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 GSR 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].

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.
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 
\[
l_i := \sigma (r_i + r_j)^2,
\]
where \( \sigma > 0 \) is a parameter. Furthermore, a generalized target distance is defined as 
\[
T_{ij} := \sqrt{\frac{1}{c_{ij}} + \varepsilon},
\]
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 
\[
F_{ij}(x_i, x_j, y_i, y_j) := \begin{cases} c_{ij}D_{ij} + \frac{t_{ij}}{T_{ij}} - 1, & D_{ij} \geq T_{ij} \\ 0 \leq D_{ij} < T_{ij} \\ \frac{1}{2\sqrt{c_{ij}T_{ij}}} - 1, & D_{ij} \leq 0 \end{cases}
\]
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:
\[
\min_{(x_i, y_i), (x_j, y_j), w_F, h_F} \sum_{1 \leq i < j \leq n} F_{ij}(x_i, x_j, y_i, y_j) - K \ln(D_{ij}/T_{ij})
\]
s.t.
\[
x_i + r_i \leq \frac{1}{2}w_F \quad \text{and} \quad r_i - x_i \leq \frac{1}{2}w_F, \quad \forall i
\]
\[
y_i + r_i \leq \frac{1}{2}h_F \quad \text{and} \quad r_i - y_i \leq \frac{1}{2}h_F, \quad \forall i
\]
\[
w_{F,\text{low}} \leq w_F \leq w_{F,\text{up}}, \quad h_{F,\text{low}} \leq h_F \leq h_{F,\text{up}},
\]
where \( h_F, w_F \) are the height and width of the floorplan; \( h_{F,\text{low}}, h_{F,\text{up}}, w_{F,\text{low}}, w_{F,\text{up}} \) 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
\[
\frac{1}{2}(w_i + w_j) \leq |x_i - x_j| \quad \text{if} \quad |y_i - y_j| \leq \frac{1}{2}(h_i + h_j)
\]
\[
\frac{1}{2}(h_i + h_j) \leq |y_i - y_j| \quad \text{if} \quad |x_i - x_j| \leq \frac{1}{2}(w_i + w_j).
\]

These non-overlap constraints for each pair of modules can be expressed as separation in the \( x \)-direction:
\[
0 \geq \frac{1}{2}(w_i + w_j) - |x_i - x_j|,
\]
or separation in the \( y \)-direction:
\[
0 \geq \frac{1}{2}(h_i + h_j) - |y_i - y_j|.
\]

**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.
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 \( \delta \) is defined to discriminate among the distances between modules. Those modules with distance at least \( \delta \) 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 \( \delta \) are modules 3, 4, and 5. Modules 2 and 6 are located farther than \( \delta \) 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).

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.

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 been 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 this section, 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]

\[
\mathcal{C}_n = \left\{ \begin{bmatrix} u \\ t \end{bmatrix} \in \mathbb{R}^{n-1}, t \in \mathbb{R} \mid \|u\| \leq t \right\}, \tag{10}
\]

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

The standard form of SOCP may be expressed as follows:

\[
\begin{align*}
\text{min} & \quad g^T x \\
\text{s.t.} & \quad \|u_i\| \leq t_i, \quad \forall i = 1, ..., L \\
& \quad u_i = A_i x + b_i, \quad \forall i = 1, ..., L \\
& \quad t_i = c_i^T x + d_i, \quad \forall i = 1, ..., L,
\end{align*}
\tag{11}
\]

where the optimization variable is $x \in \mathbb{R}^n$, $g \in \mathbb{R}^n$ is the objective function, $A_i \in \mathbb{R}^{(n-1) \times n}$, $b_i \in \mathbb{R}^{n-1}$, $c_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$, $u_i \in \mathbb{R}^{n-1}$, $t_i \in \mathbb{R}$, 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:

\[
\begin{align*}
\text{if } w_i h_i \geq a_i & \iff h_i^2 + 2hw_i + w_i^2 \geq (h_i^2 - 2hw_i + w_i^2) + 4a_i \\
& \iff (h_i + w_i)^2 \geq (h_i - w_i)^2 + 4a_i \geq 0 \\
& \iff h_i + w_i \geq \sqrt{(h_i - w_i)^2 + (2a_i)^2} \\
& \iff h_i + w_i \geq \left\| \frac{h_i - w_i}{2\sqrt{a_i}} \right\|_2, \quad \forall i.
\end{align*}
\tag{12}
\]

B. Aspect Ratio Constraints

The aspect ratio constraint $\beta_i$ 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 $w_i \geq w_i^{low} = h_i^{low} = \sqrt{a_i/\beta_i^2}$, where $a_i = w_i h_i$, then $w_i \geq w_i^{low} > 0$, $w_i^2 \geq a_i/\beta_i^2$, $a_i/\beta_i^2 \geq a_i$, $\beta_i^* \geq h_i/w_i$. 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, $a_i/\beta_i^* \geq h_i^2$.

