國立交通大學

電子工程學系 電子研究所

碩士論文

使用高斯梯度與定點技巧的佈局演算法

Quadratic Placement Algorithm Using Gaussian Blurring and Fixed Point Technique

研 究 生：黃川嘉

指導教授：陳宏明 教授

中 華 民 國 一 〇 二 年 九 月

使用高斯梯度與定點技巧的佈局演算法

Quadratic Placement Algorithm Using Gaussian Blurring and Fixed Point Technique

研 究 生：黃川嘉　　　Student：Chuan-Chia Huang

指導教授：陳宏明 教授　　Advisor：Prof. Hung-Ming Chen

國 立 交 通 大 學

電子工程學系 電子研究所

碩 士 論 文

A Thesis

Submitted to Department of Electronics Engineering and

Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electronics Engineering

September 2013

Hsinchu, Taiwan, Republic of China

中華民國一〇二年九月

使用高斯梯度與定點技巧的佈局演算法

研究生：黃川嘉　　　　指導教授：陳宏明教授

國立交通大學 電子工程學系 電子研究所

摘 要

頂尖的全域佈局演算法採用不同的邏輯閘擴散演算法來處理邏輯閘密度。在這項研究裡，我們發現頂尖的全域佈局演算法對於大型障礙物周圍以及穿越大型障礙物處理擺放行爲的不同。透過我們的研究結果，佈局演算法在大型障礙物周圍的擺放行爲對於可繞線性有顯著的影響，而這項觀察無法由傳統的評價指標去做分析。在解二次方程的佈局演算法中，SimPL [1] 和 Kraftwerk2 [2]所採用的方法代表了兩種不同的邏輯閘擴散演算法類別。我們重新實作兩種演算法類別並實際運用在邏輯閘擺放實驗，我們解釋了這兩種邏輯閘擺放演算法對於在全域與局部兩種規模的控制性有著明顯的不同。我們進一步觀察這兩種演算法對於可繞線性的影響。

爲了同時處理全域與局部兩種規模的擺放行爲，我們提出了一個兩階段的全域佈局演算法。第一階段的目的是藉由移動邏輯閘跨過大型障礙物以取得精準的可用空間分布。第二階段的目的是藉由邏輯閘在大型障礙物周圍的移動以取得邏輯閘之間正確的相對關係。我們提出的全域佈局演與 ComPLx [3]有著相同品質的實驗結果表現，而且基於在大型障礙物周圍的擺放行爲使得可繞線性的程度提高了。

Quadratic Placement Algorithm Using Gaussian Blurring and Fixed Point Technique

Student: Chuan-Chia Huang Advisor: Dr. Hung-Ming Chen

Department of Electronics Engineering

Institute of Electronics

National Chiao Tung University

ABSTRACT

State of the art global placers adopt different cell spreading algorithms to handle density of placement. In this work, we found there exist different placement behavior when placing cells *across* and *around* fixed macros among state of the art global placers. Our findings also suggest that placement behavior around fixed macros has significant impact toward routability of the design that cannot be observed through conventional evaluation metrics. For quadratic placers, the methodologies adopted in SimPL [1] and Kraftwerk2 [2] represent two distinct class of cell spreading algorithms. Based on our implementations of the two frameworks, we answer to the question on whether if there exist different level of controllability in terms of global view and local view among different cell spreading algorithms. We further investigate the impact of placement behavior on routability of the design. To address both global view and local view of the placement, we propose a two stage global placement framework. The first stage aims to assign large portion of cells with precise amount of white space by moving cells across fixed macros. The second stage aims to determine accurate relative order of cells and move cells around fixed macros. Our proposed placement framework achieves equivalent placement quality compared to ComPLx [3] with placement behavior around fixed macros that is inherently desirable to routability.

Acknowledgements

首先，感謝我的指導老師 陳宏明教授，從大學時期開始便指導著我專題的研究，引出我在電子設計自動化領域上的興趣。而在進入研究所後，讓我有機會進入這個充滿歡樂與知性學習氛圍的實驗室，VDA Lab。在更深入探討並開始實作研究的過程中，碰到困難時也總是適時地給予協助。另外在生活上的關心與照顧也是令我相當感動。

接下來我要感謝的是，兩年以來不吝指導我的博士班學長姊們，Benbean、Jerry、Sean、俊凱、雄哥和敬雨。在各位學長姊身上學到了許多東西，無論是學術上或非學術的所有技能，讓我在研究所短短兩年內的學習十分地有趣且精實！其中我要特別感謝一路帶我完成碩士論文的學長，Sean。每每在我窮途末路的時候，你總是能點出如何改善現況並且立即給了我新的方向！若沒有學長的協助，肯定是無法如此順利的完成碩士論文！在此特別感謝他！而Benbean，在實驗室主機管理、網頁編寫還有一些生活上的問題都給予許多很關鍵的協助，也透過這些協助讓我添加了許多學習機會。另外國哥、姚明、冠廷和建志學長們，在我進入實驗室第一年陪伴我修課、運動和日常生活的休閒，讓我在研究之餘也過得十分地充實！然後是天龍、孟伶、以恩和新鈞各位同屆的同學，許多實驗室的大小事都是我們一同分工完成的，在研究所最後的求學階段能和你們一起完成學業，眞的很幸運！最後是學弟妹，家麒、韋辰和柔君，感謝你們在實驗室的陪伴和協助，是你們讓實驗室更有朝氣和活力。

感謝大學時期的同學和室友，到了研究所還是互相地關心和交流。總是在我最需要協助時，跳出來替我分擔。而且也時常一起吃飯聊天，讓我更確定未來的方向。

接下來要感謝我的父母親的栽培和照顧，感謝你們給我這麼好的機會到這個頂尖的學院進修。從來都是給予支持和鼓勵，在各個方面的引導讓我有能力成爲一個獨立自主的人。感謝你們的支持，讓我在這個領域完成了碩士學位！

最後要感謝我的女朋友，在我最忙碌的這段時間裡，給我最大的鼓勵和協助！讓我能安心地順利完成學業，若只有我自己一個人是無法如此順利的。

謝謝大家的幫忙和協助，這份榮耀是屬於你們每一個人的！

Contents

Abstract (Chinese) i

Abstract ii

Acknowledgements iii

List of Tables v

List of Figures vi

1 Introduction 1
- 1.1 Prior arts based on quadratic wirelength model 2
- 1.2 Overview of the Placement Framework 4

2 Force Directed System for Quadratic Placers 7

3 White Space Allocation At Global Scale 10

4 Cell Spreading Algorithm 12
- 4.1 Bin Size Adjustment 14
- 4.2 Weight Adjustment 15
- 4.3 Determining Initial Step Size 16

5 Experimental Results 18
- 5.1 Evaluation on ISPD2005 Benchmarks 19
- 5.2 Evaluation on ICCAD2012 Benchmarks 19
- 5.3 Analysis on Distance to Optimal Region 19
- 5.4 Evaluation on Routability 20
- 5.5 Discussion and Future Improvements 20

6 Conclusion 27

List of Tables

5.1 Comparison of HPWL with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ISPD2005 benchmarks for SimPL [1], Kraftwerk2 [2], ComPLx [3], MAPLE [4], BonnPlace2 [5], NTUPlace3 [6] and mPL6 [7]. (G.M. stands for geometric mean) 22
5.2 Comparison of runtime in minutes with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ISPD2005 benchmarks for mPL6 [7] and NTUPlace [6]. Runtime is normalized to the known optimal placement solution. (G.M. stands for geometric mean) 23
5.3 Comparison of HPWL and runtime in minutes with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ICCAD2012 benchmarks for RIPPLE 2.0 [8]. (G.M. stands for geometric mean) 23
5.4 Comparison of result on distance from each cell to its optimal region with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ISPD2005 benchmarks. Unit is defined by one half of the row height. (In O.R. stands for in optimal region) 24
5.5 Comparison of result on distance from each cell to its optimal region with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ICCAD2012 benchmarks. Unit is defined by one half of the row height. (In O.R. stands for in optimal region) 25
5.6 Comparison of total overflow, max overflow and scaled wirelength with our proposed two stage global placement framework, our implementation based partition based cell spreading algorithm (P-Based) [1] and our implementation based on gradient based cell spreading algorithm [2] on ICCAD 2012 benchmarks [9]. (G.M. stands for geometric mean) 26

List of Figures

1.1 Illustration on placement solution of partition based cell spreading algorithm and gradient based cell spreading algorithm. (a) Placement solution based on partition based framework after global placement. (b) Placement solution based on partition based framework after detail placement. (c) Placement solution based on gradient based framework after global placement. (d) Placement solution based on gradient based framework after detail placement. 3

1.2 Illustration on behavior of partition based cell spreading algorithm and gradient based cell spreading algorithm when dealing with fixed macro block. (a) and (b) illustrate the ability to move cells around macro. (c) and (d) illustrate the ability to look beyond macro. 4

1.3 The flow chart of the proposed placement algorithm. 6

4.1 Illustration of the bin size adjustment scheme between rough legalization and gradient based cell spreading. (a) Cell distribution in early stage. (f) Cell distribution in latter stage. (b) Lower cluster number. (c) Larger expansion region for each cluster. (g) Higher cluster number. (h) Smaller expansion region for each cluster. (d) Lager dimension of Gaussian matrix (e) Cell spreading in larger step size. (i) Smaller dimension of Gaussian matrix (j) Cell spreading in smaller step size. 13

Chapter 1

Introduction

As design becomes increasingly complex, it becomes more apparent that using HPWL as the only evaluation metric is inadequate to meet the practical demands. To bridge the gap between academic effort and design experience, several works suggest that routability [8, 10, 11] is a more direct evaluation metric to reflect placement quality. The work done in [3] have suggested that global placers exhibit different controllability at global view of placement. However, the vague concept on global view or local view of placement is rarely discussed in prior arts.

