A Nonlinear Optimization Methodology for VLSI Fixed-Outline Floorplanning

Chaomin Luo* Miguel F. Anjos† Anthony Vannelli‡

March 14, 2006 (Revised March 14, 2007)

Abstract

Floorplanning is a critical step in the physical design of VLSI circuits. The floorplanning optimization problem can be formulated as a global optimization problem minimizing wire length, with the area of each rectangular module fixed while the module's height and width are allowed to vary subject to aspect ratio constraints. While classical floorplanning seeks to simultaneously minimize the wire length and the area of the floorplan without being constrained by a fixed outline for the floorplan, state-of-the-art technologies such as System-On-Chip require the solution of fixed-outline floorplanning. Fixing the outline of the floorplan makes the problem significantly more difficult. In this paper, we propose a two-stage nonlinear-optimization-based methodology specifically designed to perform fixed-outline floorplanning by minimizing wire length while simultaneously enforcing aspect ratio constraints on soft modules and handling a zero deadspace situation. In the first stage, a convex optimization globally minimizes an approximate measure of wire length. Using the solution of the first stage as a starting point, the second stage minimizes the wire length by sizing the modules subject to the prescribed aspect ratios, and ensuring no overlap. Computational results on standard benchmarks demonstrate that the model is competitive with other floorplanning approaches in the literature.

Keywords: circuit layout design, VLSI floorplanning, facility layout, combinatorial optimization, global optimization, convex programming.

*Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Research supported by a Postgraduate Award from the Natural Sciences and Engineering Research Council of Canada, and by an Ontario Graduate Scholarship. Email `c2luo@engmail.uwaterloo.ca`

†Department of Management Sciences, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Research partially supported by Discovery Grant 312125 and RTI Grant 314668 from the Natural Sciences and Engineering Research Council of Canada, and by a MITACS grant. Email `anjos@stanfordalumni.org`

‡Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Research partially supported by Discovery Grant 15296 from the Natural Sciences and Engineering Research Council of Canada. Email `vannelli@cheetah.vlsi.uwaterloo.ca`

1 Introduction

Floorplanning is a critical step in the physical design of Very Large Scale Integrated (VLSI) circuits. It has received much attention recently due to the increasingly high complexity of modern chip design. The rapid growth in the size, density, and complexity of VLSI circuits makes floorplanning a challenging task.

The floorplanning problem consists of arranging a set of rectangular modules on a rectangular chip area so that an appropriate measure of performance is optimized. The resulting layout is called a floorplan.

Floorplanning is becoming increasingly important as a tool to design flows in the hierarchical design of Application Specific Integrated Circuit (ASIC) and System-On-Chip (SoC) [1]. In terms of practical physical designs, floorplans can be classified as variable-die or fixed-die. The variable-die methodology uses 2-layer metal processes while the fixed-die methodology is a modern standard that is suitable for N-layer metal processes with 3 or more layers. Floorplanning dealing with fixed-die is called *fixed-outline* floorplanning, while *classical* floorplanning handles variable-die. Fixed-outline floorplanning works with a fixed floorplan outline, and aims to simultaneously minimize wire length and overlap, and possibly also timing (see [24]), while classical floorplanning seeks to place the modules inside a rectangular floorplan whose outline is variable so as to minimize the area of the floorplan.

Fixed-outline floorplanning plays an important role in state-of-the-art hierarchical approaches to multi-level fixed-die design of large scale ASICs and SoCs. At the top level of design, all the space resources should be occupied in the process of floorplanning. Hence, zero-deadspace is important and realistic for this type of practical floorplanning. However, addressing the requirements of fixed-outline floorplanning is significantly more difficult than for outline-free floorplanning.

Another important consideration is the possibility of floorplanning with soft modules, meaning that the area of each rectangular module is assumed to be fixed while its height and width are allowed to vary subject to given aspect ratio constraints [43, 55]. The floorplanning of soft modules is an important problem because it takes advantage of the fact that at this stage of the physical design process, the rectangular modules have not themselves been laid out in detail yet, and so the floorplanner can do a better job if it is allowed to change the dimensions of the modules in a controlled manner. (This control typically comes in the form of aspect ratio constraints.) However, almost all the literature on soft modules has not been done in the fixed-outline context. Two recent papers that do consider fixed-outline floorplanning of soft modules are [9] and [60]. The PATOMA method proposed in [9] requires the presence of deadspace (in particular since it uses a partitioning-based approach), and all the results reported there and in the companion technical report [10] have significant amounts of deadspace. The approach in [60] also requires the presence of deadspace (the authors use a minimum of 10% deadspace in all the results).

In this paper, we propose a mathematical-programming-based methodology for fixed-outline floorplanning that minimizes wire length while simultaneously enforcing aspect ratio constraints on soft modules and handling a zero deadspace situation. This methodology is particularly suitable for fixed-outline floorplanning, and can also be applied to classical floorplanning. An important feature of our methodology is that it achieves complete area

utilization using soft modules in a fixed-outline situation, something that had not been achieved in the literature to date.

The rest of this paper is organized as follows. In Section 2, we present an overview of previous work in the area. In Section 3, the proposed mathematical programming methodology is developed in some detail. Finally, in Section 4, computational results on the well-known Microelectronics Center of North Carolina (MCNC) benchmarks are reported.

2 Literature Review

The floorplanning problem has been tackled using various approaches. They can be classified into rectangular dualization, simulated annealing, force-directed methods, and mathematical programming methods. We focus on the latter two categories as they are most closely related to the first and second stages of our methodology.

Traditional graph theoretical methods and algorithms can be applied to the floorplanning problem. A graph theoretic rectangular dualization method may be used to construct rectangular floorplans [28]. The rectangular dualization problem is transformed into a matching problem on bipartite graphs. Therefore, rectangular dualization is used to floorplanning design [43].

Simulated annealing (SA) has been widely applied in VLSI layout design [21, 32, 41, 44, 47, 54, 56, 58]. The SA technique is particularly useful when the solution space of the problem is not well understood. When using SA, the solution is often restricted to be a slicing floorplan. A slicing floorplan is a floorplan that is obtained by recursively cutting a rectangle horizontally or vertically. The quality of floorplans obtained using SA is very competitive but SA is time-consuming and it takes substantial computation resources. Therefore, the applicability of SA is restricted to floorplans with a small number of modules [55, 56]. Murata et al. [32] proposed a sequence-pair-based SA model to implement floorplans by an adaptation strategy. Ranjan et al. [41] suggested a constructive technique for predicting floorplan metrics which is used for obtaining a fast and accurate SA based floorplan prediction. Tang et al. [47] proposed a model based on SA with sequence pair (SP) utilization. The SP is used for floorplanning where the fast SP evaluation can be achieved to improve the quality of floorplans. Adya and Markov [1] also used SP utilization to represent the topology of a floorplan, but proposed a moving technique based on slack computation and SA to optimize the wire length as well as aspect ratio of the soft modules. Ho et al. [21] proposed an orthogonal SA with an efficient generation mechanism to solve large floorplanning problems. Among the SA methods listed above, [1] is among the state-of-the-art approaches, and they reported small deadspace in their floorplans for the MCNC benchmarks. Therefore, we compare their reported results with our own results in Section 4.

Force-directed methods are analogous to Hooke's law in mechanics of a spring. The floorplanning problem is solved by a set of simultaneous linear equations that give the equilibrium positions of the modules. The force-directed method is used to find a timing and connectivity driven topological arrangement. Then, the topological arrangement is converted into a legal floorplan by minimizing the entire area of the floorplan. Usually, a competitive efficiency can be obtained by combining the force-directed method with other approaches. Brasen and Bushnell [4] implemented a timing driven MHERTZ floorplanner using a two-step strategy.

