

Large-Scale Fixed-Outline Floorplanning Design Using Convex Optimization Techniques

Chaomin Luo

Dept. of Electrical & Computer Engineering
University of Waterloo
Waterloo, ON N2L 3G1 Canada
Email: c2luo@cheetah.vlsi.uwaterloo.ca

Miguel F. Anjos

Dept. of Management Sciences
University of Waterloo
Waterloo, ON N2L 3G1 Canada
Email: anjos@stanfordalumni.org

Anthony Vannelli

School of Engineering
University of Guelph
Guelph, ON N1G 2W1 Canada
Email: vannelli@uoguelph.ca

Abstract— A two-stage optimization methodology is proposed to solve the fixed-outline floorplanning problem that is a global optimization problem for wirelength minimization. In the first stage, an attractor-repeller convex optimization model provides the relative positions of the modules on the floorplan. The second stage places and sizes the modules using second-order cone optimization. A Voronoi diagram is employed to obtain a planar graph and thus a relative position matrix to connect the two stages. *Overlap-free* and *deadspace-free* floorplans are achieved in a fixed outline and floorplans with any specified percentage of whitespace can be produced. Experimental results on GSRC benchmarks demonstrate that we obtain significant improvements on the best results known in the literature for these benchmarks. Most importantly, our methodology provides greater improvement over other floorplanners as the number of modules increases.

I. INTRODUCTION

Floorplanning is becoming increasingly important as a tool to design flows in the hierarchical design [2]. Floorplanning dealing with fixed-die is *fixed-outline* floorplanning, while *classical* floorplanning handles variable-die. Fixed-outline floorplanning attempts to pack all the modules within a given fixed floorplan outline, and aims to simultaneously minimize wirelength and overlap, and possibly also timing ([9]). It plays an important role in state-of-the-art hierarchical methods to multi-level fixed-die design of large scale ASICs and SoCs. However, addressing the requirements of fixed-outline floorplanning is significantly more difficult than for outline-free floorplanning [2, 9].

Although there have been many studies on floorplanning, most focus on area minimization for variable-die, and very few existing floorplanners can tackle fixed-die floorplanning by minimizing the total wirelength. Parquet [2] and Capo [1] are two typical floorplanners for dealing with fixed-die constraints when minimizing total wirelength. Adya and Markov [2] suggested a moving technique based on slack computation and simulated annealing (SA) to optimize the wirelength by using sequence-pair to represent the topology of a floorplan. The fixed-outline is satisfied by using a better local search. Capo [1] developed a floorplacer that effectively combines min-cut placement with SA algorithm to form a fixed-outline floorplanner. At the procedure of partitioning and placement, fixed-outline floorplanning is completed while taking routability into account. Most recently, several fixed-outline floorplanners have been developed [4, 5]. Chen et al. [4] developed a so-called IMF multilevel floorplanner using a two-phase tech-

nique. IMF consists of a top-down partitioning phase that partitions floorplan into subregions, and a bottom-up merging phase that merges subregions. The total wirelength is optimized via min-cut partitioning in the first phase and by SA-based fixed-die floorplanning implemented in each subregion in the second phase. The floorplanner IMFAFF incorporates the accelerative fixed-outline floorplanning (AFF) into IMF, and is 11 times faster than IMF but at the expense of an increase of 9% in the wirelength. Chen and Chang [5] proposed an adaptive fast-SA scheme based on B*-tree floorplan representation. The total wirelength can be effectively minimized by dynamically varying the weights of the objective function of SA. Fixed-die floorplanning is achieved by taking outline constraints into account. Zhan et al. [15] proposed an analytical approach for fixed-outline floorplanning dealing with soft modules that minimizes total wirelength in two phases. The wirelength and area distribution density of modules are minimized in the first phase. The overlap area and wirelength are optimized in the second phase to achieve an overlap-free floorplan. There have been few studies on fixed-outline floorplanning by convex optimization. Moh et al. [11] formulated the floorplanning problem as a geometric program and thus it is transformed into a convex optimization problem to exploit the advantage of convex optimization. Floorplanning with zero-deadspace ability is accomplished by SA.

