UFO: Unified Convex Optimization Algorithms for Fixed-Outline Floorplanning Considering Pre-Placed Modules

Jai-Ming Lin, *Member, IEEE,* and Zhi-Xiong Hung

***Abstract*—Fixed outline floorplanning has recently attracted more attention due to its usefulness in solving real problems in industry. This paper applies two convex optimization methods, named UFO, to solve this problem, which consists of a global distribution stage followed by a local legalization phase. In the first stage, modules are transformed into circles, and a push-pull (PP) model is proposed to uniformly distribute modules over the fixed outline with consideration of their wirelength. Due to the quality of the PP model, we obtain good results after the first stage. Therefore, it is not necessary to consider wirelength in the legalization phase. In order to maintain good results of the first stage, we propose a procedure to extract the geometric relations of the modules from the results of the first stage and store it in constraint graphs. Then, the locations and shapes of the modules are determined by second-order cone programming, which penalizes overlap and obeys the boundary constraints. Finally, we extend the UFO methodology to consider pre-placed modules in a fixed outline. We have implemented two convex functions on MATLAB, and experimental results have demonstrated that UFO clearly outperforms the results reported in the literature on the GSRC and MCNC benchmarks.**

***Index Terms*—Fixed-outline, floorplanning, pre-placed modules.**

I. Introduction

As NANOMETER integrated circuit technology advances, one single chip can contain more than millions of transistors in a modern design. To cope with the increasing design complexity, intellectual property modules are widely used for large-scale designs and hierarchical design has become an essential design technique in the real flow. These make floorplanning/placement become more important than ever.

Sha and Dutton [34] first proposed an analytical approach to deal with floorplanning/placement. They used a mathematical formulation to place arbitrarily sized modules because it can consider topological and geometrical characteristics of modules simultaneously. In order to reflect the geometrical properties of each module, they used a segment and a radius to model a rectangular module. Thus, the relative positions of two modules are measured by the relative positions of segments and radiuses for two modules. Finally, they transform the placement problem into an optimization problem, which includes an objective function and a set of constraints, and solve it by an optimization procedure. Later, Onodera *et al.* [31] used a branch-and-bound technique to find an optimal placement from the whole solution space since the straightforward way to handle floorplanning/placement is to enumerate all possible combinations of all modules in a 2-D space; however, the number of combinations becomes explosively large as the number of modules grows. Besides, they also keep additional space between modules to enable routing between them. However, the authors could only use pure branch-and-bound techniques on small circuits; they resorted to partitioning techniques to solve larger problems. Recently, many methods used different representations [4], [13], [15], [21], [25], [28], [30], [35], [39] with the simulated annealing framework to handle the problem. They have proven that the simulated annealing approach can obtain a good result in the reasonable running time.

Although many studies on floorplanning have been proposed, there exists a gap between the real problem concerned in the industry and the research aspect. Traditional researches focus on allocating modules to optimize its area and timing without restricting the resulting shape of a floorplan. However, floorplanning is usually carried out after the die size and the package have been determined, which makes it quite impractical without considering the die shape. Therefore, Kahng [18] pointed out the insufficiency of the classical floorplanning problem and proposed a formulation that is more consistent with the real requirements. In his formulation, the fixed bounding rectangle of a floorplanning/placement region is considered.

In addition to the fixed-outline constraints, the placement region often contains some pre-modules such as analog or radio frequency. These full custom modules may contain pads inside the modules. Only when the locations of all pads in a chip are assigned, the bonding rules can be checked. Thus, we should place such kind of modules earlier than others. The other reason is that designers hope to place some modules closed to each other for timing or power consideration. Thus, the ability to handle pre-placed modules (PPM) is very important to a floorplanning algorithm. What makes the problem more complex is that the shapes of PPM are usually not rectangles. Therefore, if the shapes of new placed modules are

Manuscript received February 23, 2010; revised June 25, 2010 and October 12, 2010; accepted January 7, 2011. Date of current version June 17, 2011. This work was supported by the National Science Council of Taiwan, under Grant NSC-99-2220-E-006-019. This paper was recommended by Associate Editor Evangeline Young.

The authors are with the Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: jmlin@ee.ncku.edu.tw; yanzi750618@hotmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2011.2114531

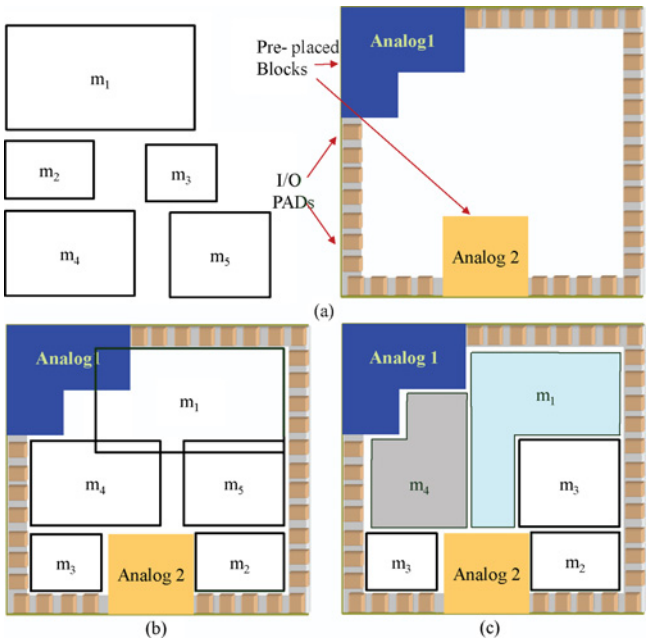


Fig. 1. Fixed-outline floorplanning with PPM. (a) Five modules to be placed into the specified region with two placed modules. (b) Illegal floorplan if only rectangular modules are allowed. (c) Feasible floorplan if arbitrary shaped modules are allowed.

restricted to the rectangles, it is inevitable to waste area. Thus, to construct an arbitrary shape modules to fit the remaining region is critical to such problem.

In the following example, we will illustrate the phenomena; Fig. 1(a) shows five modules that should be placed into a specified region in its right side. The rectangles around the boundary are I/O pads. Two full-custom modules analog1 and analog2 have been placed first since they contain their own pads (note that analog1 is L-shape). Thus, we can only place other modules in the remaining region. Fig. 1(b) shows that, even though the area of the remaining region is big enough, it is impossible to place these five modules into the region if their shapes are rectangular. Therefore, while a fixed-outlined region contains PPM, it is necessary to form non-rectangular shaped modules to fully use the space. Fig. 1(c) shows a feasible floorplan for all modules.

A. Previous Work

During past few years, many studies focusing on solving fixed-die floorplanning have been proposed. Most works handled this problem using the simulated annealing algorithm with various representations [1], [5]–[7]. Adya and Markov [1] used sequence pair to represent the topology of a floorplan. In order to speed up the convergence, they moved modules based on slack computation. The fixed-outline constraint can be satisfied more easily by using a better local search method. Chen *et al.* [5] developed an interconnect-driven multilevel floorplanning, called IMF, for the problem. IMF adopted a two-stage approach, which is top-down partitioning followed by bottom-up merging. In the first phase, a netlist and a placement region are recursively partitioned by using the min-cut algorithm under wirelength consideration. In the second stage, a B*-tree-based floorplanner is applied to respective fixed-outlined region. Then, the results of two neighboring regions are merged to generate a new result for the next iteration. Based on B*-tree representation, Chen and Chang [6] also proposed an adaptive fast simulated annealing scheme for fixed-die floorplanning, and it can be achieved by taking the outline constraint into its objective function. Recently, Yan and Chu [36] presented a fixed-outline floorplanning algorithm, named DeFer, based on a slicing tree for a non-slicing floorplan. In their method, one slicing tree actually corresponds to a huge number of slicing floorplan solutions which are recorded in one single shape curve. Based on the principle of deferred decision making, it can choose a good floorplan which is more easy to fit into the fixed-outline according to the final shape curve.

Some papers used mathematical approach for the problem. The force-directed algorithm [32], [33], [38] is one kind of mathematical method used to deal with floorplanning/placement problem. Its idea is to make connected modules exert attractive forces on one another, which can be calculated by Hook's law. Thus, a set of equations is used to reflect the interconnection relation between various modules, and the objective is to place all modules such that the total sum of the forces can be minimized. Since they can consider placement in a global view, good relative positions between modules can be obtained. However, the formulation cannot reflect the sizes or shapes of modules. Therefore, it may lead significant amount of overlaps, and thus a post-process is required to remove the overlap. Similar to the force-directed method, our pull-push (PP) model has a good property in the global view. But there exists a significant difference between the two approaches. In the force-direct method, the magnitudes of repulsive forces are not based on geometrical parameters of components. However, by incorporating the repulsive force directly into our model, our model is more realistic than force-directed method and can obtain better results.