A sequence of gradient descent manipulations based on force-directed objective functions are adopted and timing constraints are taken into consideration in the first step. The SA optimization algorithm is then employed to minimize the total wire length and overall area of the floorplan and to remove module overlaps in the second step. Youssef et al. [59] combined a force-directed approach and a constraint graph approach.

Ho et al. [7] proposed a two-stage strategy where a force directed relaxation algorithm is used in the first stage to obtain an initial placement and a rectangular Voronoi diagram is used to spread out the overlapped modules. In the second stage, each module is reshaped or moved by the independent move of each module edge based on the attractive and repulsive force exerted on it because of the overlap and the deadspace, respectively. The first stage of this approach achieves the same objective as our first stage, although our second stage is quite different, as it uses a mathematical programming formulation to finalize the floorplan.

Floorplanning solvers based on mathematical programming approaches use different techniques such as linear programming [8, 26, 45], branch-and-bound algorithms [37, 53], convex programming [31], and nonlinear programming [40].

An algorithm of Mogaki et al. [30] uses linear programming (LP) to minimize chip width and height simultaneously within given constraints on module size, relative module position, and width of intermodule routing space. Sutanthavibul et al. [45] formulated the floorplanning problem as a mixed integer LP problem that minimizes the overall area of the floorplan. The routability is simultaneously considered in the LP model. Kim and Kim [26] combined LP and SA in the following way: the floorplan is obtained from a SP by considering an LP model or an alternative construction method. (For simplicity, they assume that the pins of all the modules are located at the centers of the modules.) Chu and Young [8] applied a Lagrangian relaxation technique to solve their proposed LP model for floorplanning problem. They improved the area usage in a chip by changing the shapes and dimensions of the soft modules to fill up the empty space. Chen and Kuh [5] minimized chip area using LP. Timing constraints are also modelled in the LP problem. The method is iterative, and proceeds by solving a sequence of LP problems until it converges to a local minimum.

Prasitjutrakul and Kubitz [39] incorporated timing and geometrical constraints into the process with the formulation of the mathematical programming problem. The floorplanning problem with path-delay constraints is modelled and mathematically formulated as a constrained nonlinear programming problem [39]. Herrigel and Fichtner [20] suggested an analytic optimization technique based on nonlinear programming to minimize wire length and chip area simultaneously for floorplanning with R or L-shaped modules. The pin positions also are optimized in their second stage. Wang et al. [50] proposed Lagrangian relaxation for soft module floorplanning based on SP algorithm. They used an average-value method to compute initial values for Lagrange multipliers so as to reduce running time. Computational effort has significantly been reduced but the total wire length is longer than other models. Ying and Wong [57] proposed a two-stage approach to minimize total wire length and area by unconstrained minimization model. Their model minimizes total wire length and area using a potential energy method in the first stage, and removes overlap in the second stage. There is unavoidably much deadspace and the computational effort is high.

There have been a number of previous approaches to floorplanning that use convex op-

timization [6, 31, 33, 42, 52]. An important property of convex optimization problems is that any local minimum is a global solution of the problem. The minimal area floorplanning problem was formulated as an optimization problem by Wimer et al. [52]. The existence and uniqueness of a floorplan is proven but the aspect ratio constraints of modules are not taken into consideration in their model. Moh et al. [31] formulated the floorplanning problem as a geometric programming problem and thus it is transformed into a convex optimization problem to exploit the advantage of the convex optimization techniques. The numbers of variables and constraints in [31] convex formulation are highly decreased by Chen and Fan [6].

Murata and Kuh [33] combined the convex optimization technique with a nonslicing floorplan representation including variable modules and preplaced modules. While their method to find a solution is very time-consuming, they do achieve very small deadspace in their floorplans for the MCNC benchmarks, and thus they are used as a second comparator for our results in Section 4.

Onodera *et al.* [37], and Wimer *et al.* [53] proposed branch-and-bound techniques for floorplanning. Their approaches explore and search the solution space and find optimal solutions by a branch-and-bound technique subject to constraints on critical nets and the shape of a chip. However, the solution obtained is only near-optimal.

Finally, the recent development of semidefinite programming (SDP) has led to its application to VLSI physical design. Vandenberghe and Boyd [48] applied SDP to VLSI transistor sizing and pattern recognition by using ellipsoids. Most recently, Takouda et al. [46] proposed a mixed integer SDP model to find global lower bounds for floorplanning problem. Their technique successfully provides global lower bounds for the smaller MCNC benchmark circuits.

3 Proposed Nonlinear Optimization Models

We begin by establishing some notation. We consider the floorplanning problem in the following form. Given:

- a set of n rectangular modules $S = \{1, 2, \ldots, n\}$;
- a partition of S into sets S_1 and S_2 representing the modules with fixed and free orientation respectively;
- an interconnection matrix $C_{n\times n} = [c_{ij}], 1 \leq i, j \leq n$, where c_{ij} captures the connectivity between modules i and j (we assume C is symmetric, i.e. $c_{ij} = c_{ji}$);
- values a_i for the area of each module i;
- bounds R_i^{low} and R_i^{up} on the aspect ratio R_i of each module i;
- bounds w_F^{low}, w_F^{up}, h_F^{low}, and h_F^{up} on the width and height respectively of the floorplan, for an instance of outline-free floorplanning; and

- values w_F and h_F for the width and height of the floorplan, for an instance of fixed-outline floorplanning;

then the floorplanning problem is to determine the location, width, and height of each module on the floorplan so that:

- there is no overlap between the modules;
- $w_i h_i = a_i$ for each module i;
- $R_i^{low} \leq \frac{h_i}{w_i} \leq R_i^{up}$ for every module i with fixed orientation ($i \in S_1$);
- $R_i^{low} \leq \frac{h_i}{w_i} \leq R_i^{up}$, or $\frac{1}{R_i^{up}} \leq \frac{h_i}{w_i} \leq \frac{1}{R_i^{low}}$ for every module i with free orientation ($i \in S_2$);

and the total wire length is minimized.

3.1 Estimation of Wire Length

Total wire length is commonly used to compare the quality of different floorplans. Wire length estimation for floorplanning and placement has been extensively studied (see e.g. [17, 18, 19, 38]). A circuit is commonly represented by a hypergraph $G_h = (V, E)$ with its vertex set $V = \{v_1, v_2, \ldots, v_m\}$ representing modules and its set of hyperedges $E = \{e_1, e_2, \ldots, e_n\}$ representing the nets connecting the modules. The first step in our approach is to construct a simple graph capturing this information. A clique model is commonly employed to form a graph from a hypergraph describing a circuit. The resulting graph is represented by a symmetric $n \times n$ adjacency matrix C, where the entry $c_{ij} \geq 0$ is an aggregate measure of the connectivity between modules i and j. A positive weight W is associated with the net to indicate the criticality of the net. In the clique model, a k-pin net is typically transformed into $k(k-1)/2$ two-pin nets with certain edge weights. If W is the weight of the k-pin net, commonly used values for the edge weights are $2W/k$ [13, 27] and $W/(k-1)$ [49]. We use the clique model for transforming hypergraphs to two-pins nets, and a value of $1/(k-1)$ for the edge weights of a net with k pins. If w_{ij} represents the weight of a two-pin net between modules i and j, their total connectivity c_{ij} is obtained by summing up the weights w_{ij} over all the cliques involving both i and j. The resulting connectivities are used in the objective functions of our mathematical programming models.