In this paper, a two-stage convex optimization method is proposed for fixed-outline floorplanning. An important property of convex optimization is that any local minimum is a global solution of the problem. In the first stage, relative positions of modules are obtained by globally minimizing an estimate of the total wirelength through an attractor-repeller (AR) convex optimization model. A relative position matrix (RPM) is then built from the solution to the first stage, and the modules are placed and sized in the second stage while minimizing the total wirelength by a second convex model. The placing and sizing stage is formulated as an optimization problem with linear and second-order cone (SOC) constraints. *Overlap-free* and *deadspace-free* floorplans are achieved in fixed-outline floorplanning. Our model also produces floorplans with any specified percentage of whitespace. To the best of our knowledge, this is the first time that a completely convex-based method is used for fixed-outline floorplanning. The proposed convex-optimization-based model is particularly suitable for fixed-outline floorplanning and can also be extended to classical floorplanning.

II. FIRST STAGE CONVEX OPTIMIZATION MODEL

In the first stage, we adopt the AR model which is based on a target distance concept [3]. 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, a_i , of module i . The target distance for each pair of circles i, j is defined as $t_{ij} := \sigma(r_i + r_j)^2$, where $\sigma > 0$ is a parameter. Furthermore, a generalized target distance is defined as $T_{ij} := \sqrt{\frac{t_{ij}}{c_{ij} + \varepsilon}}$, where $\varepsilon > 0$ is a sufficiently small number, and c_{ij} is the connectivity between modules i and j . The motivation for this definition can be found in [3]. A piecewise function F_{ij} , which is convex and continuously differentiable, is defined as

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

where $D_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2$. Let (x_i, y_i) and (x_j, y_j) denote the coordinates of the centers of modules i and j . The AR model in the first stage is:

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

where h_F, w_F are the height and width of the floorplan; $h_F^{low}, h_F^{up}, w_F^{low}$, and w_F^{up} are the lower and upper bounds of the height and width of the floorplan, respectively; and K is a parameter selected empirically.

Without the term $-K \ln(D_{ij}/T_{ij})$ in formulation (1), it was proven that this problem is convex [3]. By solving formulation (1), the solution of the first stage provides relative positions within the floorplan for all the modules represented by circles.

III. NON-OVERLAP CONSTRAINTS AND RELATIVE POSITION MATRIX

If $(x_i, y_i), (x_j, y_j), w_i, h_i, w_j$, and h_j are the coordinates, widths and heights of two rectangular modules, respectively, the overlap-free constraints for each pair of modules can be expressed as

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

These non-overlap constraints for each pair of modules can be expressed as separation in the x -direction:

$$0 \geq \frac{1}{2}(w_i + w_j) - |x_i - x_j|, \quad (3)$$

or separation in the y -direction:

$$0 \geq \frac{1}{2}(h_i + h_j) - |y_i - y_j|. \quad (4)$$

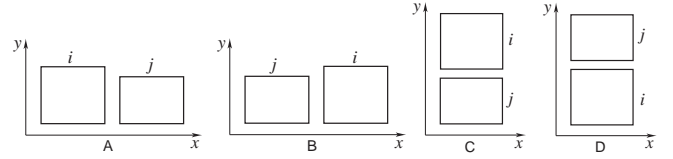


Fig. 1. Relative positions of two modules separated horizontally and vertically.

It is noted that the constraints (3) and (4) are disjunctive, nonlinear and non-convex. Once relative positions for the modules have been obtained from the first stage, we may eliminate the symbols of absolute values from inequalities (3) and (4), thus linear constraints are obtained. A relative position matrix is built with entries that represent the relative positions among modules:

If module i is horizontally separated from module j , “1” is used to represent this case as an entry in RPM, where there is only one constraint separated in the x -direction as (3). We use “11” to express that module i is on the left of module j (Fig. 1A). The following inequality is satisfied: $0 \geq \frac{1}{2}(w_i + w_j) - (x_j - x_i)$. “12” is used to denote that module i is on the right of module j (Fig. 1B). The following inequality is satisfied: $0 \geq \frac{1}{2}(w_i + w_j) - (x_i - x_j)$.

If module i is vertically separated from module j , “2” is used to represent this case, in which there is only one constraint separated in the y -direction as (4). We use “21” to express that module i is above module j (Fig. 1C). The following inequality is satisfied: $0 \geq \frac{1}{2}(h_i + h_j) - (y_i - y_j)$. “22” is used to express that module i is below module j (Fig. 1D). The following inequality is satisfied: $0 \geq \frac{1}{2}(h_i + h_j) - (y_j - y_i)$.

