Physical Synthesis for Power under Process Variation

Jason Cong  cong@cs.ucla.edu  UCLA Computer Science
Tony F. Chan  chan@math.ucla.edu  UCLA Mathematics
Lieven Vandenberghe  vandenbe@ee.ucla.edu  UCLA Electrical Engineering

March 15, 2006

1 Introduction

Power is widely considered the only real limiter for Moore’s Law in the next decade [Gelsinger, 2004]. Indeed, IBM CTO B. Meyerson has claimed that CMOS process scaling has already stopped between 130nm and 90nm due to power limitations [Clarke, 2004]. Existing methods for physical synthesis, however, have primarily targeted timing closure and are inadequate to meet the challenge of low-power physical design under process variation. The current lack of a general formulation to simultaneously consider and balance all objectives and constraints leads to potentially gross suboptimality.

The goal of the research proposed here is the formulation and implementation of a complete placement-centric physical synthesis flow to minimize total power consumption of an IC design scalably under evolving timing and netlist constraints in the presence of uncertain wire-and-device characteristics. Complete means simultaneous optimization over the following sets of design variables under tight timing constraints.

1. Locations \( z_i = (x_i, y_i) \) for modules \( i = 1, \ldots, N \)
2. Supply voltages \( V_{dd} \) and threshold voltages \( V_{th} \), assigned to modules from a fine-grain voltage-island power network
3. Module sizes \( l_i \)
4. Gate oxide thickness \( t_{ox} \)
5. Effective channel length \( L_{eff} \)

For brevity, these variables are referred to collectively as \( q = (z, v, l, t_{ox}, L_{eff}) \).

Variables \( V_{dd} \) and \( l \) impact dynamic power, while variables \( V_{th}, L_{eff}, \) and \( t_{ox} \) impact static power through current leakage. Although each module will be given a distinct location \( z_i \) and possibly a different size \( l_i \), the variables \( V_{dd}, V_{th}, t_{ox}, \) and \( L_{eff} \) are ultimately assigned not independently to individual modules but rather to subregions based on the overhead introduced by level converters and the granularity afforded by the available manufacturing technology. Nevertheless, fine-grain voltage islands have been demonstrated based on the novel design of low-cost level converters [Puri et al., 2003]. Similarly, given the demands of aggressive power minimization, it is expected that the best assignments of \( V_{th}, L_{eff}, \) and \( t_{ox} \) will vary over many regions on the chip, producing much higher complexity for optimization. It is important to consider the optimization of these variables with placement, because placement (a) defines the interconnects and provides the load and density information for sizing; and (b) is needed for assuring the spatial locality required (or preferred) for multiple \( V_{dd}, V_{th}, L_{eff}, \) and \( t_{ox} \) assignment as well as for power and clock gating. It is also important to optimize these variables simultaneously, as they all compete for the same timing slack in power minimization. Previous work on power minimization has considered (i) adjusting only proper subsets of these variables simultaneously, often under a fixed placement, or (ii) sequential optimization over multiple variables. Either of these previous approaches is likely to result in inefficient use of timing slack and a suboptimal power result.

Recent advances in both circuit modeling under uncertainty and optimization under uncertainty will also be used to explicitly incorporate uncertainty due to process variation into all objectives and constraints. This explicit modeling of uncertainty will enable both aggressive reduction in total power usage and reliable attainment of timing-yield targets.

The proposed algorithm will be built on a scalable and unified multilevel flow for a combination of strictly robust, scenario-based, and chance-constrained formulations of robust optimization under uncertainty. The multilevel flow for global optimization, successfully adapted to wirelength-driven placement by the PIs in SRC Tasks 686.001 and 1091.001, is ideally suited to the simultaneous incorporation of diverse and complex constraints. Early, explicit, and consistent handling of these constraints will ultimately match the performance of critical regions to their timing constraints by giving them sufficient power, while minimizing power everywhere else. Optimization subproblems at each level of the multilevel hierarchy will be formulated and solved rigorously, with (i) all objectives and constraints explicitly modeled in a mathematical-programming formulation, and (ii) the proposed algorithms explicitly targeting that formulation. The multilevel formulation will support thorough exploration of the trade-offs between multiple \( V_{dd} \), multiple \( V_{th} \), module sizes, module locations, \( t_{ox} \), and \( L_{eff} \) in the optimization of total dynamic and leakage power.