Moh *et al.* [24] formulated the floorplanning problem as a geometric program and then transformed it into a convex optimization problem. Then, Zhan *et al.* [40] proposed an analytical approach for fixed-outline floorplanning with soft modules. A two-stage approach is used to deal with the problem. In the first phase, they minimize the wirelength as well as distribution density of modules in each subregion. Then, the overlap problem is handled in the second phase to obtain an overlap-free floorplan. Recently, Luo *et al.* [23] also proposed a two-stage convex optimization method for the problem. In their method, relative positions of modules are first determined using a convex optimization model, named *attractor-repeller* (AR), which can minimize the total wirelength. In the second stage, the actual locations and shapes of modules are determined using second-order cone programming (SOCP) formulation, which minimizes wirelength and facilitates deadspace-free and overlap-free floorplans. However, because AR model cannot distribute modules to the placement region uniformly, the relative positions after SOCP may be completely different from that after the first stage. Thus, they needed to consider wirelength again in the second stage. Very recently, He *et al.* [14] proposed a novel fixed-outline floorplanner, named SAFFOA. They built and solved

TABLE I
COMPARISONS BETWEEN OUR METHOD AND LUO *et al.* [23]

	UFO		Luo *et al.* [23]
In the first stage	Model	PP	AR
	Wirelength	Actual length (d_{ij})	Quadratic length (D_{ij})
	Curve (see Fig. 2)	Monotonic decreasing function + monotonic increasing function	Constant value + monotonic increasing function
In the second stage	Objective function	Fixed-outline constraint	Wirelength
	Topology	Constraint graphs	Relative position matrix

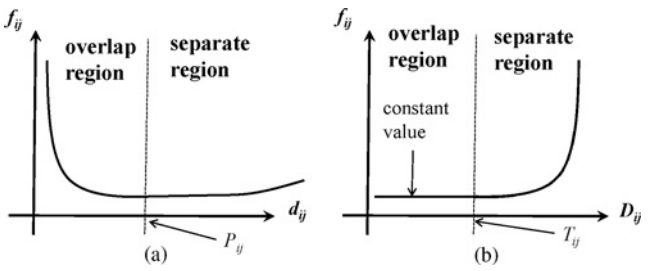


Fig. 2. (a) Curve of PP model. (b) Curve of AR model.

a group of four quadratic equations in four variables iteratively, and facilitated its integration with simulated annealing by using a topological representation, called ordered Quadtree.

Although our method looks similar to the method proposed by Luo *et al.* [23], they are different in many ways. Table I compares UFO and theirs [23]. The difference between our method and [23] in the first stage is that PP model uses actual length to measure two-pin connection while AR model uses quadratic length. Since wirelength is computed by half-perimeter wirelength (HPWL), we can get more accurate length than Luo *et al.* [23]. Besides, the two models have completely different shapes of curves. In the second stage, because we can obtain good wirelength in the first stage, we do not need to consider wirelength; thus, our objective only intends to meet the outline constraint while Luo *et al.* still needs to minimize wirelength. Besides, we uses constraint graphs to record necessary geometric relations of modules unlike Luo *et al.* uses relative position matrix which records all geometric relations between each pair of modules.

In addition, there exist a lot of papers that consider the floorplanning/placement problem in the presence of PPM, which means that some of modules (called PPM) have been placed at pre-specified locations with fixed shapes and orientations; otherwise, they are called free modules and can be placed anywhere in a chip as long as they do not overlap with the PPM. Several works have handled this problem by using different representations such as normalized polish expression [37], sequence-pair [26], [27], bounded-sliceline grid [29], B*-tree representation [16], and corner block list [8]. However, all the representation-based approach cannot place PPM at specified locations while perturbing the representation. They usually need to first get actual locations of modules by transforming a representation to a packing, and then move the illegal modules to the specified locations by the swap operation. This makes them difficult to deal with the problem.

B. Our Contribution

We propose in this paper unified convex algorithms for fixed-outline floorplanning, called UFO. The two algorithms are separately used in a global distribution stage and in a local legalization stage. In the first stage, we first transform modules into circles and then distribute circles into a specified region using a convex optimization method, called a PP model. Since distributing modules among a placement region and minimizing total wirelength are two conflicting objectives, it is very difficult to balance. PP model uses two forces, which are pull force and push force, to compromise these two factors. If two modules have larger number of interconnections, it would pull two modules to closer locations to minimize wirelength. On the contrary, when there exists a serious overlap of any two modules, it would add an additional penalty such that two modules are push away from each other. By properly formulating the two forces, the minimal value can happen when the two circles touch at one point. That is why we can uniformly distribute modules among the placement region and simultaneously minimize wirelength.

After the first stage, there still exist some overlaps between modules. Thus, we have to legalize modules in the second stage, which is considered as legalization stage. Since modules have been uniformly distributed among the region and small wirelength has been obtained, we only need to legalize modules without considering wirelength further more. In order to maintain the good result of the first stage, we propose a methodology to extract geometric relations of modules from their distribution and record them by a horizontal and a vertical constraint graphs. To maintain the necessary relations, the constraint graphs are built according to the triangles obtained by applying the Delaunay triangulation (DT) method [17] to the distribution. Then, the other convex optimization function as well as a lot of constraints transformed from the graphs are used to meet the desired objective.

Finally, we extend our method to consider PPM. Due to the mathematical formulation, it is quite easy to fix PPM in the specified locations, which makes UFO a suitable method to deal with this problem. We have implemented the two convex functions on MATLAB, and experimental results have demonstrated that UFO clearly outperforms the results reported in the literature on the GSRC benchmark.

The remainder of this paper is organized as follows. Section II formulates the fixed-outline floorplanning/placement problem. Section III presents the first convex optimization algorithm to distribute modules over a fixed-outline. Section IV introduces the other convex optimization algorithm to determine shapes and locations of modules such that no module overlaps and the fixed-outline constraint can be satisfied. Section V extends UFO to consider PPM in a fixed outline. Experimental results are reported in Section VI. Finally, we conclude our paper in Section VII.

II. PROBLEM FORMULATION

Given a die, its dimension is fixed and the width and height are denoted by W_f and H_f, respectively. Let $M = \{m_1, m_2, \ldots, m_n\}$ be a set of n modules. For each module

m_i, $1 \le i \le n$, its width, height, and area are denoted by w_i, h_i, and A_i, respectively. m_i has variable width and height and its aspect ratio is defined by w_i/h_i. The acceptable aspect ratios are in the range $[L_i, U_i]$. The goal of *fixed-outline floorplanning* is to place these modules in the given die area and determine their shapes such that no two modules overlap and the total wirelength is minimized.

III. Global Distribution Stage

In the global distribution stage, we would like to uniformly spread modules among a specified region and simultaneously minimize wirelength. We propose a convex optimization model for this purpose. Since our model is modified from the AR model by Luo *et al.* [23], we first review the AR model in the first section. Due to the deficiencies in the AR model, we propose our PP model to obtain a better distribution of modules in the second section. Finally, we would prove the convex optimization property of PP model in the last section.

