where cells need to be re-placed, then applies Capo to perform both legalization and detailed placement, simultaneously and consistently in all such regions.

3. MIXED-SIZE PLACEMENT

The number of *macros* included in modern ICs is growing [166] in response to technology and business trends. Finding locations of larger circuit modules and placing standard cells are essentially the same from an optimization viewpoint, distinguished only by (*i*) the scale relative to the size of layout regions, and (*ii*) the shaping and rotations of macros in floorplanning. *Mixed-sized placement* carefully combines floorplanning techniques — to pack (the relatively few) large blocks — and placement techniques — to handle the millions of small standard cells. Several available approaches are summarized in Table 2.

Simultaneous flows do not separate the placement of standard cells and macros into separate stages. One method to handle macros is to divide them into "shreds" comparable in size to standard cells [1]. These shreds can be connected by fake nets during wirelength optimization so as to keep them close together [1, 13], e.g., when shaping soft blocks [129]. In contrast, [88] only shreds macros during geometric spreading. Other placement-based approaches explicitly shift macros or cells during placement [31, 89, 157], or relegalize after every placement iteration [21, 45]. Techniques that simultaneously move macros and standard cells can be classified into force-directed [50, 157], non-convex [21, 31], min-cut [129] and flow-based [35]. However, many ideas are applicable in different contexts. One strategy is to legalize and fix macros that are comparable in size to the magnitude of cell displacements at the current iteration [21, 132].

Sequential flows separate macro and standard-cell placement. Some flows place all macros at once after tentatively placing the full netlist, and before standard-cell placement, such as packing macros at the chip periphery of [29]. Other approaches [1, 174] (*i*) cluster standard cells into soft blocks, (*ii*) use a floorplanner on the original macros and new soft blocks, (*iii*) dissolve soft blocks, and (*iv*) place standard cells using established methods. Many techniques [26, 29, 62, 174] account for macro flipping and rotation.

Post-placement methods remove overlap between macros and standard cells by floorplan repair [107], detailed placement [13, 44, 145], and force-directed techniques [50, 129].

Flow	Technique
simultaneous	macro shredding [13, 88, 129]
	macro or cell shifting [31, 89, 157]
	iterative re-legalization [21, 45]
	top-down legalization [21, 44, 129, 132]
	force-directed optimization [21, 31, 50, 62, 83, 88, 89, 157, 158]
	floorplacement [35, 129, 132]
sequential	periphery macro packing [29]
	macro shredding [1]
	separate floorplanning & placement steps [1, Flow 1], [26, 29, 174]
post-placement	floorplan repair [107]
	linear programming [13, 44, 145]
	force-directed optimization [1, Flow 2]

Table 2: Mixed-size placement techniques.

4. ROUTABILITY-DRIVEN PLACEMENT

With increasing design complexity, optimizing traditional placement metrics is insufficient for successful routing [6, 130]. To mitigate routing failures, *routability-driven placers* incorporate route estimation as part of their flow.

Congestion maps indicate regions where routing will be difficult, and are used to guide optimization during placement. They are generated using: (*i*) static approaches, where the congestion map is fixed for a placement instance, (*ii*) probabilistic approaches, where net topologies are not fixed, and probabilistically determined, and (*iii*) constructive approaches, where a simplified global router generates approximate net routes. Traditionally, the first two options have been the most popular, but the last option has recently been gaining acceptance thanks to advanced global routers designed to handle greater layout complexity. Empirical evidence from the ISPD 2011 Routability-driven Contest [155] suggests that both *probabilistic* and *constructive* methods are viable and scalable. Table 3 summarizes these approaches.

Placement optimizations are applied throughout the entire placement flow: (*i*) during global placement, (*ii*) modifying intermediate solutions, (*iii*) during legalization and detailed placement, and (*iv*) as a post-placement processing step (see Table 4). In global placers, the most popular techniques are *cell bloating* and *whitespace injection*. Depending on the placer type, e.g., quadratic and min-cut, The implementation of these techniques will require placer modification, including changing the optimization function. In detailed placers, the most popular techniques are *cell swapping* and *cell shifting*. Additional optimizations can be applied to intermediate (or near-final) placement solutions, and then passed on to the next step of the design flow.

Contests. Researchers from IBM organized the ISPD 2011 Routability-driven Contest [155]. The benchmarks included 483K to 1.29M movable cells and a set of routing constraints (e.g., blockages), such that many of them were intentionally difficult to route. Placement solutions were evaluated by running a global router and counting violations. The results indicate that routability can be improved by *increasing porosity*. SimPLR [86] and Ripple [58] used conges-

Approach	Technique
static	net bounding box [16, 76]
	Steiner trees [130]
	pin density [10, 179]
	counting nets in regions [164]
probabilistic	uniform wire density [58, 64, 143]
	smoothened wire density [152]
	pattern routing [168]
constructive	using A*-search [169]
	using a global router: • FastRoute [173] in IPR [38] • BFG-R [69] in SimPLR [86]

Table 3: Congestion estimation for placement.

Placement Phase	Technique
Global Placement	relocating movable objects: • moving nets [58, 76] • modifying forces [40, 143] • incorporating congestion in objective function [64, 152] • adjusting target density [86]
	cell bloating [10, 58, 61, 86]
	macro porosity [64, 76]
	pin density control [64]
	expanding/shrinking placement regions [120]
Intermediate	local placement refinement [38]
Legalization and Detailed Placement	linear placement in small windows [75, 130]
	congestion embedded in objective function [178]
	cell swapping [38, 58, 86]
	cell shifting [46, 64]
Post Placement	whitespace injection or reallocation [96, 130, 175]
	simulated annealing [30, 65, 161]
	linear programming [99]
	network flows [162, 163]
	shifting modules by expanding gcells [178]
	cell bloating [134]

Table 4: Routability-driven placement.

tion maps to *bloat cells* and modify the anchor positions during quadratic placement. NTUPlace4 [64] uses congestion maps when modeling pin density. To estimate congestion, SimPLR integrated a global router [69], whereas Ripple and NTUPlace4 adopted probabilistic congestion estimation [143]. The DAC 2012 Contest Benchmarks [156] were easier to route and emphasized different sources of congestion. The evaluation metric combined runtime and scaled HPWL. Publications describing the contest submissions are not yet available when this text was written.