In quadratic placement, using partition based cell spreading algorithm [1] combined with fixed-point technique [12] has become an increasingly popular approach [1, 3, 4] due to its precise density control and high efficiency to produce high quality placement in terms of half-perimeter wirelength (HPWL). On the contrary, although gradient based cell spreading algorithm in quadratic placement [2] may not deliver as precise density control compared to partition based approach, its unique placement behavior around fixed macro blocks is desirable when routability is taken into consideration. In this work, we refer the cell spreading technique adopted in SimPL [1] as partition based cell algorithm and cell spreading technique adopted in Kraftwerk2 [2] as gradient based cell algorithm.

Fig. 1.1 illustrates placement solutions using two distinct cell spreading algorithms. The color of the cells represents each individual design hierarchy. We have implemented two placers using partition based cell spreading algorithm and gradient based cell spreading algorithm, one is based on SimPL framework and the other is based on Kraftwerk2 framework. Fig. 1.1(a) and Fig. 1.1(b) are the placement solutions based on SimPL framework after global placement and detailed placement. Fig. 1.1(c) and Fig. 1.1(d) are the placement solutions based on Kraftwerk2 framework after global placement and detailed placement.

In Fig. 1.1, the distinct placement behavior between the two placement frameworks can be observed. The placement solution in Fig. 1.1(b) has 4% less HPWL compared to Fig. 1.1(d), but Fig. 1.1(d) has a much more sparse cell distribution. In addition, an apparent contour around macro blocks can be observed in gradient based cell spreading algorithm. From Fig. 1.1, we observe a distinct placement behavior across and around fixed macro blocks between partition based and gradient based cell spreading algorithms.

Fig. 1.2 illustrates two scenarios which demonstrates strength and weakness of partition based and gradient based cell spreading algorithm. Fig. 1.2(a) and Fig. 1.2(b) illustrate how both cell spreading algorithms move cells around fixed macros. In Fig. 1.2(a), since partition based cell spreading algorithm only calculates precise amount of white space without the knowledge on the location of fixed macros, cells are likely to place around a macro block. In Fig. 1.2(b), gradient based cell spreading algorithm knows the location of fixed macros based on density function, cells are repelled away from macro blocks. Wirelength values are the same in the examples illustrated in Fig. 1.2(a) and Fig. 1.2(b) but with different placement behavior.

Fig. 1.2(c) and Fig. 1.2(d) illustrate how both cell spreading algorithms move cell across fixed macros. For gradient based cell spreading algorithm in Fig. 1.2(d), local gradient information is likely to trap cells within a local valley between two large fixed macros. In Fig. 1.2(c), partition based cell spreading algorithm does not rely on density function to allocate cells, it progressively expands its search region until required amount of white space is found.

1.1 Prior arts based on quadratic wirelength model

For quadratic placers, two of the most promising approach to reduce the cell overlaps are gradient based approach [2] and partition based approach [1, 3–5, 13]. In Kraftwerk2 [2], each unit area is modeled as a unit charge. Full chip electrical potential can be obtained by solving Poisson equation and electric field can be derived by taking the derivative of electrical potential. Move force of each movable cell is calculated based on electric field that drives cell to move towards area with less electrical potential.

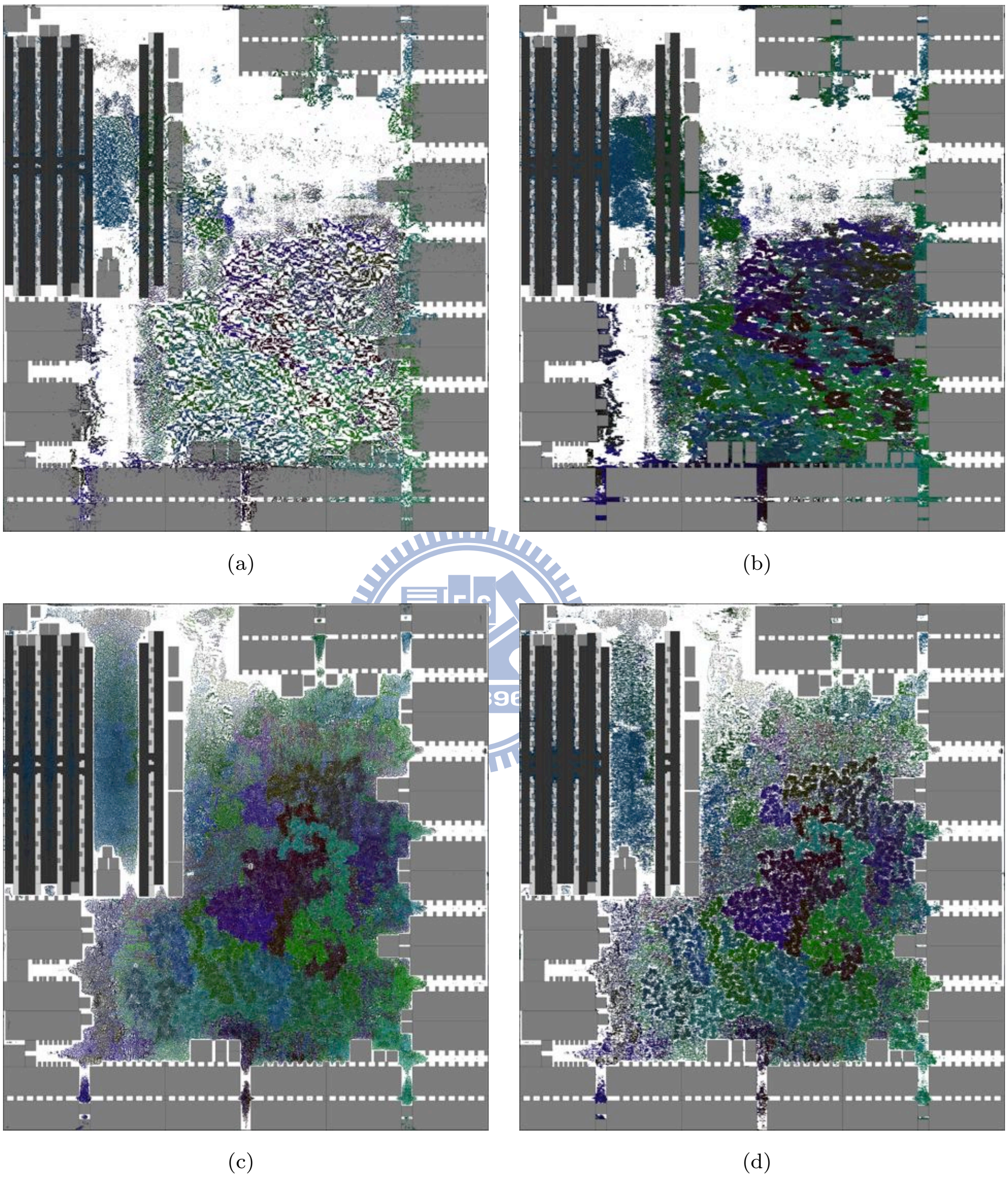

FIGURE 1.1: Illustration on placement solution of partition based cell spreading algorithm and gradient based cell spreading algorithm. (a) Placement solution based on partition based framework after global placement. (b) Placement solution based on partition based framework after detail placement. (c) Placement solution based on gradient based framework after global placement. (d) Placement solution based on gradient based framework after detail placement.

