

國立交通大學

電信工程研究所

碩士論文

對具最小擾動 *HiBinLegalizer* 之最佳化分析及網格數
的精化

Minimized Disturbance Optimal Analysis and Grid Bin Size
Refinement for *HiBinLegalizer*

研究生：鄭巧翎

指導教授：李育民 教授

中華民國 一 百 年 十 月 十 九 日

對具最小擾動 *HiBinLegalizer* 之最佳化分析及網格數的精化

Minimized Disturbance Optimal Analysis and Grid Bin Size Refinement
for HiBinLegalizer

研究生：鄭巧翎

Student : Chiao-Ling Cheng

指導教授：李育民

Advisor : Yu-Min Lee

國立交通大學

電信工程研究所

碩士論文

A Thesis

Submitted to Institute of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

August 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年十月

對具最小擾動*HiBinLegalizer*之最佳化分析及網格數的優化

學生：鄭巧翎

指導教授：李育民 博士

國立交通大學電信工程研究所碩士班

摘要

隨著積體電路製程的演進，最小線寬(feature size)的微小化，單一晶片上有數以百萬的標準元件(standard cell)以及智慧產權(Intellectual Property, 簡稱 IP)模組或巨集模組塊(macro block)，使得超大型積體電路擺置(placement)的問題愈趨於複雜。然而一個合法的擺置必須所有元件之間沒有交疊並且所有的標準元件須對齊至列上；因此在積體電路設計中，擺置的合法化(legalization)為一個重要的環節。

此篇論文主要對已發表的合法化方法*HiBinLegalizer*[1]，證明其所提出的位置解法器可得的最佳移動量，此外我們也提出數個定理改進搜尋列的尋找中位數的時間複雜度。由於*HiBinLegalizer*是一個階層式的方法，亦為將晶片分割成許多大小相同的單元格，並且透過每一個單元格的密度資訊，決定是否合併其他單元格一同納入合法化。對擺置合法化而言，分割晶片，除了可以增進執行速度，還可限制每一個標準元件的最大移動量以及減少列的搜索次數。然而初始的總單元格數量及單元格的長寬比會影響後續合併及最終結果的效能(移動量和執行時間)，因此本篇提出如何利用巨集模組塊的長寬比和最大合併區域內的標準單元個數，決定單元格長寬比以及決定總格數的範圍，最後合併兩者資訊找出好的分割網格尺寸，以取捨執行時間和最小移動量。

Minimized Disturbance Optimal Analysis and Grid Bin Size Refinement for HiBinLegalizer

Student : Chiao-Ling Cheng

Advisor : Yu-Min Lee

Department of Communication Engineering
National Chiao Tung University

ABSTRACT

With semiconductor fabrication technology developing, millions of standard cells and macros (pre-designed blocks or intellectual property (IP)) are integrated into a single chip. Legalization procedure is part of placement design in physical design automation. For a legal placement, all elements (cells and macros) are non-overlapping and all cells must be aligns to row. In this paper, we based on a method, *HiBinLegalizer*[1], which was published in ASPDAC-2010, proposed the grid bin size refinement and proved the solver for the objective function is optimal solution. The grid bin size refinement use the aspect ratio of macros to obtain the aspect ratio of grid and use the number of cells in maximum merged-bin to determine the bin size for *HiBinLegalizer*. Moreover, we proposed some lemmas to improve the procedure of trial row in *HiBinLegalizer*, so that, the order of time complexity is $O(1)$ as compared with the original $O(n)$. The experimental results demonstrate the effectiveness of our method.

誌 謝

此篇論文能夠順利完成，首要感謝我的指導老師 李育民博士，感謝老師平日指導研究時不遺餘力，當我遇到問題難點時，提供多面思考。三年期間，訓練我在看待問題時，不只狹隘的看到點，而是拓廣至面。經過老師的淬鍊，不論是在研究或是在心智上都有顯著的成長，使我面對未來工作考驗更加有自信。

在實驗室，感謝博班學長培育不時的關心研究進度，用淺顯易懂的方式解釋問題。感謝懷中、正忠和宗祐學長以及麒文、亭蓉和書含平日的關心，感謝學弟志升在研究上的討論。

感謝羽球隊的廖威彰 教練和王志全 教練以及一路上支持我的朋友，莉嬪和慧敏以及隊友們，他們的陪伴使我度過一切難關。

最後深深感恩我的父母和姐姐，感謝你們一直以來的支持與鼓勵。在我的研究生涯中，提供強力的後盾，讓我無後顧之憂的完成碩士學位。僅此將本論文獻給你們，共同分享這份喜悅以及榮耀。

目錄

1	導論	1
1.1	背景簡介	1
1.2	研究貢獻	3
2	背景與相關文獻簡介	4
2.1	標準元件的擺置合法化	4
2.1.1	標準單元	4
2.1.2	移動量的量測	4
2.2	擺置合法化的相關文獻	5
2.3	概述 <i>HiBinLegalizer</i> 的架構	6
2.3.1	問題描述	6
2.3.2	簡介 <i>HiBinLegalizer</i>	7
2.3.3	合併格程序	8
2.3.4	合法化程序	9
3	對 <i>HiBinLegalizer</i> 之最小擾動的最佳化分析以及網格數的精化	12
3.1	交疊連續性	12
3.1.1	交疊連續性的特性	13
3.1.2	邊界條件	16
3.2	部分交疊連續性之最佳化分析	17
3.2.1	問題公式化	19
3.2.2	解最佳位置	19
3.3	改善 <i>HiBinLegalizer</i> 尋找中位數的過程	24
3.4	網格數的精化	29
3.4.1	長寬比的精化	30
3.4.2	總格數的初估	31
3.4.3	決定總格數	32
4	實驗結果	33
5	結論	36
A	附錄	40

圖目錄

1.1	數位超大型積體電路設計流程圖	2
1.2	(a)實體設計流程圖 (b)實體設計之擺置設計流程圖	2
2.1	標準單元的範例	4
2.2	二維的歐式距離(綠線)為 $\sqrt{3^2 + 4^2} = 5$ ，而曼哈頓距離(紅線)為 $3 + 4 = 7$ 。	5
2.3	<i>HiBinLegalizer</i> 的流程圖	7
2.4	單元格的合併示意圖(a)選定一個單元格，其密度高於閾密度值(b)合併十字方向的單元格(c)若密度仍高於閾密度值，則合併四角方向的單元格(d)對合併區域的十字方向的單元格做合併，重複動作直到合併區域的密度低於閾密度值。	8
3.1	函數 $f(\bar{x})$ ，當 N_r 為奇數(左圖)；當 N_r 為偶數(右圖)	15
3.2	合法化之邊界條件	16
3.3	群集群組化過程的概念圖	20
3.4	將 C^{last} 分解成 m 個群集與一個標準單元 i	26
3.5	<i>HiBinLegalizer</i> 修改後的流程	30
3.6	決定寬長比的示意圖	31
3.7	總格數的切法	32



表目錄

2.1	符號說明	9
3.1	符號說明	17
3.2	此小節的符號說明	26
4.1	2005 ISPD 擺置競賽的測試電路	34
4.2	標準單元最大移動量的比較結果	35
4.3	標準單元總移動量的比較結果	35
4.4	執行時間(s)的比較結果	35



1

導論

隨著半導體製程技術快速的發展，電晶體最小線寬(feature size)的微型化，使得單一積體電路晶片上可整合百萬甚至更多的電晶體。設計上，為了縮短及降低超大型積體電路(Very Large Scale Integration, 簡稱 VLSI)¹設計的花費時間以及製造成本，電腦輔助設計(CAD)及電子設計自動化(EDA)亦隨之發展。圖1.為數位大型積體電路設計流程，從描述一個設計的功能架構，將設計的功能運作行為轉換到硬體描述語言(HDL)，再將其轉換到邏輯閘 (ex. 暫存器，乘法器)，之後描述邏輯閘如何擺置到晶片上，最後到封裝與測試。而如何將邏輯閘有效的擺置到晶片上，即為實體設計的範疇。

實體設計(physical design)流程，如圖1.2(a)所示，在此設計流程中，所有的元件不論是智慧產權(Intellectual Property, 簡稱 IP)或是邏輯閘，都將視其為一個簡單的幾何圖形，稱模組塊(block)。然而實體設計亦可分成三大部分；(1)平面規劃(floorplanning)，初估所需的面積，以及初步的排列元件。排列過程元件的長寬可變，但面積固定 (2)擺置(placement)，類似平面規劃，但此階段所有的元件長寬形狀固定，並且初估連接元件所需之線長並且元件間不可重疊(overlap) (3)繞線(routing)，為元件之間的連線找到最好的路徑。

1.1 背景簡介

超大型積體電路的擺置又可分成：(1)全域擺置(global placement) (2)擺置合法化(legalization) (3)局部擺置(detail placement)。全域擺置，隨著不同的設計要求，有不同的考量議題，例如：線長(wirelength)[2]，時序(timing)[3][4]，功率消耗(power consumption) [5][6]等議題。由於全域擺置完的結果多少會伴隨著一些不合法的情形，例如：元件間有重疊；因此，擺置合法化(Legalization)顧名思義，其目的在移除不合法的情形，但合法化的過程，必須盡可

¹積體電路晶片上整合超過百萬個電晶體，即稱超大型積體電路

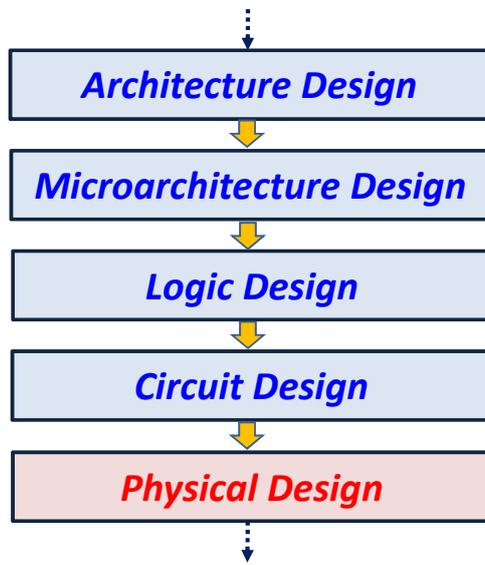


圖 1.1: 數位超大型積體電路設計流程圖

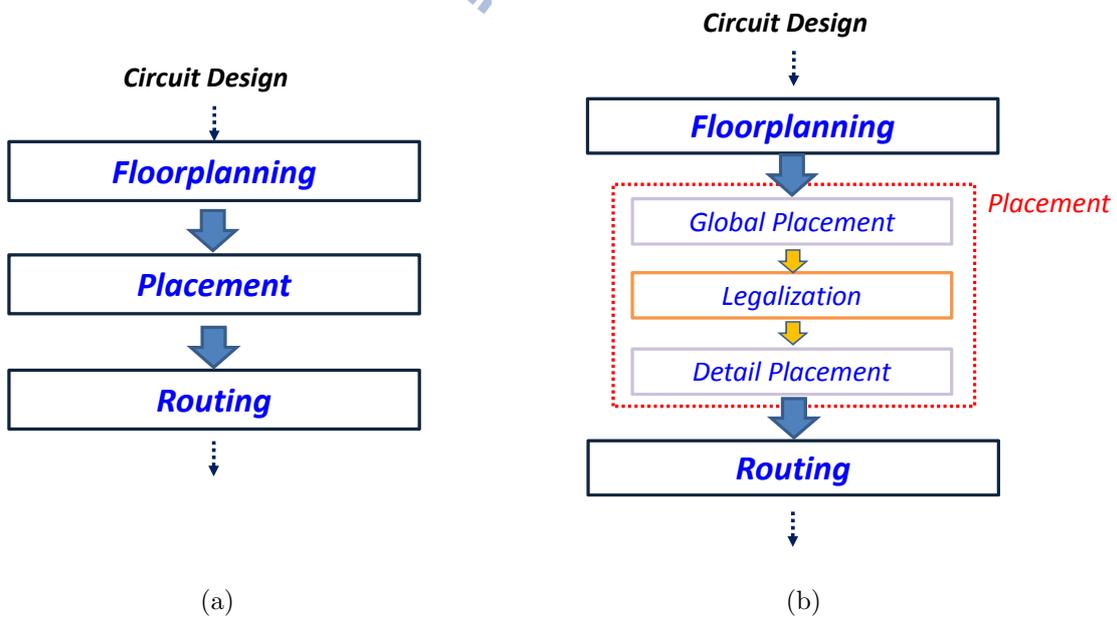


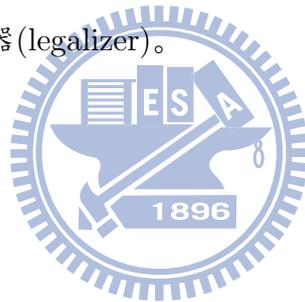
圖 1.2: (a) 實體設計流程圖 (b) 實體設計之擺置設計流程圖

能的符合全域擺置出的結果，因此合法化在擺置問題中為一重要的部分；局部擺置，則在合法的擺置下做局部的調整，例如：空間的管理等。

1.2 研究貢獻

此篇論文，我們主要延續文獻“A hierarchical bin-based legalizer for standard-cell designs with minimal disturbance” [1]，發表於第十五屆亞洲和南太平洋設計自動化國際會議²，成果貢獻可分成下列幾個部分：

- 對文獻[1]所提出位置解法器，做進一步分析與證明其最佳化。
- 我們提出數個引理，簡化尋找中位數的過程，使得尋找中位數的時間複雜度由 $O(n)$ 降低為 $O(1)$ 。
- 提出網格數的精化方法。選取網格數時在效能和時間之間做取捨；除此之外，所提出的精化法可應用在任何的合法器(legalizer)。



²15th Asia and South Pacific Design Automation Conference 2010, 簡稱 15th ASPDAC-2010

2

背景與相關文獻簡介

2.1 標準元件的擺置合法化

為了縮短設計時間，實體設計有多種不同的標準佈局型態，例如：標準單元(standard cells)，閘陣列(gate arrays)，可程式的邏輯閘陣列(programmable logic arrays)等等。此篇論文，我們選用標準單元型態作為我們的佈局型態。

2.1.1 標準單元

對標準單元的佈局型態而言，每一個邏輯閘皆可用一矩形表示，有著相同高，可變的寬稱之為標準單元。積體電路晶片可擺置的區域被分割成許多等高的列。每一個標準單元，必須擺放並且對準至列上。因此標準單元的高等於列高，如圖2.1所示。