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 constraint as follows:

\[
a_i \beta_i^* \geq h_i^2 \iff a_i + \beta_i^* \geq \left\| \frac{a_i - \beta_i^*}{2h_i} \right\|_2, \quad \forall i.
\]

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:

\[
a_i \beta_i^* \geq w_i^2 \iff a_i + \beta_i^* \geq \left\| \frac{a_i - \beta_i^*}{2w_i} \right\|_2, \quad \forall i.
\]

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:

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

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.
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 $\gamma$, as well as given aspect ratio $\alpha$ for a fixed outline, the height $H_F$ and width $W_F$ of the chip can be given as follows [2]:

$$H_F = \sqrt{(1 + \gamma)A\alpha} \quad W_F = \sqrt{(1 + \gamma)A/\alpha}.$$  \hfill (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 multipin 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 $\gamma$ 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>
<thead>
<tr>
<th>Circuit</th>
<th>Zero-whitespace</th>
<th>5% whitespace</th>
<th>10% whitespace</th>
<th>15% whitespace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU time(s)</td>
<td>HPWL</td>
<td>CPU time(s)</td>
<td>HPWL</td>
</tr>
<tr>
<td>n100</td>
<td>55.89</td>
<td>197490</td>
<td>55.32</td>
<td>200490</td>
</tr>
<tr>
<td>n200</td>
<td>1026.43</td>
<td>356520</td>
<td>1129.14</td>
<td>362070</td>
</tr>
<tr>
<td>n300</td>
<td>1654.70</td>
<td>477800</td>
<td>1669.590</td>
<td>485180</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Our HPWL</th>
<th>Parquet HPWL</th>
<th>Parquet Impv</th>
<th>Capo HPWL</th>
<th>Capo Impv</th>
<th>IMF HPWL</th>
<th>IMF Impv</th>
<th>IMFAFF HPWL</th>
<th>IMFAFF Impv</th>
</tr>
</thead>
<tbody>
<tr>
<td>n100</td>
<td>203700</td>
<td>242050</td>
<td>15.84%</td>
<td>224390</td>
<td>9.22%</td>
<td>207852</td>
<td>2.00%</td>
<td>208772</td>
<td>2.43%</td>
</tr>
<tr>
<td>n200</td>
<td>367880</td>
<td>432882</td>
<td>15.02%</td>
<td>385594</td>
<td>4.59%</td>
<td>369888</td>
<td>0.54%</td>
<td>372845</td>
<td>1.33%</td>
</tr>
<tr>
<td>n300</td>
<td>492830</td>
<td>647452</td>
<td>23.88%</td>
<td>522968</td>
<td>5.76%</td>
<td>489868</td>
<td>-0.60%</td>
<td>494480</td>
<td>0.33%</td>
</tr>
</tbody>
</table>
TABLE III
RESULTS WITH OUR MODEL (I/O PADS FIXED AT THE LOCATIONS GIVEN BY THE BENCHMARK)

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Zero-whitespace</th>
<th>5% whitespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>n100</td>
<td>290010</td>
<td>335600</td>
</tr>
<tr>
<td>n200</td>
<td>515320</td>
<td>655500</td>
</tr>
<tr>
<td>n300</td>
<td>597900</td>
<td>760500</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Our HPWL</th>
<th>Parquet4.5 HPWL</th>
<th>Imprv</th>
<th>CC HPWL</th>
<th>Imprv</th>
</tr>
</thead>
<tbody>
<tr>
<td>n100</td>
<td>290240</td>
<td>335600</td>
<td>13.58%</td>
<td>320600</td>
<td>9.55%</td>
</tr>
<tr>
<td>n200</td>
<td>519890</td>
<td>655500</td>
<td>18.91%</td>
<td>583300</td>
<td>11.65%</td>
</tr>
<tr>
<td>n300</td>
<td>604420</td>
<td>760500</td>
<td>21.38%</td>
<td>710000</td>
<td>15.79%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Our HPWL</th>
<th>Parquet4.5 HPWL</th>
<th>Imprv</th>
<th>CC HPWL</th>
<th>Imprv</th>
</tr>
</thead>
<tbody>
<tr>
<td>n100</td>
<td>292430</td>
<td>335600</td>
<td>12.86%</td>
<td>320600</td>
<td>8.79%</td>
</tr>
<tr>
<td>n200</td>
<td>519890</td>
<td>655500</td>
<td>17.48%</td>
<td>583300</td>
<td>10.87%</td>
</tr>
<tr>
<td>n300</td>
<td>604420</td>
<td>760500</td>
<td>21.38%</td>
<td>710000</td>
<td>14.87%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Our HPWL</th>
<th>ZFS HPWL</th>
<th>Imprv</th>
<th>Parquet4 HPWL</th>
<th>Imprv</th>
</tr>
</thead>
<tbody>
<tr>
<td>n100</td>
<td>292430</td>
<td>329128</td>
<td>-0.28%</td>
<td>342103</td>
<td>14.52%</td>
</tr>
<tr>
<td>n200</td>
<td>519890</td>
<td>572145</td>
<td>9.13%</td>
<td>630014</td>
<td>17.48%</td>
</tr>
<tr>
<td>n300</td>
<td>604420</td>
<td>702822</td>
<td>14.00%</td>
<td>770354</td>
<td>21.54%</td>
</tr>
</tbody>
</table>

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 GSRRC 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.

REFERENCES