In SimPL [1], placement solution is obtained using partition based cell spreading algorithm and fixed point technique within a upper bound and lower bound framework. The location of fixed points is obtained through *rough legalization* which is a partition algorithm that recursively divides cells and then allocate precise portion of white space to each partitioned cells. Objective function is solved again with additional pseudo nets

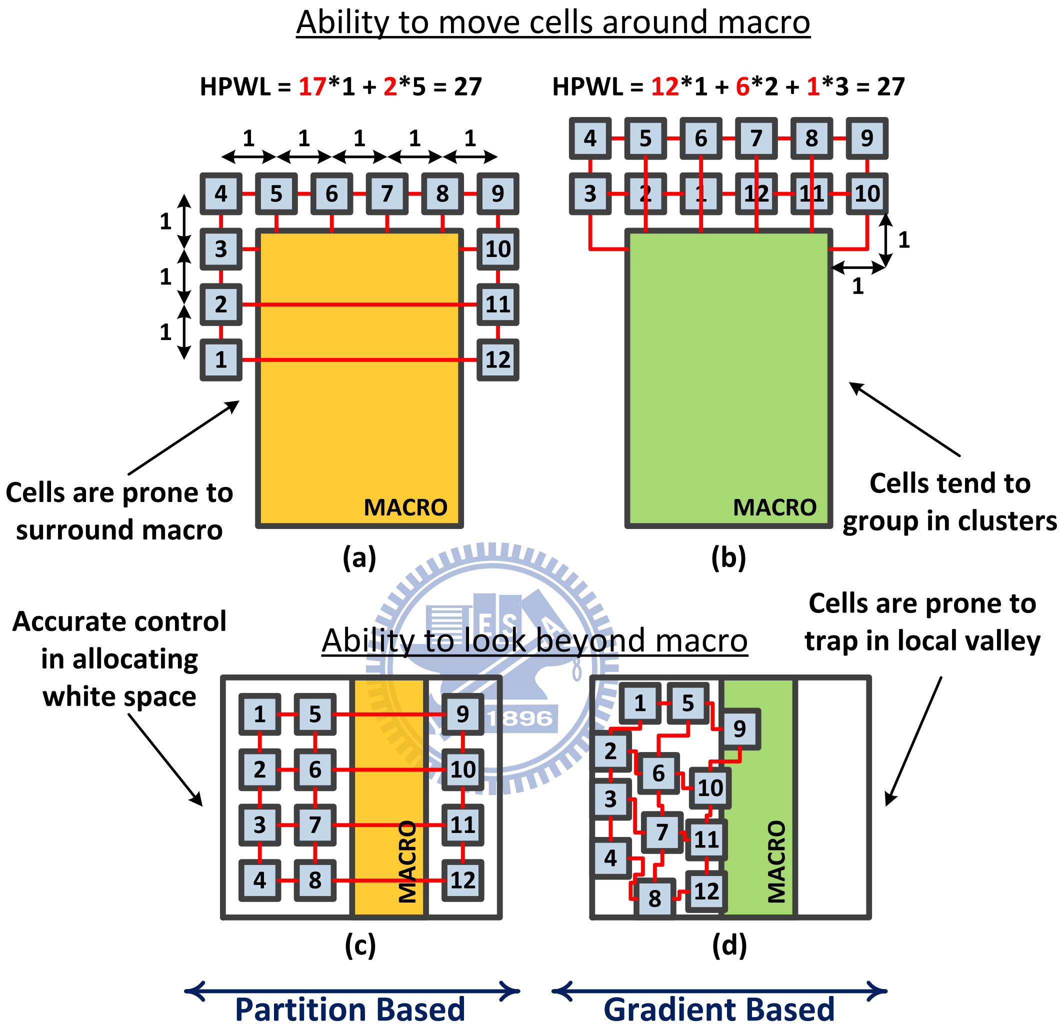

FIGURE 1.2: Illustration on behavior of partition based cell spreading algorithm and gradient based cell spreading algorithm when dealing with fixed macro block. (a) and (b) illustrate the ability to move cells around macro. (c) and (d) illustrate the ability to look beyond macro.

using iterative method. The net weight of pseudo anchors accumulates on each iteration to avoid placement collapsing to previous state.

1.2 Overview of the Placement Framework

In this work, we implement two quadratic placers using partition based and gradient based cell spreading algorithms. Both are implemented to a comparable quality compared to original works [1, 2]. By analyzing our implementation based on the two cell spreading algorithms, we found cell spreading algorithms have their strength and weakness in terms of moving cells *across* and *around* fixed macros. The ability on dealing with macro blocks significantly affects placer's global view and local view of the placement. Based on the analysis, we generalize the key elements of the two frameworks and propose a new two stage global placement framework by combining the strengths of the two cell spreading algorithms. To obtain smooth transition between the two stages, our second stage global placer is capable of handling incremental placement.

Fig. 1.3 is the flow chart of the proposed placement framework. The framework begins by obtaining an initial placement that focuses on better relative order of cells and fewer modules overlaps. The first stage of global placement applies partition based cell spreading algorithm, which focuses on white space allocation by moving cells across macro blocks. The second stage of global placement applies gradient based cell spreading algorithm, which focuses on relative order of cells and move cells around fixed macro blocks. In brief, our contributions can be summarized as follows.

- A two stage global placement framework is proposed to address both global view and local view of the placement. The global view of the placement is addressed using partition based cell spreading algorithm that allocates large portion of cells. The local view of the placement is addressed using gradient based cell spreading algorithm that focus on determining accurate relative order of cells.
- A surface model using Gaussian Blurring is proposed for gradient based cell spreading algorithm. The dimension of Gaussian Blurring can be easily adjusted to allow global placer to have global view and local view during placement iteration. To achieve faster run time for large Gaussian Matrix at finest grid, Gaussian Blurring is calculated in frequency domain through Fast Fourier Transform (FFT).
- A dynamic step size control methodology and weight adjustment scheme are proposed to handle incremental placement.

In the remainder of this thesis, Section II introduces the force directed system for quadratic placers. Section III describes the white space allocation at global scale using partition based cell spreading algorithm. Section IV compares partition based cell spreading algorithm, gradient cell spreading algorithm and our proposed two stage global placement framework. Section V presents the experimental result. Finally, Section VI concludes this work.

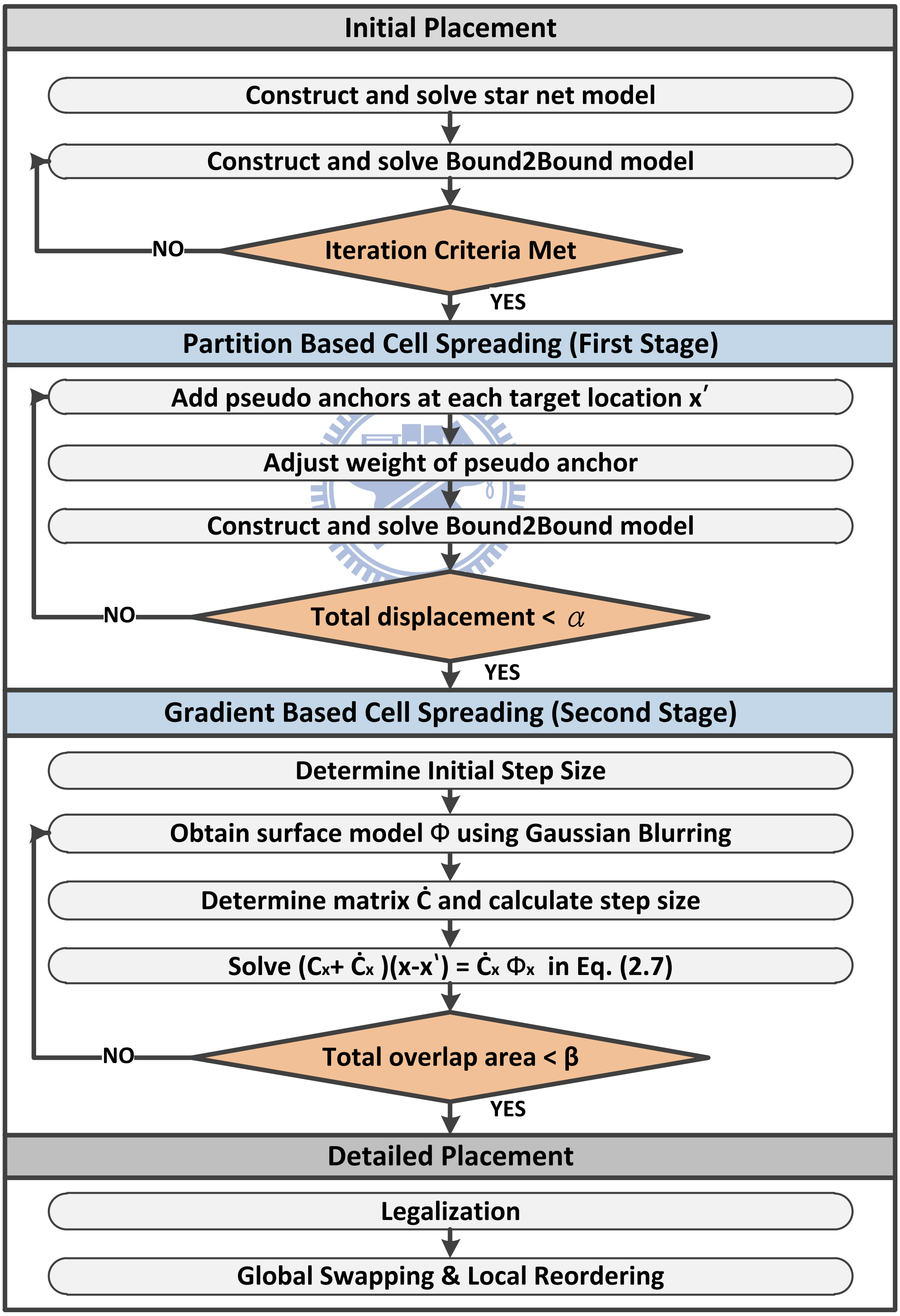

FIGURE 1.3: The flow chart of the proposed placement algorithm.