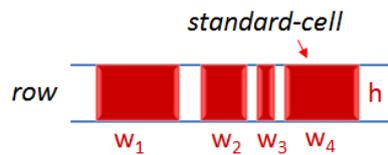


圖 2.1: 標準單元的範例

2.1.2 移動量的量測

針對擺置合法化的問題，該如何計算元件合法化前與合法化後所改變的移動量?在幾何學中，有兩個常用的計量距離方法:(1)歐幾里德距離(Euclidean distance)，簡稱歐氏距離(2)出租車距離(Taxicab distance)，亦稱為曼哈頓距離(Manhattan distance)。若 n 維度的向量空間中，有 $P = (p_1, p_2, \dots, p_n)$ 和 $Q = (q_1, q_2, \dots, q_n)$ 兩點， n 維的歐氏距離 d_E^n 定義為

$$d_E^n = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (2.1)$$

而出租車距離 d_T^m 定義成

$$d_T^m = |p_1 - q_1| + |p_2 - q_2| + \cdots + |p_n - q_n| \quad (2.2)$$

擺置合法化屬於二維的空間問題，因此歐氏距離和出租車距離分別為

$$d_E^2 = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \quad (2.3)$$

和

$$d_T^2 = |p_1 - q_1| + |p_2 - q_2| \quad (2.4)$$

由圖2.2，我們可知歐式距離即為連接 P 和 Q 兩點的直線距離，而曼哈頓距離則為 P 和 Q 兩點間水平方向差與垂直方向差的總和

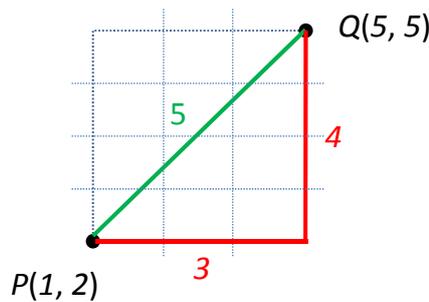


圖 2.2: 二維的歐式距離(綠線)為 $\sqrt{3^2 + 4^2} = 5$ ，而曼哈頓距離(紅線)為 $3 + 4 = 7$ 。

在此篇論文，我們使用曼哈頓距離來表示元件在合法化前後的左下角座標改變量。因此，給 N 個尚未合法化的標準單元，其合法化前後座標位置改變量，可用曼哈頓距離和表示成

$$\sum_{k=1}^N (|x'(k) - x(k)| + |y'(k) - y(k)|)$$

其中 $(x(k), y(k))$ 和 $(x'(k), y'(k))$ 分別是標準單元 k 在全域擺置和合法化後的左下角座標位置。

2.2 擺置合法化的相關文獻

近年來，有許多解決擺置合法化的方法被提出[15, 7, 8, 9, 10, 11, 12]，其中以 Tetris [10]最為著名並且廣泛應用在許多全域擺置方法的合法化上[13, 2, 14]。Tetris是一種貪婪演算法，對每一個標準元件找到一個列，使得其標準單元的移動量最小，一但標準單元擺入後，其

合法化位置就不再更動；另一方面，有數個文獻方法[11, 1, 12]，皆以 Tetris 的架構為基礎進而發展出的合法化方法，以獲得最佳的合法化位置。Tetris [10]，第一步，先對所有的標準單元的全域 x -座標位置做排序，再依序為標準單元搜尋所有的列，計算標準單元的移動量，最後擺入一個最好的列，使得標準單元的移動量最小；標準單元擺入後，即視為障礙物，因此，爾後的標準單元僅能接續擺放。Abacus [11]，以動態規劃為基礎的演算法，與Tetris類似，Abacus搜尋所有的列，找出代價最小的列擺入；不同於Tetris，Abacus的目標函數為二次方程式(quadratic equation)，標準單元擺入後，其合法化位置若需要仍可能會更動。文獻[12]，此方法亦相似於Tetris，其應用k-mediod群集法，先對列做分群，給出搜尋列的範圍，以同時得到最小的位移(displacement)以及線長(wirelength)的改變量。*HiBinLegalizer* [1]是一種階層式的演算法，將可擺置的區域分割成許多的單元格(bin)，使得搜尋列的範圍縮小，可限制最大移動量，另外更提出權重距離和的解法器(weighted sum of distance solver)使之總權重移動量為最小。由於此論文是以*HiBinLegalizer*為基礎，因此下一小節再做詳細介紹。另外文獻[15]提出對單一系列上應用動態規劃法求得標準單元的擺置位置同時合法化列上的標準單元，並說明若固定標準單元的順序，此時可求得線長最佳解。而文獻[7]將文獻[15]的想法應用在合法化上，利用最小代價流程法(minimize-cost flow)先決定標準單元的列，再應用文獻[15]的動態規劃法對列上的標準單元合法化。文獻[9]提出利用最小路徑(shortest path)法，先移除列上的標準單元面積大於列的可放置面積的情況，再對單一系列做合法化。

2.3 概述 *HiBinLegalizer*的架構

此篇論文延續文獻*The Hierarchical Bin-Based Legalizer for Standard-Cell Designs with Minimal Disturbance*(*HiBinLegalizer*[1])的方法以及理論證明。因此以下章節我們將簡介*HiBinLegalizer*所處理的問題，解決問題的作法以及相關的理論證明。

2.3.1 問題描述

對標準單元為設計基礎的合法化而言，主要三種物件(1)標準單元 (2)巨集(macro) (3)列(row)。所有列的列高相同，列寬預先給定。巨集視為不可移動的障礙物，位置長寬固定；僅標準單元可移動。合法化後，標準單元須放置在列上，與列對齊，不得與巨集或標準單元面積重疊，並且合法化的結果越接近全域擺置的結果越好。

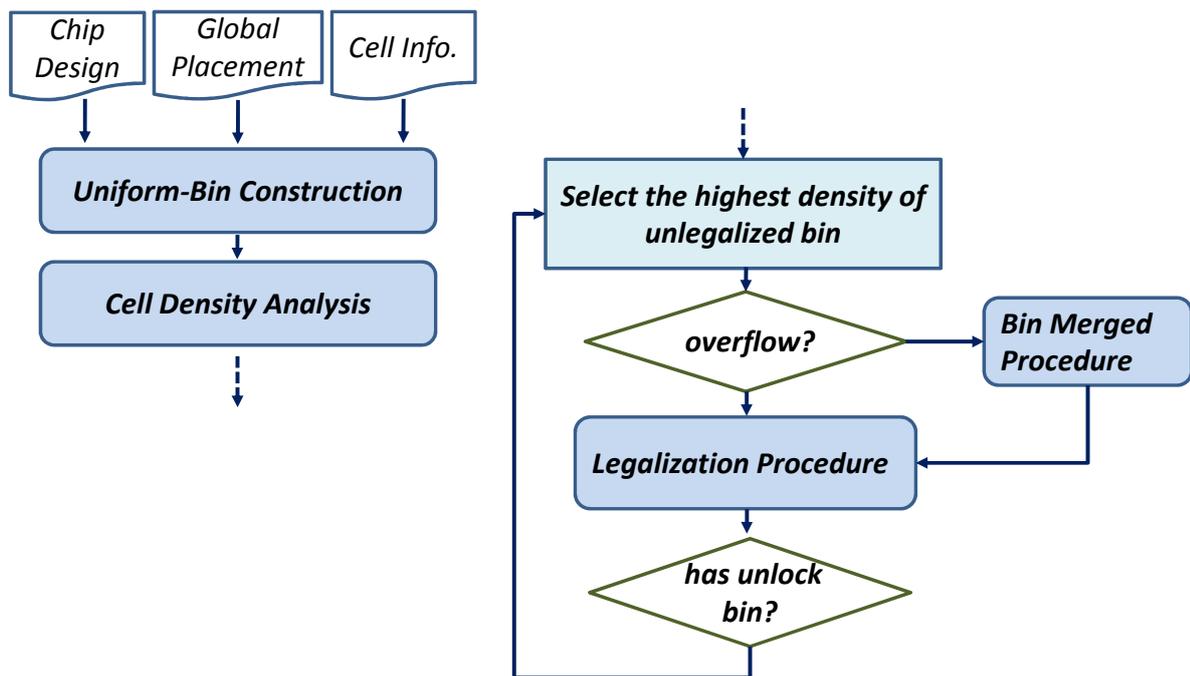


圖 2.3: *HiBinLegalizer* 的流程圖

問題公式化

由上面的問題描述，目標函數為

$$\min. \sum_{k=1}^N |x'(k) - x(k)| + |y'(k) - y(k)| \quad (2.5)$$

$$\text{s.t. } x'(k) - x'(k-1) \geq w(k-1); 2 \leq k \leq N \quad (2.6)$$

其中 N 是標準單元的總個數， $w(k)$ 是標準單元 k 的寬度， $(x(k), y(k))$ 和 $(x'(k), y'(k))$ 分別為合法化前後標準單元 k 的左下角座標。

2.3.2 簡介 *HiBinLegalizer*

圖2.3為 *HiBinLegalizer* 的概觀流程圖。給定任意的全域擺置結果，晶片的設計，元件的資訊；首先，我們先將晶片分割成 $G_x \times G_y$ 個均勻大小的單元格，其中 G_x 和 G_y 分別為水平和垂直方向的格子數。再根據每一個標準單元的全域座標位置，找其最近距離的單元格。（註：此篇所提及之距離皆為曼哈頓距離）。因此，我們可求得單元格 b 的標準單元密度(standard-cell

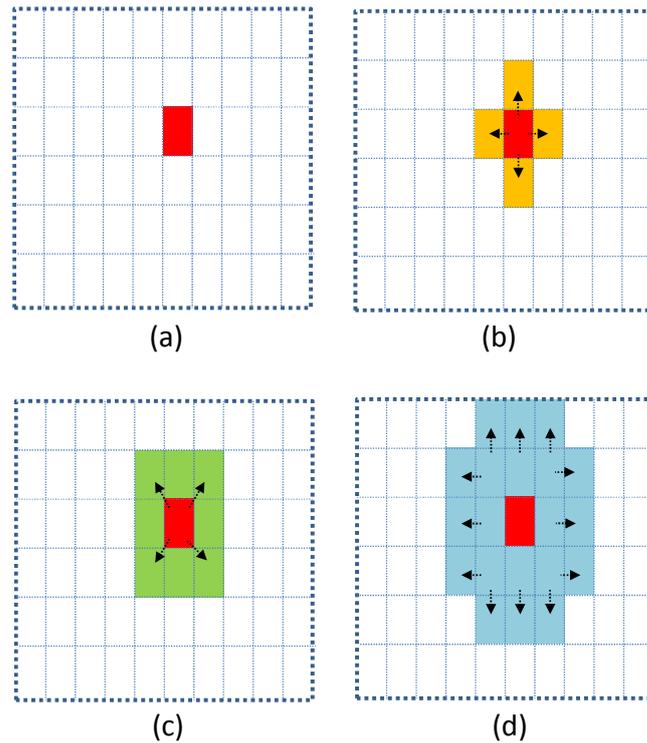


圖 2.4: 單元格的合併示意圖(a)選定一個單元格，其密度高於閾密度值(b)合併十字方向的單元格(c)若密度仍高於閾密度值，則合併四角方向的單元格(d)對合併區域的十字方向的單元格做合併，重複動作直到合併區域的密度低於閾密度值。

density),

$$D(b) = \frac{A(b)}{C(b)}$$

其中 $A(b)$ 是指派至單元格 b 的標準單元的總面積以及容量(capacity) $C(b)$ 定義為單元格 b 的面積減去障礙物(obstacle)的面積。計算完密度後，選取尚未合法化且密度最大的單元格，若其密度大於預先定義的閾密度值(threshold density)，則進入合併格程序(bin-merged procedure); 反之，程序直接進入合法化程序。重複上述過程，直到所有的單元格完成合法化。

2.3.3 合併格程序

當完成建構均勻單元格(*Uniform-bin Construction*)和標準單元密度分析(*Cell Density Analysis*)的程序後，單元格密度與單元格鄰近的單元格已知。若單元格密度大於閾密度值，則執行合併格的程序。合併單元格的程序有兩種合併規則(1)十字方向合併(東、西、南、北)(2)四角方向合併(東北、東南、西北、西南)。一開始，由一個未合法化且密度超過閾密度

值的單元格為中心，合併其十字方向未合法化的單元格形成合併區域(merged-bin region)，記為 b_m ，並重新計算合併區域的密度，若密度仍高於閾密度值，則對合併區域的四角方向的單元格做合併；依序重複上述步驟合併單元格、計算密度，直到密度小於閾密度值，如圖2.4所示。

2.3.4 合法化程序

N_r :	列 r 上標準單元的個數
N_c :	群集內標準單元的個數
$x(i)$:	標準單元 i 全域擺置的左下 x -座標
$x'(i)$:	標準單元 i 合法位置的左下 x -座標
$w(i)$:	標準單元 i 的寬度
C^{last} :	列 r 上，最右邊的群集
$w_{C^{last}}$:	群集 C^{last} 的寬度
$q_{C^{last}}$:	群集 C^{last} 的左下 x -座標位置
C^p :	前置群集，在群集 C^{last} 的前一個群集
w_{C^p} :	群集 C^p 的寬度
q_{C^p} :	群集 C^p 的左下 x -座標位置

表 2.1: 符號說明

Algorithm 1 合法化演算法的虛擬程式碼

Input: A set of target cells S_c and a set of candidate rows S_r

- 1: Sort the cells in S_c according to their x -coordinate position
- 2: **for all** cell i **do**
- 3: $c_{best} \leftarrow \infty$
- 4: **for all** candidate row r **do**
- 5: $c \leftarrow \text{LegRow}(r, \text{trial})$ // Determine the cost c
- 6: **if** $c < c_{best}$ **then**
- 7: $c_{best} \leftarrow c$
- 8: $r_{best} \leftarrow r$
- 9: **end if**
- 10: **end for**
- 11: $\text{LegRow}(r_{best}, \text{real})$ // Insert cell i into sub-row r_{best}
- 12: **end for**

演算法1列出合法化演算法的虛擬程式碼。首先，對合併區域內所有的標準單元，依其在全域擺置的左下 x -座標位置由小至大排序，其目的在於確立標準單元在 x -方向的相對位置。排序完成後，依此序為標準單元，尋找列使得標準單元擺入後有最小的代價(cost)。在尋找列的

Algorithm 2 LegRow

Input: Cell i and the legalized clusters in sub-row r

- 1: $\mathcal{C}^{last} \leftarrow$ the last legalized cluster in sub-row r
 - 2: $\mathcal{C}^p \leftarrow$ the processor of \mathcal{C}^{last}
 - 3: Remove the last cluster in sub-row r
 - 4: **if** $x(i) - q_{\mathcal{C}^{last}} \leq w_{\mathcal{C}^{last}}$ **then**
 - 5: Append cell i to \mathcal{C}^{last}
 - 6: The Optimal Position Solver(\mathcal{C}^{last})
 - 7: **while** $q_{\mathcal{C}^p} - q_{\mathcal{C}^{last}} \leq w_{\mathcal{C}^{last}}$ **do**
 - 8: Append all cells in \mathcal{C}^p to \mathcal{C}^{last} .
 - 9: The Optimal Position Solver(\mathcal{C}^{last})
 - 10: $\mathcal{C}^p \leftarrow$ the processor of \mathcal{C}^p
 - 11: Remove the last cluster in sub-row r
 - 12: **end while**
 - 13: Add \mathcal{C}^{last} into sub-row r
 - 14: **else**
 - 15: Create a cluster \mathcal{C}
 - 16: $\mathcal{C} \leftarrow$ cell i
 - 17: Add the cluster \mathcal{C} in sub-row r
 - 18: **end if**
-