A. Review of AR Model

In the AR model, each module m_i is transformed into a circle C_i. Its coordinate is denoted by (X_i, Y_i), and radius is denoted by r_i, which is proportional to $\sqrt{A_i}$. Let c_{ij} denote the connectivity between modules m_i and m_j and D_{ij} denote the square distance between the centers of the corresponding circles (i.e., $D_{ij} = (X_i - X_j)^2 + (Y_i - Y_j)^2$). Function (1) shows the AR model for two circles C_i and C_j, which consists of two equations. The first equation is composed of two terms, which, respectively, represent the attractor force and the repeller force. Two circles with a large number of connections should be placed closer such that the related wirelength can be minimized. Thus, the first term $c_{ij}D_{ij}$ can be considered as the attractor force. However, it is inevitable to cause a large overlap if there exists no repeller force. Therefore, the other term $t_{ij}/D_{ij} - 1$ is included for this purpose, where $t_{ij} = \sigma \times (r_i + r_j)^2$ and $\sigma > 0$ is a user-specified parameter, which represents the degree that two circles can overlap. In order to push two circles away when they overlap, the sign of the term would become plus when $D_{ij} < t_{ij}$. The minimum value of (1) happens at the point T_{ij}, where $T_{ij} = \sqrt{\frac{t_{ij}}{c_{ij}+\varepsilon}}$ and $\varepsilon > 0$ is a sufficiently small number, and the value is $2\sqrt{c_{ij}t_{ij}} - 1$. While $D_{ij} < T_{ij}$, they use the value as the cost in the other equation. The complete piecewise function f_{ij} is formulated as follows:

$$f_{ij}(X_i, X_j, Y_i, Y_j) = \begin{cases} c_{ij}D_{ij} + \frac{t_{ij}}{D_{ij}} - 1, & D_{ij} \ge T_{ij} \\ 2\sqrt{c_{ij}t_{ij}} - 1, & 0 \le D_{ij} < T_{ij}. \end{cases} \quad (1)$$

The second equation means that the cost becomes a constant value once two circles overlap in some degree. It may lead a large overlap of circles. Besides, the minimum value may occur when the two circles are placed at very close locations. To illustrate the phenomena, we simplify the first function of AR model as $f(z) = cz + \frac{(t-z)}{z}$, where z denotes D_{ij}, c denotes c_{ij}, and t denotes $(r_i + r_j)^2$, that is

$$f(z) = cz + \frac{(t-z)}{z} = cz + tz^{-1} - 1. \quad (2)$$

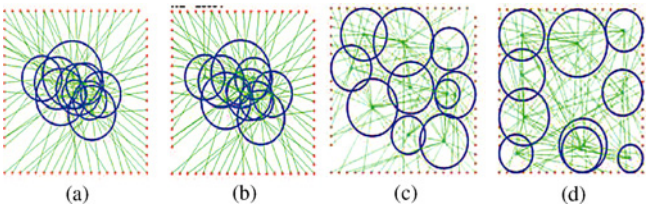


Fig. 3. Distributions of circles in AR model by multiplying different values to its second term in its first equation. The multiplied values are (a) 10^0, (b) 10^3, (c) 10^5, and (d) 10^7.

To compute the minimum value of the function, we first derive the first derivative of $f(z)$ as follows:

$$f'(z) = c - tz^{-2} = 0 \Rightarrow z = \pm\sqrt{\frac{t}{c}} \quad (3)$$

where z denotes the minimum value of the function.

For example, suppose $r_i = 30$, $r_j = 20$, and $c_{ij} = 3$. According to the above equation, the minimum value of z is 28.87, which means that the actual distance between two circle is 5.37. In such condition, there exist a serious overlap between two circles. To demonstrate the phenomena, we also show the resulting distribution of circles while AR model is applied as shown in Fig. 3. We performed the experiments on a circuit n10 from the GSRC benchmark. Fig. 3(a) shows the distribution of circles by using the first equation in AR model (*note that we do not use the second equation of AR model to perform the experiment because it is a constant value*). All circles overlap completely in this figure. By multiplying different values 10^3, 10^5, and 10^7 to the second term in the first equation of AR model, the distribution of circles are shown in Fig. 3(b)–(d), respectively. Fig. 3(b) shows that there still exists a serious overlap when the multiplied value is small. On the contrary, the circles are pushed to the boundary when the value is too large as shown in Fig. 3(d). Although the distribution of modules in Fig. 3(c) is uniform, there exists a complete overlap of two circles in the graph.

B. PP Model

We have pointed out the drawback of the AR model in the previous section. Now, we would like to introduce our PP model.

Before introducing our model, we first give the necessary notations as follows:

1) $s_{ij} = (r_i \times r_j)^2$;
2) $d_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$;
3) $p_{ij} = (r_i + r_j) - d_{ij}$.

Since we use HPWL to measure the wirelength in the floorplanning stage, our model uses d_{ij} to measure the actual distance of two-pin interconnect instead of using the quadratic length D_{ij} in AR model, which makes us get more accurate wirelength. Different from AR model, PP model wants to give a larger force to push circles away if they overlap. However, once they are separate, the force should be reduced gradually. And the minimum value happens when two circles just touch at one point, which is $d_{ij} = r_i + r_j$ (i.e., $p_{ij} = 0$).

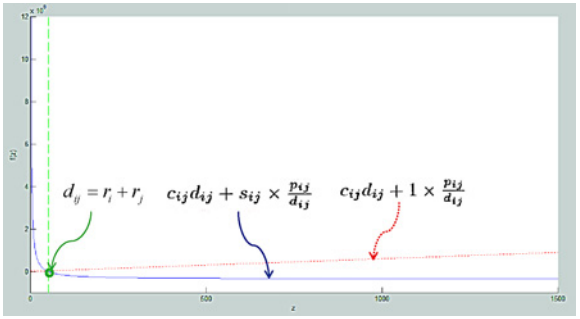


Fig. 4. Curve of the PP model (the curve of the first and second equation are denoted by a solid line and a dashed line). The minimum value happens at the point d_{ij}.

Function (4) shows our PP model for two circles C_i and C_j, which also contains two equations as follows:

$$f_{ij}(X_i,\ X_j,\ Y_i,\ Y_j) = \begin{cases} c_{ij}d_{ij} + s_{ij} \times \frac{p_{ij}}{d_{ij}}, & p_{ij} \geq 0 \\ c_{ij}d_{ij} + 1 \times \frac{p_{ij}}{d_{ij}}, & p_{ij} < 0. \end{cases} \quad (4)$$

The two equations, respectively, represent the conditions that two circles overlap ($p_{ij} \geq 0$) and do not overlap ($p_{ij} < 0$), which are both convex functions. To show its property, we use general function to represent each of them as follows:

$$f(z) = cz + s\frac{(b-z)}{z} = cz + sbz^{-1} - s \quad (5)$$

where z denotes d_{ij}, c denotes c_{ij}, and b denotes $r_i + r_j$. s may denote s_{ij} and 1. Then, the first derivative of the function is shown as follows:

$$f'(z) = c - sbz^{-2} = 0 \Rightarrow z = \pm\sqrt{\frac{sb}{c}}. \quad (6)$$

Using the same condition in the above example ($r_i = 30$, $r_j = 20$, and $c_{ij} = 3$), the minimum values happen when the distances of two circles are respective 2449.5 and 4.08 for the first equation and the second equation, which are very long and short distances. Therefore, the first equation is a monotonic decreasing when $p_{ij} \geq 0$. However, the second equation is monotonic increasing when $p_{ij} < 0$. Fig. 4 shows the curves of two equations. In the figure, the x-axis denotes the distance of the two circles and the y-axis denotes the value of the function. The dotted line denotes the position when two circles connect at one point ($p_{ij} = 0$). The resulting curve of our PP model can be obtained by combining the curve of (1) and the curve of (2). It is also a convex function and the minimum value happens when $r_i + r_j = 50$.

Our mathematical formulation for the global distribution stage is shown Fig. 5. The objective function F is to minimize the summation of PP model f_{ij} for each pair of circles C_i and C_j [see function (7)]. The four inequations are used to guarantee that the fixed-outline constraint can be satisfied for all circles.

C. Convex Property of PP Model

In this section, we will show that the objective function as stated in (7) is a convex function. Before that, we first show that each equation in PP model is a convex function as follows:

$$F = \min_{X_i, Y_i, X_j, Y_j} \sum_{1 \leq i < j \leq n} f_{ij}(X_i, Y_i, X_j, Y_j). \quad (7)$$

s.t.

$$X_i + r_i \leq W_f, \quad X_i - r_i \geq 0, \quad Y_i + r_i \leq H_f, \quad Y_i - r_i \geq 0 \quad (8)$$

Fig. 5. PP model.

Lemma 1: Let $f_{ij} : \Re^4 \mapsto \Re$ be given by

$$f_{ij}(X_i,\ X_j,\ Y_i,\ Y_j) = c_{ij}d_{ij} + s_{ij}(b/d_{ij} - 1) \quad (7)$$

where $c_{ij} \geq 0$, $d_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$, and $b = r_i + r_j$. The Hessian of f_{ij} is positive semidefinite.