Although we minimize different estimates of the wire length in our models (see Methods A, B, and C in Section 4), we always use the half-perimeter wire length (HPWL) to measure the quality of our final floorplans. The HPWL is equal to the weighted sum of half-perimeters of the bounding boxes that encompass the modules incident on each net. This technique is extensively employed because its calculation is relatively simple and accurate. It is also increasingly popular because it is correlated with shortest paths being used to route more nets in multilayer over-the-cell routing [24]. There have been a number of studies on the minimization of the HPWL. For example, the papers [23, 51] minimize the HPWL using LP, while in [22] a max-flow computation method is used for the same purpose. Kennings and Markov [25] first proposed an analytical algorithm based on a convex approximation of the HPWL. Their proposed regularization technique is able to approximate the HPWL with

arbitrarily small relative error, and thus to minimize the HPWL using unconstrained convex optimization.

3.2 First Stage Relaxation

Let (x_i, y_i) and (x_j, y_j) denote the coordinates of the centres of modules i and j. Following [12], the minimization of the quadratic wire length with respect to a given target distance may be formulated as a quartic objective function of the form:

$$\min \sum_{i,j \in \mathrm{Net}_k} c_{ij} \left[(x_i - x_j)^2 + (y_i - y_j)^2 - d_{ij}^2 \right]^2, \tag{1}$$

where c_{ij} is the connectivity between module i and j, and d_{ij} denotes the specified target distance between modules i and j. This model is an early form of the use of a target distance. Solving the wire length minimization problem (1) results in a placement without overlap if the distances represented by the d_{ij} terms are chosen appropriately. Further extensions of the concept of target distance to placement were introduced in [2, 3, 14].

We now describe the target distance methodology employed in the first stage of our approach. Let each module i be represented by a circle of radius r_i, where r_i is proportional to $\sqrt{a_i}$, the square root of the area of module i. Following [2], we define the target distance for each pair of circles i, j as:

$$t_{ij} := \alpha (r_i + r_j)^2,$$

where $\alpha > 0$ is a parameter. To prevent circles from overlapping, the target distance is enforced via the objective function by introducing a penalty term which acts as a repeller:

$$f(\frac{D_{ij}}{t_{ij}}),$$

where $f(z) = \frac{1}{z} - 1$ for $z > 0$, and $D_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2$. The objective function is thus given by

$$\sum_{1 \le i < j \le n} c_{ij} D_{ij} + f(\frac{D_{ij}}{t_{ij}}).$$

The interpretation here is that the first term is an attractor that makes the two circles move closer together and pulls them towards a layout where $D_{ij} = 0$, while the second term is a repeller that prevents the circles from overlapping. Indeed, if $D_{ij} \ge t_{ij}$ then there is no any overlap between circles and the repeller term is zero or slightly negative, while the attractor in the objective function applies an attractive force to the two circles. On the other hand, if $D_{ij} < t_{ij}$ then the repeller term is positive, and it tends to positive infinity as D_{ij} tends to zero, preventing the circles from overlapping completely. Finally, note that there is no force between i and j exactly if $D_{ij}^2 = t_{ij}/c_{ij}$. This observation leads to the definition of the generalized target distance T_{ij} below.

In summary, the model aims to ensure that $\frac{D_{ij}}{t_{ij}} = 1$ at optimality, so choosing $\alpha < 1$ sets a target value t_{ij} that allows some overlap of the areas of the respective circles, which means that a relaxed version of the non-overlap requirement of the circles is enforced. In practice,

by properly adjusting the value of α, we achieve a reasonable separation between all pairs of circles. The complete attractor-repeller (AR) model as given in [2] is:

$$\min_{(x_i,y_j),w_F,h_F} \sum_{i,j} c_{ij}D_{ij} + f(\frac{D_{ij}}{t_{ij}})$$

s.t.

$$\begin{aligned}
&x_i + r_i \le \frac{1}{2}w_F \ \text{ and } r_i - x_i \le \frac{1}{2}w_F, \ \text{for all } i,\\
&y_i + r_i \le \frac{1}{2}h_F \ \text{ and } r_i - y_i \le \frac{1}{2}h_F, \ \text{ for all } i,\\
&w_F^{low} \le w_F \le w_F^{up},\\
&h_F^{low} \le h_F \le h_F^{up},
\end{aligned}$$

where $(x_i,\ y_i)$ are the coordinates of the centre of circle i as previously defined; w_F, h_F are the width and height of the floorplan; and w_F^{low}, w_F^{up}, h_F^{low}, and h_F^{up} are the lower and upper bounds of the width and the height of the floorplan, respectively. The first two sets of constraints require that all the circles be entirely contained within the floorplan, and the remaining two pairs of inequalities bound the width and height of the floorplan. (Note that the geometric centre of the floorplan outline is at the origin of $x-y$ plane.) In particular, for fixed-outline floorplanning, we set $w_F^{low} = w_F^{up} = \bar{w}_F$ and $h_F^{low} = h_F^{up} = \bar{h}_F$, where $\bar{w}_F$ and $\bar{h}_F$ are the fixed width and height of the floorplan.

An important drawback of the AR model is that the objective function is not convex, and hence the overall model is not convex. By modifying it so as to obtain a convex problem, we expect to obtain a relaxation that captures better global information about the problem. The effectiveness of this convex approach for the closely related facility layout problem has been documented in [3].

The analysis in [2, 3] motivates the definition of the following *generalized target distance* T_{ij}:

$$T_{ij} := \sqrt{\frac{t_{ij}}{c_{ij}+\varepsilon}},$$

where $\varepsilon > 0$ is a sufficiently small number to have $D_{ij} \approx \sqrt{\frac{t_{ij}}{c_{ij}}}$ if $D_{ij} \approx T_{ij}$. Using T_{ij}, a convexified version of the AR model may be stated as:

$$\min_{(x_i,y_j),w_F,h_F} \sum_{1\le i<j\le n} F_{ij}(x_i,x_j,y_i,y_j)$$

s.t.

$$\begin{aligned}
&x_i + r_i \le \frac{1}{2}w_F \ \text{ and } r_i - x_i \le \frac{1}{2}w_F, \ \text{for all modules } i,\\
&y_i + r_i \le \frac{1}{2}h_F \ \text{ and } r_i - y_i \le \frac{1}{2}h_F, \ \text{ for all modules } i,\\
&w_F^{low} \le w_F \le w_F^{up},\\
&h_F^{low} \le h_F \le h_F^{up},
\end{aligned}$$

where

$$F_{ij}(x_i, x_j, y_i, y_j) := \begin{cases} c_{ij}z + \frac{t_{ij}}{z} - 1, & z \geq T_{ij} \\ 2\sqrt{c_{ij}t_{ij}} - 1, & 0 \leq z < T_{ij} \end{cases}$$

with $z = (x_i - x_j)^2 + (y_i - y_j)^2$. It was shown in [2] that this problem is convex, and that by construction, F_{ij} attains its minimum value whenever the positions of circles i and j satisfy $D_{ij} \leq T_{ij}$. This includes the case where $D_{ij} = 0$, i.e. both circles completely overlap. Of course, we do not want such a placement, therefore what we seek is an arrangement of the circles where $D_{ij} \approx T_{ij}$, since for such arrangements, the minimum value of F_{ij} is still attained but the resulting overlap is minimized.