過程，由於已給定目標列 r ，因此可算得 y 方向的代價。目標函數可寫成

$$\min. \sum_{k=1}^{N_r} |x'(k) - x(k)| \quad (2.7)$$

$$\text{s.t. } x'(k) - x'(k-1) \geq w(k-1); 2 \leq k \leq N_r \quad (2.8)$$

其中 N_r 是列 r 上的標準單元個數。試驗列演算法LegRow(r , trial)(行5)用來決定 x -方向的代價，而擺入列演算法LegRow(r_{best} , real)(行11)將標準單元真擺至列上。試驗列和擺入列演算法(LegRow)又可分成兩個子程序崩解(Collapse)和位置解法器(The Position Solver)，如演算法2所示。

崩解

演算法1.中，所有的標準單元會依序擺入至列上。崩解演算法(行8-12)，當試擺或擺入標準單元 i ，若其全域 x -座標位置與已擺置合法化的群集有面積重疊，則群聚已合法化的群集與標準單元，再利用位置解法器為群集內所有的標準單元找合法化位置；若合法化後的群集又與前置群集(processor cluster)面積重疊，則循環崩解、找位置解直到新位置不再與前置群集面積重疊。

位置解法器

位置解法器(行6與行9)為群集內的所有標準單元找到合法位置；此解法器假設合法化後，所有標準單元的合法位置皆緊鄰(abutted)在一起，也就是，

$$x'(k) - x'(k-1) = w(k-1), \forall 2 \leq k \leq N_c \quad (2.9)$$

換言之，

$$x'(k) = x'(1) + \sum_{i=1}^{k-1} w(i), \forall 2 \leq k \leq N_c \quad (2.10)$$

其中 $x'(1)$ 是群集內最左邊的標準單元合法的位置。將(2.9)代入(2.8)，可得

$$\begin{aligned} \sum_{k=1}^{N_c} |x'(k) - x(k)| &= |x'(1) - x(1)| + |x'(1) + w(1) - x(2)| + \cdots + |x'(1) + \sum_{i=1}^{N_c-1} w(i) - x(N_c)| \\ &= |x'(1) - \Delta x_1| + |x'(1) - \Delta x_2| + \cdots + |x'(1) - \Delta x_{N_c}|, \end{aligned} \quad (2.11)$$

將目標函數重新整理可得

$$\min. \sum_{k=1}^{N_c} |x'(k) - x(k)| = \min. \sum_{k=1}^{N_c} |x'(1) - \Delta x_k| \quad (2.12)$$

其中 $\Delta x_k = x(k) - \sum_{i=1}^{k-1} w(i)$ 。欲最小化(2.8)式，等同於最小化(2.12)式；觀察可知，問題從多變數問題簡化至單變數問題。由定理1.知，未知數 $x'(1)$ 的最佳解是數列 $P_c = \{\Delta x_k\}_{k=1}^{N_c}$ 的中位數。(備註：*HiBinLegalizer*部分證明請參見附錄，詳細證明與敘述請參見文獻[1])

3

對 *HiBinLegalizer* 之最小擾動的最佳化分析以及網格數的精化

在 *HiBinLegalizer* [1] 中，證明一群標準元件若滿足交疊連續性 (consecutively overlapped) 的條件，則位置解法器所找到的位置解是最佳解。然而交疊連續性是否為必要條件？在此章節我們將進一步證明在第二章所提及的位置解法器應用在擺置合法的問題上，若不滿足交疊連續性亦為最佳解。除此之外，*HiBinLegalizer* 在決定網格數上，僅粗略給定一個方針，因此我們提出網格數的精化方法，給出如何透過基本給定的資訊與全域擺置結果找出適合的網格數大小。

此章的組織如下：章節 3.1. 我們首先介紹交疊連續性的性質；章節 3.2. 我們接續前一章節的交疊連續性之性質，延伸至部份交疊連續性，對其證明並找出最佳的位置解，使其可推廣至一般情形；章節 3.3. 我們提出數個引理，以降低試驗列演算法找尋中位數的複雜度；章節 3.4. 我們提出網格精化的方法，透過給定的資訊，找出適當的網格尺寸。

3.1 交疊連續性

在前面的章節，介紹了在擺置合法化的問題中，我們用矩形來表示標準元件，也就是我們僅知道元件的長寬以及左下座標。而此章節我們將延續文獻 [1] 中的若滿足交疊連續性，若遇到邊界問題，如何找到最佳解。本文的開始，我們首先定義甚麼是交疊連續性

定義 1 (交疊連續性). N 個標準單元的集合，對其左下 x -座標由小至大排序，編號從 1 到 N 。對所有的 k ，若滿足下式

$$x(k) - x(k-1) \leq w(k-1), \quad 2 \leq k \leq N \quad (3.1)$$

則稱此 N 個標準單元滿足交疊連續性。其中 $x(k)$ 為元素 k 的左下 x -座標， $w(k)$ 為標準單元 k 的寬度。

3.1.1 交疊連續性的特性

定義單調遞增/遞減數列(monotonic increasing/decreasing sequence)和單調遞增/遞減函數(monotonic increasing/decreasing function)。若 $x_k \leq x_{k+1}$ ，則稱數列 $\{x_k\}$ 是單調遞增數列；反之，對所有的 $k \in \mathbb{N}$ ，若 $x_k \geq x_{k+1}$ ，則數列 $\{x_k\}$ 是單調遞減數列。函數 f ，有 $x \leq y$ 且 $f(x) \leq f(y)$ ，則函數 f 是單調遞增函數；反之，若函數 f ，有 $x \geq y$ 且 $f(x) \geq f(y)$ ，則函數 f 是單調遞減函數。

特性 1. 有 N 個單元滿足交疊連續性，

$$x(k) - x(k-1) \leq w(k-1), 2 \leq k \leq N \quad (3.2)$$

倘若定義數列 $S = \{\Delta x_k\}_{k=1}^N$ 且

$$\Delta x_k = x(k) - \sum_{j=1}^{k-1} w(j), k = 1, \dots, N \quad (3.3)$$

則數列 S 必為單調遞減數列，有

$$\Delta x_k \leq \Delta x_{k-1}, \forall k = 2, \dots, N$$

其中 $x(k)$ 和 $w(k)$ 分別是單元 k 的左下角 x -座標和單元 k 的寬。

Proof. 已知對所有的 $k = 2, \dots, N$ ， $x(k) \geq x(k-1)$ 且 $x(k) - x(k-1) \leq w(k)$ 和數列 S 的第 $k-1$ 項與第 k 項

$$\Delta x_{k-1} = x(k-1) - \sum_{i=1}^{k-2} w(i) \quad (3.4)$$

$$\Delta x_k = x(k) - \sum_{i=1}^{k-1} w(i) \quad (3.5)$$

將兩式相減，可得

$$\Delta x_k - \Delta x_{k-1} = x(k) - x(k-1) - w(k-1) \quad (3.6)$$

由於 $x(k) - x(k-1) \leq w(k-1)$ ，可得

$$\Delta x_k - \Delta x_{k-1} = x(k) - x(k-1) - w(k-1) \leq 0 \quad (3.7)$$

因此有限數列 S 是單調遞減函數，也就是對所有的 $k = 2, \dots, N$ ，有 $\Delta x_k \leq \Delta x_{k-1}$ 。 \square

就合法化的問題而言，標準單元必須對齊至列上並且合法化後，位於列上的所有標準單元的 x -座標位置順序不變。若已知某一個列上有 N_r 個標準單元，其意味著此列上的標準單元 y -座標位置固定， $\sum_{l=1}^{N_r} |y'(l) - y(l)|$ 即為已知常數。

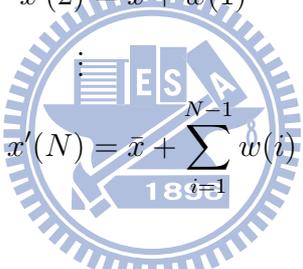
因此，目標函數可寫成

$$\min. \sum_{l=1}^{N_r} |x'(l) - x(l)| \quad (3.8)$$

$$\text{s.t. } x'(l) - x'(l-1) \geq w(l-1); 2 \leq l \leq N_r \quad (3.9)$$

根據定理 1，若 N_r 個標準單元滿足交疊連續性條件，則合法化後標準單元之間緊鄰(abutted)有最佳解。

因此，我們若 \bar{x} 是第一個標準單元的合法化位置，則其他的標準單元的合法化位置可由 \bar{x} 推得

$$\begin{aligned} x'(1) &= \bar{x} \\ x'(2) &= \bar{x} + w(1) \\ &\vdots \\ x'(N) &= \bar{x} + \sum_{i=1}^{N-1} w(i) \end{aligned}$$


此時，目標函數可改寫成

$$\begin{aligned} \sum_{l=1}^N |x'(l) - x(l)| &= |\bar{x} - \Delta x_1| + |\bar{x} - \Delta x_2| + \dots + |\bar{x} - \Delta x_N| \\ &= \sum_{l=1}^N |\bar{x} - \Delta x_l| \\ &\triangleq f(\bar{x}) \end{aligned} \quad (3.10)$$

其中 $\Delta x_l = x(l) - \sum_{i=1}^{l-1} w(i)$ 對所有的 l , $1 \leq l \leq N_r$ 。再由定理 2，可知(3.10)式有最小值，當 \bar{x} 是數列 $\{\Delta x_l\}_{l=1}^{N_r}$ 的中位數(median value)時。

接下來，我們要證明函數 $f(\bar{x})$

$$f(\bar{x}) \text{ is } \begin{cases} \text{單調遞減函數,} & \text{當 } \bar{x} \leq \Delta Q_{\frac{1}{2}} \\ \text{單調遞增函數,} & \text{當 } \bar{x} \geq \Delta Q_{\frac{1}{2}} \end{cases}$$

其中 $\Delta Q_{\frac{1}{2}}$ 是數列 $\{\Delta x_l\}_{l=1}^{N_r}$ 的中位數。根據特性1.數列 $\{\Delta x_l\}_{l=1}^{N_r}$ 是單調遞減，即對所有的 $2 \leq l \leq N_r$ ，有 $\Delta x_{l-1} \geq \Delta x_l$ ；因此，中位數 $\Delta Q_{\frac{1}{2}}$ 可以表示成

$$\Delta Q_{\frac{1}{2}} = \begin{cases} \Delta x_{\frac{N_r+1}{2}}, & \text{當 } N_r \in \text{奇數} \\ \alpha \Delta x_{\frac{N_r}{2}+1} + (1-\alpha) \Delta x_{\frac{N_r}{2}}, & \text{當 } N_r \in \text{偶數} \end{cases}$$

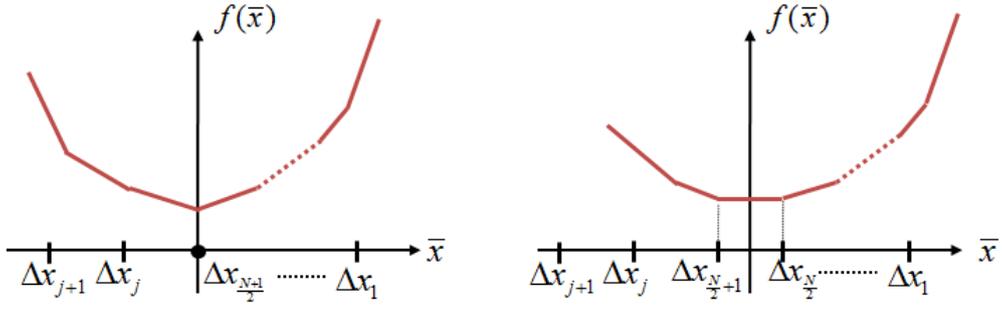


圖 3.1: 函數 $f(\bar{x})$, 當 N_r 為奇數(左圖); 當 N_r 為偶數(右圖)

, 其中 α 是介於 0 到 1 的常數。

假設 $x_s \geq x_t > \Delta Q_{\frac{1}{2}}$, 則

$$\begin{aligned}
 f(x_s) - f(x_t) &= \sum_{l=1}^{N_r} |x_s - \Delta x_l| - \sum_{l=1}^{N_r} |x_t - \Delta x_l| \\
 &= \sum_{l=1}^i (\Delta x_l - x_s) + \sum_{l=i+1}^{N_r} (x_s - \Delta x_l) \\
 &\quad - \left(\sum_{l=1}^j (\Delta x_l - x_t) + \sum_{l=j+1}^{N_r} (x_t - \Delta x_l) \right) \\
 &= (N_r - 2i)x_s - (N_r - 2j)x_t \\
 &\quad + \sum_{l=1}^i \Delta x_l - \sum_{l=1}^j \Delta x_l + \sum_{l=j+1}^{N_r} \Delta x_l - \sum_{l=i+1}^{N_r} \Delta x_l \\
 &= (N_r - 2i - 2j + 2j)x_s - (N_r - 2j)x_t - 2 \sum_{l=i+1}^j \Delta x_l \\
 &= (N_r - 2j)(x_s - x_t) + \underbrace{2 \sum_{l=i+1}^j (x_s - \Delta x_l)}_{\geq 0}
 \end{aligned}$$

由上式可知, 若所有的 $1 \leq l \leq N_r$, $l \in \mathbb{N}$, 不存在 $\Delta x_l > x_s$, 則 $f(x_s) - f(x_t)$ 恆大於零; 若存在 $\Delta x_l > x_s$, $1 \leq l \leq N_r$, $l \in \mathbb{N}$, 且已知, $x_s \geq x_t > \Delta Q_{\frac{1}{2}}$ 以及數列 $\{\Delta x_l\}$ 是單調遞減, 可知, $1 < i \leq j < \frac{N_r}{2}$, 可得上式恆大於。因此當 $\bar{x} \geq \Delta Q_{\frac{1}{2}}$ 時, $f(\bar{x})$ 是單調遞增函數; 同理可得, 當 $\bar{x} \leq \Delta Q_{\frac{1}{2}}$ 時, $f(\bar{x})$ 是單調遞減函數。