Chapter 2

Force Directed System for Quadratic Placers

The main objective for wirelength driven placer is to minimize the total half perimeter wirelength (HPWL). Since HPWL is non-differentiable, quadratic wirelength model in Eq. (2.1) is proposed to optimize the HPWL.

$$\begin{aligned}\Gamma &= \Gamma_x + \Gamma_y \\ &= \frac{1}{2}w_{i,j}(x_i - x_j)^2 + \frac{1}{2}w_{i,j}(y_i - y_j)^2 \\ &= \frac{1}{2}x^T C_x + d_x^T + \frac{1}{2}y^T C_y + d_y^T \end{aligned} \tag{2.1}$$

Eq. (2.1) optimizes the quadratic wirelength rather than the linear wirelength, Kraftwerk2 proposed the bound to bound wirelength model (B2B) to linearize the quadratic wirelength objective. The B2B model neglects interconnect of inner pins and set the net weight $w_{i,j}$ to $\frac{2}{(P-1)\ell}$ in which P stands for number of pins in a given net and ℓ is the distance between pin i and pin j. The term $2/(P-1)$ adjusts number of connections in a give net and ℓ linearize the quadratic objective.

$$F_x^{net} = C_x\boldsymbol{x} + d_x \tag{2.2}$$

Minimal wirelength can be obtained by minimizing Eq. (2.2) which can be solved efficiently using iterative method. Eq. (2.2) is generally referred as the net force. To remove the overlaps among cells, a surface model Φ can be obtained given with a density function. Target location $\boldsymbol{x'}$ can be calculated by taking derivative of Φ. Eq. (2.3) defines how to obtain the target location x' given with a surface model Φ.

$$\boldsymbol{x_i'} = \boldsymbol{x_i} - \frac{\partial}{\partial \boldsymbol{x}} \Phi(x, y)\bigg|_{x_i, y_i} = \boldsymbol{x_i} - \Phi_x \tag{2.3}$$

$$\text{Kraftwerk2:} \qquad F_x^{move} = \mathring{C}_x(\boldsymbol{x} - \boldsymbol{x'}) = \mathring{C}_x \cdot \Phi_x \tag{2.4}$$

$$\text{SimPL/ComPLx:} \qquad F_x^{move} = \mathring{C}_x(\boldsymbol{x} - \boldsymbol{x'}) \tag{2.5}$$

$$F_x^{hold} = -(C_x\boldsymbol{x'} + d_x) \tag{2.6}$$

The magnitude of step size affects the quality of the placement and execution time. Thus, implementation of a competitive placer requires precise control of step size on each iteration. Eq. (2.4) or F^{move} controls the cell spreading force by adjusting the matrix $\mathring{C}_x$ which defines the weight of step size.

To maintain stability on each iteration, F^{hold} in Eq. 2.6 is introduced to neutralize F^{net} to prevent placement collapsing to previous state. Thus, F^{move} is the only force that spreads out cells in each iteration. Eq. (2.7) defines the force directed system of Kraftwerk2 in which all three forces F^{net}, F^{hold} and F^{move} are taken into account.

$$\begin{aligned} \text{Kraftwerk2:} \qquad & F_x^{net} + F_x^{hold} + F_x^{move} = 0 \\ \Rightarrow \quad & (C_x + \mathring{C}_x)(\boldsymbol{x} - \boldsymbol{x'}) = \mathring{C}_x\Phi_x \end{aligned} \tag{2.7}$$

$$
\begin{aligned}
\text{SimPL/ComPLx:} \qquad & F_x^{net} + F_x^{move} = 0 \\
\Rightarrow \quad & C_x\boldsymbol{x} + d_x + \mathring{C}_x(\boldsymbol{x} - \boldsymbol{x'}) = 0 \\
\Rightarrow \quad & (C_x + \mathring{C}_x)\boldsymbol{x} = -(d_x + \mathring{C}_x\boldsymbol{x'})
\end{aligned} \tag{2.8}
$$

In SimPL, rather than relying on the gradient based method to obtain target location $\boldsymbol{x'}$, SimPL obtains $\boldsymbol{x'}$ using partition based method. In addition, SimPL removes hold force in Eq. (2.6) and entire system relies on the balance between F^{net} and F^{move}. Eq. (2.8) defines the force directed system used in SimPL and ComPLx.

The matrix $\mathring{C}_x$ in SimPL defines the weight of target location $\boldsymbol{x'}$ and is set uniformly for every cell. Since hold force is removed from the equation, the magnitude $\mathring{C}_x$ needs to be no less than the previous iteration in order to prevent placement collapsing to previous state. Note that the SimPL framework does NOT guarantee each cell has a corresponding target location $\boldsymbol{x'}$, if a cell is not included during rough legalization, it will not have a target location. In ComPLx [3], an adjust scheme based on the concept total displacement is proposed to derive the matrix $\mathring{C}_x$.

Regarding to Eq. (2.7) and Eq. (2.8), Kraftwerk2 and SimPL demonstrate the balance of the force directed system can be achieved using three and two forces. The stability of Kraftwerk2 relies on the balance between hold force, net force and move force. In SimPL, hold force is incorporated within the move force and stability of the system is achieved by overpowering the net weight of pseudo anchors on each iteration. The additional force in Kraftwerk2 offers better controllability of the placement structure. When move force is removed from Kraftwerk2, placement maintains its original position. However, when move force is removed in SimPL, placement collapse back to minimal wirelength solution.

Chapter 3

White Space Allocation At Global Scale

One of the key concept for cell spreading algorithm in SimPL is that relative order of cells remains unchanged. This indicates that the relative order of cells obtained during initial placement affects every subsequent placement iterations. The work done in [5, 13] also suggests that choose an intermediate wirelength model between quadratic and linear produces better wirelength result than focusing on the linear HPWL as placement objective. This is because although linear wirelength model produces better wirelength, it also has higher cell overlaps. Thus, unlike SimPL that obtains a lower bound initial placement, our initial placement is obtained by applying star net model which has less module overlaps and better relative order of cells compared to B2B wirelength model.

The strength of partition based cell spreading algorithm lies in its precise allocation of white space. The ability to search for white space across large macros is relatively inefficient for placers using local gradient information. Gradient based placers rely on the balance between density function and wirelength function to control the quality of placement. Thus, when encountering large macro blocks, the weight on density function must outweighs the weight of wirelength function for placer to overlook the presence of macro blocks.

In this work, partition based cell spreading algorithm based on SimPL [1] is adopted to allocate white space at global scale. Compared to SimPL, our partition based cell spreading algorithm does not require to generate stripes to align cells. There is no evidence that generating stripes to align cells increases precision of cell spreading. The work done in [3] also have shown that precision during rough legalization does not undermine solution quality.

During early stage of global placement, the primary focus is on white space allocation. Thus, our placer begins by searching for regions with precise amount of white space to allocate congested cells. These regions are recursively partitioned based on cell area and white space of partitioned region. Partitioning of regions stops when the white space of partitioned region is less than 4 average node area.

Different from SimPL in which cells are constantly aligned to stripes at each iteration of recursive partition, cells are assigned to each partitioned region after recursive partition is complete. Our approach removes the redundancy of cell alignment in SimPL since only final position of cells matters.

Chapter 4

Cell Spreading Algorithm

After white space allocation at global scale is complete, the cell spreading algorithm switch from partition based to gradient based. At this stage, the target location x' is obtained using gradient based method instead of using partition based method. Given with a density function of cell area $h(x, y)$, each unit cell area is regarded as an impulse. Kraftwerk2 regard each impulse as unit charge and surface model $\Phi(x, y)$ is obtained by solving the Poisson equation. However, since the primary objective is not on the accuracy of electrical potential, any distribution function is suffice to meet the supply-demand constraint.

$$g(x,y) = \exp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(x-x_o)^2}{2\sigma_x^2}\right)\right) \tag{4.1}$$

$$\begin{aligned}
-\frac{\partial}{\partial \boldsymbol{x}} g(x,y) &= \frac{x-x_o}{\sigma_x^2} \exp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(y-y_o)^2}{2\sigma_y^2}\right)\right) \\
-\frac{\partial}{\partial \boldsymbol{y}} g(x,y) &= \frac{y-y_o}{\sigma_y^2} \exp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(y-y_o)^2}{2\sigma_y^2}\right)\right)
\end{aligned} \tag{4.2}$$

In this work, the surface model is obtained using Gaussian Blurring. Each unit cell area has an amplitude of 1 unit Gaussian distribution. In terms of image processing, Gaussian blurring is equivalent to the convolution of a Gaussian function to a density function $h(x, y)$. Eq. (4.1) is the Gaussian function for two dimensional space. The term