Proof: The four-component column vector of the first derivative of f_{ij} is as follows:

$$\nabla f_{ij}(X_i,\ X_j,\ Y_i,\ Y_j) = \begin{pmatrix} \frac{\partial f_{ij}}{\partial X_i} \\ \frac{\partial f_{ij}}{\partial X_j} \\ \frac{\partial f_{ij}}{\partial Y_i} \\ \frac{\partial f_{ij}}{\partial Y_j} \end{pmatrix} = \begin{pmatrix} c_{ij}(X_i - X_j)d_{ij}^{-1} - s_{ij}b(X_i - X_j)d_{ij}^{-3} \\ -c_{ij}(X_i - X_j)d_{ij}^{-1} + s_{ij}b(X_i - X_j)d_{ij}^{-3} \\ c_{ij}(Y_i - Y_j)d_{ij}^{-1} - s_{ij}b(Y_i - Y_j)d_{ij}^{-3} \\ -c_{ij}(Y_i - Y_j)d_{ij}^{-1} + s_{ij}b(Y_i - Y_j)d_{ij}^{-3} \end{pmatrix}.$$

The Hessian of f_{ij} is a 4×4 matrix, which is the second derivative of f_{ij} and each element in the ith row and jth column is $\frac{\partial f_{ij}}{\partial X_i X_j}$. We first list each element as follows:

$$\begin{aligned}
\frac{\partial f_{ij}}{\partial {X_i}^2} = \frac{\partial f_{ij}}{\partial {X_j}^2} &= c_{ij}d_{ij}^{-1} - (c_{ij}(X_i - X_j)^2 + s_{ij}b)d_{ij}^{-3} \\
&\quad + 3s_{ij}b(X_i - X_j)^2 d_{ij}^{-5} = A \\
\frac{\partial f_{ij}}{\partial {Y_i}^2} = \frac{\partial f_{ij}}{\partial {Y_j}^2} &= c_{ij}d_{ij}^{-1} - (c_{ij}(Y_i - Y_j)^2 + s_{ij}b)d_{ij}^{-3} \\
&\quad + 3s_{ij}b(Y_i - Y_j)^2 d_{ij}^{-5} = B \\
\frac{\partial f_{ij}}{\partial X_i X_j} = \frac{\partial f_{ij}}{\partial X_j X_i} &= -c_{ij}d_{ij}^{-1} + (c_{ij}(X_i - X_j)^2 + s_{ij}b)d_{ij}^{-3} \\
&\quad - 3s_{ij}b(X_i - X_j)^2 d_{ij}^{-5} = -A \\
\frac{\partial f_{ij}}{\partial Y_i Y_j} = \frac{\partial f_{ij}}{\partial Y_j Y_i} &= -c_{ij}d_{ij}^{-1} + (c_{ij}(Y_i - Y_j)^2 + s_{ij}b)d_{ij}^{-3} \\
&\quad - 3s_{ij}b(Y_i - Y_j)^2 d_{ij}^{-5} = -B \\
\frac{\partial f_{ij}}{\partial X_i Y_i} = \frac{\partial f_{ij}}{\partial Y_i X_i} &= \frac{\partial f_{ij}}{\partial X_j Y_j} = \frac{\partial f_{ij}}{\partial Y_j X_j} \\
&= (X_i - X_j)(Y_i - Y_j)(-c_{ij}d_{ij}^{-3} + 3s_{ij}bd_{ij}^{-5}) = C \\
\frac{\partial f_{ij}}{\partial X_j Y_i} = \frac{\partial f_{ij}}{\partial Y_i X_j} &= \frac{\partial f_{ij}}{\partial X_i Y_j} = \frac{\partial f_{ij}}{\partial Y_j X_i} \\
&= (X_i - X_j)(Y_i - Y_j)(c_{ij}d_{ij}^{-3} - 3s_{ij}bd_{ij}^{-5}) = -C.
\end{aligned}$$

Thus, the Hessian matrix of f_{ij} is as follows:

$$\nabla^2 f_{ij}(X_i,\ X_j,\ Y_i,\ Y_j) = \begin{pmatrix} \frac{\partial f_{ij}}{\partial {X_i}^2} & \frac{\partial f_{ij}}{\partial X_j X_i} & \frac{\partial f_{ij}}{\partial Y_i X_i} & \frac{\partial f_{ij}}{\partial Y_j X_i} \\ \frac{\partial f_{ij}}{\partial X_i X_j} & \frac{\partial f_{ij}}{\partial {X_j}^2} & \frac{\partial f_{ij}}{\partial Y_i X_j} & \frac{\partial f_{ij}}{\partial Y_j X_j} \\ \frac{\partial f_{ij}}{\partial X_i Y_i} & \frac{\partial f_{ij}}{\partial X_j Y_i} & \frac{\partial f_{ij}}{\partial {Y_i}^2} & \frac{\partial f_{ij}}{\partial Y_j Y_i} \\ \frac{\partial f_{ij}}{\partial X_i Y_j} & \frac{\partial f_{ij}}{\partial x_j Y_j} & \frac{\partial f_{ij}}{\partial Y_i Y_j} & \frac{\partial f_{ij}}{\partial {Y_j}^2} \end{pmatrix} = \begin{pmatrix} A & -A & C & -C \\ -A & A & -C & C \\ C & -C & B & -B \\ -C & C & -B & B \end{pmatrix}.$$

It is easy to show that $\nabla^2 f_{ij}(X_i,\ X_j,\ Y_i,\ Y_j) = 0$. ■

Corollary 1: The piecewise function $f_{ij} : \Re^4 \rightarrow \Re$ in (7) is a convex function on an open convex set $C \subset \Re^4$.

Proof: According to the well-known fact [9] stated as follows: if $f : \Re^n \rightarrow \Re$ is twice continuously differentiable, then f is convex on an open convex set $C \subset \Re^n$ if $\bigtriangledown^2 f$ is positive semidefinite for all points in C.

Thus, f_{ij} is a convex function based on Lemma 1. ■

We have shown f_{ij} is a convex function. Now, we can derive that PP model [see (4)] is a also a convex function as follows.

Corollary 2: PP model is a convex function on an open convex set $C \subset \Re^4$.

Proof: PP model combines the left part of the function $c_{ij}d_{ij}+s_{ij}p_{ij}/d_{ij}$ and the right part of the function $c_{ij}d_{ij}+p_{ij}/d_{ij}$ from the input domain separated at the point d_{ij}. Since we have proven that the two functions are convex in Corollary 1, we can say PP model is a convex function according to the following fact [9]: if $f : \Re^n \rightarrow \Re$ and $g : \Re^n \rightarrow \Re$ are convex functions, then so is $h = max(f, g)$. ■

Finally, we can show the objective function F is also a convex function.

Theorem 1: Let $F : \Re^{2n(n-1)} \mapsto \Re$ be given by

$$F = \sum_{1 \leq i < j \leq n} f_{ij}(X_i, Y_i, X_j, Y_j) \tag{8}$$

is a convex function, where n is the number of circles.

Proof: We have derived that f_{ij} is a convex function in Corollary 2. Since the objective function is a summation for each pair of circles C_i and C_j, the objective function F is a convex function based on the following fact [9]: if $f : \Re^n \rightarrow \Re$ and $g : \Re^n \rightarrow \Re$ are convex functions, then so is $h = (f + g)$. ■

IV. Local Legalization Stage

After the first stage, the circles that represent modules have been uniformly distributed over a specified region and the total wirelength is minimized. Now, we have to determine the exact locations and shapes of modules so that no two modules overlap and all modules are placed inside the outline. To maintain the optimization result of the first stage, we first show a procedure to extract geometric relations of modules from the circles' distribution and record them by constraint graphs.

A. Constraint Graphs Construction

We utilize the DT method [17] to find neighboring modules. Given a distribution of objects, neighboring objects can be identified by an edge in the resulting graph, named a DT graph, after applying the method. Then, we assign the geometric relation between each pair of objects according to their exact locations. Finally, a horizontal constraint graph (denoted by C_h) and a vertical constraint graph (C_v) are built according to these geometric relations.

A DT graph is composed of a set of triangles. For each triangle [denoted by $\triangle(n_i, n_j, n_k)$] in a DT graph, node n_i denotes a module m_i, and an edge (n_i, n_j) denotes that m_i and m_j are in the neighboring locations. For each pair of modules corresponding to an edge in a triangle, their geometric relation needs to be assigned. To get the geometric relations between the modules, we first recover the shape of each module from a circle to a square and place it at the location with its center overlapping original circle's center. Depending on whether two modules overlap, we can respectively determine their geometric relation as follows.

1) m_i *and* m_j *separate:* The rectangle, denoted by $R^s_{i,j}$, formed by the centers of the modules is taken. Let d_w and d_h denote the width and height of the rectangle. If $d_w > d_h$, m_i, and m_j have *horizontal* relation (denote by $m_i \vdash m_j$). Otherwise, m_i and m_j have *vertical* relation (denote by $m_i \perp m_j$).
2) m_i *and* m_j *overlap:* The rectangle, denoted by $R^o_{i,j}$, formed by the overlapped area is taken. Opposite of the previous case, $m_i \perp m_j$ if $d_w > d_h$; otherwise, $m_i \vdash m_j$.