特性 2. 給定 N 個標準單元滿足交疊連續性, 合法化後, N 個標準單元順序不變, 目標函數可寫

成

$$\begin{aligned} \min. \quad & \sum_{k=1}^N |\bar{x} - \Delta x_k|; \Delta x_k = x(k) - \sum_{i=1}^{k-1} w(i) \\ \text{s.t.} \quad & x'(k) - x'(k-1) = w(k-1); 2 \leq k \leq N \end{aligned}$$

其中 $x'(k)$ ($x(k)$) 和 $w(k)$ 分別為標準單元 k 的合法化(全域)位置和單元 k 的寬度, \bar{x} 是標準單元 1 的合法化位置。已知當 \bar{x} 是數列 $\{\Delta x_k\}$ 的中位數時, $\bar{x} = \Delta Q_{\frac{1}{2}}$, 有最小值。若改變 \bar{x} 愈遠離 $\Delta Q_{\frac{1}{2}}$, 則目標函數值愈大。

3.1.2 邊界條件

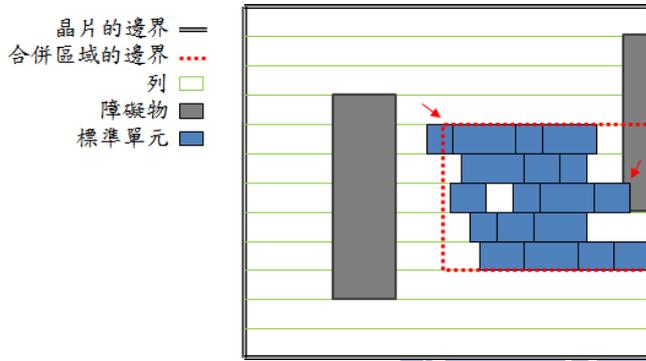


圖 3.2: 合法化之邊界條件

在前面章節, 合法化的位置僅受限於兩個條件, 標準單元之間沒有面積交疊以及合法化後的順序不變性。然而實際的問題中, 可能會觸及邊界的問題, 例如: 標準單元擺置的位置不可超出晶片可擺置的邊界範圍或超出單元格邊界以及標準單元合法最佳化後與障礙物有面積交疊, 如圖3.2所示。因此, 滿足邊界條件下, 我們該如何修正最佳位置, 使得目標函數值為最小? 首先, 目標函數及邊界條件改寫成

$$\begin{aligned} \min. \quad & \sum_{l=1}^N |x'(l) - x(l)| \\ \text{s.t.} \quad & x'(l) - x'(l-1) \geq w(l-1), \\ & x'(l) \geq Bx_{min}, \\ & x'(l) \geq Rx_{min}, \\ & x'(l) + w(l) \leq Bx_{max}, \\ & x'(l) + w(l) \leq Rx_{max}, 2 \leq l \leq N \end{aligned}$$

表 3.1: 符號說明

*	所有的上標代表第 i 次合法化的結果
N_r	列上標準單元的個數
C_k^i	根據第 i 次合法化結果, 所得的第 k 個群集
$N_{c,k}^i$	C_k^i 內的群集數
C^i	群集簇, 多個群組的集合
N_c^i	群集簇中的未合法的群集個數
$x^i(l)$	標準單元 l 經過第 i 次合法化的左下角 x -位置
$w(l)$	第 l 個標準單元的寬度
$X_c^i(C_k^i)$	第 i 次合法化, 群集 C_k^i 的左下角 x -座標位置
$W_c^i(C_k^i)$	群組 C_k^i 的總寬度

其中 Bx_{min} 和 Bx_{max} 分別是單元格的最左 x -邊界及最右 x -邊界; Rx_{min} 和 Rx_{max} 是列的最左及最右的 x 邊界。

利用引理7.知, 合法化後, 當所有的標準單元緊鄰(abutted)目標函數有最小值, 再利用特性2.可知若第一個單元的合法化位置, $x'(1)$, 遠離最佳位置其代價越大。因此若遇邊界問題, 則第一個單元的合法化位置將修正為 $x'(1) = \max\{Bx_{min}, Rx_{min}\}$, 當 $x'(1) < \max\{Bx_{min}, Rx_{min}\}$ 和 $x'(N) = \min\{Bx_{max}, Rx_{max}\}$, 當 $x'(N) > \min\{Bx_{max}, Rx_{max}\}$, 目標函數有最小值。

3.2 部分交疊連續性之最佳化分析

在前一章節, 我們知道列上 N_r 標準單元滿足交疊連續性的性質, 合法化後, 所有的單元緊鄰且沿 x -方向的順序不變有最小的擾動量。此章節主要討論, 若列上的 N_r 的標準單元不滿足交疊連續性時, 該如何找到最佳解?

延用交疊連續性的概念, 先對 N_r 個已排序的標準單元根據交疊連續性分組, 分別對各組合法最佳化後, 若組與組之間有面積重疊, 將其合併處理, 重複此步驟直到完全合法。在詳細說明及解決此想法前, 我們先定義符號, 見圖表3.1., 並且闡釋群集(Cluster), 群集簇(a set of Cluster)的定義及引理。

細部來看, 群集是由許多的標準單元所組成, 也可看成是許多有“關聯性”的群集所組成。事實上, 此關聯性就是沿用上一章節的交疊連續性, 定義如下。

定義 2 (群集簇). 群集簇, C^i , 為群組的集合 $C^i = \{C_1^i, C_2^i, \dots, C_{N_c^i}^i\}$ 且 $C_m^i \cap C_n^i = \emptyset$ 對所有的 $1 \leq m < n \leq N_c^i$ 。

定義 3 (群集). 群集 \mathcal{C}_k^i 由第 i 次合法化的位置為依據, 若有

$$X_c^i(\mathcal{C}_j^{(i-1)}) - X_c^i(\mathcal{C}_{j-1}^{(i-1)}) \leq W_c(\mathcal{C}_{j-1}^{(i-1)}), \quad J_s + 1 \leq j \leq J_t$$

則將 \mathcal{C}^{i-1} 中的第 J_s 個群集到第 J_t 個群集群組在一起。也就是

$$\mathcal{C}_k^i = \left\{ \mathcal{C}_{J_s}^{(i-1)}, \mathcal{C}_{J_s+1}^{(i-1)}, \dots, \mathcal{C}_{J_t}^{(i-1)} \right\}$$

特例, 經過第 i 次合法化後, 群集不再與其他群集重疊, 此時群集自己組成一個群集, 有

$$N_{c,k}^i = 1 \text{ 且 } X_c^{i+1}(\mathcal{C}_k^i) = X_c^i(\mathcal{C}_k^i)$$

簡單來說群集 \mathcal{C}_k^i 就是根據群組簇, \mathcal{C}^{i-1} , 做完第 i 次合法化的位置重新分組。

引理 1. 給一個群集, \mathcal{C}_k^i , 由 $N_{c,k}^i$ 個群集組成, 有

$$X_c^i(\mathcal{C}_j^{(i-1)}) - X_c^{i'}(\mathcal{C}_{j-1}^{(i-1)}) \leq W_c(\mathcal{C}_{j-1}^{(i-1)}), \quad J_s + 1 \leq j \leq J_t$$

經過合法化後, 滿足

$$X_c^{i+1}(\mathcal{C}_j^{(i-1)}) - x_c^{i+1}(\mathcal{C}_{j-1}^{(i-1)}) \geq W_c(\mathcal{C}_{j-1}^{(i-1)}), \quad J_s + 1 \leq j \leq J_t$$

假設第 J_c 個群集, 相較於第 i 次合法化位置, 第一個往右平移的群集, 也就是,

$$X_c^{i+1}(\mathcal{C}_{J_c}^{(i-1)}) \geq X_c^i(\mathcal{C}_{J_c}^{(i-1)})$$

因此, 合法化後, 對所有的群集 \mathcal{C}_j^{i-1} , $j \geq J_c$ 其合法化位置必向右平移, $X_c^{i+1}(\mathcal{C}_j^{i-1}) \geq X_c^i(\mathcal{C}_j^{i-1})$, $j \geq J_c$. 同理, 對所有的群集 \mathcal{C}_j^{i-1} , $j < J_c$ 有 $X_c^{i+1}(\mathcal{C}_j^{i-1}) < X_c^i(\mathcal{C}_j^{i-1})$ 。

Proof. 對 $\mathcal{C}_{J_c+1}^{(i-1)}$, 經過第 i 此合法化後, 與其他群集有

$$X_c^i(\mathcal{C}_{J_c+1}^{(i-1)}) - X_c^i(\mathcal{C}_{J_c}^{(i-1)}) \leq W_c(\mathcal{C}_{J_c}^{(i-1)}) \quad (3.11)$$

因此, 第 $i+1$ 次合法化後, 其必滿足

$$X_c^{i+1}(\mathcal{C}_{J_c+1}^{(i-1)}) - X_c^{i+1}(\mathcal{C}_{J_c}^{(i-1)}) \geq W_c(\mathcal{C}_{J_c}^{(i-1)}) \quad (3.12)$$

比較(3.12)式和(3.11)式, 我們有

$$X_c^{i+1}(\mathcal{C}_{J_c+1}^{(i-1)}) - x_c^{i+1}(\mathcal{C}_{J_c}^{(i-1)}) \geq X_c^i(\mathcal{C}_{J_c+1}^{(i-1)}) - X_c^i(\mathcal{C}_{J_c}^{(i-1)}) \quad (3.13)$$

對(3.13)式移項, 可得

$$X_c^{i+1}(\mathcal{C}_{J+1}^{(i-1)}) - X_c^i(\mathcal{C}_{J+1}^{(i-1)}) \geq X_c^{i+1}(\mathcal{C}_{J_c}^{(i-1)}) - X_c^i(\mathcal{C}_{J_c}^{(i-1)}) \quad (3.14)$$

由於 $X_c^{i+1}(\mathcal{C}_{J_c}^{(i-1)}) \geq X_c^i(\mathcal{C}_{J_c}^{(i-1)})$, 因此(3.14)式意味著

$$X_c^{i+1}(\mathcal{C}_{J_c+1}^{(i-1)}) \geq X_c^i(\mathcal{C}_{J_c+1}^{(i-1)}). \quad (3.15)$$

重複相同的步驟可得, 對所有的 $k > J_c$, 有 $X_c^{i+1}(\mathcal{C}_k^{(i-1)}) \geq X_c^i(\mathcal{C}_k^{(i-1)})$ 同理, 我們可得, 對所有的 $k < J_c$, 有 $X_c^{i+1}(\mathcal{C}_k^{(i-1)}) < X_c^i(\mathcal{C}_k^{(i-1)})$ □

3.2.1 問題公式化

給定列上的 N_r 個已排序的標準單元, 編號從 1 到 N_r , 及每個標準單元的全域位置, 合法化後, 在滿足標準單元順序不變的條件下, 使得目標函數為最小值。目標函數可寫成

$$\min. \sum_{l=1}^N |x^i(l) - x^0(l)| \quad (3.16)$$

$$\text{s.t. } x^i(l) - x^i(l-1) \geq w(l-1), \quad 2 \leq l \leq N \quad (3.17)$$

其中 $x^0(l)$ 與 $x^i(l)$ 分別為標準單元 l 的全域擺置和第 i 次合法化的左下角 x -座標位, $w(l)$ 是標準單元 l 的寬度。

3.2.2 解最佳位置

由定義3., 我們可將 N_r 個標準單元可以被分成 N_c^0 個群集。對每一個群集 \mathcal{C}_k^0 , $k = 1, \dots, N_c^0$, 有 N_k^0 個群集組成, 編號從 $J_k + 1$ 到 $J_k + N_{c,k}^0$ 。

已知

$$x^0(j) - x^0(j-1) \leq w(j), \quad J_k + 2 \leq j \leq J_k + N_{c,k}^0 \quad (3.18)$$

此外, 群組的最後一個群集與後一個群組的第一個群集之間沒有交疊, 也就是

$$x^0(J_k + 1) - x^0(J_{k-1} + N_{c,k-1}^0) \geq w(J_{k-1} + N_{c,k-1}^0), \quad k = 2, \dots, N_c^0 \quad (3.19)$$

其中 $J_k = \sum_{n=1}^{k-1} N_{c,n}^0$

由於每一個群集 \mathcal{C}_k^0 是由全域擺置的結果群組而成的, 此時, 內部的標準單元尚符合前一章節所提的交疊續性, 因此可直接使用上一章節所提及的方法, 可求得群集的第一次合法化的最佳位置 $X_c^1(\mathcal{C}_k^0)$, 對所有的 $k = 1, \dots, N_c^0$ 。因此目標函數改寫成

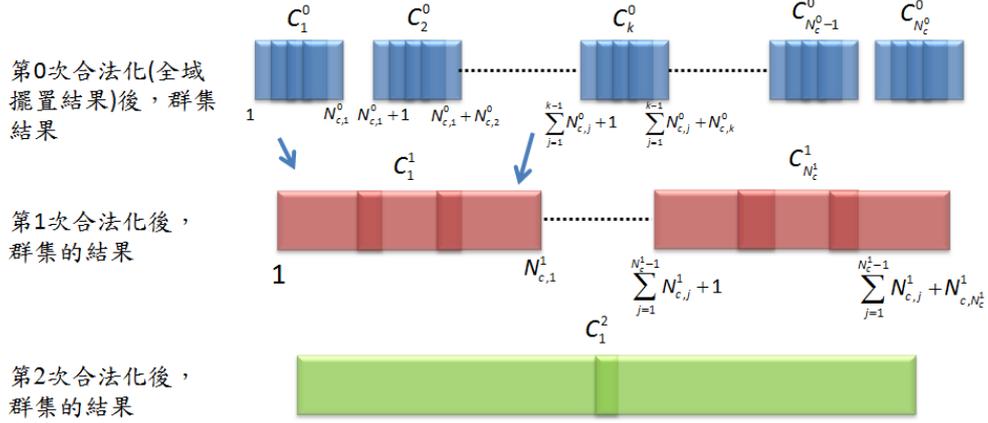


圖 3.3: 群集群組化過程的概念圖

$$\sum_{j=1}^{N_r} |x^1(j) - x^0(j)| = \sum_{j=1}^{J_1} |x^1(j) - x^0(j)| + \sum_{j=J_1+1}^{J_2} |x^1(j) - x^0(j)| + \cdots + \sum_{j=J_{N_c^0-1}+1}^{J_{N_c^0}} |x^1(j) - x^0(j)| \quad (3.20)$$

$$= \sum_{j=1}^{J_1} |x^1(1) - \Delta x_j| + \sum_{j=J_1+1}^{J_2} |x^1(J_1+1) - \Delta x_j| + \cdots + \sum_{j=J_{N_c^0-1}+1}^{J_{N_c^0}} |x^1(J_{N_c^0-1}+1) - \Delta x_j| \quad (3.21)$$

$$= \sum_{j=1}^{J_1} |X_c^1(C_1^0) - \Delta x_j| + \sum_{j=J_1+1}^{J_2} |X_c^1(C_2^0) - \Delta x_j| + \cdots + \sum_{j=J_{N_c^0-1}+1}^{J_{N_c^0}} |X_c^1(C_{N_c^0}^0) - \Delta x_j| \quad (3.22)$$

$$\triangleq f_1(X_c^1(C_1^0)) + f_2(X_c^1(C_2^0)) + \cdots + f_{N_c^0}(X_c^1(C_{N_c^0}^0)) \quad (3.23)$$

$$\text{where } J_k = \sum_{n=1}^{k-1} N_{c,n}^0$$

第一次合法化後的群集，若仍會有不合法的情形，則再做第二次的群集，因此 C^0 可再分成 N_c^1 個群集並且 $N_c^1 < N_c^0$ 。

特性2., 我們知道若 C_k^0 的位置越遠離 $X_c^1(C_k^0)$, 則成本(cost)越大。因此對群集 C_l^1 而言, $l = 1, \dots, N_c^1$ 目標函數可改寫成

$$\min. \sum_{k=L_l+1}^{L_l+N_{c,l}^1} |f_k(X_c^2(C_k^0)) - f_k(X_c^1(C_k^0))| \quad (3.24)$$

$$\text{s.t. } X_c^2(C_k^0) - X_c^2(C_{k-1}^0) \geq W_c(C_{k-1}^0), L_l + 2 \leq k \leq L_l + N_{c,l}^1 \quad (3.25)$$