The remainder of the proposal is organized as follows. Section 2 states the mathematical formulation of physical synthesis in both its deterministic and robust forms. Section 3 describes the outer multilevel strategy for building a scalable and robust solution method. Section 4 describes the optimization algorithms used at each level of the multilevel hierarchy. Section 5 describes how the robust, timing-yield constrained formulation will be reduced to deterministic form and solved.

2 Problem Formulation

Given a provisional netlist with timing constraints and net-switching statistics, we seek a complete geometrical specification of its layout such that total power consumption is minimized and the likelihood that all timing constraints are satisfied under the uncertainty of process variation is greater than a prescribed lower limit. The formulation under uncertainty is presented as an extension of the deterministic formulation below.
2.1 Deterministic Form

In terms of leakage power \( P_{\text{leak}} \), dynamic power \( P_{\text{dyn}} \), and an a-priori estimate \( K \) of the ratio \( P_{\text{dyn}}/P_{\text{leak}} \), the proposed deterministic formulation of power-driven physical synthesis is

\[
\begin{align*}
\text{minimize} & \quad P_{\text{leak}}(q) + KP_{\text{dyn}}(q) \\
\text{subject to} & \quad u_j(q) \leq \bar{u}_j \quad \text{for } j = 1, 2, \ldots \\
& \quad t_i(q) + d_i(q) + e_i(q) \leq t_j(q) \quad \text{all edges } (i, j) \\
& \quad P \left( t_k(q) \leq r_k \quad \text{all } k \in \text{PO} \right) \geq Y_{\text{min}}.
\end{align*}
\]

where \( q = (z, v, l, t_{\text{ox}}, L_{\text{eff}}) \) is the set of design variables defined in Section 1. The functions \( u_j \) with respective upper bounds \( \bar{u}_j \) represent generalized Poisson-based density constraints for modeling global features such as area utilization, routability, heat distribution, etc. [Eisenmann and Johannes, 1998, Chan et al., 2005c]. As usual, required arrival times \( r_k \) at primary outputs \( k \in \text{PO} \) are back propagated to the modules in order to obtain a tractable formulation; \( t_i(q) \) denotes the arrival time of a signal at node \( i \), \( d_i(q) \) is the delay at node \( i \), and \( e_i(q) \) is a precise a-priori estimate of the optimized propagation delay from node \( i \) to node \( j \) attainable after buffering and sizing [Cong and Pan, 2001].

Static power is dominated by leakage power \( P_{\text{leak}} \). For a gate with effective channel length \( L_{\text{eff}} \), a standard empirical model of leakage power [Kao et al., 2002, Mani et al., 2005]

\[
P_{\text{leak}} = V_{\text{gs}}^2 e^\frac{1}{2} \mu_{\text{ox}} e \frac{V_{\text{dd}} - V_{\text{th}}}{L_{\text{eff}}} \left( 1 - e^{-V_{\text{dd}}/V_T} \right) V_{\text{dd}},
\]

where \( V_{\text{gs}} \) denotes voltage between gate and source, \( V_{\text{dd}} \) denotes voltage between drain and source, \( V_T \) denotes thermal voltage, and \( \mu \) denotes carrier mobility.

Dynamic power is dominated by net-switching power, which can be modeled as \( kCV_{\text{dd}}\alpha f \), where \( k \) is a constant, \( C = C(q) \) is the total parasitic and load capacitance due to wires and gate-input pins to be charged and discharged, and \( \alpha f \) is the switching rate; i.e., the expected number of transition events per unit time [Cheon et al., 2005]. Capacitance is a function of module positions, module sizes, and net lengths; total weighted half-perimeter wirelength is a standard and effective approximation to the objective at early stages [Sarrafzadeh et al., 2002].