An edge (n_i, n_j) is called a transitive edge (n_i, n_j) if there exists a path between n_i and n_j except the edge (n_i, n_j); otherwise, it is a reduction edge. In order to decrease the number of constraints, the transitive edge (n_i, n_j) should not be added to the constraint graphs during construction. Such edge can be easily eliminated during building constraint graphs. In each triangle, if the three edges have the same geometric relation, we only need to ignore the edge with the longest length. However, after this procedure, some pairs of modules still do not have geometric relations even through the transitive relations between modules (e.g., if $m_i \perp m_j$ and $m_j \perp m_k$, then $m_i \perp m_k$). To avoid overlaps, we need to add geometric relations for these modules. However, in real condition, the probability that two modules separate in a long distance to overlap is quite low if the distribution of modules is uniform. Thus, in order to reduce the number of constraints, we only add additional geometric relations for those modules corresponding to the nodes in the diagonal of the proximity triangles. If they originally do not have a geometrical relation, an additional edge is added according to the rule introduced above.

Fig. 6 gives an example to construct horizontal and vertical constraint graphs. The circles in Fig. 6(a) show the initial distribution of circles and the dotted lines enclosing the circles form the initial shapes of modules. After applying the DT method on the centers of the circles, a set of triangles is obtained as shown in Fig. 6(b). Note that each pair of neighboring modules are described by an edge in a triangle, and their geometric relations should be determined. For example, take the edge (n_1, n_{10}) in the triangle $\triangle(n_1, n_3, n_{10})$. Since the modules m_{10} and m_1 overlap in the initial placement, we use the rectangle $R^o_{10,1}$ formed by the overlap area of the two modules to determine their relation. Because $d_w < d_h$, we consider their relation as $m_{10} \vdash m_1$ since it is more efficient to separate them in horizontal direction. Since all the three edges in $\triangle(n_1, n_3, n_{10})$ have horizontal geometric relations, the longest edge (n_{10}, n_3) should be ignored directly. For an example of the other case, see the edge (n_7, n_9) in the triangle $\triangle(n_2, n_7, n_9)$. Because m_7 and m_9 do not overlap, the rectangle $R^s_{7,9}$ formed by the centers of m_7 and m_9 is used to determine their relation. Since $d_w < d_h$, we consider

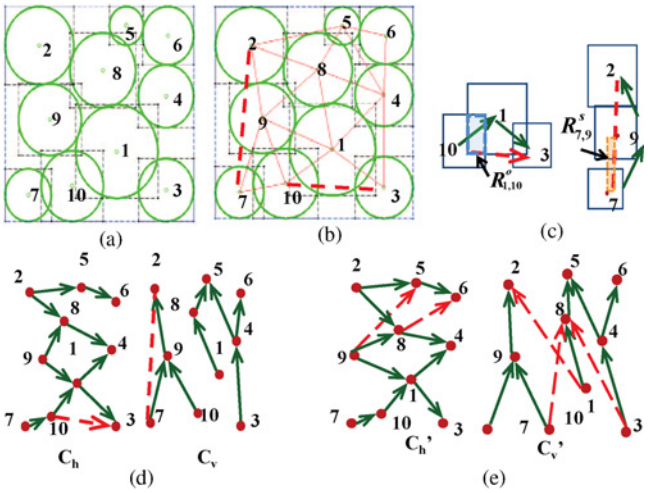


Fig. 6. (a) Distribution of circles after the global distribution stage. (b) DT graph after applying DT method on the result shown in (a). (c) Two cases for determining the geometric relation of two modules. (d) Horizontal and vertical constraint graphs C_h and C_v. (e) Resulting constraint graphs C_h' and C_v' after adding additional edges.

$m_7 \perp m_9$ because it has larger space in the vertical direction. Similarly, the longest edge (n_7, n_2) is also ignored because the three edges all have vertical relations. The initial constraint graphs are shown in Fig. 6(d). Note that some pairs of modules still do not have geometric relations after this procedure such as m_2 and m_3 or m_1 and m_2. Although m_2 and m_3 have no geometric relation in the constraint graphs, we do not add an additional edge since they are far from each other. (There exist several modules around m_2 and m_3. The two modules can overlap if these modules are moved away. This cannot happen in the fixed outline floorplan with zero dead space if the distribution is uniform in the global distribution.) For modules in neighboring triangles, we may try to add a geometric relation to avoid overlap during legalization. For example, we would try to add the geometric relation for modules m_1 and m_2 which correspond to the nodes in the diagonal of the parallelogram formed by $\triangle(n_2, n_8, n_9)$ and $\triangle(n_1, n_8, n_9)$. Because their geometric relation is not defined in the original constraint graphs, the edge (n_1, n_2) is added to C_v based on their actual locations. The resulting constraint graphs C_h' and C_v' are shown in Fig. 6(e).

B. SOPL: SOCP for Legalization

Before introducing second-order cone programming for legalization (SOPL), we need the following definitions.

1) (x_i, y_i): (x_i, y_i) denotes the bottom-left coordinate of a module m_i.
2) N_s^h (or N_e^h): N_s^h (N_e^h) denotes a set of nodes with zero in-degree (out-degree) in C_h.
3) N_s^v (or N_e^v): N_s^v (N_e^v) denotes a set of node with zero in-degree (out-degree) in C_v.

According to the constraints defined in the constructed constraint graphs introduced in the previous section, we use a SOCP [22] to reshape modules so that modules can be placed into a fixed-outline and no two modules overlap as shown in Fig. 7.

$$\min \{|W - W_f| + |H - H_f|\} \tag{11}$$

s.t.

$$x_i + w_i \le x_j, \quad \forall (n_i, n_j) \in C_h \tag{12}$$
$$y_i + h_i \le y_j, \quad \forall (n_i, n_j) \in C_v \tag{13}$$
$$x_s \ge 0, \quad \forall n_s \in N_s^h \tag{14}$$
$$x_e + w_e \le W \quad \forall n_e \in N_e^h \tag{15}$$
$$y_s \ge 0, \quad \forall n_s \in N_s^v \tag{16}$$
$$y_e + h_e \le H, \quad \forall n_e \in N_e^v \tag{17}$$
$$w_i \times h_i \ge A_i, \quad \forall n_i \in C_h \tag{18}$$

Fig. 7. SOCP for Legalization (SOPL).

To satisfy the *fixed-outline constraint*, the objective function tries to minimize the difference between predefined width (height) and the resulting width (height) as shown in the objective function [see (11)] in Fig. 7. By defining $D_x = |W - W_f|$ and $D_y = |H - H_f|$, we can substitute the objective function by $D_x + D_y$. However, we have to add the following four constraints $D_x > W - W_f$, $D_x > W_f - W$, $D_y > H - H_f$, and $D_y > H_f - H$ to make sure the values of D_x and D_y positive. Inequations (12) and (13) are *non-overlap constraints* for neighboring modules in horizontal and vertical constraint graphs, respectively. Inequations (14)–(17) are *boundary constraints* that restrict all boundary modules are placed inside the outline. Inequality (14) ((15)) makes modules in the left (right) boundary whose x-coordinates larger (smaller) than zero (W). Similarly, the y-coordinates of modules in the bottom (top) boundary are larger (smaller) than zero (H) by Inequality (16) ((17)). Note that once those modules in the boundary are constrained, other modules can be placed inside the boundary as well. Finally, to guarantee that each module m_i has enough space, we need to add the constraint $w_i \times h_i = A_i$. Similar to Luo *et al.* [23], we would relax the constraint as $w_i \times h_i \ge A_i$ [see Inequality (18)], and it can be further transformed it into a second-order constraint [3] as follows:

$$\begin{aligned}
&\iff && {h_i}^2 + 2h_i w_i + {w_i}^2 && \ge {h_i}^2 - 2h_i w_i + {w_i}^2 + 4A_i \\
&\iff && (h_i + w_i)^2 && \ge (h_i - w_i)^2 + 4A_i \ge 0 \\
&\iff && h_i + w_i && \ge \sqrt{(h_i - w_i)^2 + \left(2\sqrt{A_i}\right)^2} \\
&\iff && h_i + w_i && \ge \left\| \begin{pmatrix} h_i - w_i \\ 2\sqrt{A_i} \end{pmatrix} \right\|_2.
\end{aligned}$$

C. Hierarchical UFO Flow

We do not add the aspect ratio constraints in the above formulation. The fixed-outline floorplanning with zero dead space is not easy to be satisfied if the aspect ratio constraints are constrained. Thus, we use hierarchical flow to deal with the problem in this section.