Following the approach used in [3], we use a slightly modified model whose minima satisfy $D_{ij} \approx T_{ij}$ at optimality. It can be viewed as a compromise between convexity and computational practice, in the sense that we lose the convexity of the model above, but we gain a model which can be solved efficiently and still aims to achieve the generalized target distances. The idea is to add to the objective function a term of the form $-\ln(D_{ij}/T_{ij})$ for each pair i, j of circles. (This particular choice of function is inspired by the log-barrier functions in interior-point methods for convex optimization.) Hence, the model we solve as the first stage of our methodology is:

$$\begin{aligned} \min_{(x_i,y_j),w_F,h_F} & \sum_{1\leq i<j\leq n} F_{ij}(x_i, x_j, y_i, y_j) - \beta K \ln(\frac{D_{ij}}{T_{ij}}) \\ \text{s.t.} & \\ & x_i + r_i \leq \frac{1}{2}w_F \text{ and } r_i - x_i \leq \frac{1}{2}w_F, \text{ for all modules } i, \\ & y_i + r_i \leq \frac{1}{2}h_F \text{ and } r_i - y_i \leq \frac{1}{2}h_F, \text{ for all modules } i, \\ & w_F^{low} \leq w_F \leq w_F^{up}, \\ & h_F^{low} \leq h_F \leq h_F^{up}, \end{aligned} \tag{2}$$

where β is a parameter selected empirically, and

$$K = \sum_{i<j} c_{ij}. \tag{3}$$

3.3 Second Stage Formulation

The solution of the first stage provides relative locations within the floorplan for all the modules. In the second stage, we determine the precise location and dimensions of the modules while minimizing the total wire length. At this point, classical floorplanning performs a multi-objective minimization which seeks to minimize both the wire length and the area of the floorplan. We focus on fixed-outline floorplanning, and use a mathematical program with complementarity constrains (MPCC) which only minimizes wire length, and yields deadspace-free and overlap-free floorplans. (Note that it is straightforward to also incorporate the minimization of the area of the floorplan in the objective function, but we use only the wire length since our focus is on fixed-outline floorplanning.)

The no-overlap constraints for each pair of modules can be expressed as

$$\frac{1}{2}(w_i + w_j) \leq |x_i - x_j| \quad \text{or} \quad \frac{1}{2}(h_i + h_j) \leq |y_i - y_j|.$$

However, these constraints are disjunctive, nonlinear and non-convex. They were reformulated in [3] by introducing two new variables X_{ij} and Y_{ij} and expressing them as:

$$\begin{cases} X_{ij} \geq \frac{1}{2}(w_i + w_j) - |x_i - x_j| & X_{ij} \geq 0, \\ Y_{ij} \geq \frac{1}{2}(h_i + h_j) - |y_i - y_j| & Y_{ij} \geq 0, \\ X_{ij} Y_{ij} = 0. \end{cases} \tag{4}$$

It is straightforward to check that the constraints (4) require no overlap between modules i and j.

We also require that the area requirement $w_i h_i = a_i$ be satisfied for every module i.

Incorporating all these constraints, the problem of minimizing wire length for fixed-outline floorplanning is formulated as:

$$\min_{(x_i,y_i),w_i,h_i} \quad \sum_{1 \leq i < j \leq n} c_{ij} L(x_i, x_j, y_i, y_j)$$

s.t.

$$\begin{aligned}
& x_i + \frac{1}{2} w_i \leq \frac{1}{2} \bar{w}_F \quad \forall\, i, \\
& y_i + \frac{1}{2} h_i \leq \frac{1}{2} \bar{h}_F \quad \forall\, i, \\
& \frac{1}{2} w_i - x_i \leq \frac{1}{2} \bar{w}_F \quad \forall\, i, \\
& \frac{1}{2} h_i - y_i \leq \frac{1}{2} \bar{h}_F \quad \forall\, i, \\
& w_i h_i = a_i \quad \forall\, i, \\
& w_i^{low} \leq w_i \leq w_i^{up} \quad \forall\, i, \\
& h_i^{low} \leq h_i \leq h_i^{up} \quad \forall\, i, \\
& \frac{1}{2}(w_i + w_j) - |x_i - x_j| \leq X_{ij} \quad \forall\, 1 \leq i < j \leq n, \\
& X_{ij} \geq 0 \quad \forall\, 1 \leq i < j \leq n, \\
& \frac{1}{2}(h_i + h_j) - |y_i - y_j| \leq Y_{ij} \quad \forall\, 1 \leq i < j \leq n, \\
& X_{ij} \geq 0, Y_{ij} \geq 0 \quad \forall\, 1 \leq i < j \leq n, \\
& X_{ij} Y_{ij} = 0 \quad \forall\, 1 \leq i < j \leq n,
\end{aligned} \tag{5}$$

where $\bar{w}_F$ and $\bar{h}_F$ are the fixed width and height of the floorplan, and $L(x_i, x_j, y_i, y_j)$ is the distance between modules i and j. (Different choices of L lead to different methods as discussed in Section 3.3.2 below.) The above formulation is an instance of an MPCC due to the presence of the complementarity constraints:

$$X_{ij} \geq 0, Y_{ij} \geq 0, X_{ij} Y_{ij} = 0.$$

One consequence of these constraints is that the problem lacks a strictly feasible point (because at any feasible point $X_{ij} = 0$ or $Y_{ij} = 0$ must be satisfied). To address this difficulty, we penalize the complementarity constraints $X_{ij}Y_{ij} = 0$ in the objective function and solve the resulting problem. If $X_{ij}Y_{ij} = 0$ is satisfied for all pairs of modules i and j, then the computed solution is feasible for the formulation (5).

This leads to the model used in the second stage of our methodology:

$$\begin{aligned}
\min_{(x_i,y_i),w_i,h_i} \quad & \sum_{1\le i<j\le n} c_{ij}L(x_i,x_j,y_i,y_j) + \gamma K X_{ij}Y_{ij} \\
\text{s.t.} \quad & \\
& x_i + \frac{1}{2}w_i \le \frac{1}{2}w_F \quad \forall\, i, \\
& y_i + \frac{1}{2}h_i \le \frac{1}{2}h_F \quad \forall\, i, \\
& \frac{1}{2}w_i - x_i \le \frac{1}{2}w_F \quad \forall\, i, \\
& \frac{1}{2}h_i - y_i \le \frac{1}{2}h_F \quad \forall\, i, \\
& w_i h_i = a_i \quad \forall\, i, \\
& w_i^{low} \le w_i \le w_i^{up} \quad \forall\, i, \\
& h_i^{low} \le h_i \le h_i^{up} \quad \forall\, i, \\
& \delta\left(\frac{1}{2}(w_i+w_j) - |x_i - x_j|\right) \le X_{ij} \quad \forall\, 1 \le i < j \le n, \\
& X_{ij} \ge 0 \quad \forall\, 1 \le i < j \le n, \\
& \delta\left(\frac{1}{2}(h_i+h_j) - |y_i - y_j|\right) \le Y_{ij} \quad \forall\, 1 \le i < j \le n, \\
& Y_{ij} \ge 0 \quad \forall\, 1 \le i < j \le n
\end{aligned} \tag{6}$$

where K is as in (3), and γ and δ are parameters.