2.2 Statistical Form

In sub-100nm IC’s, random and systematic variation in gate channel length, doping density, wire length and width must be explicitly modeled in order to obtain reliable yields. The statistical formulation of physical synthesis is obtained by (a) interpreting \( v_i, t_i, d_i, l, t_{\text{ox}}, \) and \( L_{\text{eff}} \) in (1) as spatially correlated random variables and (b) requiring that the arrival-time constraints hold only with a certain minimum yield probability \( Y_{\text{min}} \).

\[
\begin{align*}
\text{minimize} & \quad P_{\text{leak}}(q) + KP_{\text{dyn}}(q) \\
\text{subject to} & \quad u_j(q) \leq \bar{u}_j \quad \text{for } j = 1, 2, \ldots \\
& \quad t_i(q) + d_i(q) + e_i(q) \leq t_j(q) \quad \text{all edges } (i, j) \\
& \quad P \left( t_k(q) \leq r_k \quad \text{all } k \in \text{PO} \right) \geq Y_{\text{min}}.
\end{align*}
\]

Many of the random variables can be modeled as Normal or Log-Normal [Mani et al., 2005] with known means and variances; but in general, their distributions are unknown. Systematic effects in manufacturing are manifested as spatial correlations. Accurate formulation of the variations in \( v_i, t_i, d_i, l, t_{\text{ox}}, \) and \( L_{\text{eff}} \) as determined by fundamental physical processes is another goal of the proposed research. Proposed techniques for the deterministic reformulation and solution of (2) are described in Section 5.

3 Combined Multilevel Algorithm

The outer flow of the core algorithm combines optimization by simultaneous placement, module sizing, voltage assignment, and \( t_{\text{ox}} \) and \( L_{\text{eff}} \) optimization in a multilevel formulation. 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.

Multilevel optimization consists of four main elements [Brandt, 1986, Cong and Shinner, 2003]: coarsening, relaxation, interpolation, and iteration flow.

Coarsening — recursive aggregation, or generalized clustering, produces order \( \log(N) \) approximations or cluster levels of the original problem (1), each smaller than its predecessor by a prescribed factor. Each cluster level is defined by an explicit mapping of design variables, objective, and constraints at its adjacent finer level to corresponding variables and functions at the cluster level. Thus, at each level, all modules in the same cluster will have one location, the same \( V_{\text{dd}}, V_{\text{th}}, \) and \( t_{\text{ox}} \), and the same scaling factor for gate sizes and \( L_{\text{eff}} \), resulting in a much simplified problem.

Relaxation — optimization within each level begins with a solution inherited from an adjacent level and adjusts all design variables either simultaneously or in some sequence in order to reduce the objective until a stopping criterion is met.

Interpolation — A solution at one level is transformed to a solution at the adjacent finer level by mapping values of the design variables associated with coarser-level aggregates to values of design variables associated with their finer-level components. Thus, values for clusters’ locations, \( V_{\text{dd}}, V_{\text{th}}, l, L_{\text{eff}}, \) and \( t_{\text{ox}} \) are refined to values for the modules or sub-clusters composing them.

Iteration Flow — In the simplest V-cycle form of multilevel optimization, a single pass of recursive aggregation is followed by aggressive coarsest-level optimization and a single pass of recursive interpolation and relaxation. However, recursive correction by frequent recoarsening under evolving aggregation criteria is widely cited in the literature as a more effective approach for difficult problems [Brandt and Ron, 2003].

Proposed strategies for relaxation are derived from (i) generalized Poisson-based force-directed placement [Chan et al., 2005c] and (ii) recent advances in linear and geometric programming. These strategies are discussed in more detail in Section 4. Formulations of coarsening and interpolation specific to low-power physical synthesis are considered below.

Although placement, sizing, and voltage assignment may well be considered either separately or jointly within each cluster level, the multilevel flow ensures that solution of Formulation (1) proceeds jointly over all design variables.
3.1 Coarsening

The multilevel hierarchy is built by recursive aggregation [Brandt and Ron, 2003]. The aggregation algorithm first quantifies the affinity each module has for its netlist neighbors. Affinities between vertices can be based on logic dependency or logic hierarchy (the logic hierarchy is preferred, if possible, for ease of verification, test, and ECQ); timing constraints (modules on the critical paths will be clustered), constraints on \( V_{dd}, V_{th}, L_{eff}, \) and \( t_{ox} \) assignments (some modules may be required or preferred to have the same assignment), netlist connectivity (such as by First-Choice [Karaypis, 2003] or Best-Choice [Alpert et al., 2005]), and/or geometry derived from intermediate placement results (especially in a multi-V-cycle flow [Briggs et al., 2000]). The use of different affinity functions and their impact on the final quality of result will be investigated. Initial experiments with clustering schemes will focus on the logic hierarchy and timing constraints.