If there exist modules violating the aspect ratio constraint after applying SOPL, we only have to deal with the modules in the corresponding subregion. Therefore, we divide the region quadratically into four subregions. For the subregion which has violating modules, we first globally redistribute the modules in the subregion by applying PP model. Then, these modules can be legalized again by applying SOPL. Experiments

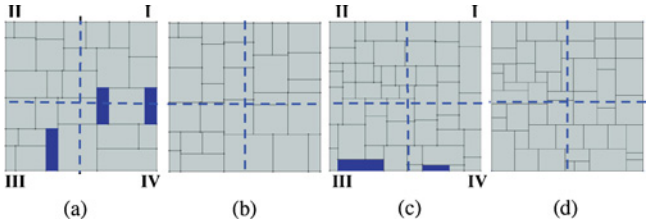


Fig. 8. (a) Initial floorplan of n30. (b) Floorplan of n30 after the hierarchical flow. (c) Initial floorplan of n50. (d) Floorplan of n50 after the hierarchical flow.

have shown that we only have to perform this procedure once before the aspect ratio constraints can be satisfied. This procedure not only repairs violations but also improves wirelength.

For example, Fig. 8(a) is an initial floorplan of n30 after first applying SOPL with zero dead space. There exist three modules violating the aspect ratio constraint, which are denoted by dark colors. We divide the region into four subregions I, II, III, and IV as labels in the figure. Since the violated modules are in the subregions III and IV, we only redistribute modules in these subregions. Then, SOPL is applied again to obtain a legal floorplan as shown in Fig. 8(b). Note that the locations of modules in subregions I and II are almost not changed. Fig. 8(c) and (d) shows another example for the circuit n50.

V. Floorplanning With PPM

In the real design flow, there may exist several modules pre-placed on a die before floorplanning, which can be considered as blockages. In order to get a feasible floorplan, it is necessary to keep away from these blockages while placing modules. Thus, we would like to handle fixed-outline floorplanning with pre-placed modules in this section.

UFO can deal with PPM easily due to its mathematical formulation. Since the coordinates of PPM have been obtained, we only need to assign the values to the centers of the corresponding circles in the global distribution stage. Suppose the bottom-left coordinate of a PPM m_p is $(x_p,\ x_p)$. Then, the coordinate $(X_p,\ Y_p)$ of the corresponding circle is assigned to $(x_p+r,\ y_p+r)$, where r is the radius of the circule. If the circle is outside the outline in the x (y) direction [i.e., $X_p+r \geq W_f$ $(Y_p+r \geq H_f)$], we adjust X_p (Y_p) as $W_f - r$ as $(H_f - r)$ to make it inside the outline. In this way, all circles can be moved arbitrarily except these circles which are placed on the fixed locations.

Similar to other modules, we do not give aspect ratio constraints for PPM during SOPL. Therefore, they may overlap with other modules when their shapes are recovered. Thus, we need to adjust the shapes of modules overlapped by them. For each triangle $\triangle$ associated with a PPM m_p in a DT graph, the module m_i corresponding to another node in the triangle are considered as a surrounding module. After identifying these modules, we can obtain a legal floorplan by reshaping these modules as rectilinear modules using a horizontal or a vertical cut line.

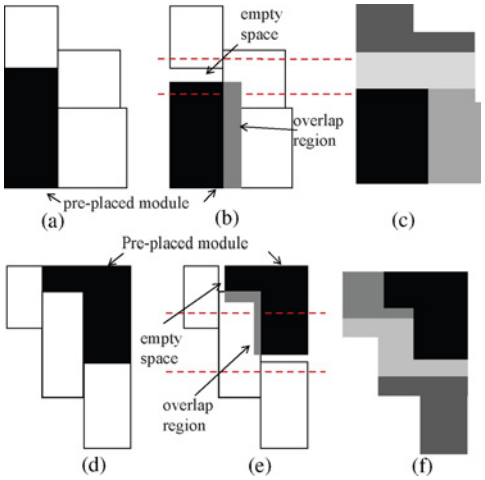


Fig. 9. Legalization of PPM. (a) Initial shapes of modules after SOPL (i.e., the black module denotes a PPM). (b) Recovering of the pre-placed module. (c) Resulting floorplan. (d)–(f) Another example with a rectilinear PPM.

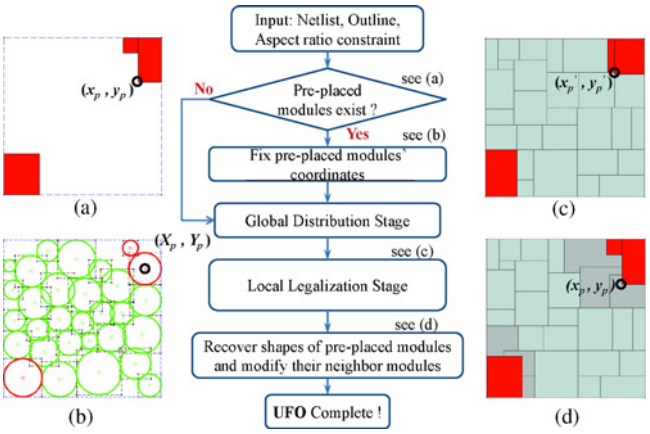


Fig. 10. Overall flow of UFO with PPM. (a) Design with two pre-placed modules. (b) Fixing the locations of the circles corresponding to pre-placed modules. (c) Shapes of modules after local legalization stage. (d) Recovering shapes of pre-placed modules and modifying the shapes of their neighboring modules.

Fig. 9 shows the procedure to deal with PPM. Fig. 9(a) shows the inial shapes of modules after SOPL. The black module denotes a PPM. After the shape of the PPM is recovered, it may overlap with neighboring modules and release some space as shown in Fig. 9(b). We can get a feasible floorplan by reshaping these modules by dividing the space horizontally as shown in Fig. 9(c). Fig. 9(d) shows another example when the shape of PPM is rectilinear. The shape of module is recovered in Fig. 9(e). Similar procedure can be applied. The resulting floorplan is shown in Fig. 9(f).

Fig. 10 shows the design flow for PPM. First, all modules are transformed into circles. If a design contains PPM, we first set the locations of corresponding circles. Except these circles, other circles are globally distributed among a placement region according to our PP mode. Based on the resulting distribution, we perform SOPL to determine the exact locations and shapes of modules. Finally, we recover the shapes of PPM as well as other modules to obtain a legal floorplan. In order to model modules with extreme aspect ratios correctly, we can divide

the modules into multiple pieces and use multiple circles to model each part accurately. With this modification, it is easy to recover the original shape of the module.

VI. Experimental Results

This section shows our experimental results, which can be divided into two parts depending on whether a chip contains PPM. We have used MATLAB to implement our UFO system on a 1.6 GHz SUN Blade-2500 workstation with 4 GB memory. The wirelength was estimated by HPWL in all experiments.

A. Fixed-Outline Floorplanning Without PPM

In Table II, we first compare UFO with all the best published results of the state-of-art fixed-outline floorplanners such as ZDS (PATOMA) [10], SAFFOA [14], and Luo *et al.* [23], which can achieve zero dead space floorplanning using soft modules. The experiments were taken on the circuits from the GSRC and MCNC benchmarks, and the aspect ratio constraint for modules is between 1/3 and 3. I/O pads were fixed at the locations given by the benchmark for fair comparison with previous methods. First, we show the results of UFO with hierarchical flow and without hierarchical flow in columns 2–4 and 5–8, respectively. No module violates the aspect ratio constraint after hierarchical flow while there exist a small number of modules violating it without the flow as shown in columns 4 and 8, respectively. Note that the number does not increase proportional to the number of modules in a circuit, which demonstrates that our global distribution stage can actually distribute modules to the placement region, and its optimization results can be maintained during the local legalization stage. This is why UFO can obtain better results comparing to previous works. Although the runtime is longer with the hierarchical flow, it can further reduce wirelength by 11.37%. Comparing to SAFFOA [14], ZDS [11], and Luo *et al.* [23], hierarchical UFO can greatly reduce wirelength by 31.20%, 41.50%, and 38.22%. The value $Avg.\Delta$ is computed by the equation [(each wire length)/(wire length of UFO with hierarchical flow)-1].