其中 $L_l = \sum_{n=1}^{l-1} N_{c,n}^1$; 此時, $f_k(X_c^1(C_k^0))$ 已知, 且有 $N_{c,l}^1$ 個變數

對群集簇 C^1 目標函數可寫成

$$\min. \sum_{l=1}^{N_c^1} \left(\sum_{k=L_l+1}^{L_l+N_{c,l}^1} |f_k(X_c^2(C_k^0)) - f_k(X_c^1(C_k^0))| \right) \quad (3.26)$$

欲求(3.26)的最小值等價於求(3.24)的最小值。因此，第二次合法化後，假設存在 L_c 是第一個向右平移的群集，也就是， $X_c^2(\mathcal{C}_{L_c}^0) \geq X_c^1(\mathcal{C}_{L_c}^0)$ 。

- $\sum_{k=L_l+1}^{L_c-1} f_k(X_c^2(\mathcal{C}_k^0)) - f_k(X_c^1(\mathcal{C}_k^0)) : X_c^2(\mathcal{C}_k^0) \leq X_c^1(\mathcal{C}_k^0), L_l + 1 \leq k \leq L_c - 1$; 經過第 2 次合法化後，每一個群集的位置， $X_c^2(\mathcal{C}_k^0)$ ，可表示成

$$X_c^2(\mathcal{C}_{k+1}^0) = X_c^2(\mathcal{C}_k^0) + W_c(X_c^2(\mathcal{C}_k^0)) + \delta_k, L_l + 1 \leq k \leq L_c - 1$$

其中 δ_k 是兩群集 \mathcal{C}_k^0 和 \mathcal{C}_{k+1}^0 之間間隙，($\delta_k = X_c^2(\mathcal{C}_{k+1}^0) - X_c^2(\mathcal{C}_k^0) - W_c(\mathcal{C}_k^0)$)

$$\begin{cases} X_c^2(\mathcal{C}_{L_c-1}^0) = X_c^2(\mathcal{C}_{L_c-2}^0) + W_c(\mathcal{C}_{L_c-2}^0) + \delta_{L_c-2}, \\ X_c^2(\mathcal{C}_{L_c-2}^0) = X_c^2(\mathcal{C}_{L_c-3}^0) + W_c(\mathcal{C}_{L_c-3}^0) + \delta_{L_c-3}, \\ \vdots \\ X_c^2(\mathcal{C}_{k+1}^0) = X_c^2(\mathcal{C}_k^0) + W_c(\mathcal{C}_k^0) + \delta_k \end{cases}$$

將上列式子相加總，可得到每一個 $X_c^2(\mathcal{C}_k^0)$ 皆可用 $X_c^2(\mathcal{C}_{L_c-1}^0)$ 表示之，

$$X_c^2(\mathcal{C}_k^0) = X_c^2(\mathcal{C}_{L_c-1}^0) - \sum_{j=l}^{L_c-1} W_c(\mathcal{C}_j^0) - \sum_{j=l}^{L_c-1} \delta_j$$

對所有的 $L_l + 1 \leq k \leq L_c - 1$ 而言，由於 $X_c^2(\mathcal{C}_k^0) < X_c^1(\mathcal{C}_k^0)$ 且 f_k 是單調遞增，欲求 $\sum_{k=L_l+1}^{L_c-1} f_k(X_c^1(\mathcal{C}_k^0)) - f_k(X_c^2(\mathcal{C}_k^0))$ 的最小值等價於 $X_c^2(\mathcal{C}_{L_c-1}^0)$ 愈最大化使得 $X_c^2(\mathcal{C}_k^0)$ 越接近 $X_c^1(\mathcal{C}_k^0)$ 並且同時將 $\sum_{j=l}^{L_c-1} \delta_j$ 最小化，也就是令每一項的 δ_j 等於零。換句話說，

$$X_c^2(\mathcal{C}_k^0) - X_c^2(\mathcal{C}_{k-1}^0) = W_c(\mathcal{C}_{k-1}^0) \quad (3.27)$$

- $\sum_{k=L_c}^{L_l+N_{c,l}^1} f_k(X_c^2(\mathcal{C}_k^0)) - f_k(X_c^1(\mathcal{C}_k^0)) : X_c^2(\mathcal{C}_k^0) \geq X_c^1(\mathcal{C}_k^0), L_c \leq k \leq L_l + N_{c,l}^1$ 。每一項 $X_c^2(\mathcal{C}_k^0)$ 可表示成

$$X_c^2(\mathcal{C}_k^0) = X_c^2(\mathcal{C}_{k-1}^0) + W_c(\mathcal{C}_{k-1}^0) + \delta_{k-1}, L_c \leq k \leq L_l + N_{c,l}^1$$

其中 δ_{k-1} 是 \mathcal{C}_k^0 和 \mathcal{C}_{k-1}^0 之間間隙 ($\delta_{k-1} = X_c^2(\mathcal{C}_k^0) - X_c^2(\mathcal{C}_{k-1}^0) - W_c(\mathcal{C}_{k-1}^0)$)。

因此，

$$\begin{cases} X_c^2(\mathcal{C}_{L_c+1}^0) = X_c^2(\mathcal{C}_{L_c}^0) + W_c(\mathcal{C}_{L_c}^0) + \delta_{L_c}, \\ X_c^2(\mathcal{C}_{L_c+2}^0) = X_c^2(\mathcal{C}_{L_c+1}^0) + W_c(\mathcal{C}_{L_c+1}^0) + \delta_{L_c+1}, \\ \vdots \\ X_c^2(\mathcal{C}_k^0) = X_c^2(\mathcal{C}_{k-1}^0) + W_c(\mathcal{C}_{k-1}^0) + \delta_{k-1} \end{cases}$$

將上列式子相加總, 考得每一 $X_c^2(\mathcal{C}_k^0)$ 都可用 $X_c^2(\mathcal{C}_{L_c}^0)$, 來表示之,

$$X_c^2(\mathcal{C}_k^0) = X_c^2(\mathcal{C}_{L_c}^0) + \sum_{j=L_c}^{L_l+N_{c,l}^1} W_c(\mathcal{C}_j^0) + \sum_{j=L_c}^{L_l+N_{c,l}^1} \delta_j$$

對所有的 $L_c \leq k \leq L_l + N_{c,l}^1$ 而言, 由於 $X_c^2(\mathcal{C}_k^0) \geq X_c^1(\mathcal{C}_k^0)$ 且 f_k 是單調遞增; 因此, 欲求 $\sum_{k=L_c}^{L_l+N_{c,l}^1} f_k(X_c^2(\mathcal{C}_k^0)) - f_k(X_c^1(\mathcal{C}_k^0))$ 的最小值等價於 $X_c^2(\mathcal{C}_{L_c-1}^0)$ 愈小使得 $X_c^2(\mathcal{C}_k^0)$ 越接近 $X_c^1(\mathcal{C}_k^0)$ 並且同時將 $\sum_{j=L_c}^{L_l+N_{c,l}^1} \delta_j$ 最小化, 也就是令每一項的 δ_j 等於零。換句話說,

$$X_c^2(\mathcal{C}_k^0) - X_c^2(\mathcal{C}_{k-1}^0) = W_c(\mathcal{C}_j^0), \quad L_c \leq k \leq L_l + N_{c,l}^1 \quad (3.28)$$

- $\sum_{k=L_c}^{L_l+N_{c,l}^1} f_k(X_c^2(\mathcal{C}_k^0)) - f_k(X_c^1(\mathcal{C}_k^0))$ 的最小化等同於將 $X_c^2(\mathcal{C}_{L_c}^0)$ 最小並且同時 $X_c(\mathcal{C}_{L_c-1}^0)$ 最大。經過第二次合法化後, $X_c^2(\mathcal{C}_{L_c}^0)$ 和 $X_c^2(\mathcal{C}_{L_c-1}^0)$ 受限於

$$X_c^2(\mathcal{C}_{L_c}^0) - X_c^2(\mathcal{C}_{L_c-1}^0) \geq W_c(\mathcal{C}_{L_c}^0)$$

的條件, 因此

$$X_c^2(\mathcal{C}_{L_c}^0) = X_c^2(\mathcal{C}_{L_c-1}^0) + W_c(\mathcal{C}_{L_c-1}^0) + \delta_{\mathcal{C}_{L_c-1}^0}$$

又基於 $\sum_{k=L_l+1}^{L_l+N_{c,l}^1} f_k(X_c^2(\mathcal{C}_k^0)) - f_k(X_c^1(\mathcal{C}_k^0))$ 等同於最小化 $x_c^2(\mathcal{C}_{L_c}^0)$, 可得到

$$X_c^2(\mathcal{C}_{L_c}^0) - X_c^2(\mathcal{C}_{L_c-1}^0) = W_c(\mathcal{C}_{L_c-1}^0) \quad (3.29)$$

因此, 由(3.27),(3.28)和(3.29), 可得結論

$$X_c^2(\mathcal{C}_k^0) - X_c^2(\mathcal{C}_{k-1}^0) = W_c(\mathcal{C}_k^0) \quad (3.30)$$

由(3.30)式, 對所有的 $1 \leq l \leq N_c^1$, 群集 C_l^1 的成本函數可重新表示成

$$\sum_{j=1}^{N_r} |x^2(j) - x^0(j)| = \cdots + \underbrace{\sum_{j=J_n+1}^{J_n+N_{c,n}^0} |x^2(j) - x^0(j)| + \cdots + \sum_{j=J_{(n+N_{c,l}^1)+1}}^{J_{(n+N_{c,l}^1)}+N_{c,(n+N_{c,l}^1)}^0} |x^2(j) - x^0(j)| + \cdots}_{\text{有 } N_{c,l}^1 \text{ 項}} + \cdots \quad (3.31)$$

$$= \cdots + \sum_{j=J_n+1}^{J_n+N_{c,n}^0} |x^2(J_n+1) + \sum_{i=J_n+1}^{j-1} w(i) - x^0(j)| \quad (3.32)$$

$$+ \cdots + \sum_{j=J_{(n+N_{c,l}^1)+1}}^{J_{(n+N_{c,l}^1)}+N_{c,(n+N_{c,l}^1)}^0} |x^2(J_{(n+N_{c,l}^1)}+1) + \sum_{i=J_{(n+N_{c,l}^1)+1}}^{j-1} w(i) - x^0(j)| + \cdots \quad (3.33)$$

$$= \cdots + \sum_{j=J_n+1}^{J_n+N_{c,n}^0} |X_c^2(C_{L_l+1}^0) + \sum_{i=J_n+1}^{j-1} w(i) - x^0(j)| \quad (3.34)$$

$$+ \cdots + \sum_{j=J_{(n+N_{c,l}^1)+1}}^{J_{(n+N_{c,l}^1)}+N_{c,(n+N_{c,l}^1)}^0} |X_c^2(C_{L_l+N_{c,l}^1}^0) + \sum_{i=J_{(n+N_{c,l}^1)+1}}^{j-1} w(i) - x^0(j)| + \cdots \quad (3.35)$$

$$= \cdots + \sum_{j=J_n+1}^{J_n+N_{c,n}^0} |X_c^2(C_{L_l+1}^0) + \underbrace{\sum_{i=J_n+1}^{j-1} w(i)}_{\Delta x'_j} - x^0(j)| \quad (3.36)$$

$$+ \cdots + \sum_{j=J_{(n+N_{c,l}^1)+1}}^{J_{(n+N_{c,l}^1)}+N_{c,(n+N_{c,l}^1)}^0} |X_c^2(C_{L_l+1}^0) + \underbrace{\sum_{i=L_l+1}^{L_l+N_{c,l}^1-1} W_c(C_i^0) + \sum_{i=J_{(n+N_{c,l}^1)+1}}^{j-1} w(i)}_{\Delta x'_j} - x^0(j)| + \cdots \quad (3.37)$$

$$= \cdots + \sum_{j=J_n+1}^{J_{(n+N_{c,l}^1)}+N_{c,(n+N_{c,l}^1)}^0} |X_c^2(C_{L_l+1}^0) - \Delta x'_j| + \cdots \quad (3.38)$$

$$\text{其中 } J_n = \sum_{j=1}^{n-1} N_{c,j}^0, \quad L_l = \sum_{j=1}^{l-1} N_{c,j}^1$$

由上式可知, 對所有的 $1 \leq l \leq N_c^1$, 群集 C_l^1 的成本函數(3.38)式, 只有一個未知數 $X_c^2(C_{L_l+1}^0)$ 且數列 $\{\Delta x'_j\}$ 已知, 因此根據定理2., 當 $X_c^2(C_{L_l+1}^0)$ 等於數列 $\{\Delta x'_j\}$ 的中位數時, 目標函數有最小值。

然而, 對所有的 $1 \leq l \leq N_c^1$, 群集 C_l^1 合法化後, 並找到其最佳解($X_c^2(C_{L_l+1}^0)$), 群集之間仍可能出現面積重疊(不合法), 此時重複上述步驟, 直到群集簇 C^{i+1} 的個數等於群集簇 C^i 的個數 ($N_c^{i+1} = N_c^i$), 表示群集間不再有重疊面積, 完成合法化。