Although modules in the given netlist are generally assumed to have single outputs, aggregates of modules defining coarse-level variables must be given multiple outputs in order to maintain an accurate timing view at all levels of hierarchy. “Timing views” of each cluster will be constructed, and the constraints (1) will be aggregated to correctly propagate timing data both within clusters and between clusters.

3.2 Interpolation

Interpolation may be viewed approximately as the inverse of coarsening. However, rather than simply transfer each cluster’s variables’ values to all its components, a more sophisticated mapping can be used that takes the finer-level view of the circuit into account [Chan et al., 2003]. In order for the multilevel formulation to improve quality over the traditional, sequential application of placement, voltage assignment, sizing, etc. to a flat netlist, two key conditions must be met. First, the interpolation of a solution from one level to the next must satisfy any necessary restrictions, such as timing feasibility. Second, a solver at one level of hierarchy must make good use of a solution interpolated from an adjacent level as starting point; i.e., the ability of the solver to make “warm starts,” good initial guesses of the optimal solution, is crucial. If these conditions are not met, there is no reason to expect that the multilevel solution will outperform the traditional approach. Warm starts are considered in more detail in Section 4. Finally, when the variables \( V_{dd}, V_{th}, L_{ox}, \) and \( t_{eff} \) are refined for cluster \( i \), the minimum allowed cluster or region sizes due to the restrictions from manufacturing and/or the design of the power supply networks will be considered. Assignments of these variables will not be interpolated beyond their minimum allowed sizes.

4 Intralevel Optimization

Relaxation at each cluster level consists of three major steps in the following sequence.

(i) Global placement under multiple density constraints (for area, routability, heat density, etc.);

(ii) simultaneous module sizing and assignments of \( V_{dd}, V_{th}, L_{eff}, \) and \( t_{ox} \) based on a generalized geometric-programming formulation similar to that in [Boyd et al., 2005] in order to reduce power as much as possible by making efficient use of available slacks;

(iii) legalization for exact module locations and use of feasible discrete \( V_{dd}, V_{th}, L_{eff}, \) and \( t_{ox} \) values.

At each intermediate configuration of an iterative algorithm, the change in power with respect to changes in the separate variables can be modeled following the Zuyban and Strenski marginal-cost/sensitivity model of hardware intensity:

\[
\theta(X) = \sum_{i} -D \frac{\partial E}{\partial x_i} \bigg|_{x=X}.
\]

Energy-efficient design is achieved when the marginal costs of all the tuning variables are balanced [Brodersen et al., 2002].

4.1 Global Placement

Global placement has three objectives (i) minimize the total interconnect capacitance weighted netwise by switching activity; (ii) maximize the total slack for subsequent optimization 1, and (iii) enhance spatial locality for \( V_{dd}, V_{th}, L_{eff}, \) and \( t_{ox} \) assignment as well as power and clock gating. How to balance the three factors will be important part of the research. It is expected that objectives (i) and (ii) are somewhat consistent and can be combined in a weighted sum. The third objective may be expressed in terms of the additional affinity functions to be used in coarsening and placement. Sequential circuit elements may be clustered near clock-tree leaves, and nets may be weighted by switching activity [Cheon et al., 2005].

The placement engine will build on the mP.6 package of the PIs’ current SRC Task 1091.001. mP.6’s approach to placement generalizes the analytical force-directed framework of Eisenmann and Johannes in two ways [Chan et al., 2005c]. First, mP.6 incorporates force-directed placement within a multilevel-placement engine as intralevel relaxation. This approach leads to improvement in both scalability and solution quality. Second, mP.6 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 adhoc tuning. For further details, please see the paper on mP.5 [Chan et al., 2005c].