5. TIMING- & POWER-DRIVEN PLACEMENT

Timing-driven placement (TDP) seeks to optimize *circuit delay*. TDP identifies *critical nets* using *Static Timing Analysis (STA)* [79, Section 8.2], and typically minimizes *total negative slack (TNS)*, *worst negative slack (WNS)*, or both. Table 5 outlines timing-driven placement.

Net-based approaches optimize circuit delay by translating STA results and timing constraints into *net weights* and *net constraints*, respectively. A higher net weight encourages interconnect optimization to preferentially shorten the net, whereas a net constraint limits net delay. *Static net weights* remain constant during placement, and are typically based on negative slack [15, 24, 49, 90] or sensitivity [53, 128, 172], where placers attempt to predict the impact each net has on timing. A net weight that is too high may shorten a net at the expense of upstream or downstream nets, increasing circuit delay. To avoid this, *dynamic net weights* are gradually updated based on slack change [15, 125] or net criticality [50, 125]. While more flexible, dynamic net weights may cause nets to *oscillate* between critical and non-critical [43]. To dampen oscillations, net weights are accumulated based on histories, with non-increasing increments.

Net constraints limit net length or delay, and do not require as accurate timing predictions as net weights. Common methods to generate delay budgets include the *zero-slack algorithm (ZSA)* [79, Section 8.2.2], [101, 108] and the *Iterative-Minmax-PERT* algorithm [177]. The usage of net constraints is placer-dependent. Min-cut placers [51] modify cut costs, and analytic placers modify forces [124] or Lagrange multipliers [151]. In detailed placement, net constraints have also been integrated in cell movement [71], primary objective functions [55], and as a separate local-move step [35]. Net constraints are supported by *differential timing analysis* [127], which generalizes incremental STA [117].

Technique	Implementation
static net weights	slack [15, 24, 49, 90]
	sensitivity [53, 128, 172]
dynamic net weights	incremental timing analysis [15, 125]
	based on previous iterations [50, 125]
net constraints	in global placement [51, 54, 124, 149, 151]
	in detailed placement [35, 55, 71, 127]
path-based	partitioning [74]
	simulated annealing [147]
	Lagrangian relaxation [56, 146]
	differential timing analysis [37]
compound	net weights & constraints [102]

Table 5: Timing-driven placement approaches.

Path-based approaches directly model timing on critical paths, and *explicitly* ensure that each considered path meets timing constraints. These approaches typically achieve better solutions than net-based approaches because of their global scope. However, accurate path-based optimization due to numerous signal paths in large ICs. To facilitate scalability, path-based approaches (*i*) embed a graph-based timing model and (*ii*) formulate a mathematical program that maintains intermediate timing variables. Auxiliary techniques such as partitioning [74] and Lagrangian relaxation [56, 146] can solve the program and improve quality. Other approaches include solving linear programs in local neighborhoods [37], and using simulated annealing [147].

Compound approaches are illustrated by [102], which uses a hybrid path-based delay sensitivity function for net weights and minimizes critical nets' wirelength.

In the context of IC power optimization, *static power* does not directly depend on cell locations. With multiple voltages, static power can be reduced by changing voltage-island assignments. In contrast, *dynamic power* depends on interconnect lengths, which are determined by placement. To support higher clock frequencies, modern designs are heavily pipelined, and require more-capacitive clock networks, which can contribute 30% of total IC power [106]. To support design scaling [73], placers must co-optimize (*i*) the hundreds of millions of signal nets, each consuming a small amount of power, and (*ii*) clock networks with significant power consumption, as illustrated in [32, 92]. Table 6 outlines power-driven placement.

Static-power reduction techniques trade positive timing slack for power. When multiple supply voltages are present, cells can be moved closer to voltage sources in *rows* [176], where cells are in interleaving (half) rows of high- and low-V_{DD} rows, and in *regions* [68, 91, 122], where cells are powered by the closest voltage island. Other approaches include using cell hierarchy and clustering [100], and locality [123].

Dynamic-power reduction can be accomplished by (*i*) reducing net activity, and (*ii*) optimizing register locations and optimizing the clock tree. These classes are not mutu-

Power	Technique
static	multiple supply voltages [68, 91, 100, 122, 176]
	logic and physical adjacency [123]
dynamic	weights on signal nets [32, 110, 136, 153]
	register clustering [23, 32, 77, 140]
	explicit register relocation [33, 105, 119]
	clock-tree co-synthesis [39, 92, 119, 140, 165]

Table 6: Power-driven placement techniques.

ally exclusive. Clustering registers at the clock-tree leaves facilitates inverter sharing [23,77] and reduces clock-tree capacitance [32, 92]. Such optimizations can be performed using net weights based on activity factors [32, 110, 136, 153]. Register placement is described in [33, 92, 105, 119], whereas [39, 92, 119, 140, 165] incorporate clock-tree synthesis into global placement. In [92], this integration reduces clock trees by 30% and total dynamic power of the netlist and clock tree by 7%. The authors of [140] add clock-gating logic, and further refine the tree with incremental placement techniques. An IBM physical-synthesis flow that accounts for gated clocks in high-performance ICs is described in [119].

6. PLACEMENT IN PHYSICAL SYNTHESIS

Physical synthesis [4] modifies the netlist based on placement information so as to (*i*) fix timing violations and (*ii*) optimize performance metrics. After a round of optimizations, the design can be re-placed to facilitate further optimizations. Hence, the interaction with placement algorithms is crucial for timing closure. Table 7 lists physical synthesis techniques that heavily interact with placement.

Logic transformations [79, Section 8.5.3] such as *cloning*, *gate decomposition*, and *combinational restructuring* manipulate area-power-timing trade-offs in combinational circuits. Newly inserted gates must be given valid locations, which can make or break a given transformation [48], as illustrated by (*i*) restructuring fanin trees [171] and cones [167], and (*ii*) simulation-driven restructuring that uses *controllability* and *observability don't-cares* [121].