If we have the relative positions of two module separated diagonally (Fig. 2), a rule is defined to determine how the two modules are separated. If $\Delta y \geq \Delta x$, then these two modules should be separated in the y -direction axis. If $\Delta x > \Delta y$, then these two modules should be separated in the x -direction.

IV. RELATIVE POSITION MATRIX AND VORONOI DIAGRAM

In this section, the construction of the RPM is discussed. The nine-module *apte* circuit from the MCNC benchmark is used to show how a RPM is generated. The relative position graph of *apte* is illustrated by dashed circles obtained from the first stage as Fig. 3A. The RPM is an $m \times m$ non-negative, symmetric matrix with zeros on the principal diagonal; m is the number of modules. The information in the corresponding upper triangular matrix is sufficient. Therefore, the RPM is usually expressed as an upper triangular matrix. The representation such as “11”, “21” of each entry of the matrix has been described in the previous section.

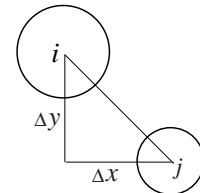


Fig. 2. Relative position of two modules separated diagonally.

$$RPM = \begin{pmatrix} 0 & 11 & 11 & 12 & 11 & 11 \\ & 0 & 12 & 12 & 12 & 12 \\ & & 0 & 12 & 12 & 11 \\ & & & 0 & 11 & 11 \\ & & & & 0 & 11 \\ & & & & & 0 \end{pmatrix} \quad (5)$$

A one-dimension six-module case is given as an example in Fig. 4 to show how a sparse relative position matrix (SRPM) can be built up. The relative position for these six modules will be arranged on one row. With the notation above, the RPM is shown in (5).

The distances between modules are obtained from the relative position of modules and included in a distance matrix (DM), which indicates the intervals among modules. The DM for the one-dimension six-module case in Fig. 4 is constructed based on the distance between modules shown in (6). A parameter δ is defined to discriminate among the distances between modules. Those modules with distance at least δ from any given module are not significant for the relative position of the module. If module 1, for instance, is the current module in Fig. 4, and $\delta = 2$, its neighboring modules within a distance of δ are modules 3, 4, and 5. Modules 2 and 6 are located farther than δ from module 1 thus the relative positions between module 1 and module 2, as well as between module 1 and module 6 are considered insignificant. The insignificant distances have been highlighted in the first row in the DM in (6).

$$DM = \begin{pmatrix} 0 & 4 & 2 & 1 & 1 & \mathbf{3} \\ & 0 & 2 & \mathbf{5} & \mathbf{3} & 1 \\ & & 0 & \mathbf{3} & 1 & 1 \\ & & & 0 & 2 & \mathbf{4} \\ & & & & 0 & 2 \\ & & & & & 0 \end{pmatrix} \quad (6)$$

By replacing those entries by zeros, a sparse DM (SDM) is formed as (7). The RPM shown in (5) becomes SRPM in (8) by substituting less significant entries of RPM by zeros. For instance, $DM(2,4) = 5$ denotes that the interval between module 2 and module 4 is 5. It is only necessary to record the relative position if two modules have distance less than 3. Therefore, the element in the SRPM corresponding to (2,4) is recorded to be zero, i.e., $SRPM(2,4) = 0$. The mechanism of construction of DM and SRPM for one dimension can easily be extended to the two-dimension case.

$$SDM = \begin{pmatrix} 0 & \mathbf{0} & 2 & 1 & 1 & \mathbf{0} \\ & 0 & 2 & \mathbf{0} & \mathbf{0} & 1 \\ & & 0 & \mathbf{0} & 1 & 1 \\ & & & 0 & 2 & \mathbf{0} \\ & & & & 0 & 2 \\ & & & & & 0 \end{pmatrix} \quad (7)$$

$$SRPM = \begin{pmatrix} 0 & \mathbf{0} & 11 & 12 & 11 & \mathbf{0} \\ & 0 & 12 & \mathbf{0} & \mathbf{0} & 12 \\ & & 0 & \mathbf{0} & 12 & 11 \\ & & & 0 & 11 & \mathbf{0} \\ & & & & 0 & 11 \\ & & & & & 0 \end{pmatrix} \quad (8)$$