Poisson-based methods for density-constraint satisfaction directly applies only to density constraints formulated as inequalities. The density inequalities in (1) are therefore replaced by equalities via the introduction of nonnegative artificial “density slack” variables \( d_{SI} \), one for each bin of a uniform grid laid over the placement region. These can be interpreted as deriving from artificial unconnected “filler” cells added to underutilized regions in order to allow the given, interconnected cells to assume non-uniform configurations.

4.2 Incorporating Multilevel Linear and Geometric Programming

Scalable, high-quality placement under generalized density constraints has already been successfully developed by the PIs in mP.6, and its extension to maximize timing slack is not expected to cause difficulty. The challenge lies in incorporating sizing, voltage assignment, and interconnect performance optimization into the multilevel flow in a way that minimizes total power. Thus, although efficient, high-quality multilevel algorithms for device sizing, voltage assignment and performance estimation are a by-product of the proposed research, the focus here is on the incorporation of such algorithms into a unified multilevel placement flow as well as on the fast solution of the individual problems themselves. Fast approximating algorithms for large-scale linear-programming (LP) and
geometric-programming (GP) formulations of sizing and voltage assignment will be investigated, and multilevel algorithms for them will be developed. Of particular interest is the ability to construct loosely-coupled convex subproblems in which the coupling itself is optimized.

Classical decomposition methods

Decomposition methods in large-scale optimization decompose a large optimization problem into smaller problems that can be solved independently, in sequence or in parallel [Bertsekas and Tsitsiklis, 1989, Lasdon, 1970]. These techniques are useful for problems that possess a structure that makes them nearly separable, and where the subproblems (at the leaf nodes) can be solved very efficiently. Primal-dual formulations lead to convex subproblems for optimizing the choice of coupled variables. These techniques will be investigated both as a means of enhancing relaxation efficiency and as a means of improving clustering.

Multilevel approach to linear and geometric programming

As an example, consider optimal gate sizing where the goal is to minimize the critical path delay by varying individual gate sizes subject to limits on sizes, power and area. This problem can be formulated as a geometric program [Boyd et al., 2005]. If the entire problem is too expensive to solve using standard geometric programming algorithms, one can attempt to first solve a simplified problem obtain by clustering nodes in the graph and treating each of these clusters of nodes as a single gate. From the solution of the coarsened problem, one obtains an approximation of the solution of the original problem. This approximation can then be refined efficiently.

A rigorous implementation will require specialized solution methods for the subproblems. For example, standard interior-point methods for linear and geometric programming are not well suited for exploiting warm starts. Cutting-plane methods [Ye, 1997] are more attractive when a series of closely related problems need to be solved. Recent techniques for solving constrained optimization problems via smooth unconstrained minimization [Nesterov, 2004, Nesterov, 2005] are also very important in this context.

Gate channel-length $L_{eff}$ control and gate oxide thickness $t_{ox}$ control are also formulated as generalized geometric programs [Boyd et al., 2005] via approximation by polynomials of standard expressions for their impact on capacitance and leakage.

4.3 Legalization

Global placement may produce results with partial module overlap. The overlap may further increase after module sizing. In addition, simultaneous $V_{dd}$, $V_{th}$, $L_{eff}$, and $t_{ox}$ assignment using geometric programming is based on the assumption that these design variables can vary continuously within their prescribed ranges, which is not realistic. Legalization includes module overlap removal and mapping continuous design variables to discrete variables. Legalization algorithms for overlap removal in mixed-size placement have been developed by the PIs [Cong et al., 2005, Cong and Xie, 2006] under current SRC Task 1091.001 and have been shown to be very effective. They will be extended to consider a power-optimization requirement (e.g. insertion of level converters and constraints due to power supply network, etc.). Legalization of $V_{dd}$, $V_{th}$, $L_{eff}$, and $t_{ox}$ assignments will leverage existing techniques for approximate integer programming by linear and geometric programming. Initially, randomized rounding [Raghavan and Tompson, 1987] and sensitivity-based methods [Nguyen et al., 2003] will be considered.