Interconnect buffering [98, 154] improves circuit timing by speeding up signal transitions in long nets. (Approximately) equal-spaced buffers break down timing-critical nets into shorter segments. Due to technology scaling, buffers comprise 10-44% of standard-cell instances in large designs [118]. As the final buffer requirement is unknown in advance, placers reserve unused space (whitespace) throughout the layout [2]. *Virtual buffering* [118,138] assigns buffers to long interconnects without changing the netlist, but accounts for their impact on area and timing. Porosity-aware buffer-planning integrated in an analytical-placement framework adds buffer density to the objective function [27]. Porosity-aware Steiner trees place buffers in available sites [5].

Technique	Implementation
logic transformations	fanin restructuring [167, 171]
	cloning and decomposition [48]
	simulation-driven restructuring based on don't-cares [121]
interconnect buffering	fanout restructuring [48]
	delay-optimal [118, 138]
	Steiner tree construction [5]
gate sizing	discrete [67, 113, 117]
	continuous [148, 159]
compound transformations	area management [95]
	retiming-based physical synth. [116]
	design flows [117, 119, 150, 159]

Table 7: Placement-aware physical synthesis.

Gate sizing [67, 113] does not change connectivity but impacts timing-power trade-offs, facilitating other optimizations. The authors of [117] develop placement-aware branch-and-bound search using a discrete cell library. Alternatively, gate sizes can be extrapolated using continuous delay models within performance-driven physical design [148, 159].

Physical retiming moves registers through combinational logic in the netlist in order to ease timing constraints. Despite numerous publications on retiming in the logic domain, practical implementations must account for interconnect delay, and therefore gate locations and buffering. A retiming-based physical-transformation system [116] uses virtual buffering and exploits the interaction between retiming, placement and cloning in a unified mixed integer-linear program.

Compound optimizations perform sophisticated area-timing-power trade-offs along multiple design dimensions while limiting design iterations and turnaround time. Logic transformations, buffer insertions and gate sizing may require legalizaton and detailed placement. Additionally, [95] adjusts floorplans to accommodate area changes. Industrial physical-synthesis flows are reported in [117, 119, 150, 159].

7. OPEN CHALLENGES

As modern ICs continue to grow [73], flat placement will require new algorithms and data structures to support physical design and physical synthesis with numerous macros and multiple clock domains [7]. Most gate-level techniques for 3D placement remain impractical due to high TSV costs [84].

Automatic generation of datapath layout currently remains inferior to manual placement. Addressing this suboptimality requires accurate identification of datapath logic, alignment of bit slices, careful spacing of aligned groups, as well as structure-aware legalization algorithms [36, 160].

More integrated timing and power optimizations. Given that timing-critical nets are typically identified by sign-off quality timing engines *after* placement, significant placement modification can be required in presence of a large number of near-timing critical nets. Removing all slack violations on these critical nets *during* placement, however, can undermine placement quality, generate new critical nets, and hamper timing closure. Moreover, as the distance between metal layers (and neighboring nets) is reduced, *coupling delay* further complicates timing analysis, as it can now be induced by nets *above* and *below* in addition to the parallel wires on the same layer. Further challenges arise when integrating placement with clock-network synthesis to better account for process variation, useful skew, hold constraints, clock gating and other low-power optimizations.

Layout-friendly high-level synthesis [41] promises improvement in IC power and performance by initiating physical optimization much earlier than is currently done.

Lithography-aware physical synthesis [112,170] enables early manufacturability optimization by, e.g., controlling wire density to enhance CMP [28] and improve timing yield [81].

Quantifying the impact. Given significant investment in placement algorithms and tools, as well the tangible progress achieved, it is important to quantify their impact on the cost and quality of ICs and semiconductor products. To this end, experiments in [134] show that more effective congestion-driven placement facilitates die-size reductions, which translate into lower manufacturing costs and higher profits.

8. REFERENCES

[1] S. N. Adya, I. L. Markov, "Combinatorial Techniques for Mixed-size Placement", *TODAES* 10(1) 2005, pp. 58-90.

[2] S. N. Adya, I. L. Markov, P. G. Villarrubia, "On Whitespace and Stability in Physical Synthesis", *Integration* 39(4) 2006, pp. 340-362.

[3] A. Agnihorti, M. C. Yildiz, A. Khatkhate, A. Mathur, S. Ono, P. H. Madden, "Fractional Cut: Improved Recursive Bisection Placement", *ICCAD* 2003, pp. 307-310.

[4] C. J. Alpert et al., "Techniques for Fast Physical Synthesis", *IEEE* 95(3) 2007,pp. 573-599.

[5] C. J. Alpert, G. Gandham, M. Hrkic, J. Hu, S. T. Quay, "Porosity Aware Buffered Steiner Tree Construction", *ISPD* 2003, pp. 158-165.

[6] C. J. Alpert, Z. Li, M. D. Moffitt, G.-J. Nam, J. A. Roy, G. Tellez, "What Makes a Design Difficult to Route", *ISPD* 2010, pp. 7-12.

[7] C. J. Alpert, Z. Li, G.-J. Nam, C. N. Sze, N. Viswanathan, S. Ward, "Placement: Hot or Not?" *ICCAD* 2012.

[8] C. J. Alpert, D. P. Mehta, S. S. Sapatnekar (eds.), *Handbook of Algorithms for VLSI Physical Design Automation*, CRC Press 2008.

[9] U. Brenner, "VLSI Legalization with Minimum Perturbation by Iterative Augmentation", *DATE* 2012 pp. 1385-1390.

[10] U. Brenner, A. Rohe, "An Effective Congestion Driven Placement Framework", *ISPD* 2002, pp. 6-11.

[11] U. Brenner, J. Vygen, "Faster Optimal Single-row Placement with Fixed Ordering", *DATE* 2000, pp. 117-121.

[12] U. Brenner, J.Vygen,"Legalizing a Placement with Minimum Total Movement", *TCAD* 23(12) 2004, pp.1597-1613.

[13] U. Brenner, M. Struzyna, J. Vygen, "BonnPlace: Placement of Leading-edge Chips by Advanced Combinatorial Algorithms", *TCAD* 27(9) 2008, pp.1607-1620.

[14] M. Breuer, "Min-cut Placement", *Journal of Design Automation and Fault-tolerant Computing* 10 (1977), pp. 343-382.

[15] M. Burstein, M. N. Youssef, "Timing Influenced Layout Design", *DAC* 1985, pp. 124-130.