We performed another experiments by changing the aspect ratios of chips. We compared UFO with SAFFOA [14] and DeFer [36]. For fair comparisons, I/O pads were also fixed at the locations given by the benchmark. DeFer [36] used 1% whitespace in their experiments. Table III shows that UFO can outperform SAFFOA and DeFer by 19.33% and 1.08% in average, respectively. Fig. 11(a) and (b) shows the resulting floorplan of UFO for n300 (n200) while the aspect ratio of a chip is 1:1 (2:1).

B. Fixed-Outline Floorplanning With PPM

We also used UFO to handle fixed-outline floorplanning with PPM and the results are shown in Table IV. The circuits were generated from GSRC circuits by randomly picking some modules and placing them in arbitrary locations of a die area. The experiments were performed on outlines with different dead spaces 0%, 5%, and 10%, and the number of PPM, wirelength, and runtime are shown in the table. Note that the wirelength becomes longer when the dead space is larger since the modules are not compacted in such condition. Fig. 12 shows the fixed-outline floorplans with PPM for n100 and n200 with zero dead space. The PPM are denoted by the red color and their neighbor modules are reshaped. While the dead space is larger than 5%, UFO could accomplish floorplanning without reshaping those modules connected to pre-placed ones. Fig. 13 shows the experimental results of n100 and n300 with 10% dead space. Note that all modules are rectangular in the figures. Fig. 14 shows that UFO could handle the condition that some modules do not abut to boundaries.

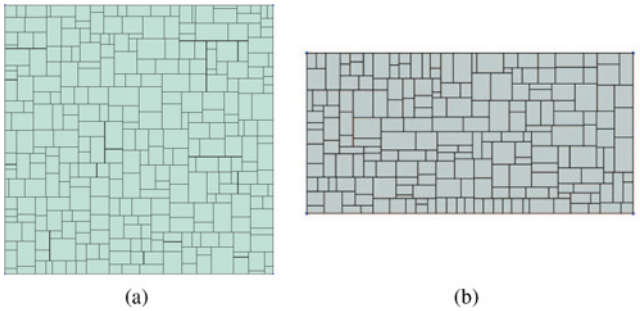


Fig. 11. Fixed-outline floorplans with zero dead space for (a) n300 (aspect ratio of a chip = 1:1, HPWL: 476560μm) and (b) n200 (aspect ratio of a chip = 2:1, HPWL: 381320μm).

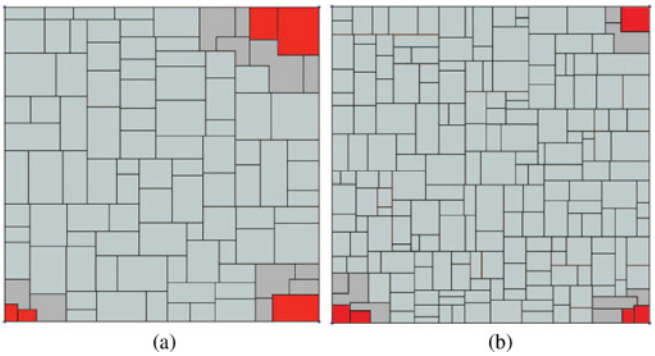


Fig. 12. Fixed-outline floorplans with PPM. (a) n100 (no. of PPM is 5, HPWL: 243080μm). (b) n200 (no. of PPM is 5, HPWL: 399510μm).

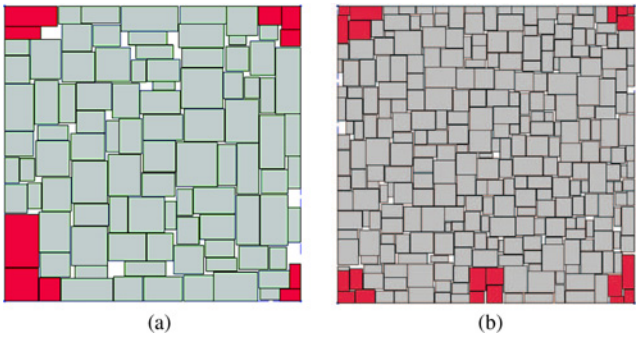


Fig. 13. Fixed-outline floorplans with PPM with 10% dead space. (a) n100 (no. of PPM is 10, HPWL: 259430μm). (b) n300 (no. of PPM is 20, HPWL: 598520μm).

TABLE II

WIRELENGTH AND RUNTIME COMPARISONS BETWEEN UFO AND OTHER APPROACHES WHEN THE ASPECT RATIO OF A CHIP IS 1:1

Circuit	UFO							ZDS (PATOMA) [10]			SAFFOA [14]			Luo *et al.* [23]		
	With Hierarchical Flow			Without Hierarchical Flow												
	Wire (μm)	Time (s)	No. of violated modules	Wire (μm)	Δ (%)	Time (s)	No. of violated modules	Wire (μm)	Δ (%)	Time (s)	Wire (μm)	Δ (%)	Time (s)	Wire (μm)	Δ (%)	Time (s)
n10	36 398	3	0	45 193	24.16	3	0	52 258	43.57	1	46 207	26.94	2	–	–	–
n30	102 100	87	0	117 420	15.00	45	3	156 921	53.69	1	138 218	35.37	14	–	–	–
n50	124 300	204	0	145 330	16.91	124	2	180 115	44.90	1	165 366	33.03	39	–	–	–
n100	195 200	677	0	206 440	5.75	452	5	283 452	45.21	2	262 469	34.46	158	285 070	46.00	–
n200	346 660	3306	0	358 200	3.32	2310	9	505 736	45.88	3	480 573	38.63	690	506 070	45.98	–
n300	476 560	4718	0	491 480	3.13	2785	10	566 242	15.77	4	551 720	18.81	1563	584 640	22.67	–
Avg. Δ	0			11.37				41.50			31.20			38.22		

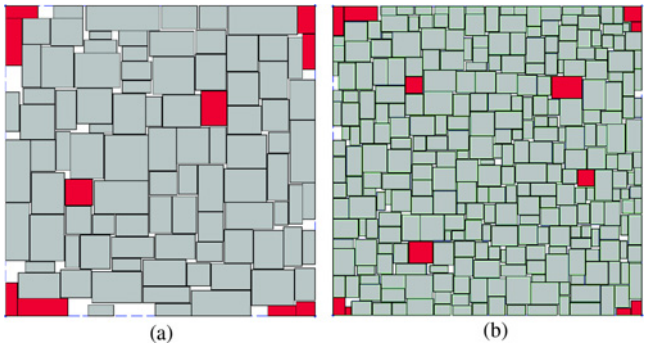
I/O pads are fixed at the locations given by the benchmark.

TABLE III

WIRELENGTH COMPARISONS FOR UFO AND OTHER APPROACHES WHILE DIFFERENT ASPECT RATIOS 1:1, 2:1, AND 3:1 OF CHIPS ARE APPLIED

Circuit	1:1			2:1			3:1		
	UFO	SAFFOA	DeFer	UFO	SAFFOA	DeFer	UFO	SAFFOA	DeFer
	Wire (μm)	Wire (μm)	Wire (μm)	Wire (μm)	Wire (μm)	Wire (μm)	Wire (μm)	Wire (μm)	Wire (μm)
n100	195 200	263 200	197 568	214 430	281 509	215 486	235 210	296 229	235 675
n200	346 660	480 014	358 230	381 320	520 802	382 190	406 610	536 040	412 226
n300	476 560	554 240	478 531	510 020	615 713	513 198	541 240	617 554	551 704
ami33	50 699	59 431	–	52 252	62 272	–	55 827	67 364	–
ami49	671 920	625 541	–	687 611	641 746	–	699 540	675 540	–
Avg. Δ	0	19.96%	1.65%	0	20.21%	0.44%	0	17.81%	1.16%

I/O pads are fixed at the locations given by the benchmark.



(a) (b)

Fig. 14. Fixed-outline floorplans with PPM with 10% dead space. (a) n100 (no. of PPM is 10, HPWL: 256520μm). (b) n300 (no. of PPM is 12, HPWL: 599310μm).