Complete legalization can be performed only at the finest level of the multilevel minimization flow. However, experience with large-scale mixed size placement strongly suggests that it is important to carry out “progressive” legalization at every level for macros with much larger sizes than the average cluster sizes in that level [Chan et al., 2005a, Chan et al., 2005b]. It is expected that similar techniques can be applied to legalization of $V_{dd}$, $V_{th}$, $L_{eff}$, and $t_{ox}$ assignments as well. For larger macros or clusters that will not be further refined for $V_{dd}$, $V_{th}$, $L_{eff}$, and $t_{ox}$ assignments, it is likely beneficial to determine their discrete legal values at intermediate levels of the multilevel optimization flow.

5 Robust Algorithm

Maximizing total slack enables aggressive power reduction by slack redistribution. But maximal total slack may be attained by allowing many paths to become near-critical. In this case, the deterministic formulation loses usefulness, and enforcing yield constraints in the statistical formulation becomes imperative.

Variations in feature dimensions and doping density cause variables $d_i$, $l_i$, and $t_i$ to be random variables. The variations are tightly correlated due to systematic error in the manufacturing process such as lithographic model angle, doping, chemical process, etc. These correlations in the random variables will be modeled explicitly and hierarchically in the multilevel flow. The precise relationship will be investigated.

In order to solve Formulation (2), its transformation to a related deterministic form is necessary. Several techniques drawn from leading paradigms for robust algorithms will be investigated. Because slight differences in formulations for optimization under uncertainty can amount to huge differences in applicability or results, some of these differences are explained here.

The general formulation of optimization under uncertainty can be written as

\[
\begin{align*}
\text{minimize} & \quad h(x) \\
\text{subject to} & \quad f_i(x, \delta_i) \leq 0, \quad (UCP)
\end{align*}
\]

where $x$ is the vector of deterministic variables, and $\delta_i$ is the vector of uncertainties for constraint $i$. Note that $h(x)$ is a deterministic objective. If each such constraint $i$ is required to hold for each admissible value of $\delta_i$, then there are infinitely many deterministic constraints for each $f_i$, one for each fixed value of each $\delta_i$. Typically, the $\delta_i$ can vary continuously.

Robust Convex Programming (RCP) is formulated the same as UCP above and is based on a direct, deterministic minimization of $h(x)$ subject to all infinitely many constraints over all possible values of $\delta_i$. In some cases, depending on the set of possible values $\delta_i$, RCP is a tractable optimization problem. For example, if the coefficients of the inequality constraints in an LP are restricted to an ellipsoid, the robust counterpart of the LP is a nonlinear convex optimization problem (a second-order cone program) [Ben-Tal and Nemirovski, 1998, Ben-Tal and Nemirovski, 2000]. Realistic ellipsoidal descriptions of the uncertainty can be derived, for example, from the error covariance and confidence ellipsoids of a least-squares estimation [Goldfarb and Iyengar, 2003a, Goldfarb and Iyengar, 2003b]. A literal interpretation of RCP
is not generally applicable to IC design, because direct enforcement of all constraints under all possible uncertainties is overly pessimistic; i.e., RCP is also not naturally suited to yield optimization. However, RCP is still a useful model (i) for understanding the often unreasonable sensitivity of a deterministic solution to constraint perturbations and (ii) as an effective heuristic technique for incorporating correlated uncertainties into a deterministic optimization model.

Probability-Constrained Programming (PCP) [Charnes and Cooper, 1959] is more directly applicable to VLSI physical synthesis:

\[
\begin{align*}
\text{minimize} & \quad h(x) \\
\text{subject to} & \quad \mathbf{P}\left(\text{violations } f_i(x, \delta) > 0 \right) < \epsilon.
\end{align*}
\]

PCP is, however, extremely difficult to solve exactly for most distributions. (An important exception is an LP with normally distributed coefficients. In this case the corresponding PCP problem is again a second-order cone program.) Relaxations form an “outer (infeasible) approximation” to the PCP problem is again a second-order cone program.)

Relaxations form an “outer (infeasible) approximation” to the solution space. Even when the original feasible (constraint) region is convex, the corresponding PCP feasible region may not be convex. PCP is the preferred yield-driven model for most IC design problems, but a unified algorithm framework for it has yet to be introduced.