[16] A. E. Caldwell, A. B. Kahng, S. Mantik, I. L. Markov, A. Zelikovsky, "On Wirelength Estimations for Row-based Placement", *TCAD* 18(9) 1999, pp. 1265-1278.

[17] A. E. Caldwell, A. B. Kahng, I. L. Markov, "Can Recursive Bisection Alone Produce Routable Placements?" *DAC* 2000, pp. 477-482.

[18] A. E. Caldwell, A. B. Kahng, I. L. Markov, "Optimal Partitioners and End-case Placers for Standard-cell Layout", *TCAD* 19(11) 2000, pp. 1304-1313.

[19] S. Cauley, V. Balakrishnan, Y. C. Hu, C.-K. Koh, "A Parallel Branch-and-Cut Approach for Detailed Placement", *TODAES* 16(2) 2011, no. 18.

[20] T. F. Chan, J. Cong, K. Sze, "Multilevel Generalized Force-directed Method for Circuit Placement", *ISPD* 2005, pp. 185-192.

[21] T. F. Chan, J. Cong, M. Romesis, J. R. Shinnerl, K. Sze, M. Xie, "mPL6: Enhanced Multilevel Mixed-size Placement with Congestion Control", *Modern Circuit Placement*, 2007, pp. 247-288.

[22] T. F. Chan, J. Cong, E. Radke, "A Rigorous Framework for Convergent Net Weighting Schemes in Timing-driven Placement", *ICCAD* 2009, pp. 288-294.

[23] Y.-T. Chang, C.-C. Hsu, M. P.-H. Lin, Y.-W. Tsi, S.-F. Chen, "Post-placement Power Optimization with Multi-bit Flip-flops", *ICCAD* 2010, pp. 218-223.

[24] H. Chang, E. Shragowitz, J. Liu, H. Youssef, B. Lu, S. Sutanthavibul, "Net Criticality Revisited: An Effective Method to Improve Timing in Physical Design", *ISPD* 2002, pp. 155-160.

[25] C. Chang, J. Cong, X. Yuan, "Multi-level Placement for Large-scale Mixed-size IC Design", *ASPDAC* 2003, pp. 325-330.

[26] H.-C. Chen, Y.-L. Chuang, Y.-W. Chang, Y.-C. Chang, "Constraint Graph-based Macro Placement for Modern Mixed-size Circuit Designs", *ICCAD* 2008, pp. 218-223.

[27] T.-C. Chen, A. Chakraborty, D. Z. Pan, "An Integrated Nonlinear Placement Framework with Congestion and Porosity Aware Buffer Planning", *DAC* 2008, pp. 702-707.

[28] T.-C. Chen, M. Cho, D. Z. Pan, Y.-W. Chang, "Metal-density-driven Placement for CMP Variation and Routability", *TCAD* 27(12) 2008, pp. 2145-2155.

[29] T.-C. Chen, P.-H. Yuh, Y.-W. Chang, F.-J. Huang, T.-Y. Liu, "MP-Trees: A Packing-based Macro Placement Algorithm for Modern Mixed-size Designs", *TCAD* 27(9) 2008, pp. 1621-1634.

[30] C. E. Cheng, "RISA: Accurate and Efficient Placement Routability Modeling", *ICCAD* 1994, pp. 650-695.

[31] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, Y.-W. Chang, "NTUPlace3: An Analytical Placer for Large-scale Mixed-size Designs with Preplaced Blocks and Density Constraints", *TCAD* 27(7) 2008, pp.1228-1240.

[32] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda, Q. Wang, "Power-aware Placement", *DAC* 2005, pp. 795-800.

[33] M.-F. Chiang, T. Okamoto, T. Yoshimura, "Register Placement for High-performance Circuits", *DATE* 2009, pp. 1470-1475.

[34] M. Cho, H. Ren, H. Xiang, R. Puri, "History-based VLSI Legalization using Network Flow", *DAC* 2010, pp. 286-291.

[35] W. Choi, K. Bazargan, "Hierarchical Global Floorplacement using Simulated Annealing and Network Flow Area Migration",*DATE* 2003,pp. 1104-5.

[36] S. Chou, M.-K. Hsu, Y.-W. Chang, "Structure-aware Placement for Datapath-intensive Circuit Designs", *DAC* 2012, pp. 762-767.

[37] A. Chowdhary et al., "How Accurately Can We Model Timing in a Placement Engine", *DAC* 2005, pp. 801-806.

[38] C. Chu, M. Pan, "IPR: An Integrated Placement and Routing Algorithm", *DAC* 2007, pp. 59-62.

[39] Y.-L. Chuang, H.-T. Lin, T.-Y. Ho, Y.-W. Chang, D. Marculescu, "PRICE: Power Reduction by Placement and Clock-network Co-synthesis for Pulsed-latch Designs", *ICCAD* 2011, pp. 85-90.

[40] Y.-L. Chuang, G.-J. Nam, C. J. Alpert, Y.-W. Chang, J. A. Roy, N. Viswanathan, "Design-hierarchy Aware Mixed-size Placement for Routability Optimization", *ICCAD* 2010, pp. 663-668.

[41] J. Cong, B. Liu, G. Luo, R. Prabhakar, "Towards Layout-friendly High-level Synthesis", *ISPD* 2012, pp. 165-172.

[42] J. Cong, G. Luo, E. Radke, "Highly Efficient Gradient Computation for Density-constrained Analytical Placement",*TCAD* 27(12) 2008,pp.2133-44.

[43] J. Cong, J. R. Shinnerl, M. Xie, T. Kong, X. Yuan, "Large-scale Circuit Placement", *TODAES* 10(2) 2005, pp. 389-430.

[44] J. Cong, M. Xie, "A Robust Detailed Placement for Mixed-size IC Designs", *ASPDAC* 2006, pp. 188-194.

[45] J. Cong, M. Romesis, J. R. Shinnerl, "Robust Mixed-size Placement Under Tight White-space Constraints", *ICCAD* 2005, pp. 165-172.

[46] K.-R. Dai, C.-H. Lu, Y.-L. Li, "GRPlacer: Improving Routability and Wirelength of Global Routing with Circuit Replacement", *ICCAD* 2009, pp. 351-356.

[47] K. Doll, F. M. Johannes, K. J. Antreich, "Iterative Placement Improvement by Network Flow Methods", *TCAD* 13(10) 1994, pp. 1189-1200.