3.3 改善 *HiBinLegalizer* 尋找中位數的過程

前一節，我們證明利用 *HiBinLegalizer* 所提出的最佳位置解法器，所找到的最佳位置解，無論全域擺置的位置是否具備交疊連續性的性質，一旦，群集或標準單元與已合法化的群集面積交疊，合法化後，在群集內所有的標準單元，其彼此間緊鄰，此時，每一個標準單元 k 的合法化位置 $x^i(k)$ ，皆可由第一個標準單元的合法化位置 $x^i(1)$ 與左方所有的標準單元寬度總和， $\sum_{j=1}^{k-1} w(j)$ 表示之。再利用最佳位置解法器，求得 $x^i(1)$ ；當 $x^i(1)$ 是數列 $\{\Delta x_k\}_{k=1}^{k=N}$ 的中位數，目標函數為最小；其中 N 是有連續面積交疊群集的總標準單元個數， $\Delta x_k \triangleq x^0(k) - \sum_{j=1}^{k-1} w(j)$ ， $x^0(k)$ 是標準單元 k 的全域擺置的左下 x -座標。然而，*HiBinLegalizer* 演算法的試擺列的過程，需不斷的試驗列，對列上的群集尋找中位數。在此章節，我們提出並且證明數個引理，以簡化試驗列尋找中位數的過程。

中位數(median)，對 N 個已排序的數而言，位於最中間的數即為中位數，記為 Q_m 。因此，當 N 是奇數時，位於最中間的數恰有一個；然而，當 N 是偶數時，位於最中間的數有兩個，較小的數稱之為下界中位數(lower median)和另一則稱上界(upper median)[16]，分別記為 Q_L 和 Q_U 。一般而言，當 N 是偶數時，中位數會取下界中位數和上界中位數的算數平均數 ($Q_m = \frac{Q_L + Q_U}{2}$)。觀察可發現，中位數對 N 個數而言，恰好大於二分之一個數且小於二分之一個數；因此，當 N 是奇數時，中位數大於且小於 $[\frac{N}{2}]$ 個數；當 N 是偶數時，下界中位數小於 $\frac{N}{2}$ 且大於等於 $\frac{N}{2}$ 個數，上界中位數大於 $\frac{N}{2}$ 且小於等於 $\frac{N}{2}$ 個數。

就此篇所定義的目標函數而言，若群集內的標準單元個數是偶數時，不論中位數的選其上界中位數或下界中位數，皆不影響目標函數的成本；若選擇下界中位數表示希望群集盡可能往左擺置，以減少新插入的標準單元與群集面積重疊的發生，然而，愈往左擺則會增加與已擺置的群集面積重疊的發生機會，因此，我們取

$$Q_m = \alpha Q_L + (1 - \alpha) Q_U, \alpha \in [0, 1]$$

其中 α 是介於 0 到 1 的隨機常數。

在說明簡化尋找中位數的引理前，我們先證明，若新加入標準單元與已擺置的群集面積重疊，則合法化後，群集的合法化位置會往左移。

引理 2. 已知群集 C^{last} ，有 $N_{C^{last}}$ 個標準單元，與最佳位置 $q_{C^{last}}$ 。若插入一個標準單元 i 與群集有面積重疊，則經合法化後， C^{last} 的最佳合法位置 $q'_{C^{last}}$ ，會向左平移，

$$q'_{C^{last}} \leq q_{C^{last}} \quad (3.39)$$

Proof. 假設 $N_{c^{last}}$ 是偶數, $q_{c^{last}} = \alpha Q_L + (1 - \alpha)Q_H$, 其中 Q_L 和 Q_H 分別是數列 $P = \{\Delta x_k\}_{k=1}^{N_{c^{last}}}$ 的下界與上界中位數, α 是介於 0 到 1 的常數。由於標準單元 i 與 C^{last} 面積重疊, 有

$$q_{c^{last}} + \sum_{j=1}^{N_{c^{last}}} w(j) > x(i) \quad (3.40)$$

合法化後,

$$x'(i) = q'_{c^{last}} + \sum_{j=1}^{N_{c^{last}}} w(j) \quad (3.41)$$

由於 $q'_{c^{last}}$ 是 $\{\Delta x_k\}_{k=1}^{N_{c^{last}}} \cap \{\Delta x_i\}_{i=1}$ 的中位數, 故 $q'_{c^{last}}$ 大於並小於 $\lfloor \frac{N_{c^{last}}+1}{2} \rfloor = \frac{N_{c^{last}}}{2}$ 個數, 其中 $\Delta x_i = x(i) - \sum_{j=1}^{N_{c^{last}}} w(j)$ 。

從 (3.40) 式, $q_{c^{last}} > \Delta x_i$, 考慮兩個極端值 $\alpha = 1$ 和 $\alpha = 0$

$$\Rightarrow \begin{cases} \text{當 } \alpha = 1, & \Delta x_i < Q_L < Q_H \\ \text{當 } \alpha = 0, & \Delta x_i < Q_H \end{cases} \quad (3.42)$$

可推得 Q_H 大於 $(\frac{N_{c^{last}}}{2} + 1)$ 個數, 因此 $q'_{c^{last}}$ 必小於 Q_H 。我們已知, 扣除 Δx_i , Q_L 大於 $\frac{N_{c^{last}}}{2} - 1$ 個數並且小於 $\frac{N_{c^{last}}}{2}$ 。倘若, 我們假設 $\Delta x_i < Q_L$, 則 Q_L 恰大於並小於 $\frac{N_{c^{last}}}{2}$ 個數; 反之, $\Delta x_i > Q_L$, 則 $Q_H > \Delta x_i > Q_L$, Δx_i 恰大於並小於 $\frac{N_{c^{last}}}{2}$ 個數。整理可得

$$q'_{c^{last}} = \max\{Q_L, \Delta x_i\} \quad (3.43)$$

又

$$Q_L \leq q_{c^{last}} \leq Q_H \quad (3.44)$$

因此, 結合(3.43)(3.44)兩式, 得 $q'_{c^{last}} \leq q_{c^{last}}$ 得證。

假設 $N_{c^{last}}$ 是奇數, $q_{c^{last}}$ 是數列 $P = \{\Delta x_k\}_{k=1}^{N_{c^{last}}}$ 的中位數, 大於並小於 $\lfloor \frac{N_{c^{last}}}{2} \rfloor$ 個數; 加入標準單元 i 後, 同(3.40)式, 得 $\Delta x_i < q_{c^{last}}$, 此時, $q_{c^{last}}$ 恰大於 $\frac{N_{c^{last}}+1}{2}$ 個數且小於 $\frac{N_{c^{last}}-1}{2}$ 個數, 為上界中位數, 記為 Q'_H 。尋找下界中位數, Q'_L 前, 假設已知 Δx_{m-1} , $\Delta x_{m-1} \in P$, 其大於 $\lfloor \frac{N_{c^{last}}}{2} \rfloor - 1$ 且小於 $\lfloor \frac{N_{c^{last}}}{2} \rfloor + 1$ 個數。今加入標準單元 i 後, 若 $\Delta x_i < \Delta x_{m-1}$, 則 Δx_{m-1} 大於 $\lfloor \frac{N_{c^{last}}}{2} \rfloor$ 且小於 $\lfloor \frac{N_{c^{last}}}{2} \rfloor + 1$ 個數, $Q'_L = \Delta x_{m-1}$; 反之, 若 $\Delta x_i < \Delta x_{m-1}$, 則 $Q'_L = \Delta x_i$, 整理可得,

$$Q'_H = q_{c^{last}} \quad (3.45)$$

$$Q'_L = \max\{\Delta x_i, \Delta x_{m-1}\} \quad (3.46)$$

又 $q'_{c^{last}}$ 介於 Q'_L 和 Q'_H 之間, 因此可得 $q'_{c^{last}} \leq q_{c^{last}}$, 得證。 \square

C^{last}	列上最右邊的群集
$N_{C^{last}}$	群集 C^{last} 中, 標準單元的個數
$\{\Delta x_k\}_{k=1}^{N_{C^{last}}}$	有 $N_{C^{last}}$ 項的數列, 編號從 1 到 $N_{C^{last}}$
$P_{C^{last}}$	對 C^{last} , 其可能位置解的解集合
$q_{C^{last}}$	群集 C^{last} 的合法化位置
$W_{C^{last}}$	群集 C^{last} 的寬度
C^p	前置群集
N_{C^p}	前置群集內標準單元的個數
W_{C^p}	群集 C^p 的寬度
P_{C^p}	前置群集的解集合
$\Delta x_U^p, \Delta x_L^p$	當 N_{C^p} 為偶數時, 前置群集的上界與下界中位數
$\Delta x_U^p, \Delta x_L^p$	當前置群集合併後的標準單元個數是偶數, 其分別代表為上界與下界中位數
C^k	第 k 個群集
q_k	群集 C^k 的合法化位置
W_{C^k}	群集 C^k 的寬度

表 3.2: 此小節的符號說明

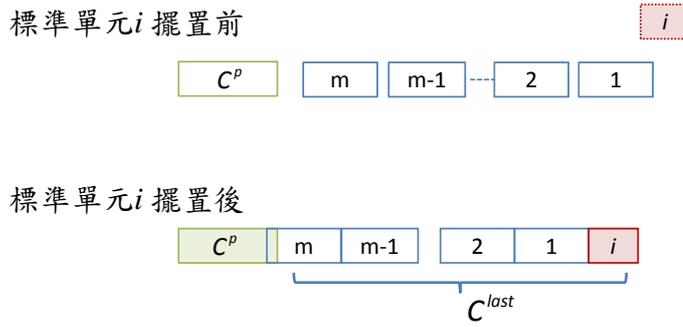


圖 3.4: 將 C^{last} 分解成 m 個群集與一個標準單元 i

由引理 2 知, 若標準單元 i 擺入後與列上最右邊的群集, C^{last} , 有

$$x(i) - q_{C^{last}} \leq W_{C^{last}} \quad (3.47)$$

合法化後, 標準單元會併入群集且群集會向左平移, ($q'_c \leq q_c$), 可能會與前置群集, C^p , 面積重疊, 並且在合法化後又與前置群集面積重疊, 我們稱此效應為骨牌效應(domino effect)。

接下來, 我們將提出四個引理, 說明標準單元 1 觸發骨牌效應後, 何時會停止以及說明如何簡化試驗列找尋中位數。我們先列舉出符號的定義, 以方便說明與證明引理, 參見表格 3.3

引理 3. 若 N_{C^p} 是偶數且 $N_{C^{last}}$ 是奇數, 合併後, 更新的位置為

$$q'_{C^p} = \max\{\Delta x_L^p, \Delta q_{C^{last}}\} \quad (3.48)$$

Proof. 假設 C^{last} 由 m 個群集和標準單元 i 所構成, 如圖3.4.所示; 解法器的解集合數列分解成兩數列 $P_{c^{last}} = \{\Delta x_k\}_{k=1}^{N_{c^{last}}-1} \cup \{\Delta x_i\}$, 在標準單元 i 觸發前, 群集間沒有面積重疊

$$q_i - q_{i-1} \geq W_{c^i}, i = 2, \dots, m \quad (3.49)$$

以及第 m 個群集與前置群集亦沒有面積重疊,

$$q_m - q_{c^p} \geq W_{c^p} \quad (3.50)$$

令 q_H 是數列 $\{\Delta x_k\}_{k=1}^{N_{c^{last}}-1}$ 的上界中位數, 其大於 $\frac{N_{c^{last}}-1}{2}$ 且小於 $\frac{N_{c^{last}}-1}{2} - 1$ 個數。

由(3.49)(3.50)兩式, 得

$$q_H - q_{c^p} \geq W_{c^p} \quad (3.51)$$

標準單元觸發後, 姑且不論如何求得 $q_{c^{last}}$, 我們知道 $q_{c^{last}} - q_{c^p} \leq W_{c^p}$ 且 $q_{c^{last}}$ 大於並小於 $\frac{N_{c^{last}}-1}{2}$ 個數。此時, 更新 $P'_{c^{last}} = \{\Delta x_k - W_{c^p}\}_{k=1}^{N_{c^{last}}}$ (註: 此舉僅將數列向左平移, 其項之間的相關位置不變)。

另一方面, 已知 q_{c^p} 介於 Δx_L^p 和 Δx_H^p 之間, 有 $q_{c^{last}} - \Delta x_H^p \leq W_{c^p}$ 和 $q_H - \Delta x_L^p \geq W_{c^p}$, 整理得

$$q_{c^{last}} - W_{c^p} \leq \Delta x_H^p \quad (3.52)$$

$$\Delta x_L^p \leq q_H - W_{c^p} \quad (3.53)$$

$$q_{c^{last}} - W_{c^p} \leq q_H - W_{c^p} \quad (3.54)$$

今假設, $\Delta x_L^p \leq q_{c^{last}} - W_{c^p}$, 由(3.52)式可知, $q_{c^{last}} - W_{c^p}$ 對 P_{c^p} 而言, 大於且小於 $\frac{N_{c^p}}{2}$ 個項; 另一方面, 對 $P_{c^{last}}$ 而言, 大於且小於 $\frac{N_{c^{last}}-1}{2}$ 個數。再由中位數的定義, $q'_{c^p} = q_{c^{last}}$ 。反之, 若 $\Delta x_L^p \geq q_{c^{last}} - W_{c^p}$, 由(3.53)式可得, 對 $P'_{c^{last}}$ 而言, 大於 $\frac{N_{c^{last}}-1}{2} + 1$ 和小於 $\frac{N_{c^{last}}-1}{2}$ 個項; 對 P_{c^p} 而言, 大於 $\frac{N_{c^p}}{2} - 1$, 小於 $\frac{N_{c^p}}{2}$, 此時 $q'_{c^p} = \Delta x_L^p$ 。總結以上得

$$q'_{c^p} = \max\{\Delta x_L^p, q_{c^{last}} - W_{c^p}\} \quad (3.55)$$

$$q'_{c^p} \leq q_{c^p} \quad (3.56)$$

□

引理 4. 若 N_{c^p} 是奇數且 $N_{c^{last}}$ 是奇數, 更新的位置為

$$q'_{c^p} = \alpha' \max\{\Delta x_{m-1}^p, \Delta q_{c^{last}}\} + (1 - \alpha')q_{c^p} \quad (3.57)$$