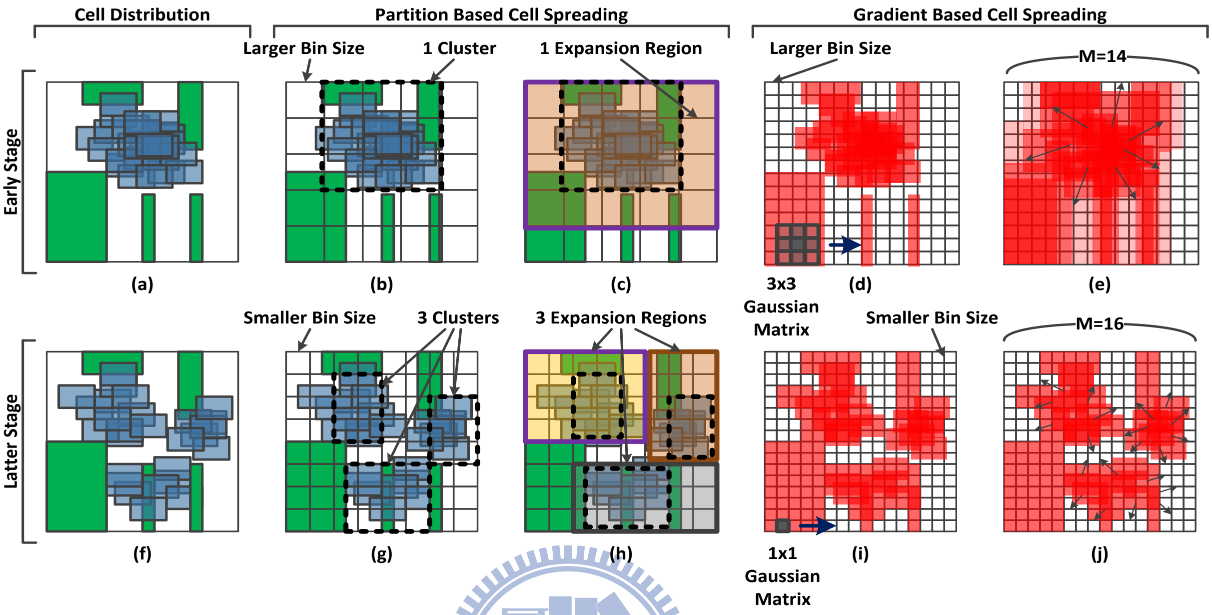

FIGURE 4.1: Illustration of the bin size adjustment scheme between rough legalization and gradient based cell spreading. (a) Cell distribution in early stage. (f) Cell distribution in latter stage. (b) Lower cluster number. (c) Larger expansion region for each cluster. (g) Higher cluster number. (h) Smaller expansion region for each cluster. (d) Lager dimension of Gaussian matrix (e) Cell spreading in larger step size. (i) Smaller dimension of Gaussian matrix (j) Cell spreading in smaller step size.

σ defines the affected range for each unit cell area. Larger value σ translates to larger dimension of Gaussian matrix. Eq. (4.2) is the derivative or the gradient function of the surface model. The discrete surface model $\Phi[m,n]$ can be obtained by solving Eq. (4.3). In Eq. (4.3), $g[m,n]$ and $h[m,n]$ are the discrete representation to $g(x,y)$ and $h(x,y)$ respectively. According to convolution property: $\frac{\partial}{\partial \boldsymbol{x}}(f * g) = f * \frac{\partial}{\partial \boldsymbol{x}} g$ the gradient of the surface model Φ_x and Φ_y can be obtained by solving Eq. (4.4).

$$\Phi[m,n] = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} h[i,j]g[x-i,y-j] \tag{4.3}$$

$$\begin{aligned}\Phi_x[m,n] &= \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} h[i,j]\frac{\partial}{\partial \boldsymbol{x}}g[x-i,y-j]\\ \Phi_y[m,n] &= \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} h[i,j]\frac{\partial}{\partial \boldsymbol{y}}g[x-i,y-j]\end{aligned} \tag{4.4}$$

4.1 Bin Size Adjustment

A given design is generally partitioned in to set of bins. The size of bin significantly affects the quality of placement. Fig. 4.1 illustrates the bin size adjustment scheme between rough legalization and gradient based cell spreading adopted in this work. Fig. 4.1(a) represents a cell distribution that is highly congested which typically is at the early stage of placement iterations. Fig. 4.1(f) represents a cell distribution that is more spread out which typically is at the latter stage of placement iterations.

During rough legalization in the SimPL framework, bin size is decreased by a factor of 0.97 on each iteration of rough legalization. Decrease in bin size has several implicit benefits. Larger bin size illustrated in Fig. 4.1(b) and Fig. 4.1(c) has lower cluster number and larger expansion region for each cluster. Vice versa for smaller bin size illustrated in Fig. 4.1(g) and Fig. 4.1(h) which produces higher cluster number and smaller expansion region for each cluster. Larger bin size at the beginning allows placer to have a global view of the placement, which can effectively allocate cells to regions with precise amount of white space. As bin size decreases, more local congestion spots can be identified and less perturbation of cells can be expected since size of expansion region decreases.

In this work, since dimension of Gaussian Blurring can be arbitrary adjusted, such characteristic can be achieved by using larger value of σ and larger bin size to capture global view of the placement at the beginning. In Fig. 4.1(d), larger bin size is used and larger dimension of Gaussian matrix is used to blur local white space region. Local congestion at the latter stage can be revealed by decreasing the value of σ and bin size as illustrated in Fig. 4.1(i). Note that in both Fig. 4.1(d) and Fig. 4.1(i), the summation on dimension of density matrix and Gaussian matrix is a constant in which $M + N = 17$.

The execution time of two dimensional convolution in spatial domain increases quadratically with the dimension of Gaussian matrix and density matrix. Suppose a $M \times M$ Gaussian matrix is convolved with a $N \times N$ density matrix, the complexity in spatial domain is $O(M^2N^2)$. The complexity can be reduced to $O(N^2 logN)$ by convolving two functions in frequency domain.

Since larger value of σ implies larger dimension of Gaussian matrix and larger bin size implies smaller dimension of density matrix, the summation on dimension of Gaussian matrix and dimension of density matrix can be set to equal to a constant. To fully exploit the efficiency of convolution at frequency domain, we use Eq. (4.5) to determine the dimension of a $M \times M$ Gaussian matrix and a $N \times N$ density matrix. Thus, if we set the constant to 1025 and initialize the dimension of Gaussian matrix to a quarter of density matrix, then M equals to 205 and N equals to 820.

$$M + N = 2^n + 1 \tag{4.5}$$

4.2 Weight Adjustment

In quadratic placement, the step size from x to x' controls the trade-off curve between quality of placement and convergence rate. Smaller step size leads to better quality placement at the expense of longer execution time. The magnitude of step size is controlled by the matrix $\mathring{C}_x = \{\mathring{w}_0, \mathring{w}_1, ..., \mathring{w}_M\}$, which is the matrix that defines the weight of move force. Eq. (4.6) is the weight adjustment scheme in Kraftwerk2. In Kraftwerk2, a target step size μ_T is determined initially, and then weight of move force w_i is adjusted such that average step size μ can be approximated as close to μ_T as possible.

Eq. (4.7) is the weight adjustment scheme in SimPL and Eq. (4.8) is the weight adjustment scheme in ComPLx. In Eq. (4.8), Π is total displacement from cell location to its target location and h is a predefined constant. Since both Eq. (4.7) and Eq. (4.8) are based on accumulation of move force, weight of move force can only be increased throughout the iterations. In addition, step size in cell spreading process can only be indirectly affected and not be accurately controlled.

$$\text{Kraftwerk2:} \qquad \mathring{w}_i^{k+1} = \mathring{w}_i^k \cdot (1 + \tanh(\ln(\mu_T/\mu))) \tag{4.6}$$

$$\text{SimPL:} \qquad \mathring{w}^{k+1} = \frac{0.01 \cdot (1 + \text{iterationNumber})}{|x - x'|} \tag{4.7}$$

$$\text{ComPLx:} \qquad \mathring{w}^{k+1} = min\{2\mathring{w}^k, \mathring{w}^k + (\Pi^{k+1}/\Pi^k)h\} \tag{4.8}$$

Observing the three different weight adjustment schemes defined in Eq. (4.6) to Eq. (4.8), Kraftwerk2 has different weight $\mathring{w}_i$ for each individual cell with precise control over the average step size. On the contrary, SimPL and ComPLx has same weight $\mathring{w}$ for every cell and can only indirectly determine step size. This indicate gradient based cell spreading algorithm is more delicate when dealing with relative order of cells at local scale. To accelerate convergence rate and allowing cells to step across macro blocks at early stage of placement, we adopt a target step size adjustment scheme described in Eq. (4.9).