[48] W. E. Donath, P. Kudva, L. Stok, P. G. Villarrubia, L. N. Reddy, A. Sullivan and K. Chakraborty, "Transformational Placement and Synthesis", *DATE* 2000, 194-201.

[49] A. E. Dunlop, V. D. Agrawal, D. N. Deutsch, M. F. Jukl, P. Kozak, M. Wiesel, "Chip Layout Optimization using Critical Path Weighting", *DAC* 1984, pp. 133-136.

[50] H. Eisenmann, F. M. Johannes, "Generic Global Placement and Floorplanning", *DAC* 1998, pp. 269-274.

[51] T. Gao, P. M. Vaidya, C. L. Liu, "A Performance Driven Macro-cell Placement Algorithm", *DAC* 1992, pp. 147-152.

[52] S. Ghiasi, E. Bozorgzadeh, P.-K. Huang, R. Jafari, M. Sarrafzadeh, "A Unified Theory of Timing Budget Management", *TCAD* 25(11) 2006, pp. 2364-2375.

[53] B. Halpin, C. Y. R. Chen, N. Sehgal, "A Sensitivity Based Placer for Standard Cells", *GSLVLSI*, 2000, pp. 193-196.

[54] B. Halpin, C. Y. R. Chen, N. Sehgal, "Timing Driven Placement using Physical Net Constraints", *DAC* 2001, pp. 780-783.

[55] B. Halpin, C. Y. R. Chen, N. Sehgal, "Detailed Placement with Net Length Constraints", *IWSOC* 2003, pp. 22-27.

[56] T. Hamada, C. K. Cheng, P. M. Chau, "Prime: A Timing-driven Placement Tool using a Piecewise Linear Resistive Network Approach", *DAC* 1993, pp. 531-536, 1993.

[57] P. S. Hauge, R. Nair, E. J. Yoffa, "Circuit Placement for Predictable Performance", *ICCAD* 1987, pp. 88-91.

[58] X. He, T. Huang, L. Xiao, H. Tian, G. Cui, E. F. Young, "Ripple: An Effective Routability-driven Placer by Iterative Cell Movement", *ICCAD* 2011, pp. 74-79.

[59] D. Hill, *Method and System for High Speed Detailed Placement of Cells within an Integrated Circuit Design*, U.S. Patent 6370673, 2001.

[60] T.-Y. Ho, S.-H. Liu, "Fast Legalization for Standard Cell Placement with Simultaneous Wirelength and Displacement Minimization", *VLSI-SoC* 2010, pp. 369-374.

[61] W. Hou, H. Yu, X. Hong, Y. Cai, W. Wu, J. Gu, W. H. Kao, "A New Congestion-driven Placement Algorithm Based on Cell Inflation", *ASPDAC* 2001, pp. 723-728.

[62] M.-K. Hsu, Y.-W. Chang, "Unified Analytical Global Placement for Large-scale Mixed-size Circuit Designs", *ICCAD* 2010, pp. 657-662.

[63] M.-K. Hsu, Y.-W. Chang, V. Balabanov, "TSV-aware Analytical Placement for 3D IC Designs", *DAC* 2011, pp. 664-669.

[64] M.-K. Hsu, S. Chou, T.-H. Lin, Y.-W. Chang, "Routability-driven Analytical Placement for Mixed-size Circuit Designs", *ICCAD* 2011, pp. 80-84.

[65] B. Hu, M. Marek-Sadowska, "Congestion Minimization during Placement without Estimation", *ICCAD* 2002, pp. 739-745.

[66] B. Hu, M. Marek-Sadowska, "Multilevel Fixed-point-addition-based VLSI Placement", *TCAD* 24(8) 2005, pp. 1188-1203.

[67] J. Hu, A. B. Kahng, S.-H. Kang, M.-C. Kim, I. L. Markov, "Sensitivity-guided Metaheuristics for Accurate Discrete Gate Sizing", *ICCAD* 2012.

[68] J. Hu, Y. Shin, N. Dhanwada, R. Marculescu, "Architecting Voltage Islands in Core-based System-on-a-Chip Designs", *ISLPED* 2004, pp. 180-5.

[69] J. Hu, J. A. Roy, I. L. Markov, "Completing High-quality Routes", *ISPD* 2010, pp. 35-41.

[70] T. C. Hu, K. Moerder, "Multiterminal Flows in a Hypergraph", *VLSI Circuit Layout: Theory and Design (T. C. Hu, E. S. Kuh, eds.)*, IEEE, 1985.

[71] S. Hur, T. Cao, K. Rajagopal, Y. Parasuram, A. Chowdhary, V. Tiourin, B. Halpin, "Force Directed Mongrel with Physical Net Constraints", *DAC* 2003, pp. 214-219.

[72] S.-W. Hur, J. Lillis, "Mongrel: Hybrid Techniques for Standard Cell Placement", *ICCAD* 2000, pp. 165-170.

[73] International Technology Roadmap for Semiconductors (ITRS). `http://public.itrs.net`

[74] M. A. B. Jackson, E. S. Kuh, "Performance-driven Placement of Cell Based ICs", *DAC* 1989, pp. 370-375.

[75] D. Jariwala, J. Lillis, "RBI: Simultaneous Placement and Routing Optimization Technique", *TCAD* 26(1) 2007, pp. 127-141.

[76] Z.-W. Jiang, B.-Y. Su, Y.-W. Chang, "Routability-driven Analytical Placement by Net Overlapping Removal for Large-scale Mixed-size Designs", *DAC* 2008, pp. 167-172.

[77] H.-R. Jiang, C.-L. Chang, Y.-M. Yang, "INTEGRA: Fast Multibit Flip-flop Clustering for Clock Power Saving", *TCAD* 31(2) 2012, pp. 192-204.

[78] A. B. Kahng, "Classical Floorplanning Harmful?" *ISPD* 2000, pp. 207-213.

[79] A. B. Kahng, J. Lienig, I. L. Markov, J. Hu, "VLSI Physical Design: From Graph Partitioning to Timing Closure", Springer 2011.

[80] A. B. Kahng, I. L. Markov, S. Reda, "On Legalization of Row-based Placements", *GSLVLSI*, 2004, pp. 214-219.