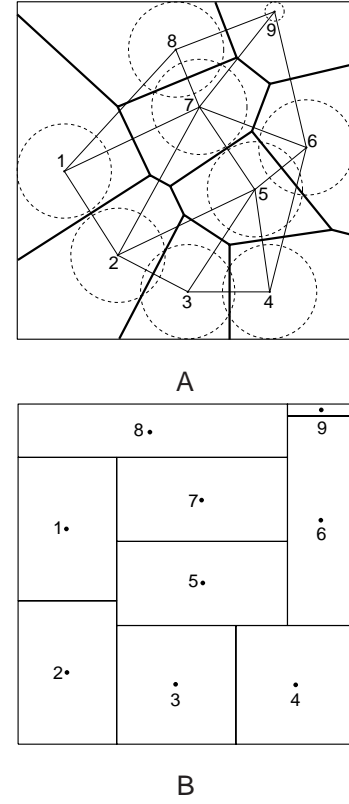


Fig. 3. Example for the apte circuit. A: The relative position graph, its Voronoi diagram and Delaunay triangulation; B: The final floorplan.

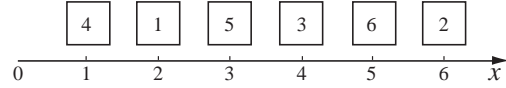


Fig. 4. An example of one-dimension (one-row) six-module case.

$$RPM = \begin{pmatrix} 0 & 21 & 11 & 11 & 11 & 11 & 11 & 22 & 11 \\ 0 & 0 & 11 & 11 & 11 & 11 & 22 & 22 & 22 \\ 0 & 0 & 0 & 11 & 22 & 22 & 22 & 22 & 22 \\ 0 & 0 & 0 & 0 & 22 & 22 & 22 & 22 & 22 \\ 0 & 0 & 0 & 0 & 0 & 11 & 22 & 22 & 22 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 12 & 22 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & 22 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (9)$$

A planar graph that can reflect the relationship of relative position of modules is needed to build up a RPM for the second stage use. An efficient solution to convert the relative position graph into a planar graph is to employ the Voronoi diagram (VD) obtained from the Delaunay triangulation (DT) [8]. Note that the DT of sites S is the geometric dual of the VD of S . A VD induces a subdivision of the total area of the layout.

The central points of the circles in the relative position graph are regarded as sites of the DT. By connecting any two central points, i.e. sites, with a line segment, a DT is formed from the relative position graph. In Fig. 3A, the relative position graph consisting of the dashed circles obtained from the first stage is converted into its corresponding DT illustrated by fine lines and corresponding VD depicted by black bold lines.

As can be seen, by using the VD (see Fig. 3A), the layout is partitioned into nine cells that represent the relationship

of relative position of nine modules. After solving the second stage optimization model described in the following section, a zero-deadspace fixed-outline floorplan can be obtained as Fig. 3B for the *apte* circuit.

V. SECOND STAGE CONVEX OPTIMIZATION MODEL

So far, a high-quality solution with regard to the relative positions of modules has been obtained from the first stage. In the second stage, we determine the precise locations and dimensions of the modules while minimizing the total wirelength. We focus on fixed-outline floorplanning, and use a Second-Order Cone Programming (SOCP) formulation, which minimizes wirelength, and yields deadspace-free and overlap-free floorplans.

SOCP is a special class of convex programming problems, in which a linear objective function is minimized subject to SOC constraints [10]. As a special case of convex optimization, SOCPs have attracted much attention recently [10]. In this section, a new model is formulated by applying SOCP techniques to incorporate area and aspect ratio constraints. SOCP problems can be efficiently solved by specialized interior-point methods. Additionally, the disjunctive constraints in (3) and (4) are replaced by linear constraints using SRPM. Using the SRPM instead of the full RPM results in fewer constraints for the second stage model, and thus in faster computation of the solution.

Also known as quadratic, ice-cream or Lorentz cone, a *standard* SOC of dimension n is expressed as [10]

$$\mathcal{C}_n = \left\{ \left[\begin{array}{c} u \\ t \end{array} \right] \mid u \in \mathbf{R}^{n-1}, t \in \mathbf{R}, \|u\| \leq t \right\}, \quad (10)$$

where $u \in \mathbf{R}^{n-1}, t \in \mathbf{R}$, and the SOC defines a convex set.