3.3.1 Inclusion of the Aspect Ratio Constraints

We now show how the aspect ratio constraint for each module is easily incorporated in the formulation (6). We assume that we are given lower and upper bounds $R_i^{low} > 0$ and $R_i^{up} > 0$ on the aspect ratios R_i of module i. Aspect ratios restrict modules from becoming excessively narrow in either direction. By definition, the aspect ratio R_i for module i is

$$R_i := \max\{h_i, w_i\} / \min\{h_i, w_i\}.$$

If we set $w_i^{low} = h_i^{low} = \sqrt{a_i/R_i^{up}}$ where $a_i = w_i h_i$, then

$$w_i \ge w_i^{low} \Rightarrow w_i^2 \ge a_i/R_i^{up} \Rightarrow R_i^{up} w_i^2 \ge a_i \Rightarrow R_i^{up} \ge h_i/w_i$$

since $w_i \ge w_i^{low} > 0$. Similarly, $h_i \ge h_i^{low} > 0$ implies $R_i^{up} \ge w_i/h_i$.

By the same line of argument, setting $w_i^{up} = h_i^{up} = \sqrt{a_i R_i^{low}}$ enforces $R_i^{low} \le w_i/h_i$ and $R_i^{low} \le h_i/w_i$.

3.3.2 Minimization of Different Wire Lengths

The formulations (5) and (6) offer the ability to use different distance functions $L(x_i, x_j, y_i, y_j)$ to estimate the wire length between modules i and j. Common choices of distance function are:

- the rectilinear distance $|x_i - x_j| + |y_i - y_j|$;
- the quadratic distance $(x_i - x_j)^2 + (y_i - y_j)^2$; and
- the HPWL. This method requires additional linear constraints as well as a specific choice of L. This is described in Section 3.3.3 below.

Computational results for these three choices are reported in Section 4.

3.3.3 Minimization of Half Perimeter Wire Length

We describe here how the formulations (5) and (6) can be used to minimize the HPWL. We use a four-module net in Figure 1 to illustrate the approach.

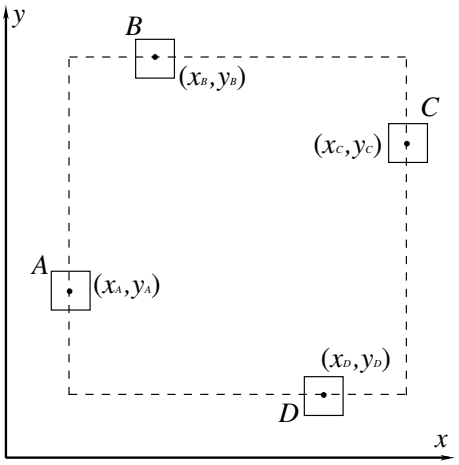


Figure 1: *Illustration of the bounding box for a four-module net*

In Figure 1, the HPWL is equal to the half-perimeter of the bounding box of the four-module net that includes the modules A, B, C, and D. We introduce two new variables, wl_x and wl_y, which will represent the components of the HPWL in the x and y directions respectively. Clearly,

$$wl_x = \max\{|x_A - x_B|, |x_A - x_C|, |x_A - x_D|, |x_B - x_C|, |x_B - x_D|, |x_C - x_D|\}$$

and

$$wl_y = \max\{|y_A - y_B|, |y_A - y_C|, |y_A - y_D|, |y_B - y_C|, |y_B - y_D|, |y_C - y_D|\}$$

Therefore, we can minimize the HPWL for this small example by solving

$$
\begin{array}{ll}
\min & c(wl_x + wl_y) \\
\text{s.t.} & \\
wl_x \geq x_A - x_B, & wl_x \geq x_B - x_A, \\
wl_x \geq x_A - x_C, & wl_x \geq x_C - x_A, \\
wl_x \geq x_A - x_D, & wl_x \geq x_D - x_A, \\
wl_x \geq x_B - x_C, & wl_x \geq x_C - x_B, \\
wl_x \geq x_B - x_D, & wl_x \geq x_D - x_B, \\
wl_x \geq x_C - x_D, & wl_x \geq x_D - x_C, \\
wl_y \geq y_A - y_B, & wl_y \geq y_B - y_A, \\
wl_y \geq y_A - y_C, & wl_y \geq y_C - y_A, \\
wl_y \geq y_A - y_D, & wl_y \geq y_D - y_A, \\
wl_y \geq y_B - y_C, & wl_y \geq y_C - y_B, \\
wl_y \geq y_B - y_D, & wl_y \geq y_D - y_B, \\
wl_y \geq y_C - y_D, & wl_y \geq y_D - y_C,
\end{array}
\tag{7}
$$

where c is the connectivity of the four-module net.

Therefore, starting with the formulations (5) and (6), adding a similar set of constraints for each net in the circuit, and setting up the objective function as the sum of the weighted half-perimeters, we can directly minimize the HPWL in the second stage of our methodology.

We note that due to the large number of linear constraints that HPWL minimization requires, it is computationally expensive to solve the placement problem when there are a huge number of modules [23]. However, the number of linear constraints is technically acceptable for floorplanning since the number of modules is relatively small.

4 Computational Results

In this section, the proposed methodology is applied to standard MCNC benchmarks to demonstrate its effectiveness and flexibility. We use the MCNC benchmark problems *apte*, *xerox*, *hp*, *ami*33 and *ami*49 [29]. Table 1 shows the characteristics of these problems. All

Circuit	# of modules	# of nets	# of I/O pads	# of pins
apte	9	97	73	287
xerox	10	203	2	698
hp	11	83	45	309
ami33	33	123	42	522
ami49	49	408	22	953

Table 1: The standard MCNC benchmark circuits

the modules are taken to be soft modules with fixed areas and variable dimensions, and (as an approximation) all the pins are assumed to be at the centre of the modules. The aspect ratio of every module is constrained to lie between 0.1 and 10, so we set $R_i^{low} = R_i^{up} = 10$

for every i. The multi-pin nets are transformed into cliques using the clique model discussed in Section 3, and taking $W = 1$ for all the nets.

Both stages of our methodology were solved by means of an AMPL model [16] and the optimization package MINOS [34, 35, 36] available via the NEOS server [11, 15].

In the second stage, we apply the proposed optimization model in three different ways:

Method A: Solve (6) with the rectilinear distance $|x_i - x_j| + |y_i - y_j|$, and calculate the HPWL of the resulting floorplan;

Method B: Solve (6) with the quadratic distance $(x_i - x_j)^2 + (y_i - y_j)^2$, and calculate the HPWL of the resulting floorplan;

Method C: Minimize the HPWL directly. This was described in Section 3.3.3.

Note that we always use the HPWL to compare the quality of the floorplans obtained.

For these instances of fixed-outline floorplanning, we respect the dimensions of the floorplans provided in the MCNC data, so that the fixed layout area is a sum of the area of all of the modules included in a circuit.

The parameters in our models took values in the following ranges:

- $\alpha \in [0.10, 2.90]$;
- $\beta = 10$;
- $\gamma \in [0.10, 2.00]$;
- $\delta \in [0.01, 5.00]$.

4.1 Detailed Computational Results for the MCNC Benchmarks

Our methods are compared with two state-of-the-art academic floorplanners [1, 33]. All reported wire lengths are measured using the HPWL, which was also used by [1] and [33]. We recall that each of methods A, B, and C actually uses a different measure of wire length as the objective function for the optimization, and once the final floorplan is obtained, the corresponding HPWL is computed for solution, and reported below.

Each of methods A, B, and C was run 20 times, and we report the best floorplan out of the 20 runs, as well as the average HPWL and runtime over the 20 runs.