[81] A. B. Kahng, C.-H. Park, P. Sharma, Q. Wang, "Lens Aberration Aware Placement for Timing Yield", *TODAES* 14(1) 2009, no. 16.

[82] A. B. Kahng, P. Tucker, A. Zelikovsky, "Optimization of Linear Placements for Wirelength Minimization with Free Sites",*ASPDAC* 1999,pp. 241-4.

[83] A. B. Kahng, Q. Wang, "A Faster Implementation of APlace", *ISPD* 2006, pp. 218-220.

[84] J. Knechtel, I. L. Markov, J. Lienig, "Assembling 2-D Blocks Into 3-D Chips", *TCAD* 31(2) 2012, pp. 228-241.

[85] M.-C. Kim, D.-J. Lee, I. L. Markov, "SimPL: An Effective Placement Algorithm", *ICCAD* 2010, pp. 649-656.

[86] M.-C. Kim, J. Hu, D.-J. Lee, I. L. Markov, "A SimPLR Method for Routability-driven Placement", *ICCAD* 2011, pp. 67-73.

[87] M.-C. Kim, D.-J. Lee, I. L. Markov, "SimPL: An Effective Placement Algorithm", *TCAD* 31(1) 2012, pp.50-60.

[88] M.-C. Kim, I. L. Markov, "ComPLx: A Competitive Primal-dual Lagrange Optimization for Global Placement", *DAC* 2012, pp. 747-752.

[89] M.-C. Kim, N. Viswanathan, C. J. Alpert, I. L. Markov, S. Ramji, "MAPLE: Multilevel Adaptive PLacEment for Mixed-size Designs", *ISPD* 2012, pp. 193-200.

[90] T. Kong, "A Novel Net Weighting Algorithm for Timing-driven Placement", *ICCAD* 2002, pp. 172-176.

[91] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D. W. Stout, S. W. Gould and J. M. Cohn, "Managing Power and Performance for System-on-Chip Designs using Voltage Islands", *ICCAD* 2002, pp. 195-202.

[92] D.-J. Lee, I. L. Markov, "Obstacle-aware Clock-tree Shaping during Placement", *TCAD* 31(2) 2012, pp. 205-216.

[93] Y.-M. Lee, T.-Y. Wu, P.-Y. Chang, "A Hierarchical Bin-based Legalizer for Standard-cell Designs with Minimal Disturbance", *ASPDAC* 2010, pp. 568-573.

[94] C. Li, C.-K. Koh, "Recursive Function Smoothing of Half-perimeter Wirelength for Analytical Placement", *ISQED*, 2007, pp. 829-834.

[95] C. Li, C.-K. Koh, P. H. Madden, "Floorplan Management: Incremental Placement for Gate Sizing and Buffer Insertion",*ASPDAC* 2005,pp. 349-54.

[96] C. Li, M. Xie, C.-K. Koh, J. Cong, P. H. Madden, "Routability-driven Placement and White Space Allocation", *TCAD* 26(5) 2007, pp. 858-871.

[97] S. Li, C.-K. Koh, "Mixed Integer Programming Models for Detailed Placement", *ISPD* 2012, pp. 87-94.

[98] Z. Li, C. N. Sze, C. J. Alpert, J. Hu, W. Shi, "Making Fast Buffer Insertion Even Faster via Approximation Techniques", *ASPDAC* 2005, pp. 13-18.

[99] Z. Li, W. Wu, X. Hong, "Congestion Driven Incremental Placement Algorithm for Standard Cell Layout", *ASPDAC* 2003, pp. 723-728.

[100] B. Liu, Y. Cai, Q. Zhou, X. Hong, "Power Driven Placement with Layout Aware Supply Voltage Assignment for Voltage Island Generation in Dual-Vdd Designs", *ASPDAC* 2006, pp. 582-587.

[101] W. K. Luk, "A Fast Physical Constraint Generator for Timing Driven Placement", *DAC* 1991, pp. 626-631.

[102] T. Luo, D. Newmark, D. Z. Pan, "A New LP Based Incremental Timing Driven Placement for High Performance Designs", *DAC* 2006, pp. 1115-20.

[103] T. Luo, D. Z. Pan, "DPlace2.0: A Stable and Efficient Analytical Placement Based on Diffusion", *ASPDAC* 2008, pp. 346-351.

[104] T. Luo, H. Ren, C. J. Alpert, D. Z. Pan, "Computational Geometry Based Placement Migration", *DAC* 2007, pp. 41-47.

[105] Y. Lu, C. N. Sze, X. Hong, Q. Zhou, Y. Cai, L. Huang, J. Hu, "Navigating Registers in Placement for Clock Network Minimization", *DAC* 2005, pp. 176-181.

[106] N. Magen, A. Kolodny, U. Weiser, N. Shamir, "Interconnect-power Dissipation in a Microprocessor", *SLIP* 2004, pp. 7-13.

[107] M. D. Moffitt, J. A. Roy, I. L. Markov, M. E. Pollack, "Constraint-driven Floorplan Repair", *TODAES* 13(4) 2008, pp. 1-13.

[108] R. Nair, C. L. Berman, P. S. Hauge, E. J. Yoffa, "Generation of Performance Constraints for Layout", *TCAD* 8(8) 1989, pp. 860-874.

[109] G.-J. Nam, J. Cong, *Modern Circuit Placement: Best Practices and Results*, Springer 2007.

[110] B. Obermeier, F. M. Johannes, "Temperature-aware Global Placement", *ASPDAC* 2004, pp. 143-148.

[111] B. Obermeier, H. Ranke, F. M. Johannes, "Kraftwerk: a Versatile Placement Approach", *ISPD* 2005, pp. 242-244.

[112] M. Orshansky, S. Nassif, D. Boning, *Design for Manufacturability and Statistical Design: A Constructive Approach (Integrated Circuits and Systems)*, Springer 2010.

[113] M. M. Ozdal, C. Amin, A. Ayupov, S. Burns, G. Wilke, C. Zhuo, "The ISPD-2012 Discrete Cell Sizing Contest and Benchmark Suite", *ISPD* 2012, pp. 161-164.

[114] M. Pan, C. Chu, "FastRoute: A Step to Integrate Global Routing into Placement", *ICCAD* 2006, pp. 59-62.