In Eq. (4.9), μ_T^{Hi} is the upper bound of step size and μ_T^{Lo} is the lower bound of step size. Larger step size in beginning motivates cells to move across macro blocks while smaller step size motivates cells to have more accurate relative order. This is because larger step size implies that $\mathring{C}$ has larger influence in term $(C + \mathring{C})(\boldsymbol{x} - \boldsymbol{x'})$ which means density function has more influence to the objective function. Vice versa for smaller step size that translates to smaller $\mathring{C}$.

$$\mu_T^{k+1} = \mu_T^k - \frac{\mu_T^{Hi} - \mu_T^{Lo}}{x}, k = 2, 3..., x - 1 \tag{4.9}$$

$$\mathring{w}_i^{k+1} = \mathring{w}_i^k \cdot (1 + \tanh(\ln((x\mu_T^{k-1} - \mu_T^{Hi} + \mu_T^{Lo})/x\mu))) \tag{4.10}$$

4.3 Determining Initial Step Size

One of the critical factor to handle incremental placement is to determine the initial step size based on the current state of placement (e.g. cell density, total overlapping area, current value of Γ). Eq. (4.11) defines the initial step size in Kraftwerk2 and Eq. (4.12) defines the initial step size in ComPLx. Eq. (4.11) is inadequate to handle incremental placement since its a constant value in which it neglects the current state of placement. Given with an almost legal placement solution where cell density is roughly flat, using Eq. (4.11) produces very large step size where forces at local gradient are exaggerated.

On the contrary, although Eq. (4.12) in ComPLx considers the current value of objective function and total displacement, the absence of hold force in the ComPLx framework creates certain degree of instability. Since move force are accumulated in ComPLx framework, its very difficult to determine an appropriate step size such the placement will not collapse to its previous state without the information from previous iteration.

$$\text{Kraftwerk2:} \qquad \mathring{w}^1 = \frac{A_{mod,i}}{A_{avg}} \cdot \frac{1}{M} \tag{4.11}$$

$$\text{ComPLx:} \qquad \mathring{w}^1 = \frac{\Gamma}{100\Pi} \tag{4.12}$$

In this work, to deal with placement at different level of cell density and different amount of overlapping area, the initial weight $\mathring{w}^1$ is set inversely proportional to the average bounded length of each net. Eq. (4.13) is the initial step size used in this work for gradient based cell spreading. In Eq. (4.13), N denotes total number of nets and Γ is the value of the objective function in Eq. (2.1).

$$\mathring{w}^1 = \frac{A_{mod,i}}{A_{avg}} \cdot \frac{N}{\Gamma} \tag{4.13}$$

Chapter 5

Experimental Results

In this section, experimental results of our implementations are presented. All of our implementations are self contained and implemented using standard C++ language and performed on a Intel Xeon E5620 machine running at 2.4Ghz. To evaluate the HPWL quality of our placer, we compared our implementations with state of the art global placers including Kraftwerk2 [2], SimPL [1], ComPLx [3], MAPLE [4], BonnPlace [5], NTUplace3 [6], mPL6 [7] on ISPD2005 benchmarks [14]. We also compared our implementation with RIPPLE 2.0 [8] on ICCAD 2012 Placement Benchmarks [9]. Latest binaries from NTUplace3 (v12.06.05) and mPL6 are obtained from original authors and evaluate on same machine. We do not have the access to Kraftwerk2 [2], SimPL [1], ComPLx [3], MAPLE [4], RIPPLE2.0 [8] and BonnPlace [5], so execution time from these global placers are omitted. Result of our proposed framework is presented using default setting without specific tuning for each individual testcase.

For ISPD 2005 Placement Benchmarks, FastDP [15] is used as the detailed placer. FastDP is not compatible with ICCAD 2012 Placement Benchmarks. Since we do not have access to the source code of FastDP, we implement legalization algorithm based on Abacus [16], detailed placer based on FastDP [15] and used them on ICCAD 2012 Placement Benchmarks. NCTUgr 2.0 [17] is used as the global router to evaluate routability of the placement.

The discussion in this section is divided to three parts. The first part presents quality of global placers using HWPL as evaluating metrics. The second part analyze our implementation on partition based cell spreading algorithm, gradient based cell spreading algorithm and our two stage global placement framework. Analysis is performed using routability metric and evaluating distance from each cell to its optimal region after legalization and detailed placement. The third part of this section concludes our findings and discusses future improvement of proposed framework.

5.1 Evaluation on ISPD2005 Benchmarks

Table 5.1 compares the performance of HPWL on ISPD 2005 Placement benchmarks. Our implementation using partition based cell spreading algorithm (P-Based) achieves quality within 1.56% compared to SimPL. Our implementation using gradient based cell spreading algorithm (G-Based) achieves quality within 1.18% compared to Kraftwerk2. This demonstrate our implementation based on the two frameworks achieves comparable quality compared to the original work.

Our proposed two stage global placement framework achieves equivalent quality compared to ComPlx [3], outperforms Kraftwerk2 [2], mPL6 [7], NTUplace3 [6] and SimPL [1] by 6.15%, 3.50%, 6.97%, 0.40% respectively, and trail behind MAPLE [4] and BonnPlace [13] by 1.39% and 1.74% respectively.

5.2 Evaluation on ICCAD2012 Benchmarks

We compared our implementations with RIPPLE 2.0 [8] on ICCAD 2012 Placement Benchmarks [9] since only RIPPLE 2.0 released their wirelength driven result on this set of benchmarks. Results are presented in Table 5.3. Our two-stage framework outperforms RIPPLE2.0 by 3.14% in HPWL. Our implementation on P-Based cell spreading algorithm leads by a marginal 0.28% and P-Based cell spreading algorithms trails behind by 7.38%.

5.3 Analysis on Distance to Optimal Region

To analyze placement behavior among different global placers, we analyze the distance from each cell to its optimal region. Our assumption is that if a given placement has more percentage of cells within optimal region, it implies that this placement has better local view. Vice versa for a given placement has more percentage of cells that are far apart from its optimal region, which implies worse global view.

Table 5.4 and Table 5.5 presents the result on distance from each cell to its optimal region on ISPD 2005 benchmarks and ICCAD 2012 benchmarks respectively. Unit is defined by one half of the row height. In Table 5.4 and Table 5.5, G-Based cell spreading algorithm has the highest percentage of cells within optimal region on 16 out 16 testcases while P-Based cell spreading algorithm has the least percentage of cells on 14 out 16 testcases. Our two stage placement framework has more percentage of cells within optimal region on 14 out 16 testcases compared to P-Based cell spreading algorithm.

When analyzing percentage of cells that are placed more than 10*0.5 row height away from its optimal region, G-Based cell spreading algorithm has the highest percentage on 11 out 16 benchmarks while our two-stage framework has the least percentage of cells on 15 out of 16 testcases.

5.4 Evaluation on Routability

Evaluation on routability of partition based, gradient based cell spreading algorithm and our two-stage framework is presented in Table 5.6. While gradient based cell spreading algorithm has better local view, it also has the worst global view. Routability of gradient based cell spreading algorithm is significantly compromised due to its higher HPWL. This also shows that our implementation on gradient based cell spreading algorithm exhibit inherent difficulty in controlling density at global scale.

An interesting phenomenon can be observed by comparing partition based cell spreading algorithm and our two stage framework. Both placers achieves nearly equivalent quality in terms of HPWL, but our two-stage framework has 43% less total overflow and 34% reduction in maximum overflow compared to partition based cell spreading algorithm. This shows that two placements with same HPWL can exhibit entirely different routability. Our explanation to this phenomenon lies in controllability of cell spreading algorithms at local view of placement. Improving local view of placement significantly improves routability.

5.5 Discussion and Future Improvements

Experimental results presented in Table 5.4, Table 5.5 and Table 5.6 supports our original hypothesis. Our findings are discussed as follows. (1) There exhibit different controllability among different cell spreading algorithms at global view and local view of placement. (2) Gradient based cell spreading algorithm exhibit better controllability at local view and worse controllability at global view. (3) A two stage global placement framework can have better controllability on both global and local view of placement. (4) Improving global view of placement have more obvious improvement on HPWL while improving local view of placement have more obvious improvement on routability.