Proof. 同上一引理證明, 我們推得

$$q_{c^{last}} - W_{c^p} \leq q_{c^p} \leq q_H - W_{c^p} \quad (3.58)$$

對 $P'_{c^{last}}$ 而言, q_{c^p} 大於 $\frac{N_{c^{last}}-1}{2} + 1$, 小於 $\frac{N_{c^{last}}-1}{2}$ 個項; 對 P_{c^p} 而言, q_{c^p} 大於且小於 $\frac{N_{c^p}-1}{2}$ 個項, 因此 q_{c^p} 是前置群集更新位置後的上界中位數, 共大於 $\frac{N_{c^{last}}+N_{c^p}}{2}$, 小於 $\frac{N_{c^{last}}+N_{c^p}}{2} - 1$ 個項。對 P_{c^p} 而言, 令 Δx_{m-1}^p 大於 $\frac{N_{c^p}-1}{2} - 1$ 個項, 小於 $\frac{N_{c^p}-1}{2} + 1$ 個項 $\Delta x_{m-1}^p \leq q_{c^p}$ 。若 $q_{c^{last}} - W_{c^p} \leq \Delta x_{m-1}^p$, 則由(3.58)推得

$$q_{c^{last}} - W_{c^p} \leq \Delta x_{m-1}^p \leq q_H - W_{c^p} \quad (3.59)$$

因此, 可知對 $P'_{c^{last}}$ 而言, Δx_{m-1}^p 大於 $\frac{N_{c^{last}}-1}{2} + 1$, 小於 $\frac{N_{c^{last}}-1}{2}$ 個項。得 Δx_{m-1}^p 是下界中位數, 共大於 $\frac{N_{c^{last}}+N_{c^p}}{2} - 1$, 小於 $\frac{N_{c^{last}}+N_{c^p}}{2}$ 個項。反之, 若 $q_{c^{last}} - W_{c^p} \geq \Delta x_{m-1}^p$, 對 P_{c^p} 而言, $q_{c^{last}} - W_{c^p}$ 大於 $\frac{N_{c^p}-1}{2}$ 個項, 小於 $\frac{N_{c^p}-1}{2} + 1$ 個項。對 $P'_{c^{last}}$ 而言, $q_{c^{last}} - W_{c^p}$ 大於 $\frac{N_{c^{last}}-1}{2}$ 個項, 小於 $\frac{N_{c^{last}}-1}{2} + 1$ 個項。得 $q_{c^{last}} - W_{c^p}$ 是下界中位數, 共大於 $\frac{N_{c^{last}}+N_{c^p}}{2} - 1$, 小於 $\frac{N_{c^{last}}+N_{c^p}}{2}$ 個項。

因此, 更新後的前置群集的位置為

$$q'_{c^p} = \alpha' \max\{\Delta x_{m-1}^p, q_{c^{last}} - W_{c^p}\} + (1 - \alpha')q_{c^p} \quad (3.60)$$

□

引理 5. 若 N_{c^p} 和 $N_{c^{last}}$ 是皆是偶數時, 更新後的位置為

$$q'_{c^p} = \alpha' \max\{\Delta x_L^p, \Delta x_L^c\} + (1 - \alpha') \min\{\Delta x_U^p, \Delta x_U^c\}, \alpha' \in [0, 1] \quad (3.61)$$

Proof. 同引理3., 我們將 C^{last} 分成 m 個群集和一個標準單元 i ; 解法器的解集合數列 $P_{c^{last}}$ 亦分解成兩數列 $\{\Delta x_k\}_{k=1}^{N_{c^{last}}}$ 和 $\{\Delta x_i\}$ 。令 q_T 是數列 $\{\Delta x_k\}_{k=1}^{N_{c^{last}}}$ 的中位數, 由於在標準單元 i 觸發前, 群集間沒有面積重疊, 且第 m 個群集與前置群集亦沒有面積重疊, 重寫 (3.49) 式與(3.50)式

$$q_m = q_{c^p} + W_{c^p} + d_{c^p}$$

$$q_{m-1} = q_m + W_{c^m} + d_{m-1}$$

$$q_{m-2} = q_{m-1} + W_{c^{m-1}} + d_{m-1}$$

⋮

$$q_1 = q_2 + W_{c^2} + d_2$$

整理得,

$$q_k = q_{c^p} + \sum_{j=k+1}^m W_{c^j} + \sum_{j=k+1}^m d^j + d_{c^p}, k = m-1, \dots, 1$$

由上式可知, 若 N_{c^k} 是奇數則至少有 $\lfloor \frac{N_{c^k}}{2} \rfloor + 1$ 的項數大於 q_{c^p} , 若 N_{c^k} 是偶數則至少有 $\frac{N_{c^k}}{2}$ 的項數大於 q_{c^p} , 因此 q_T 必大於 q_{c^p} 。假設 $q_{c^{last}}$ 由 Δx_L^{last} 和 Δx_H^{last} 組成, 當標準單元 i 觸發後, $q_{c^{last}} - q_{c^p} \leq W_{c^p}$, 意味著, $\Delta x_L^{last} - W_{c^p} \leq q_{c^p} \leq \Delta x_H^{last}$, 若 $\Delta x_H^{last} - W_{c^p} \leq \Delta x_H^p$, 則對 P_{c^p} 而言, $\Delta x_H^{last} - W_{c^p}$ 大於和小於 $\frac{N_{c^p}}{2}$ 個項, 因此 $\Delta x_H^p = \Delta x_H^{last} - W_{c^p}$ 是上界中位數, 共大於 $\frac{N_{c^p} + N_{c^{last}}}{2} + 1$, 小於 $\frac{N_{c^p} + N_{c^{last}}}{2}$ 。反之, 若 $\Delta x_H^{last} - W_{c^p} \geq \Delta x_H^p$, 則對 $P_{c^{last}}$ 而言, Δx_H^p 大於 $\frac{N_{c^{last}}}{2} + 1$, 小於 $\frac{N_{c^{last}}}{2}$ 個項, 因此 $\Delta x_H^p = \Delta x_H^{last}$ 是上界中位數, 共大於 $\frac{N_{c^p} + N_{c^{last}}}{2} + 1$, 小於 $\frac{N_{c^p} + N_{c^{last}}}{2}$ 。得 $\Delta x_H^p = \min\{\Delta x_H^{last} - W_{c^p}, \Delta x_H^p\}$; 同理, 可推得下界中位數, $\Delta x_L^p = \max\{\Delta x_L^{last} - W_{c^p}, \Delta x_L^p\}$ 。□

引理 6. 若 N_{c^p} 是奇數且 $N_{c^{last}}$ 是偶數, 則前置群集更新後的位置不變

$$q_{c^p}' = q_{c^p} \quad (3.62)$$

Proof. 由上一引理可知, $\Delta x_L^{last} - W_{c^p} \leq q_{c^p} \leq \Delta x_H^{last} - W_{c^p}$, 對 $P_{c^{last}}$ 而言, q_{c^p} 大於 $\frac{N_{c^{last}}}{2}$, 小於 $\frac{N_{c^{last}}}{2}$ 個項。因此 q_{c^p} 共大於且小於 $\lfloor \frac{N_{c^p} + N_{c^{last}}}{2} \rfloor$ 個項。我們可以直接得到

$$q_{c^p}' = q_{c^p}$$

□

3.4 網格數的精化

格分(Grid-based)演算法其優點在於降低執行時間; 對擺置合法化而言, 還可限制元件的最大移動量, 最大移動量的最小化意味著可最總移動量; 然而若網格數過多會迫使標準單元的移動量增加。另一個影響移動量的因素為網格的長寬比(Aspect Ratio)。在同樣地總網格數下, 改變單元格的長寬比, 總移動量亦有差異性。文獻[1]中, 所提及之網格的大小的求法, 僅與晶片的寬度和列的個數有關, 此決定晶片內的標準單元與巨集的位置分布改變, 其網格大小不改變; 因此我們提出如何決定網格數的大小以及網格的長寬比, 以取捨總移動量與執行時間。如圖3.5.所示, 此方法的概念適用於任意的合法器(legalizer)。

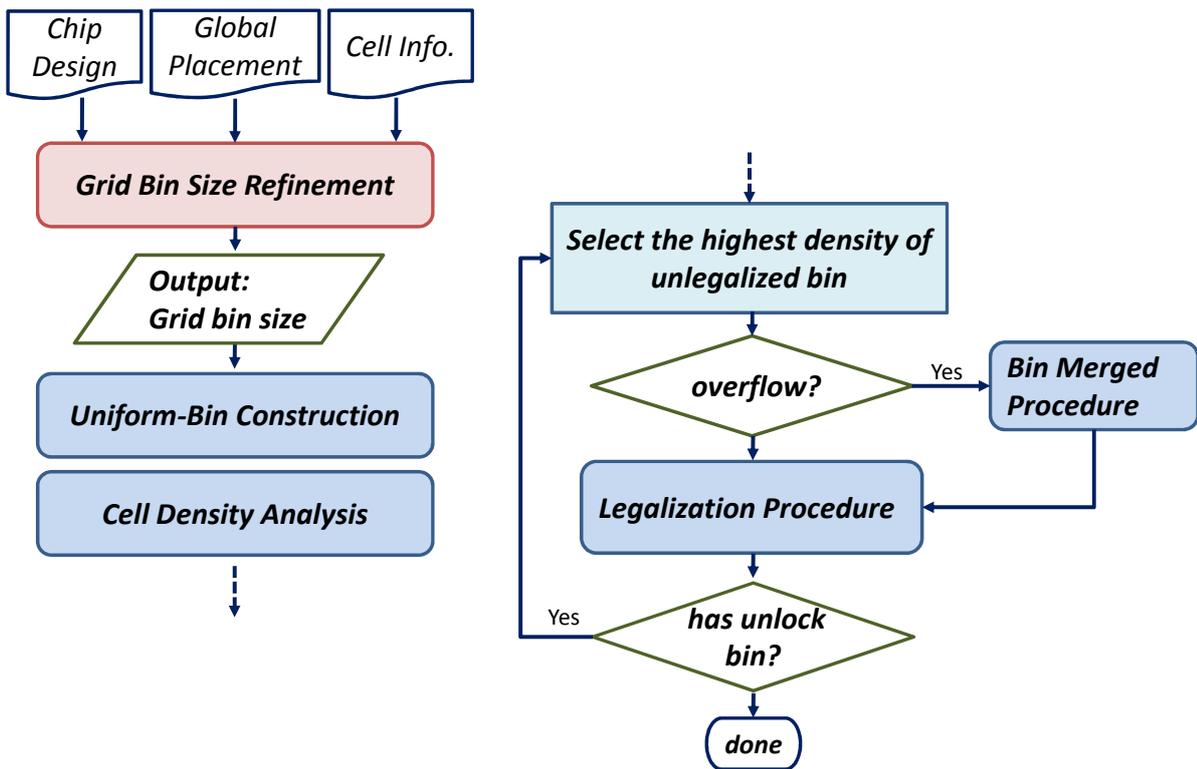


圖 3.5: *HiBinLegalizer*修改後的流程

3.4.1 長寬比的精化

標準單元合法化的過程中，所有巨集的位置固定，因此對標準單元而言，視其為障礙物，可想而知，巨集的分佈會影響最終合法化的結果；再者，倘若我們將晶片分割成均勻的格子，不但會分割可放置的列亦會切割障礙物，想當然而，會直接影響標準單元的移動量。假設單元格的面積相同且單元格的密度低於閾密度值，但單元格的寬長比不同，如圖3.6.所示。明顯地，可知左圖所得到的移動量較右圖所得到的移動量要來的好。然而，一個設計中有許多的巨集，但單元格只可有一個長寬比，此時該如何決定長寬比？而哪些巨集主宰整體的移動量？我們首先想到的是巨集的平均長寬比，然而全域擺置的結果中，標準單元與巨集也可能有面積重疊。由圖我們可以知道，若巨集與越多的標準單元面積重疊，其影響標準單元的移動量則越大。因此，我們利用加權平均(weighted average)來求得單元格長寬比。加權平均的定義如

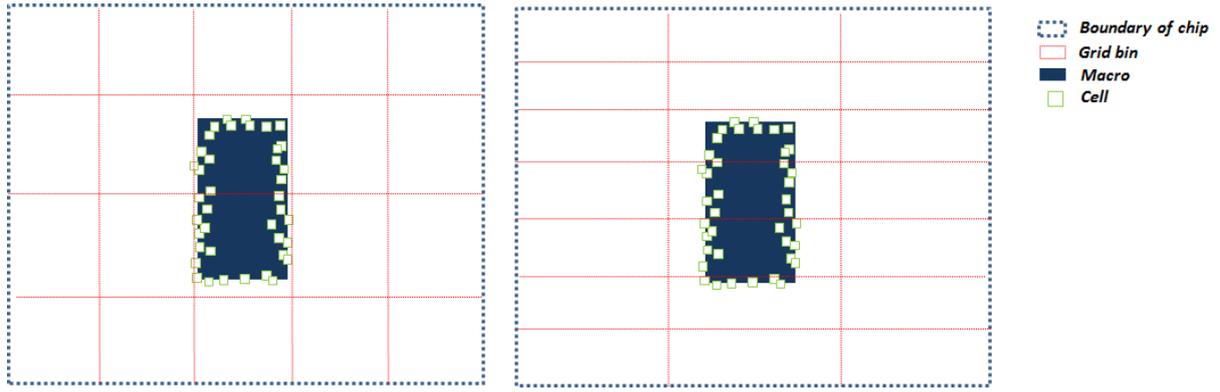


圖 3.6: 決定寬長比的示意圖

下:

$$\bar{X} = \frac{w(t_1)X(t_1) + w(t_2)X(t_2) + \cdots + w(t_M)X(t_M)}{w(t_1) + w(t_2) + \cdots + w(t_M)} \quad (3.63)$$

$$= \frac{\sum_{i=1}^M w(t_i)X(t_i)}{\sum_{i=1}^M w(t_i)} \quad (3.64)$$

其中 $w(t_i)$ 是非負的權重值， $X(t_i)$ 是輸入資料。

給定 N 標準單元和 M 巨集，令 $A_{overlap}$ 是巨集 t_i 與 N cells 之間的總交疊面積，而令為 $w(t_i)$ 等於 $1 + A_{overlap}$ ， $X(t_i)$ 是巨集 t_i 的寬或長。求得寬與長加權平均數， \bar{W} 和 \bar{H} 。此外，由圖 3.6，我們可以發現倘若格子的邊界沿著巨集切割，會迫使標準單元遠離最近的列。因此未避免此情況發生，網格的寬長比等於加權平均寬長比的倒數，也就是，

$$AR = \frac{W_b}{H_b} = \frac{\bar{H}}{\bar{W}} \quad (3.65)$$

其中 W_b 和 H_b 分別為單元格的寬度與長度

3.4.2 總格數的初估

給定的全域擺置結果，倘若標準單元與巨集的分佈不均勻，則經分割合併的程序後，合併區域的面積大小與目標標準單元個數亦可能分佈不均。再者，由於時間複雜度與合併區域內的標準單元個數有關，因此，擁有愈多標準單元的合併區域，花費時間愈久。因此，我們以最多標準單元個數的最大合併區域(maximum merged-bin)，作為猜測格子總數的依據。