4.1.1 Comparison with the Model of MK

First, we compare our approach with the model of Murata and Kuh (MK) [33]. Recall that Murata and Kuh employed a SA algorithm to improve the SP, and a convex optimization algorithm to optimize the aspect ratios of the soft modules.

Note that while in Murata and Kuh's experiment, the chip aspect ratio for every benchmark was modified to 1, thus requiring the chip to be square in shape, the dimensions of the floorplans in our experiments comply with the original ones provided in the MCNC benchmark. Like us, Murata and Kuh also assumed that every module is soft with aspect ratio being constrained to lie in the same range of 0.1 to 10.

MCNC circuit	Total area (mm^2)	Our Methodology						
		Our area (mm^2)	Runtime			HPWL		
			Method A min/avg (s)	Method B min/avg (s)	Method C min/avg (s)	Method A min/avg (mm)	Method B min/avg (mm)	Method C min/avg (mm)
apte	46.56	46.56	0.093/0.69	0.084/1.04	0.11/0.94	384.30/425.09	386.81/436.59	397.70/438.82
xerox	19.35	19.35	0.34/1.23	0.33/2.03	0.25/0.98	420.11/462.12	433.27/475.87	427.61/469.75
hp	8.30	8.30	0.37/1.17	0.21/1.65	0.42/1.72	131.83/154.84	139.80/149.64	130.50/151.28
ami33	1.16	1.16	8.11/14.16	7.41/10.03	7.53/9.51	60.36/65.31	60.25/62.37	61.40/62.83
ami49	35.4	35.4	37.91/66.09	38.78/55.53	38.90/56.46	684.62/720.65	681.72/706.06	681.70/709.46

Table 2: Experimental results with our methodology

MCNC circuit	Total area (mm^2)	MK [33]		
		Area (mm^2)	Runtime (s)	HPWL (mm)
apte	46.56	46.55	789	344.36
xerox	19.35	19.50	1198	401.25
hp	8.30	8.83	1346	118.82
ami33	1.16	1.16	75684	53.39
ami49	35.4	35.58	612103	775.10

Table 3: Results reported by MK

Table 2 gives the results obtained using our methodology, while Table 3 lists the areas, computation times, and total wire lengths reported by Murata and Kuh. The comparisons with regard to wire length and runtime are reported in Table 4. The results show that any of the methods A, B, or C requires much less CPU time than Murata and Kuh's method. Furthermore, the HPWL is competitive for the three methods, and in fact we obtain an improvement in wire length over MK for the largest benchmark problem, namely *ami*49. On average, our three methods are 8000 to 9000 times faster, and yield results that are only 5.73% to 7.75% worse than the SA results reported by Murata and Kuh.

MCNC circuit	Our Methodology vs MK					
	Speed-up			WL		
	Method A min	Method B min	Method C min	Method A min	Method B min	Method C min
apte	8483.87	9392.86	7172.73	-11.60%	-12.33%	-15.49%
xerox	3523.53	3630.30	4792.00	-4.70%	-7.98%	-6.57%
hp	3637.84	6409.52	3204.76	-10.95%	-17.66%	-9.83%
ami33	9332.18	10213.77	10051.00	-13.05%	-12.85%	-15.00%
ami49	16146.21	15783.99	15735.30	+11.67%	+12.05%	+12.05%
Average	8224.73	9086.09	8191.16	-5.73%	-7.75%	-6.97%

Table 4: Improvements in Runtime and Wire Length Compared with MK

4.1.2 Comparison with the Model of AM

We also compare our results with the results reported by Adya and Markov (AM) [1]. Recall that Adya and Markov use sequence pair utilization to represent the topology of a floorplan, together with a moving technique based on slack computation and SA.

MCNC circuit	Total area (mm^2)	AM [1]		
		Area min/avg (mm^2)	Runtime avg (s)	WL min/avg (mm)
apte	46.56	46.97/48.95	15.4	464/560
xerox	19.35	19.51/20.62	20.1	373/468
hp	8.30	8.96/9.72	15.3	177/214
ami33	1.16	1.18/1.24	31.0	62.5/75.4
ami49	35.4	36.07/37.8	31.9	673/812

Table 5: Results reported by AM

Table 5 summarizes the areas, computation times, and total wire lengths reported by Adya and Markov. The comparisons with regard to wire length and runtime are reported in Table 6. The results show that any of the methods A, B, or C requires amounts of CPU time comparable to Adya and Markov's method. Furthermore, our average HPWL for the

MCNC circuit	Our Methodology vs AM					
	Speed-up			WL		
	Method A avg	Method B avg	Method C avg	Method A min/avg	Method B min/avg	Method C min/avg
apte	22.32	14.81	16.38	+17.18/+24.09%	+16.64/+22.04%	+14.29/+21.64%
xerox	16.34	9.90	20.51	-12.63/+1.26%	-16.16/-1.68%	-14.64/-0.37%
hp	13.08	9.27	8.90	+25.52/+27.65%	+21.02/+30.07%	+26.27/+29.31%
ami33	2.19	3.09	3.26	+3.42/+12.48%	+3.60/+17.28%	+1.76/+16.67%
ami49	0.48	0.57	0.57	-1.73/+9.22%	-1.29/+13.05%	-1.29/+12.63%
Average	10.88	7.53	9.92	+6.35/+14.94%	+4.76/+16.15%	+5.28/+15.97%

Table 6: Improvements in Runtime and Wire Length Compared with AM

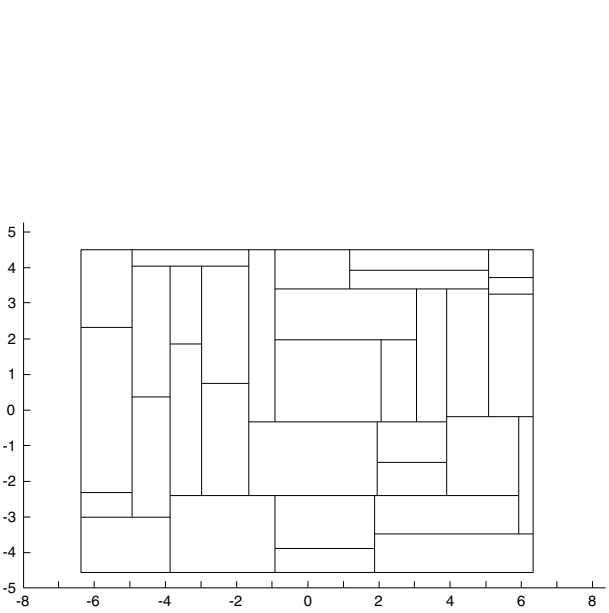
three methods is consistently better, and the best floorplan improves on their best floorplan for several benchmarks.