[115] M. Pan, N. Viswanathan, C. Chu, "An Efficient and Effective Detailed Placement Algorithm", *ICCAD* 2005, pp. 48-55.

[116] D. A. Papa, S. Krishnaswamy, I. L. Markov, "SPIRE: A Retiming-based Physical-synthesis Transformation System", *ICCAD* 2010, pp. 373-380.

[117] D. A. Papa, M. D. Moffitt, C. J. Alpert, I. L. Markov, "Speeding Up Physical Synthesis with Transactional Timing Analysis", *Design and Test* 27(5) 2010, pp. 14-25.

[118] D. A. Papa et al., "RUMBLE: An Incremental, Timing-driven, Physical-synthesis Optimization Algorithm", *TCAD* 27(12) 2008, pp. 2156-2168.

[119] D. A. Papa, N. Viswanathan, C. N. Sze, Z. Li, G.-J. Nam, C. J. Alpert, I. L. Markov, "Physical Synthesis with Clock-Network Optimization for Large Systems on Chips", *Micro* 31(4) 2011, pp. 51-62.

[120] P. N. Parakh, R. B. Brown, K. A. Sakallah, "Congestion Driven Quadratic Placement", *DAC* 1998, pp. 275-278.

[121] S. Plaza, I. L. Markov, V. Bertacco, "Optimizing Non-Monotonic Interconnect using Functional Simulation and Logic Restructuring", *TCAD* 27(12) 2008, pp. 2107-2119.

[122] R. Puri, L. Stok, S. Bhattacharya, "Keeping Hot Chips Cool", *DAC* 2005, pp. 285-288.

[123] R. Puri et al., "Pushing ASIC Performance in a Power Envelope", *DAC* 2003, pp. 788-793.

[124] K. Rajagopal et al., "Timing Driven Force Directed Placement with Physical Net Constraints", *ISPD* 2003, pp. 60-66.

[125] B. M. Reiss, G. G. Ettelt, "SPEED: Fast and Efficient Timing Driven Placement", *ISCAS* 1995, pp. 377-380.

[126] H. Ren, D. Z. Pan, C. J. Alpert, P. G. Villarrubia, "Diffusion-based Placement Migration", *DAC* 2005, pp. 515-520.

[127] H. Ren, D. Z. Pan, C. J. Alpert, G.-J. Nam, P. G. Villarrubia, "Hippocrates: First-Do-No-Harm Detailed Placement", *ASPDAC* 2007, pp. 141-146.

[128] H. Ren, D. Z. Pan, D. S. Kung, "Sensitivity Guided Net Weighting for Placement Driven Synthesis", *TCAD* 24(5) 2005, pp. 711-721.

[129] J. A. Roy, S. N. Adya, D. A. Papa, I. L. Markov, "Min-cut Floorplacement", *TCAD* 25(7) 2006, pp.1313-1326.

[130] J. A. Roy, I. L. Markov, "Seeing the Forest and the Trees: Steiner Wirelength Optimization in Placement", *TCAD* 23(4) 2007, pp. 632-644.

[131] J. A. Roy, I. L. Markov, "ECO-System: Embracing the Change in Placement", *TCAD* 26(12) 2007, pp. 2173-2185.

[132] J. A. Roy, A. N. Ng, R. Aggarwal, V. Ramachandran, I. L. Markov, "Solving Modern Mixed-size Placement Instances", *Integration* 42(2) 2009, pp. 262-275.

[133] J. A. Roy et al., "Capo: Robust and Scalable Open-source Min-Cut Floorplacer", *ISPD* 2005, pp. 224-226, `vlsicad.eecs.umich.edu/BK/PDtools/`.

[134] J. A. Roy, N. Viswanathan, G.-J. Nam, C. J. Alpert, I. L. Markov, "CRISP: Congestion Reduction by Iterated Spreading during Placement", *ICCAD* 2009, pp. 357-362.

[135] A. E. Ruehli, P. K. Wolff, G. Goertzel, "Analytical Power/Timing Optimization Technique for Digital Systems", *DAC* 1977, pp. 142-146.

[136] S. M. Sait, M. R. Minhas, J. A. Khan, "Performance and Low Power Driven VLSI Standard Cell Placement using Tabu Search", *Congress on Evolutionary Computation*, 2002, pp. 372-377.

[137] M. Sarrafzadeh, M. Wang, "NRG: Global and Detailed Placement", *ICCAD* 1997, p. 532-537.

[138] P. Saxena, "On Controlling Perturbation Due to Repeaters during Quadratic Placement", *TCAD* 25(9) 2006, pp. 1733-1743.

[139] N. Selvakkumaran, P. N. Parakh, G. Karypis, "Perimeter-degree: A Priori Metric for Directly Measuring and Homogenizing Interconnection Complexity in Multilevel Placement", *SLIP* 2003, pp. 53-59.

[140] W. Shen, Y. Cai, X. Hong, J. Hu, "Activity and Register Placement Aware Gated Clock Network Design", *ISPD* 2008, pp. 182-189.

[141] R. E. Showalter, "Monotone Operators in Banach Space and Nonlinear Partial Differential Equations", *Mathematical Surveys and Monographs* 49. Providence, RI: American Mathematical Society, 1997, pp. 162-163.

[142] G. Sigl, K. Doll, F. Johannes,"Analytical Placement: A Linear or a Quadratic Objective Function?" *DAC* 1991, pp. 427-432.

[143] P. Spindler, F. M. Johannes, "Fast and Accurate Routing Demand Estimation for Efficient Routability-driven Placement",*DATE* 2007, pp. 1226-31.

[144] P. Spindler, U. Schlichtmann, F. M. Johannes, "Abacus: Fast Legalization of Standard Cell Circuits with Minimal Movement", *ISPD* 2008, pp. 47-53.

[145] P. Spindler, U. Schlichtmann, F.M. Johannes, "Kraftwerk2 - A Fast Force-directed Quadratic Placement Approach using an Accurate Net Model", *TCAD* 27(8) 2008, pp. 1398-1411.

[146] A. Srinivasan, K. Chaudhary, E. S. Kuh, "Ritual: A Performance Driven Placement Algorithm for Small Cell ICs", *ICCAD* 1991, pp. 48-51.