TABLE IV

WIRELENGTH AND RUNTIME COMPARISONS WITH PPM

Circuit	0% deadspace			5% deadspace			10% deadspace		
	No. of PPM	Wire (μm)	Time (s)	No. of PPM	Wire (μm)	Time (s)	No. of PPM	Wire (μm)	Time (s)
n10	1	45 701	3	1	43 533	3	1	41 089	3
n30	2	117 430	22	3	127 660	20	3	123 370	21
n50	3	154 510	42	5	163 040	39	5	164 300	39
n100	5	243 080	164	10	248 270	138	10	259 430	136
n200	5	399 510	815	15	417 110	737	15	472 680	725
n300	5	554 380	1841	20	585 030	1522	20	598 520	1539

VII. CONCLUSION

We have proposed an analytical-based approach, named UFO, for fixed-outline floorplanning. The method not only can achieve zero dead space floorplanning but also can handle PPM. Experimental results on the GSRC and MCNC benchmarks have demonstrated that our methodology outperforms the results reported in the literature.

REFERENCES

[1] S. N. Adya and I. L. Markov. "Fixed-outline floorplanning: Enabling hierarchical design," *IEEE Trans. Very Large Scale Integr.*, vol. 11, no. 6, pp. 1120–1135, Dec. 2003.

[2] S. N. Adya, S. Chaturvedi, J. A. Roy, D. A. Papa, and I. L. Markov, "Unification of partitioning, placement and floorplanning," in *Proc. ICCAD*, 2004, pp. 550–557.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[4] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu. "B*-trees: A new representation for non-slicing floorplans," in *Proc. DAC*, 2000, pp. 458–463.

[5] T.-C. Chen, Y.-W. Chang, and S.-C. Lin, "A new multilevel framework for large-scale interconnect-driven floorplanning," *IEEE Trans. Comput.-Aided Des.*, vol. 27, no. 2, pp. 286–294, Feb. 2008.

[6] T. Chen and Y. W. Chang, "Modern floorplanning based on B*-tree and fast simulated annealing," *IEEE Trans. Comput.-Aided Des.*, vol. 25, no. 4, pp. 637–650, Apr. 2006.

[7] S. Chen and T. Yoshimura, "Fixed-outline floorplanning: Block-position enumeration and a new method for calculating area costs," *IEEE Trans. Comput.-Aided Des.*, vol. 27, no. 5, pp. 858–871, May 2008.

[8] S. Dhamdhere, N. Zhou, and T.-C. Wang, "Module placement with pre-placed modules using the corner block list representation," in *Proc. ISCAS*, 2002, pp. 349–352.

[9] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*, 2nd ed. New York: Wiley, 2001.

[10] J. Cong, M. Romesis, and J. Shinnerl, "Fast floorplanning by look ahead enabled recursive bipartitioning," *IEEE Trans. Comput.-Aided Des.*, vol. 25, no. 9, pp. 1719–1732, Sep. 2006.

[11] J. Cong, G. Nataneli, M. Romesis, and J. Shinnerl, "An area-optimality study of floorplanning," in *Proc. ISPD*, 2004, pp. 78–83.
[12] Y. Feng, D. P. Mehta, and H. Yang, "Constrained modern floorplanning," in *Proc. ISPD*, 2003, pp. 128–135.
[13] P.-N. Guo, T. Takahashi, C.-K. Cheng, and T. Yoshimura, "Floorplanning using a tree representation," *IEEE Trans. Comput.-Aided Des.*, vol. 20, no. 2, pp. 281–289, Feb. 2001.
[14] O. He, S. Dong, J. Bian, S. Goto, and C.-K. Chen, "A novel fixed-outlined floorplanner with zero dead space for hierarchical design," in *Proc. ICCAD*, 2008, pp. 16–23.
[15] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C.-K. Cheng, and J. Gu, "Corner block list: An effective and efficient topological representation of non-slicing floorplan," in Proc. ICCAD, 2000, pp. 8–12.
[16] Y.-H. Jiang, J. La, and T.-C. Wang, "Module placement with pre-placed modules using the B*-tree representation," in *Proc. ISCAS*, 2001, pp. 347–350.
[17] L. Jin, D. Kim, L. Mu, D.-S. Kim, and S.-M. Hu, "A sweepline algorithm for Euclidean Voronoi diagram of circules," *IEEE Trans. Comput.-Aided Des.*, vol. 38, no. 3, pp. 260–272, Mar. 2006.
[18] A. B. Kahng, "Classical floorplanning harmful?" in *Proc. ISPD*, 2000, pp. 207–213.
[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[20] C. Lin, D. Chen, and Y. Wang, "Robust fixed-outline floorplanning through evolutionary search," in *Proc. ASP-DAC*, 2004, pp. 42–44.
[21] J.-M. Lin and Y.-W. Chang, "TCG: A transitive closure graph based representation for general floorplans," *IEEE Trans. Very Large Scale Integr.*, vol. 13, no. 4, pp. 288–292, Apr. 2005.
[22] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second order cone programming," *Linear Algebra Its Applicat.*, vol. 284, pp. 193–228, Nov. 1998.
[23] C. Luo, M. F. Anjos, and A. Vannelli, "Large-scale fixed-outline floorplanning design using convex optimization techniques," in *Proc. ASP-DAC*, 2008, pp. 198–203.
[24] T.-S. Moh, T.-S. Chang, and S. L. Hakimi, "Globally optimal floorplanning for a layout problem," *IEEE Trans. Circuits Syst.*, vol. 43, no. 9, pp. 713–720, Sep. 1996.
[25] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-packing based module placement," in *Proc. ICCAD*, 1995, pp. 472–479.
[26] H. Murata, K. Fujiyoshi, and M. Kaneko, "VLSI/PCB placement with obstacles based on sequence-pair," in *Proc. ISPD*, 1997, pp. 26–31.
[27] H. Murata and E. S. Kuh, "Sequence-pair based placement method for hard/soft/pre-placed modules," in *Proc. ISPD*, 1998, pp. 167–172.
[28] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module placement on BSG-structure and IC layout applications," in *Proc. ICCAD*, 1996, pp. 484–491.
[29] S. Nakatake, M. Furuya, and Y. Kajitani, "Module placement on BSG-structure with pre-placed modules and rectilinear modules," in *Proc. ASP-DAC*, 1998, pp. 571–576.
[30] R. H. J. M. Otten, "Automatic floorplan design," in *Proc. DAC*, Jun. 1982, pp. 261–267.
[31] H. Onodera, Y. Taniguchi, and K. Tamaru, "Branch-and-bound placement for building block layout," in *Proc. DAC*, 1991, pp. 433–439.
[32] N. R. Quinn, Jr., "The placement problem as viewed from the physics of classical mechanics," in *Proc. DAC*, 1975, pp. 173–178.
[33] N. R. Quinn, Jr., and M. A. Breuer, "A forced directed component placement procedure for printed circuit boards," *IEEE Trans. Circuits Syst.*, vol. CAS-26, no. 3, pp. 377–388, Jun. 1979.
[34] L. Sha and R. W. Dutoon, "An analytical algorithm for placement of arbitrary sized rectangular blocks," in *Proc. DAC*, 1985, pp. 602–608.
[35] D. F. Wong and C.-L. Liu, "A new algorithm for floorplan design," in *Proc. DAC*, Jun. 1986, pp. 101–107.
[36] J. Z. Yan and C. Chu, "DeFer: Deferred decision making enabled fixed-outline floorplanning algorithm," *IEEE Trans. Comput.-Aided Des.*, vol. 29, no. 3, pp. 367–381, Mar. 2010.
[37] F. Y. Yong and D. F. Wong, "Slicing floorplans with pre-placed modules," in *Proc. ICCAD*, 1998, pp. 252–258.
[38] H. Youssef, S. M. Sait, and K. J. Al-Farra, "Timing influenced force directed floorplanning," in *Proc. Eur. Des. Automat. Conf.*, 1995, pp. 156–161.
[39] H. Zhou and J. Wang, "ACG: Adjacent constraint graph for general floorplans," in *Proc. ICCD*, 2004, pp. 572–575.
[40] Y. Zhan, Y. Feng, and S. Sapatnekar, "A fixed-die floorplanning algorithm using an analytical approach," in *Proc. ASP-DAC*, 2006, pp. 771–776.

Jai-Ming Lin (M'08) received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1996, 1998, and 2002, respectively, all in computer science.

From 2002 to 2007, he was an Assistant Project Leader with the Computer-Aided Design Team, Realtek Corporation, Science Park, Hsinchu. Currently, he is an Assistant Professor with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. His current research interests include floorplan, placement, routing, and clock tree synthesis.

Zhi-Xiong Hung received the B.S. degree from the National University of Arousing, Arousing, Taiwan, in 2008, and the M.S degree from National Cheng Kung University, Tainan, Taiwan, in 2010, both in electrical engineering.

He is currently fulfilling the compulsory military service. His current research interests include floorplanning and placement.