4.1.3 Obtaining Zero-Deadspace Floorplans

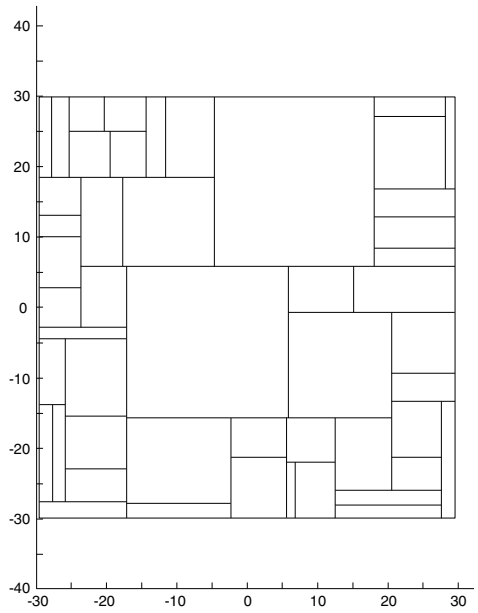
The results reported above show that our methodology is always competitive with, and frequently outperforms, the methods of Murata and Kuh, and Adya and Markov. Beyond these features, a very important feature of our methodology is that not only do the dimensions of the floorplans in our experiments comply with the original ones provided in the MCNC benchmark, but also zero-deadspace floorplans were obtained for all five problems, as shown in Table 7. In Figure 2, we depict the best zero-deadspace floorplans for the benchmarks $ami33$ and $ami49$ obtained using our methodology. In fact, the floorplans obtained using our method have little overlap, as demonstrated by the average deadspace results in Table 7. We illustrate this by presenting in Figure 3 the floorplan with the best HPWL for $ami49$; its deadspace is only 0.044% of the total area. Thus, our approach is able to guarantee complete area utilization in a fixed-outline situation, while MK and AM both have some deadspace in their floorplans.

MCNC circuit	Total area (mm^2)	MK [33]		AM [1]		Our Methodology			
		Area (mm^2)	Deadspace	Area min/avg (mm^2)	Deadspace min/avg	Area (mm^2)	Deadspace min/avg		
							Method A	Method B	Method C
apte	46.56	46.55	-0.02%	46.97/48.95	0.87%/4.88%	46.56	**0%/0%**	**0%/0%**	**0%/0%**
xerox	19.35	19.50	0.77%	19.51/20.62	0.82%/6.16%	19.35	**0%/0%**	**0%/0%**	**0%/0%**
hp	8.30	8.83	6.0%	8.96/9.72	7.40%/14.60%	8.30	**0%/0%**	**0%/0%**	**0%/0%**
ami33	1.16	1.16	0%	1.18/1.24	1.70%/6.45%	1.16	**0%/0.11%**	**0%/0.034%**	**0%/0.013%**
ami49	35.4	35.58	0.5%	36.07/37.8	1.86%/6.35%	35.4	**0%/0.11%**	**0%/0.063%**	**0%/0.094%**

Table 7: Deadspace comparisons with MK and AM



Floorplan for *ami*33 with HPWL = 62.65
($\alpha = 1.02, \beta = 10, \gamma = 1.08, \delta = 1$)



Floorplan for *ami*49 with HPWL = 716.74
($\alpha = 0.15, \beta = 10, \gamma = 1, \delta = 0.128$)

Figure 2: Best zero-deadspace floorplans for the two largest benchmarks

5 Conclusion

We proposed a two-stage nonlinear-optimization-based methodology for floorplanning that can be applied to both classical floorplanning and fixed-outline floorplanning. The first stage consists of a convex relaxation of the problem which globally minimizes an approximate measure of wire length. Using the solution of the first stage as a starting point, the second stage minimizes the wire length by sizing the modules subject to the prescribed aspect ratios, and ensuring no overlap. Computational results on standard benchmarks demonstrate that our methodology is always competitive with, and frequently outperforms, the results reported in the literature. An important feature of our methodology is that not only do the dimensions of the floorplans in our experiments comply with the original ones provided in the MCNC benchmark, but also zero-deadspace floorplans were obtained for all five problems. Thus, our approach guarantees complete area utilization in a fixed-outline situation, something that had not been achieved in the literature to date.

References

[1] S. N. Adya and I. L. Markov. Fixed-outline floorplanning: Enabling hierarchical design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 11(6):1120–1135, 2003.

[2] M. F. Anjos and A. Vannelli. An attractor-repeller approach to floorplanning. *Mathematical Methods of Operations Research*, 56:3–27, 2002.

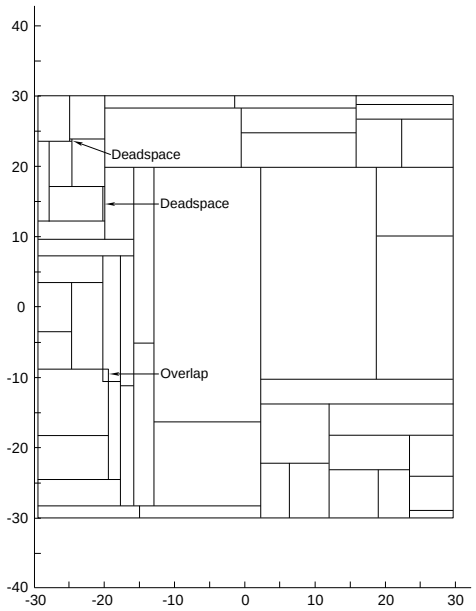


Figure 3: Floorplan for *ami*49 with best HPWL (HPWL = 681.70, deadspace = 0.044%; obtained with $\alpha = 0.15, \beta = 10, \gamma = 1, \delta = 0.11$)

[3] M. F. Anjos and A. Vannelli. A new mathematical-programming framework for facility-layout design. *INFORMS Journal on Computing*, 18(1):111–118, 2006.

[4] D. R. Brasen and M. L. Bushnell. MHERTZ: A new optimization algorithm for floorplanning and global routing. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 107–110, 1990.

[5] P. Chen and E. S. Kuh. Floorplan sizing by linear programming approximation. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 468–471, 2000.

[6] T. Chen and M. K. H. Fan. On convex formulation of the floorplan area minimization problem. In *Proc. of ACM Intl Symp. on Physical Design*, pages 124–128, 1998.

[7] S.-G. Choi and C.-M. Kyung. A floorplanning algorithm using rectangular Voronoi diagram and force-directed block shaping. In *Proc. of IEEE/ACM Intl Conf. on Computer-Aided Design*, pages 56–59, 1991.

[8] C. C. N. Chu and E. F. Y. Young. Non-rectangular shaping and sizing of soft modules for floorplan design improvement. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(1):71–79, 2004.

[9] J. Cong, M. Romesis, and J. Shinnerl. Fast floorplanning by look-ahead enabled recursive bipartioning. In *Proc. of Asia and South Pacific Design Automation Conf.*, pages 1119–1122, 2005.

[10] J. Cong, M. Romesis, and J. Shinnerl. Fast floorplanning by look-ahead enabled recursive bipartioning. Technical report, 2005.

[11] J. Czyzyk, M. Mesnier, and J. Moré. The NEOS server. *IEEE J. on Computational Science and Engineering*, 5:68–75, 1998.

[12] Y. Du and A. Vannelli. A nonlinear programming and local improvement method for standard cell placement. In *Proc. of IEEE Custom Integrated Circuit Conf.*, pages 597–600, 1998.

[13] H. Eisenmann and F. Johannes. Generic global placement and floorplanning. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 269–274, 1998.

[14] H. Etawil, S. Areibi, and A. Vannelli. Attractor-repeller approach for global placement. In *Proc. of IEEE/ACM Intl Conf. on Computer-Aided Design*, pages 20–24, 1999.

[15] M. Ferris, M. Mesnier, and J. Moré. NEOS and Condor: Solving optimization problems over the Internet. *ACM Trans. on Math. Softw.*, 26(1):1–18, 2000.

[16] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Brooks/Cole Publishing, Pacific Grove, CA, 2003.

[17] A. A. E. Gamal. Two-dimensional stochastic model for interconnections in master slice integrated circuits. *IEEE Trans. on Circuits and Systems*, 28:127–138, 1981.