[147] W. Swartz, C. Sechen, "Timing Driven Placement for Large Standard Cell Circuits", *DAC* 1995, pp. 211-215.

[148] H. Tennakoon, C. Sechen, "Nonconvex Gate Delay Modeling and Delay Optimization", *TCAD* 27(9) 2003, pp. 1583-1594.

[149] M. Terai, K. Takahashi, K. Sato, "A New Min-cut Placement Algorithm for Timing Assurance Layout Design Meeting Net Length Constraint", *DAC* 1990, pp. 96-102.

[150] L. Trevillyan, D. Kung, R. Puri, L. N. Reddy, M. A. Kazda, "An Integrated Environment for Technology Closure of Deep-Submicron IC Designs", *Design and Test* 21(1) 2004, pp. 14-22.

[151] R. S. Tsay, J. Koehl, "An Analytic Net Weighting Approach for Performance Optimization in Circuit Placement", *DAC* 1991, pp. 636-639.

[152] K. Tsota, C. Koh, V. Balakrishnan, "Guiding Global Placement with Wire Density", *ICCAD* 2008, pp. 212-217.

[153] H. Vaishnav, M. Pedram, "PCUBE: A Performance Driven Placement Algorithm for Low Power Designs", *DAC with EURO-VHDL*, 1993, pp.72-77.

[154] L. P. P. P. van Ginneken, "Buffer Placement in Distributed RC-tree Network for Minimal Elmore Delay", *ISCAS* 1990, pp. 865-868.

[155] N. Viswanathan, C. J. Alpert, Z. Li, G.-J. Nam, J. A. Roy, "The ISPD-2011 Routability-driven Placement Contest and Benchmark Suite", *ISPD* 2011, pp. 141-146.

[156] N. Viswanathan, C. J. Alpert, C. N. Sze, Z. Li, Y. Wei, "The DAC 2012 Routability-driven Placement Contest and Benchmark Suite", *DAC* 2012, pp. 774-782.

[157] N. Viswanathan, M. Pan, C. Chu, "FastPlace 3.0: A Fast Multilevel Quadratic Placement Algorithm with Placement Congestion Control", *ASPDAC* 2007, pp. 135-140.

[158] N. Viswanathan, G.-J. Nam, C. J. Alpert, P. G. Villarrubia, H. Ren, "RQL: Global Placement via Relaxed Quadratic Spreading and Linearization", *DAC* 2007, pp. 453-458.

[159] M. Vujkovic, D. Wadkins, B. Swartz, C. Sechen, "Efficient Timing Closure without Timing Driven Placement and Routing",*DAC* 2004, pp. 268-73.

[160] S. Ward, D. Ding, D. Z. Pan, "PADE: A High-performance Placer with Automatic Datapath Extraction and Evaluation Through High Dimensional Data Learning", *DAC* 2012, pp. 756-761.

[161] M. Wang, M. Sarrafzadeh, "On the Behaviour of Congestion Minimization during Placement", *ISPD* 1999, pp. 145-150.

[162] M. Wang, M. Sarrafzadeh, "Model and Minimization of Routing Congestion", *ASPDAC* 2000, pp. 185-190.

[163] M. Wang X. Yang, M. Sarrafzadeh, "Congestion Minimization during Placement", in *TCAD* 19(10) 2000, pp. 1140-1148.

[164] M. Wang, X. Yang, K. Eguro, M. Sarrafzadeh, "Multicenter Congestion Estimation and Minimization during Placement",*ISPD* 2000, pp. 147-152.

[165] Y. Wang, Q. Zhou, X. Hong, Y. Cai, "Clock-tree Aware Placement Based on Dynamic Clock-tree Building", *ISCAS* 2007, pp. 2040-2043.

[166] E. Wein, J. Benkoski, "Hard Macros Will Revolutionize SoC Design", *EE Times Online*, August 20, 2004.
`http://www.eetimes.com/news/design/showArtical.jhtml?articalID=26807055`

[167] J. Werber, D. Rautenbach, C. Szegedy, "Timing Optimization by Restructuring Long Combinatorial Paths", *ICCAD* 2007, pp. 536-543.

[168] J. Westra, C. Bartels, P. Groeneveld, "Probabilistic Congestion Prediction", *ISPD* 2004, pp. 204-209.

[169] J. Westra, P. Groeneveld, "Is Probabilistic Congestion Estimation Worthwhile?" *SLIP* 2005, pp. 99-106.

[170] B. P. Wong, A. Mittal, G. W. Starr, F. Zach, V. Moroz, A. Kahng, *Nano-CMOS Design for Manufacturability*, John Wiley and Sons 2009.

[171] H. Xiang, H. Ren, L. Trevillyan, L. Reddy, R. Puri, M. Cho, "Logical and Physical Restructuring of Fan-in Trees", *ISPD* 2010, pp. 67-74.

[172] Zhong Xiu, R. Rutenbar, "Timing-driven Placement by Grid-warping", *DAC* 2005, pp. 585-590.

[173] Y. Xu, Y. Zhang, C. Chu, "FastRoute 4.0: Global Router with Efficient Via Minimization", *ASPDAC* 2009, pp. 576-581.

[174] J. Z. Yan, N. Viswanathan, C. Chu, "Handling Complexities in Modern Large-scale Mixed-size Placement", *DAC* 2009, pp. 436-441.

[175] X. Yang, B.-K. Choi, M. Sarrafzadeh, "Routability-driven White Space Allocation for Fixed-die Standard-cell Placement", *TCAD* 22(4) 2003, pp. 410-419.

[176] C. Yeh, Y. Kang, S. Shieh, J. Wang, "Layout Techniques Supporting the Use of Dual Supply Voltages for Cell-based Designs",*DAC* 1999, pp. 62-67.

[177] H. Youssef, E. Shragowitz, "Timing Constraints for Correct Peformance", *ICCAD* 1990, pp. 24-27.

[178] Y. Zhang, C. Chu, "CROP: Fast and Effective Congestion Refinement of Placement", *ICCAD* 2009, pp. 344-350.

[179] K. Zhong, S. Dutt, "Algorithms for Simultaneous Satisfaction of Multiple Constraints and Objective Optimization in a Placement Flow with Application to Congestion Control", *DAC* 2002, pp. 854-859.