Scenario-based (sampling-based) convex programming (SCP_N) [Calafiore and Campi, 2005] can be written as follows.

\[
\begin{align*}
\text{minimize} & \quad h(x) \\
\text{subject to} & \quad f_i(x, \delta_{i_k}) \leq 0 \quad k = 1, ..., N
\end{align*}
\]

Deterministic optimization is done over a random sample of the infinitely many constraints. The uncertainties are sampled at random. Equivalently, each of the deterministic constraints is replicated \(N\) times, once for each randomly sampled instance of its uncertainties \(\delta_i\). If known, the specific distributions for them can be exploited, but they are not required in an explicit parameterized form. It is sufficient to be able to generate samples from the distribution, so much more general and more realistic distributions can be handled than in PCP.

The value of this method is supported by general theoretical results on the required number of samples [Calafiore and Campi, 2005]. Compared to semi-infinite optimization, note that the required number of samples \(N\) for SCP_N is independent of the dimension of the parameter space \(\Delta\) in which \(\delta \equiv (\delta_i)\) lies. In semi-infinite optimization, the required number of discretization points grows exponentially with the dimension of the parameter space \(\Delta\).

Under both PCP and SCP_N, a well-motivated heuristic [Boyd et al., 2005, Mani et al., 2005] is used to transform the probalistic timing-yield constraint into a deterministic constraint by replacing each primitive random variable \(r\) by an estimate of the form \(\mu(s) + \kappa \sigma(s)\) based on its mean \(\mu(s)\) and standard deviation \(\sigma(s)\). This heuristic is simple and effective. The incorporation of correlations between its random parameters will be investigated.

Although sampling significantly increases the number of constraints present in the formulation, adaptive constraint generation can be used to limit computation at each iteration to those constraints closest to being violated. Recent methods suitable for constraint generation include the Analytic Centering Cutting-Plane Method (ACCPM) [Goffin and Vial, 1999, Luo and Sun, 1998, Ye, 1997], which is popular both as a general-purpose convex optimization algorithm, and, when combined with decomposition techniques, in distributed optimization. Other examples are the analytic centering techniques for recursive parameter estimation in signal processing and control [Bai et al., 1999, Bai et al., 2000]. Sequential analytic centering methods can be analyzed rigorously using the techniques developed for interior-point methods (in particular, the convergence analysis of Newton’s method for logarithmic barrier functions) [Nesterov and Nemirovsky, 1994, Ye, 1997].

Modeling the uncertainty of the parameters in an optimization model involves a trade-off between tractability and accuracy. It is critical to use realistic models (e.g., models incorporating correlations) which can be handled efficiently. For example, when estimating a covariance matrix from the sample covariances, it is important to impose structural restrictions that make the resulting robust optimization problems easier to solve. Of particular interest are low-rank structure, band- edness, sparsity and sparsity in the inverse [Dempster, 1972].

Technology Transfer

Anticipated results of the proposed research include technical reports, published papers in major EDA conferences and journals, and a software prototype of a novel physical synthesis flow for power optimization under process variation.

The PIs’ research group has a strong track record in delivering results and transferring technology to SRC member companies. Their early work on interconnect optimization (TRIO: http://cadlab.cs.ucla.edu/ trio) and estimation (IPEM: http://cadlab.cs.ucla.edu/software_release/ipem/htdocs) has been used and customized by multiple SRC companies, including Intel and IBM. The group’s PEKO placement sub-optimality benchmarks [Chang et al., 2004, Goering, 2003c] have been downloaded by over 310 different individuals at EDA companies and research universities and have been covered by multiple EE Times articles [Goering, 2003c, Goering, 2003b, Goering, 2003a]. Their SRC placement project, Task 1091.001 ending June 2006, promised a Moore’s-law generation 30% reduction of wirelength and has already exceeded that goal considerably [Chan et al., 2005c]. The resulting mPL placement package [Chan et al., 2005c] has over 250 downloads, including many from SRC member companies such as Cadence, Intel and Mentor Graphics.

References

References and PI bios are available at http://cadlab.cs.ucla.edu/cpmo/wp06refs.html.
References