The standard form of SOCP may be expressed as follows:

$$\begin{aligned} & \min \quad g^T x \\ & \text{s.t.} \\ & \|u_i\| \leq t_i, \quad \forall i = 1, \dots, L \\ & u_i = A_i x + b_i, \quad \forall i = 1, \dots, L \\ & t_i = c_i^T x + d_i, \quad \forall i = 1, \dots, L, \end{aligned} \quad (11)$$

where the optimization variable is $x \in \mathbf{R}^n$, $g \in \mathbf{R}^n$ is the objective function, $A_i \in \mathbf{R}^{(n_i-1) \times n}$, $b_i \in \mathbf{R}^{n_i-1}$, $c_i \in \mathbf{R}^n$, $d_i \in \mathbf{R}$, $u_i \in \mathbf{R}^{n_i-1}$, $t_i \in \mathbf{R}$, and $\|u\| \leq t$ is the standard SOC constraint.

A. Area Constraints

The area constraint, $w_i h_i = a_i$, for each soft module can be relaxed as $w_i h_i \geq a_i$. For $a_i > 0$, this area constraint can be formulated as an SOC constraint:

$$\begin{aligned} & w_i h_i \geq a_i \\ \iff & h_i^2 + 2h_i w_i + w_i^2 \geq (h_i^2 - 2h_i w_i + w_i^2) + 4a_i \\ \iff & (h_i + w_i)^2 \geq (h_i - w_i)^2 + 4a_i \geq 0 \\ \iff & h_i + w_i \geq \sqrt{(h_i - w_i)^2 + (2\sqrt{a_i})^2} \\ \iff & h_i + w_i \geq \left\| \begin{array}{c} h_i - w_i \\ 2\sqrt{a_i} \end{array} \right\|_2, \quad \forall i. \end{aligned} \quad (12)$$

B. Aspect Ratio Constraints

The aspect ratio β_i for module i is defined as $\beta_i = \max\{h_i, w_i\} / \min\{h_i, w_i\}$. To avoid excessively narrow modules in either direction in the floorplan, we assume that the aspect ratio of module i must be bounded above by a given value $\beta_i^* > 0$.

If given $w_i^{low} = h_i^{low} = \sqrt{a_i / \beta_i^*}$, where $a_i = w_i h_i$, then $w_i \geq w_i^{low} > 0$, $w_i^2 \geq a_i / \beta_i^*$, $\beta_i^* w_i^2 \geq a_i$, $\beta_i^* \geq h_i / w_i$. Similarly, as $h_i \geq h_i^{low} > 0$, therefore, $\beta_i^* \geq w_i / h_i$. With inequality $\beta_i^* \geq h_i / w_i$ and $w_i h_i = a_i$, we obtain $\beta_i^* \geq h_i^2 / a_i$. Further, we obtain $a_i \beta_i^* \geq h_i^2$. Combining inequality $\beta_i^* \geq w_i / h_i$ and $w_i h_i = a_i$ yields $\beta_i^* \geq w_i^2 / a_i$. Therefore, $a_i \beta_i^* \geq w_i^2$.

The aspect ratio constraint of height of every module denoted by inequality $a_i \beta_i^* \geq h_i^2$ can be formulated by an SOC cone constraint as follows.

$$a_i \beta_i^* \geq h_i^2 \iff a_i + \beta_i^* \geq \left\| \begin{array}{c} a_i - \beta_i^* \\ 2h_i \end{array} \right\|_2, \quad \forall i$$

The aspect ratio constraint of width of every module denoted by inequality $a_i \beta_i^* \geq w_i^2$ can be, similarly, formulated by an SOC constraint as follows:

$$a_i \beta_i^* \geq w_i^2 \iff a_i + \beta_i^* \geq \left\| \begin{array}{c} a_i - \beta_i^* \\ 2w_i \end{array} \right\|_2, \quad \forall i$$

By incorporating the SOC constraints above for the area and aspect ratio of the modules, the complete problem of minimizing the total wirelength for fixed-outline floorplanning is formulated as:

$$\begin{aligned} & \min_{(x_i, y_i), w_i, h_i} \quad \sum_{1 \leq i < j \leq n} c_{ij} L(x_i, x_j, y_i, y_j) \\ & \text{s.t.} \\ & x_i + \frac{1}{2} w_i \leq \frac{1}{2} \bar{w}_F \quad \text{and} \quad y_i + \frac{1}{2} h_i \leq \frac{1}{2} \bar{h}_F \quad \forall i, \\ & \frac{1}{2} w_i - x_i \leq \frac{1}{2} \bar{w}_F \quad \text{and} \quad \frac{1}{2} h_i - y_i \leq \frac{1}{2} \bar{h}_F \quad \forall i, \\ & w_i^{low} \leq w_i \leq w_i^{up} \quad \text{and} \quad h_i^{low} \leq h_i \leq h_i^{up} \quad \forall i, \\ & \left\| \begin{array}{c} h_i - w_i \\ 2\sqrt{a_i} \end{array} \right\|_2 \leq h_i + w_i \quad \forall i, \\ & \left\| \begin{array}{c} a_i - \beta_i^* \\ 2h_i \end{array} \right\|_2 \leq a_i + \beta_i^* \quad \forall i, \\ & \left\| \begin{array}{c} a_i - \beta_i^* \\ 2w_i \end{array} \right\|_2 \leq a_i + \beta_i^* \quad \forall i, \end{aligned} \quad (13)$$

where $1 \leq i < j \leq n$, and \bar{w}_F and \bar{h}_F are the fixed width and height of the floorplan, and L is the rectilinear distance $|x_i - x_j| + |y_i - y_j|$, between modules i and j . Obviously, L is not a linear function, thus we need to linearize it. By defining $u = |x_i - x_j|$ and $v = |y_i - y_j|$ and substituting for L using u and v , the objective function of (13) becomes $\sum_{1 \leq i < j \leq n} c_{ij}(u + v)$. The following four constraints are enforced in (13): $u \geq x_i - x_j$, $u \geq x_j - x_i$, $v \geq y_i - y_j$ and $v \geq y_j - y_i$.

Now we obtain a convex SOCP model that can be efficiently solved using the commercially available conic software package MOSEK [12]. Note that the non-overlap constraints are not shown in (13). They are included in this formulation in the form of linear constraints derived from the SRPM.

TABLE I
RESULTS WITH OUR MODEL WITH I/O PADS FIXED AT THE BOUNDARY OF THE CHIPS

Circuit	Zero-whitespace		5% whitespace		10% whitespace		15% whitespace	
	CPU time(s)	HPWL	CPU time(s)	HPWL	CPU time(s)	HPWL	CPU time(s)	HPWL
n100	55.89	197490	55.32	200490	55.96	203700	55.03	207180
n200	1026.43	356520	1129.14	362070	990.79	367880	881.39	373840
n300	1654.70	477800	1669.590	485180	1428.92	492830	1461.95	500910

TABLE II
COMPARISONS FOR HPWL WIRELENGTH AMONG CAPO, PARQUET(PQ), IMF, AND IMFAFF WITH I/O PADS FIXED AT THE BOUNDARY OF THE CHIPS (10% WHITESPACE)

Circuit	Our HPWL	Parquet		Capo		IMF		IMFAFF	
		HPWL	Imprv	HPWL	Imprv	HPWL	Imprv	HPWL	Imprv
n100	203700	242050	15.84%	224390	9.22%	207852	2.00%	208772	2.43%
n200	367880	432882	15.02%	385594	4.59%	369888	0.54%	372845	1.33%
n300	492830	647452	23.88%	522968	5.76%	489868	-0.60%	494480	0.33%

VI. EXPERIMENTAL RESULTS

We use the clique model for transforming hypergraphs to two-pins nets and the half-perimeter wirelength (HPWL) to measure the quality of all our floorplans. For a set of modules with total area A and maximum whitespace fraction γ , as well as given aspect ratio α for a fixed outline, the height H_F and width W_F of the chip can be given as follows [2]:

$$H_F = \sqrt{(1 + \gamma)A\alpha} \quad W_F = \sqrt{(1 + \gamma)A/\alpha}. \quad (14)$$

The proposed methodology is applied to the standard GSRC benchmark to demonstrate its effectiveness and flexibility. In this section, our model is compared with several state-of-the-art floorplanners: Parquet [2, 14], Capo [1], IMF and IMFAFF [4].

All the modules are chosen to be soft modules with fixed areas and variable dimensions, and (as an approximation) all the pins are assumed to be at the center of the modules. The multi-pin nets are transformed into cliques using the clique model.

The first stage was solved using the optimization package MINOS [13] via the modeling language AMPL [7] accessed on the NEOS server [6]. The second stage was solved by MOSEK [12] on a Linux SUN Fire-V890 server with 16 1200-MHz processors and 32 GB RAM. We report only the CPU time for the second stage. Because solving the first stage takes significantly less time than the second stage, the CPU time for the first stage is negligible. The final total wirelength was computed by 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 GSRC benchmarks, so that the fixed layout area is a sum of the area of all of the modules included in a circuit if there are no whitespace. We chose $\alpha=1$ in (14).