TABLE 5.1: Comparison of HPWL with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ISPD2005 benchmarks for SimPL [1], Kraftwerk2 [2], ComPLx [3], MAPLE [4], BonnPlace2 [5], NTUPlace3 [6] and mPL6 [7]. (G.M. stands for geometric mean)

#	Two Stage	SimPL [1]	P-Based	Kraft.2 [2]	G-Based	ComPLx [3]	MAPLE [4]	BonnPlace [5]	NTUP3 [6]	mPL6 [7]
AD1	76.90	77.42	80.27	82.43	82.15	77.75	76.36	76.87	80.29	77.93
AD2	88.74	91.01	89.40	92.85	95.36	88.76	86.95	86.36	90.18	92.04
AD3	209.95	203.84	208.34	227.22	223.50	206.57	209.78	202.00	233.77	214.16
AD4	183.29	184.70	193.92	199.43	195.82	184.07	179.91	181.53	215.02	193.89
BB1	95.97	94.66	98.31	97.67	99.87	95.30	93.74	94.85	98.65	96.80
BB2	146.25	145.87	142.77	154.74	158.64	145.87	144.55	144.21	158.27	152.34
BB3	323.24	351.55	334.02	343.32	364.31	330.74	323.05	317.71	346.33	344.10
BB4	798.82	790.28	818.12	852.40	846.69	789.45	775.71	781.79	829.09	829.44
G.M.	**100.00%**	**100.40%**	**101.96%**	**106.15%**	**107.33%**	**100.01%**	**98.61%**	**98.26%**	**106.97%**	**103.50%**

Table 5.2: Comparison of runtime in minutes with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ISPD2005 benchmarks for mPL6 [7] and NTUPlace [6]. Runtime is normalized to the known optimal placement solution. (G.M. stands for geometric mean)

#	Two Stage	P-Based	G-Based	NTUPlace3 [6]	mPL6 [7]
AD1	6.51	4.27	8.96	7.82	24.22
AD2	7.81	6.34	11.73	8.87	26.57
AD3	15.20	11.72	19.04	19.93	76.02
AD4	13.31	12.64	17.66	25.40	71.34
BB1	8.51	6.67	10.31	14.45	31.60
BB2	12.39	9.28	18.14	35.13	81.01
BB3	27.85	26.52	50.98	38.88	110.68
BB4	53.53	49.35	76.61	111.38	253.58
G.M.	**122.25%**	**100.00%**	**172.80%**	**198.15%**	**544.48%**

Table 5.3: Comparison of HPWL and runtime in minutes with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ICCAD2012 benchmarks for RIPPLE 2.0 [8]. (G.M. stands for geometric mean)

	Two Stage		**P-Based**		**G-Based**		**RIPPLE 2.0 [8]**
	HPWL	**Time**	**HPWL**	**Time**	**HPWL**	**Time**	**HPWL**
superblue1	259625987	16.17	260008361	11.96	290051349	17.74	272906304
superblue3	307662108	17.29	305324913	12.86	319436744	18.25	307528119
superblue4	211549386	10.99	210924662	7.66	229463294	10.94	218230511
superblue5	342149887	13.78	340303341	9.84	365500638	14.02	335332413
superblue7	402442619	27.04	398444616	20.80	462238462	36.83	395288349
superblue10	533745854	20.47	535118536	15.32	559797179	20.09	565020331
superblue16	251757882	13.32	256752919	9.23	258444606	14.47	249202445
superblue18	146527792	11.49	143596983	8.19	152419258	12.51	171609483
G.M.	**100.28%**	**137.58%**	**100.00%**	**100.00%**	**107.38%**	**148.63%**	**103.43%**

TABLE 5.4: Comparison of result on distance from each cell to its optimal region with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ISPD2005 benchmarks. Unit is defined by one half of the row height. (In O.R. stands for in optimal region)

#	Method	In O.R.	Distance to Optimal Region 1	2	3	4	5	6	7	8	9	10	>10
adaptec1	Two Stage	21.35%	**4.20%**	**16.60%**	**19.57%**	**14.25%**	8.46%	4.83%	2.77%	1.69%	1.09%	0.73%	4.45%
	P-Based	21.22%	4.07%	16.20%	19.05%	14.12%	**8.61%**	5.03%	**2.98%**	**1.83%**	1.18%	0.85%	4.86%
	G-Based	**22.06%**	4.04%	15.71%	18.81%	13.78%	8.53%	**5.07%**	2.90%	1.83%	**1.25%**	**0.88%**	**5.13%**
adaptec2	Two Stage	26.11%	**9.17%**	**19.29%**	**17.36%**	**10.39%**	5.52%	3.03%	1.86%	1.23%	0.81%	0.62%	4.61%
	P-Based	26.23%	9.08%	19.05%	17.05%	10.19%	5.51%	3.11%	1.90%	1.26%	0.93%	0.66%	5.03%
	G-Based	**26.87%**	8.18%	17.69%	16.20%	10.22%	**5.62%**	**3.28%**	**2.09%**	**1.46%**	**1.06%**	**0.79%**	**6.53%**
adaptec3	Two Stage	25.91%	7.16%	16.69%	**17.11%**	**10.64%**	**6.47%**	3.83%	2.37%	1.58%	1.09%	0.80%	6.35%
	P-Based	26.07%	**7.18%**	**16.70%**	16.90%	10.53%	6.43%	**3.91%**	**2.44%**	**1.65%**	**1.15%**	**0.86%**	6.18%
	G-Based	**27.05%**	7.11%	16.54%	16.65%	10.20%	6.32%	3.74%	2.34%	1.55%	1.11%	0.81%	**6.57%**
adaptec4	Two Stage	27.20%	**8.58%**	**18.89%**	**16.49%**	**10.10%**	**6.07%**	**3.47%**	2.10%	1.29%	0.87%	0.63%	4.32%
	P-Based	27.14%	8.51%	18.73%	16.20%	9.79%	5.99%	3.45%	**2.11%**	**1.40%**	**0.94%**	**0.71%**	**5.02%**
	G-Based	**28.31%**	8.48%	18.66%	16.05%	9.63%	5.85%	3.35%	2.03%	1.32%	0.92%	0.65%	4.75%
bigblue1	Two Stage	23.01%	4.98%	**19.29%**	**20.41%**	**13.00%**	**7.21%**	3.77%	2.11%	1.26%	0.86%	0.59%	3.52%
	P-Based	22.98%	4.92%	18.86%	19.88%	12.98%	7.11%	**3.86%**	**2.22%**	**1.44%**	**0.95%**	**0.69%**	4.11%
	G-Based	**24.47%**	**5.04%**	18.67%	19.55%	12.39%	6.88%	3.68%	2.08%	1.34%	0.93%	0.66%	**4.33%**
bigblue2	Two Stage	22.95%	8.35%	**19.10%**	**17.61%**	11.35%	**6.99%**	4.06%	2.35%	1.56%	1.04%	0.74%	3.89%
	P-Based	22.47%	**8.41%**	18.89%	17.12%	**11.42%**	6.96%	**4.08%**	**2.53%**	**1.64%**	**1.17%**	**0.83%**	4.47%
	G-Based	**23.95%**	8.36%	18.62%	16.99%	11.05%	6.79%	3.92%	2.34%	1.59%	1.06%	0.76%	**4.57%**
bigblue3	Two Stage	21.83%	14.76%	**24.70%**	**17.23%**	**8.73%**	4.58%	2.28%	1.26%	0.79%	0.53%	0.37%	2.95%
	P-Based	21.55%	**15.36%**	24.32%	16.42%	8.53%	4.54%	2.39%	**1.41%**	**0.91%**	**0.63%**	0.45%	3.49%
	G-Based	**22.23%**	14.44%	23.97%	16.40%	8.56%	**4.69%**	**2.39%**	1.41%	0.90%	0.63%	**0.45%**	**3.94%**
bigblue4	Two Stage	20.26%	13.23%	**25.35%**	**17.18%**	**9.87%**	**5.30%**	2.64%	1.49%	0.92%	0.61%	0.43%	2.71%
	P-Based	19.96%	**13.78%**	24.81%	16.25%	9.43%	5.14%	**2.79%**	**1.67%**	**1.11%**	**0.79%**	**0.58%**	**3.70%**
	G-Based	**20.79%**	13.20%	24.79%	16.74%	9.58%	5.19%	2.69%	1.58%	1.02%	0.70%	0.51%	3.21%

TABLE 5.5: Comparison of result on distance from each cell to its optimal region with Our proposed framework, Partition based (P-Based) and Gradient based (G-Based) on ICCAD2012 benchmarks. Unit is defined by one half of the row height. (In O.R. stands for in optimal region)