[18] T. Hamada, C. K. Cheng, and P. M. Chau. A wire length estimation technique utilizing neighborhood density equations. *IEEE Trans. on Computer-Aided Design*, 15:912–922, 1996.

[19] W. Hebgen and G. Zimmermann. Hierarchical netlength estimation for timing prediction. In *Proc. of Physical Design Workshop*, pages 118–125, 1996.

[20] A. Herrigel and W. Fichtner. An analytic optimization technique for placemeent of macro-cells. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 376–381, 1989.

[21] S.-Y. Ho, S.-J. Ho, Y.-K. Lin, and W. C.-C. Chu. An orthogonal simulated annealing algorithm for large floorplanning problems. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 12(8):874–877, 2004.

[22] S. W. Hur and J. Lillis. Relaxation and clustering in a local search framework: Application to linear placement. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 360–366, 1999.

[23] M. A. B. Jackson and E. S. Kuh. Performance-driven placement of cell based IC's. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 370–375, 1989.

[24] A. B. Kahng. Classical floorplanning harmful? In *Proc. of ACM Intl Symp. on Physical Design*, pages 207–213, 2000.

[25] A. Kennings and I. Markov. Analytical minimization of half-perimeter wirelength. In *Proc. of Asia and South Pacific Design Automation Conf.*, pages 179–184, 2000.

[26] J.-G. Kim and Y.-D. Kim. A linear programming-based algorithm for floorplanning in VLSI design. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2(5):584–592, 2003.

[27] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich. Gordian: VLSI placement by quadratic programming and slicing optimization. *IEEE Trans. on Computer-Aided Design*, 10(3):356–365, 1991.

[28] Y. Lai and S. M. Leinwand. Algorithms for floorplan design via rectangular dualization. *IEEE Trans. on Computer-Aided Design*, 7(12):1278–1289, 1988.

[29] MCNC. http://www.cse.ucsc.edu/research/surf/gsrc/mcncbench.html. 2004. MCNC Floorplan Benchmark Suite, University of California, Santa Cruz.

[30] M. Mogaki, C. Miura, and H. Terai. Algorithm for block placement with size optimization technique by the linear programming approach. In *Proc. of IEEE/ACM Intl Conf. on Computer-Aided Design*, pages 80–83, 1987.

[31] T.-S. Moh, T.-S. Chang, and S. L. Hakimi. Globally optimal floorplanning for a layout problem. *IEEE Trans. on Circuits and Systems*, 43(9):713–720, 1996.

[32] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. VLSI/PCB placement with obstacles based on sequence pair. *IEEE Trans. on Computer-Aided Design*, 17(1):61–68, 1998.

[33] H. Murata and E. S. Kuh. Sequence-pair based placement method for hard/soft/pre-placed modules. In *Proc. of ACM Intl Symp. on Physical Design*, pages 167–172, 1998.

[34] B.A. Murtagh and M.A. Saunders. Large-scale linearly constrained optimization. *Math. Programming*, 14(1):41–72, 1978.

[35] B.A. Murtagh and M.A. Saunders. A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints. *Math. Programming Stud.*, (16):84–117, 1982.

[36] B.A. Murtagh and M.A. Saunders. MINOS 5.0 User's Guide. Technical Report SOL 83-20, Department of Operations Research, Stanford University, 1983. Revised as MINOS 5.1 User's Guide, Report SOL 83-20R, 1987.

[37] H. Onodera, Y. Taniguchi, and K. Tamaru. Branch-and-bound placement for building block layout. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 433–439, 1991.

[38] M. Pedram and B. Preas. Interconnection length estimation for optimized standard cell layouts. In *Proc. of IEEE/ACM Intl Conf. on Computer-Aided Design*, pages 390–393, 1989.

[39] S. Prasitjutrakul and W. J. Kubitz. Path-delay constrained floorplanning: A mathematical programming approach for initial placement. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 364–369, 1989.

[40] X. Qi, Z. Feng, and X. Yan. An algorithm of timing driven floorplanning for VLSI layout design. In *Proc. of Intl Conf. on Computer-Aided Drafting, Design and Manufacturing Technology*, pages 642–646, 1994.

[41] A. Ranjan, K. Bazargan, S. Ogrenci, and M. Sarrafzadeh. Fast floorplanning for effective prediction and construction. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 9(2):341–351, 2001.

[42] E. Rosenberg. Optimal module sizing in VLSI floorplanning by nonlinear programming. *Methods and Models of Operations Research*, 33:131–143, 1989.

[43] S. M. Sait and H. Youssef. *VLSI physical design automation : theory and practice*. IEEE Press, New York, USA, 1995.

[44] C. Sechen. *VLSI placement and global routing using simulated annealing*. Kluwer Academic Publishers, Boston, USA, 1988.

[45] S. Sutanthavibul, E. Shragowitz, and J. B. Rosen. An analytical approach to floorplan design and optimization. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 10(6):761–769, 1991.

[46] P. L. Takouda, M. F. Anjos, and A. Vannelli. Global lower bounds for the VLSI macrocell floorplanning problem using semidefinite optimization. In *Proc. of the Fifth International Workshop System-on-Chip for Real-Time Applications*, pages 275–280, 2005.

[47] X. Tang, R. Tian, and D. F. Wong. Fast evaluation of sequence pair in block placement by longest common subsequence computation. *IEEE Trans. on Computer-Aided Design*, 20(12):1406–1413, 2001.

[48] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[49] J. Vygen. Algorithms for large-scale flat placement. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 746–751, 1997.

[50] B. Wang, M. Chrzanowska-Jeske, and M. Jeske. Methods for efficient use of lagrangian relaxation for soc soft-module floorplanning. In *Proc. of IEEE Intl Conf. on Systems-on-Chip*, pages 293–294, 2003.

[51] B. X. Weis and D. A. Mlynski. A new relative placement procedure based on MSST and linear programming. In *Proc. of IEEE Intl Symp. on Circuits and Systems*, pages 564–567, 1987.

[52] S. Wimer, I. Koren, and I. Cederbaum. Floorplans, planar graphs, and layouts. In *IEEE Trans. on Circuits and Systems*, pages 267–278, 35, 3 1988.

[53] S. Wimer, I. Koren, and I. Cederbaum. Optimal aspect ratios of building blocks in VLSI. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 8(2):139–145, 1989.

[54] D. F. Wong, H. W. Leong, and C. L. Liu. *Simulated annealing for VLSI design*. Kluwer Academic Publishers, Norwell, MA, USA, 1988.

[55] D. F. Wong and C. L. Liu. A new algorithm for floorplan design. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 101–107, 1986.

[56] D. F. Wong and C. L. Liu. Floorplan design of VLSI circuits. *Algorithmica*, (4):263–291, 1989.

[57] C.-S. Ying and J. S.-L. Wong. An anslytical approach to floorplanning for hierarchical building blocks layout. *IEEE Trans. on Computer-Aided Design*, 8(4):403–412, 1989.

[58] F. Y. Young, D. F. Wong, and H. H. Yang. Slicing floorplans with range constraint. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(2):272–278, 2000.

[59] H. Youssef, S. M. Sait, and K. J. Al-Farra. Timing influenced force directed floorplanning. In *Proc. of Design Automation Conf. with EURO-VHDL*, pages 156–161, 1995.

[60] Y. Zhan, Y. Feng, and S.S. Sapatnekar. A fixed-die floorplanning algorithm using an analytical approach. In *Proc. of Asia and South Pacific Design Automation Conf.*, pages 771–776, 2006.