Firstly, we compare our model with Parquet [2, 14], Capo [1], IMF and IMFAFF [4], where I/O pads were shifted to the boundary of the chips. In our experiment, I/O pads also were fixed on the boundary of the chips for comparison. We use the three largest circuits n100, n200, and n300 from the GSRC benchmarks. All reported wirelengths were measured using the HPWL, which was also used by Parquet, Capo, IMF

and IMFAFF. Each circuit was run once and the floorplans obtained are reported in Table I. For all the circuits, our model can achieve complete area utilization, i.e., zero-deadspace, in a fixed-outline. The results of Parquet, Capo, IMF and IMFAFF described in [4] only stated that the whitespace allowed is less than 15%. Therefore, we chose our case of 10% whitespace to compare with these floorplanners for HPWL wirelength in Table II. Our model can obtain better wirelength solutions than Parquet, Capo, IMF and IMFAFF, except IMF for n300a circuit. Our model gives an improvement of 0.33% to 23.88% in terms of the total wirelength in comparison with these floorplanners. Parquet and Capo can handle hard modules and soft modules. Our methodology is currently focused on the soft modules but in principle it is applicable to the hard module case. We are uncertain which kind of modules IMF and IMFAFF used.

Secondly, our model is compared with the Chen and Chang model (CC) [5], and Parquet [2, 14], with the I/O pads fixed at the locations originally given by the benchmarks. In fact, in this case, the experimental results of Parquet were also reported in [5] for comparison. In our experiments, the HPWL wirelength computation method and the model itself are the same as above, but I/O pads were fixed at the original locations in the benchmarks for comparisons.

Our model allows the flexibility to have any amount of whitespace. Not only can our model implement zero-deadspace floorplanning, but also it implements any percentage of whitespace, by setting the whitespace fraction γ to the desired value. Our test results for zero-deadspace and 5% whitespace are reported in Table III, and the comparisons with CC model and Parquet for 10% and 15% whitespace, are listed in Table IV and Table V, respectively. The data of zero-deadspace and 5% whitespace cases for CC model and Parquet are not available in their papers. The results reported in their papers were obtained by defining the maximum percentage of whitespace as 10% and 15%. However, the specific percentage of whitespace they adopted was not pointed out in the papers. Therefore, we compare our experimental results under both percentages of whitespace, 10% and 15%, with theirs. Table IV shows that our model achieves improvements of 9.55%, 11.65%, and 15.79% in the total wirelength

TABLE III

RESULTS WITH OUR MODEL (I/O PADS FIXED AT THE LOCATIONS GIVEN BY THE BENCHMARK)

Circuit	Zero-whitespace	5% whitespace
n100	285070	287560
n200	506070	510700
n300	584640	591320

TABLE IV

COMPARISON WITH *Parquet4.5(SP)* AND *CC*, 10% WHITESPACE (I/O PADS FIXED AT THE LOCATIONS GIVEN BY THE BENCHMARK)

Circuit	Our HPWL	Parquet4.5		CC	
		HPWL	Imprv	HPWL	Imprv
n100	290010	335600	13.58%	320600	9.55%
n200	515320	635500	18.91%	583300	11.65%
n300	597900	760500	21.38%	710000	15.79%

compared to CC model for n100, n200, and n300, respectively, with 10% whitespace. In comparison with Parquet, our model further improves the total wirelength by 13.58%, 18.91%, and 21.38% for n100, n200, and n300, respectively, with 10% whitespace. It is a superior feature that our model can produce greater improvement in terms of the total wirelength with increasing number of modules compared with the other two floorplanners (see Table IV and Table V). Particularly compared with Parquet, the total wirelength is reduced by 13.58% to 21.38% from n100 to n300, respectively.

Our methodology is also compared with the model of Zhan et al. [15] (in short, ZFS). The floorplans with 15% whitespace obtained by Zhan et al.'s model and Parquet4 dealing with soft modules are reported in [15]. Table VI lists the comparison result when I/O pads are fixed at the original locations in the benchmarks. Table VI again shows that for soft modules, we can produce greater total wirelength improvement as the number of modules increases compared with these two floorplanners.

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 GSRC benchmark, but also zero-deadspace floorplans can be obtained. Thus, our approach is able to guarantee complete area utilization in a fixed-outline situation.