#	Method	In O.R.	Distance to Optimal Region										
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**>10**
superblue1	Two Stage	14.34%	**7.55%**	**17.67%**	**17.52%**	**13.27%**	9.97%	6.01%	3.83%	2.27%	1.49%	0.97%	5.12%
	P-Based	13.36%	6.40%	15.89%	16.53%	12.84%	**10.22%**	**6.54%**	**4.43%**	**2.88%**	**2.00%**	1.37%	7.52%
	G-Based	**15.48%**	6.19%	15.30%	15.11%	11.68%	9.05%	5.75%	4.03%	2.68%	1.95%	**1.40%**	**11.37%**
superblue3	Two Stage	15.74%	**6.63%**	**16.82%**	**18.14%**	**13.62%**	9.11%	5.90%	3.40%	2.10%	1.30%	0.90%	6.34%
	P-Based	14.65%	5.83%	15.72%	17.03%	13.28%	**9.20%**	**6.17%**	**3.89%**	**2.58%**	**1.74%**	**1.24%**	8.67%
	G-Based	**17.18%**	5.85%	15.40%	16.58%	12.75%	8.62%	5.83%	3.61%	2.41%	1.65%	1.18%	**8.93%**
superblue4	Two Stage	23.17%	**6.14%**	**14.16%**	**14.88%**	**10.69%**	8.52%	5.29%	3.58%	2.33%	1.68%	1.17%	8.38%
	P-Based	21.57%	5.43%	13.06%	13.76%	10.37%	**8.58%**	**5.54%**	**4.00%**	**2.69%**	**2.03%**	**1.46%**	**11.52%**
	G-Based	**24.46%**	5.90%	13.54%	13.83%	9.85%	7.96%	5.01%	3.39%	2.28%	1.69%	1.22%	10.87%
superblue5	Two Stage	17.43%	**5.42%**	**12.41%**	**14.73%**	**12.00%**	**9.91%**	6.55%	4.43%	2.81%	1.97%	1.32%	11.02%
	P-Based	16.24%	4.57%	11.00%	13.49%	11.42%	9.77%	**6.83%**	**4.94%**	**3.36%**	**2.48%**	**1.74%**	**14.15%**
	G-Based	**19.48%**	4.86%	11.33%	13.50%	11.15%	9.54%	6.41%	4.60%	3.14%	2.32%	1.67%	12.01%
superblue7	Two Stage	13.90%	**5.55%**	**13.34%**	**16.47%**	**14.07%**	**10.50%**	7.22%	4.60%	2.98%	1.96%	1.36%	8.05%
	P-Based	12.95%	4.74%	11.87%	14.98%	13.24%	10.19%	**7.38%**	**5.06%**	3.54%	2.52%	1.87%	11.65%
	G-Based	**14.54%**	4.27%	10.72%	13.20%	11.74%	9.21%	6.88%	4.94%	**3.62%**	**2.72%**	**2.13%**	**16.03%**
superblue10	Two Stage	16.99%	**7.29%**	**15.92%**	**16.55%**	**12.27%**	**9.73%**	6.00%	3.82%	2.36%	1.57%	1.06%	6.44%
	P-Based	15.59%	5.82%	13.48%	14.72%	11.55%	9.73%	**6.51%**	4.60%	3.10%	2.28%	1.63%	10.99%
	G-Based	**17.28%**	5.41%	12.44%	13.62%	10.76%	9.19%	6.41%	**4.73%**	**3.30%**	**2.50%**	**1.89%**	**12.47%**
superblue16	Two Stage	19.67%	**6.33%**	**14.96%**	**15.93%**	**12.26%**	9.39%	6.18%	4.01%	2.53%	1.66%	1.12%	5.96%
	P-Based	17.90%	5.01%	12.32%	13.85%	11.58%	**9.56%**	**6.84%**	**4.88%**	**3.41%**	**2.50%**	**1.81%**	**10.34%**
	G-Based	**20.92%**	5.83%	13.87%	14.75%	11.64%	8.93%	6.11%	4.16%	2.79%	1.96%	1.40%	7.64%
superblue18	Two Stage	14.19%	**5.21%**	**11.92%**	**13.75%**	13.20%	10.62%	8.12%	5.40%	3.84%	2.57%	2.00%	9.18%
	P-Based	13.55%	4.67%	11.22%	13.40%	**13.37%**	**10.73%**	**8.20%**	**5.57%**	3.98%	2.77%	2.14%	10.38%
	G-Based	**15.24%**	4.10%	9.78%	11.39%	11.31%	9.33%	7.52%	5.33%	**4.11%**	**3.05%**	**2.54%**	**16.30%**

Table 5.6: Comparison of total overflow, max overflow and scaled wirelength with our proposed two stage global placement framework, our implementation based partition based cell spreading algorithm (P-Based) [1] and our implementation based on gradient based cell spreading algorithm [2] on ICCAD 2012 benchmarks [9]. (G.M. stands for geometric mean)

#	Total_Overflow			Max_Overflow			Scaled_Wirelength		
	Two Stage	**P-Based**	**G-Based**	**Two Stage**	**P-Based**	**G-Based**	**Two Stage**	**P-Based**	**G-Based**
superblue1	814160	1008500	1600318	40	68	212	640485109	707363368	940180310
superblue3	1182478	1646296	2026156	70	72	104	635630014	701229372	1025784456
superblue4	321944	452100	1012890	44	80	70	450960373	555543686	735745688
superblue5	1161770	1280734	775058	88	160	68	756611579	839204971	755856202
superblue7	551202	950280	2294126	36	48	72	672576461	771577691	1069387797
superblue10	2873914	3553926	2889218	128	148	192	1836829999	2062457141	1852100146
superblue16	416620	1085342	556812	32	40	26	372002748	467132401	418233295
superblue18	1649578	1972954	1906914	64	56	54	474572539	511244329	480174593
G.M.	**100.00%**	**143.25%**	**161.25%**	**100.00%**	**132.93%**	**145.34%**	**100.00%**	**114.24%**	**127.63%**

Chapter 6

Conclusion

In this work, we implement two state of the art placement frameworks based on SimPL and Kraftwerk2. Our observations indicate there exist a distinct difference in controllability at global view and local view of placement between the two frameworks. Partition based cell spreading algorithm adopted in the SimPL framework has better knowledge at *how much* macro blocks during placement, and gradient based cell spreading algorithm adopted in the Kraftwerk2 framework has better knowledge at *where* are the macro blocks during placement. Based on our implementation experience, this leads to better controllability at global view for SimPL framework and better controllability at local view for the Kraftwerk2 framework. The SimPL framework resolves relative order at local view by aligning cells to stripes. The Kraftwerk2 framework allocates white space at global view by imposing a demand supply constraint. While both placement frameworks can cover both global and local view of the placement, we propose a two stage global placement framework by combining the strength of partition based and gradient based cell spreading algorithm.

Bibliography

[1] M.-C. Kim, D.-J. Lee, and I. L. Markov, "SimPL: an algorithm for placing VLSI circuits," *Communication of the ACM*, vol. 56, pp. 105–113, June 2013.

[2] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Kraftwerk2 - A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1398–1411, Aug. 2008.

[3] M.-C. Kim and I. L. Markov, "ComPLx: A Competitive Primal-Dual Lagrange Optimization for Global Placement," in *Proceedings of the Design Automation Conference*, pp. 747–752, 2012.

[4] M.-C. Kim, N. Viswanathan, C. J. Alpert, I. L. Markov, and S. Ramji, "MAPLE: Multilevel Adaptive Placement for Mixed-Size Designs," in *Proceedings of the International Symposium on Physical Design*, pp. 193–200, 2012.

[5] M. Struzyna, "Sub-quadratic objectives in quadratic placement," in *Design, Automation Test in Europe Conference Exhibition*, pp. 1867–1872, 2013.

[6] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, "NTUplace3: An Analytical Placer for Large-Scale Mixed-Size Designs With Preplaced Blocks and Density Constraints," *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1228–1240, July 2008.

[7] T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze, and M. Xie, "mPL6: Enhanced Multilevel Mixed-Size Placement," in *Proceedings of the International Symposium on Physical Design*, pp. 212–214, 2006.

[8] X. He, T. Huang, W.-K. Chow, J. Kuang, K.-C. Lam, W. Cai, and E. F. Y. Young, "Ripple 2.0: high quality routability-driven placement via global router integration," in *Proceedings of the Design Automation Conference*, pp. 1521–1526, 2013.

[9] "Design Hierarchy Aware Routability-Driven Placement Contest." http://cad_contest.cs.nctu.edu.tw/CAD-contest-at-ICCAD2012/problems/p2/p2.html. [Online; accessed 19-Nov.-2012].

[10] J. A. Roy, S. N. Adya, D. A. Papa, and I. L. Markov, "Min-Cut Floorplacement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp. 1313–1326, July 2006.

[11] M.-C. Kim, J. Hu, D.-J. Lee, and I. L. Markov, "A SimPLR method for routability-driven placement," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 67–73, 2011.

[12] B. Hu, Y. Zeng, and M. Marek-Sadowska, "mFAR: Fixed-Points-Addition-Based VLSI Placement Algorithm," in *Proceedings of the International Symposium on Physical Design*, pp. 239–241, 2005.

[13] U. Brenner, M. Struzyna, and J. Vygen, "BonnPlace: Placement of Leading-Edge Chips by Advanced Combinatorial Algorithms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 9, pp. 1607–1620, 2008.

[14] G.-J. Nam, "ISPD 2005 Placement Benchmarks." http://archive.sigda.org/ispd2005/contest.htm.

[15] M. Pan, N. Viswanathan, and C. Chu, "An Efficient and Effective Detailed Placement Algorithm," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 48–55, Nov. 2005.

[16] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Abacus: fast legalization of standard cell circuits with minimal movement," in *Proceedings of the International Symposium on Physical Design*, pp. 47–53, 2008.

[17] W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. Chao, "Multi-threaded collision-aware global routing with bounded-length maze routing," in *Proceedings of the Design Automation Conference*, pp. 200–205, 2010.