VII. CONCLUSION

We proposed a two-stage completely convex optimization methodology for floorplanning that can be applied to both classical floorplanning and fixed-outline floorplanning. Computational results on the GSRC benchmark demonstrate that our methodology clearly outperforms the results reported in the literature. Our methodology guarantees complete area utilization in a fixed-outline situation, and it also produces floorplans with any specified percentage of whitespace. Most importantly, our methodology provides greater improvement over other floorplanners as the number of modules increases.

TABLE V

COMPARISON WITH *Parquet4.5(SP)* AND *CC*, 15% WHITESPACE (I/O PADS FIXED AT THE LOCATIONS GIVEN BY THE BENCHMARK)

Circuit	Our HPWL	Parquet4.5		CC	
		HPWL	Imprv	HPWL	Imprv
n100	292430	335600	12.86%	320600	8.79%
n200	519890	635500	18.19%	583300	10.87%
n300	604420	760500	20.52%	710000	14.87%

TABLE VI

COMPARISON WITH *ZFS* AND *Parquet4* WITH 15% WHITESPACE (I/O PADS FIXED AT THE LOCATIONS GIVEN BY THE BENCHMARK)

Circuit	Our HPWL	ZFS		Parquet4	
		HPWL	Imprv	HPWL	Imprv
n100	292430	291628	-0.28%	342103	14.52%
n200	519890	572145	9.13%	630014	17.48%
n300	604420	702822	14.00%	770354	21.54%

REFERENCES

- [1] S. N. Adya, S. Chaturvedi, J. A. Roy, D. A. Papa and I. L. Markov, "Unification of partitioning, placement and floorplanning," *Proc. of IEEE/ACM Intl. Conf. on Computer-Aided Design*, pp. 550-557, 2004.
- [2] S. N. Adya and I. L. Markov, "Fixed-outline floorplanning: enabling hierarchical design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol.11, No.6, pp. 1120-1135, 2003.
- [3] M. F. Anjos and A. Vannelli, "A new mathematical-programming framework for facility-layout design," *INFORMS Journal on Computing*, Vol.18, No.1, pp. 111-118, 2006.
- [4] T.-C. Chen, Y.-W. Chang and S.-C. Lin, "IMF: interconnect-driven multilevel floorplanning for large-scale building-module designs," *Proc. of IEEE/ACM Intl. Conf. on Computer-Aided Design*, pp. 159-164, 2005.
- [5] T.-C. Chen and Y.-W. Chang, "Modern floorplanning based on B*-tree and fast simulated annealing," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.25, No.4, pp. 637-650, 2006.
- [6] M. Ferris, M. Mesnier and J. Moré, "NEOS and Condor: Solving optimization problems over the Internet," *ACM Trans. on Math. Softw.*, Vol.26, No.1, pp. 1-18, 2000.
- [7] R. Fourer, D. M. Gay and B. W. Kernighan, *AMPL: A modeling language for mathematical programming*, Pacific Grove, CA: Thomson/Brooks/Cole, 2003.
- [8] L. Jin, D. Kim, L. Mu, D.-S. Kim and S.-M. Hu, "A sweepline algorithm for Euclidean Voronoi diagram of circles," *Computer-Aided Design*, Vol.38, pp. 260-272, 2006.
- [9] A. B. Kahng, "Classical floorplanning harmful?," *Proc. of ACM Intl. Symp. on Physical Design*, pp. 207-213, 2000.
- [10] M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, "Applications of second order cone programming," *Linear Algebra and its Applications*, Vol.284, pp. 193-228, 1998.
- [11] T.-S. Moh, T.-S. Chang and S. L. Hakimi, "Globally optimal floorplanning for a layout problem," *IEEE Trans. on Circuits and Systems*, Vol.43, No.9, pp. 713-720, 1996.
- [12] MOSEK, "<http://www.mosek.com/documentation.html>"
- [13] B. A. Murtagh and M. A. Saunders, "MINOS 5.4 User's Guide," *Report SOL 83-20R*, Systems Optimization Laboratory, Stanford University, Dec 1983 (revised Feb 1995).
- [14] PARQUET, "<http://vlsicad.eecs.umich.edu/BK/parquet/>"
- [15] Y. Zhan, Y. Feng and S. S. Sapatnekar, "A fixed-die floorplanning algorithm using an analytical approach," *Proc. of Asia and South Pacific Design Automation Conf*, pp. 771-776, 2006.