首先，令 G_x 和 G_y 分別為水平和垂直方向的格子數；總格數 G 即可表示成 $G_x \times G_y$ 。我們規律的改變總格數，經過多次試驗，比較最大合併區域內的標準單元個數，記錄有最少標準單元個數的 $G_{b,x}$ 和 $G_{b,y}$ 。圖 3.7 是依個簡單找總格數的例子。

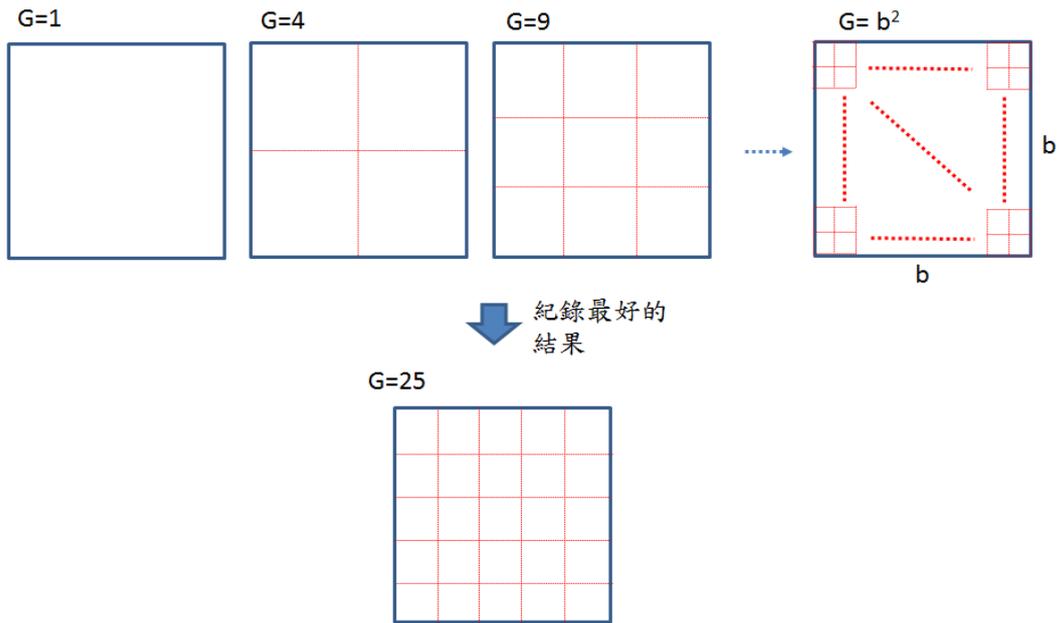


圖 3.7: 總格數的切法

3.4.3 決定總格數

前一小節，我們給出一個總格數 $G_{b,x} \times G_{b,y}$ ，然而為了提供更多的選擇，我們將單一的初估總格數擴展定義成總格數區間，如下式

$$(G_{b,x} - 1)(G_{b,y} - 1) \leq G_x G_y \leq (G_{b,x} + 1)(G_{b,y} + 1), \forall G_x, G_y \in \mathbb{N} \quad (3.66)$$

其中 G_x 和 G_y 分別為水平和垂直方向的格子數。我們希望給出一個合適的格子尺寸，使得 G_x 和 G_y 接近前一小節給出的寬長比 AR 以及接近初估的總格數。我們定義誤差量

$$Est = a \frac{|AR - r|}{AR} + b \frac{|G_{b,x} G_{b,y} - G_x G_y|}{G_{b,x} G_{b,y}} \quad (3.67)$$

$$r = \frac{G_x}{G_y} \quad (3.68)$$

其中 a 和 b 是介於 0 到 1 的常數。因此，在總格數區間下，對所有可能的 G_x 和 G_y ，記錄有最小誤差量 G_x 和 G_y ，作為最終的格子尺寸。

4

實驗結果

我們將Abacus以及*HiBinLegalizer*所提出的合法化方法用 C++ 的程式語言實現，並且執行於Intel Xeon 5160 3GHz 中央處理器、32G記憶體的系統工作站。為了驗證實驗結果，我們利用國際實體設計研討會(International Symposium on Physical Design, 簡寫 ISPD)[17]所釋出的測試電路，並且以NTUplace3 [2] 所擺置出的全域結果作為合法化方法的輸入。

根據*HiBinLegalizer*所提的方法，定義閾密度值 0.97 和網格尺寸

$$G_y = \sqrt{N_{row}} \quad (4.1)$$

$$G_x = 10 \quad (4.2)$$

其中 G_x 和 G_y 分別為水平方向與垂直方向的格子數，而 N_{row} 是列的總個數。由於我們應用*HiBinLegalizer*作為網格精化的合法器，因此我們利用所提出的精化方法，求出網格尺寸後，再給入*HiBinLegalizer*。

圖表4.1 列出測試電路的基本資訊，其中“Chip Width”和“Chip Height”分別代表晶片的寬度和長度，“Num. Rows”是晶片上的列個數，“#Total Objects”是所有的單元個數，包含可移動的標準單元與預先給定的巨集，“#Mov. Objects”是可移動的標準單元個數。實驗結果表.4.3, 表.4.2和表.4.4中，欄位一皆為測試電路的名稱、欄位二到四為比較Abacus, *HiBinLegalizer*, 和插入網格精化的*HiBinLegalizer*方法的真實數據，表.4.3: 標準單元在合法化前後的總移動量(total movement)無單位，表.4.2: 比較所有的標準單元的移動量，找出最大移動量(maximum movement)無單位，表.4.4: 執行時間(runtime)單位為秒(s)；而欄位五到七則是以將欄位二到四的數值除以欄位四所得的比值。

前面提及的*HiBinLegalizer*方法，格分法目的在於加速合法化的執行速度，還可限制最大移動量進而降低總移動量；然而我們所提出的網格精化法搭配在*HiBinLegalizer*下，給出總網

2005 ISPD Placement Contest and Benchmark					
Circuit	Chip Width	Chip Height	Num. Rows	#Total Objects	#Mov. Objects
adaptec1	10692	10680	890	211447	210904
adaptec2	14054	14040	1170	255023	254457
adaptec3	23190	23328	1944	451650	450927
adaptec4	23190	23328	1944	496045	494716
bigblue1	10692	10680	890	278164	277604
bigblue2	18690	18792	1566	557866	534782
bigblue4	32190	32328	2694	2177353	2169180

表 4.1: 2005 ISPD 擺置競賽的測試電路

格數以加速合法化的執行速度, 透過給定網格的寬長比改善網格邊界對總移動量的影響, 最後求得總移動量與執行速度的平衡。在最大移動量的實驗結果中, 我們可以看到大部分的電路可得到較小的移動量, 然而在電路adaptec1和adaptec4的結果中發現最大移動量並沒有降低, 此原因在於, 具最大移動量的標準單元落在網格的右邊界上, 又剛好剩餘的列空間不夠放置此標準單元, 因此迫使標準單元移到較遠的列上。雖說如此但是由實驗結果表.4.3可發現總移動量並沒有比其他文獻要來得大, 表示平均的移動量有降低, 但最大移動量被落於邊界的標準單元主宰。在實驗結果表.4.4中, 我們所提的方法雖然部分電路的執行時間大於*HiBinLegalizer*, 但總移動量小於*HiBinLegalizer*, 原因在決定網格大小的過程中, 我們取捨了時間與總移動量。比較Abacus, 在執行時間上有明顯的改善, 除了網格加速外, 在搜尋列的做法中, Abacus僅在 y 方向有停止搜尋的機制, 也就是若 y 方向的代價超過最小代價的列, 則不再搜尋。而我們另外也加入了 x 方向停止搜尋的機制, 因此大幅改善執行速度。由於格分法限制標準單元在固定區域移動以限制最大移動量, 使得總移動量變小, 然而實驗上卻有些出入, 部分電路, 最大移動量大於Abacus 但總移動量卻小於Abacus, 原因在於*HiBinLegalizer*在合法化前, 單元格會經過合併, 因此可能造成最大移動量變大, 也因此網格邊界變少, 使得總移動量變好。

circuit	Abacus	<i>HiBinLegalizer</i>	Our	Abacus	<i>HiBinLegalizer</i>	Our
adaptec1	7.29E+02	1.02E+03	2.65E+03	0.28	0.38	1.00
adaptec2	2.60E+03	2.49E+03	2.49E+03	1.05	1.00	1.00
adaptec3	6.01E+03	8.95E+03	6.67E+03	0.90	1.34	1.00
adaptec4	1.94E+03	1.90E+03	4.07E+03	0.48	0.47	1.00
bigblue1	4.59E+02	6.93E+02	4.90E+02	0.94	1.41	1.00
bigblue2	1.14E+03	6.71E+02	1.13E+03	1.01	0.59	1.00
bigblue4	6.19E+03	8.86E+03	4.92E+03	1.26	1.80	1.00

表 4.2: 標準單元最大移動量的比較結果

circuit	Abacus	<i>HiBinLegalizer</i>	Our	Abacus	<i>HiBinLegalizer</i>	Our
adaptec1	4.88E+06	5.98E+06	5.98E+06	0.82	1.00	1.00
adaptec2	9.78E+06	1.03E+07	9.83E+06	1.00	1.05	1.00
adaptec3	6.99E+07	7.64E+07	5.67E+07	1.23	1.35	1.00
adaptec4	1.78E+07	1.62E+07	1.49E+07	1.20	1.08	1.00
bigblue1	8.91E+06	8.98E+06	9.13E+06	0.98	0.98	1.00
bigblue2	1.64E+07	5.55E+06	5.47E+06	2.99	1.01	1.00
bigblue4	4.29E+08	3.73E+08	3.53E+08	1.21	1.06	1.00

表 4.3: 標準單元總移動量的比較結果

circuit	Abacus	<i>HiBinLegalizer</i>	Our	Abacus	<i>HiBinLegalizer</i>	Our
adaptec1	11.16	1.94	2.1	5.31	0.92	1.00
adaptec2	29.83	2.74	3.14	9.50	0.87	1.00
adaptec3	81.71	11.03	10.22	8.00	1.08	1.00
adaptec4	54.69	6.51	7.59	7.21	0.86	1.00
bigblue1	9.83	21.51	20.33	0.48	1.06	1.00
bigblue2	47.70	5.28	6.76	7.06	0.78	1.00
bigblue4	1994.67	732.71	573.15	3.48	1.28	1.00

表 4.4: 執行時間(s)的比較結果

5

結論

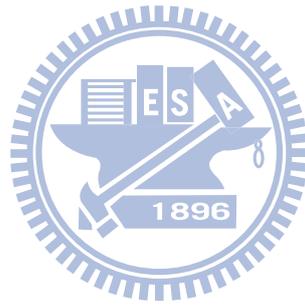
此篇論文，我們首先對*HiBinLegalizer*所提出的最佳位置解法器作理論上的分析與討論，證明此解法器所求得的解是最佳解，不僅適用於一群標準單元具交疊連續性，也適用於一群標準單元具部分交疊連續性。我們也提出了數個定理，簡化試驗列找尋中位數的過程，使得試驗列的演算法複雜度由 $O(n)$ 降為 $O(1)$ 。最後，我們沿用*HiBinLegalizer*的階層式概念，除了保留格分法可限制最大移動量以降低總移動量的概念，我們又另外提出網格數的精化法，透過障礙物的寬長比以及利用最大合併區域的總標準單元個數找出總網格數的初估值，進而給出適當的網格數及網格的尺寸，藉此取捨總移動量與執行時間。然而，我們所提出的網格精化法，亦可搭配其他的合法器。實驗結果顯示，應證我們的網格精化方法所求出的網格尺寸，相較於文獻[1]所提出的網格尺寸，平均可改善1.08倍的總移動量以及平均加速執行時間達1.25倍，應證，網格精化過後的網格尺寸的可行性。

Reference

- [1] Yu-Min Lee, Tsung-You Wu, and Po-Yi Chiang. A hierarchical bin-based legalizer for standard-cell designs with minimal disturbance. In *Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific*, pages 568 –573, Jan. 2010.
- [2] Tung-Chieh Chen, Zhe-Wei Jiang, Tien-Chang Hsu, Hsin-Chen Chen, and Yao-Wen Chang. Ntuplace3: An analytical placer for large-scale mixed-size designs with replaced blocks and density constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(7):1228–1240, July 2008.
- [3] Carl Sechen William Swartz. Timing driven placement for large standard cell circuits. In *Design Automation, 1995. DAC '95. 32nd Conference on*, pages 211 –215, 1995.
- [4] A.B. Kahng and Q. Wang. An analytic placer for mixed-size placement and timing-driven placement. In *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pages 565 – 572, nov. 2004.
- [5] Yongseok Cheon, Pei-Hsin Ho, A.B. Kahng, S. Reda, and Qinke Wang. Power-aware placement. In *Design Automation Conference, 2005. Proceedings. 42nd*, pages 795 – 800, june 2005.
- [6] Kimiyoshi Usami and Mark Horowitz. Clustered voltage scaling technique for low-power design. In *Proceedings of the 1995 international symposium on Low power design, ISLPED '95*, pages 3–8, New York, NY, USA, 1995. ACM.
- [7] U. Brenner and J. Vygen. Legalizing a placement with minimum total movement. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(12):1597 – 1613, dec. 2004.

- [8] Ulrich Brenner, Anna Pauli, and Jens Vygen. Almost optimum placement legalization by minimum cost flow and dynamic programming. In *Proceedings ACM/SIGDA International Symposium on Physical design*, pages 2–9, 2004.
- [9] Andrew B. Kahng. On legalization of row-based placements. In *in Proceedings 14th Great Lakes Symposium on VLSI*, pages 214–219, 2004.
- [10] Dwight Hill. Method and system for high speed detailed placement of cells within an integrated circuit design. *U.S. Patent 6370673*, April 2002.
- [11] Peter Spindler, Ulf Schlichtmann, and Frank M. Johannes. Abacus: Fast legalization of standard cell circuits with minimal movement. In *Proceedings ACM/SIGDA International Symposium on Physical Design*, pages 47–53, April 2008.
- [12] Tsung-Yi Ho and Sheng-Hung Liu. Fast legalization for standard cell placement with simultaneous wirelength and displacement minimization. In *VLSI System on Chip Conference (VLSI-SoC), 2010 18th IEEE/IFIP*, pages 369–374, Sept. 2010.
- [13] Andrew B. Kahng, Sherief Reda, and Qinke Wang. Architecture and details of high quality large-scale analytical placer. In *Proceedings Asia and South Pacific Design Automation Conference*, pages 890–897, 2005.
- [14] Tao Luo and David Z. Pan. Dplace2.0: A stable and efficient analytical placement based on diffusion. In *Proceedings Asia and South Pacific Design Automation Conference*, pages 346–351, 2008.
- [15] U. Brenner and J. Vygen. Faster optimal single-row placement with fixed ordering. In *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, pages 117–121, 2000.
- [16] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, USA, 2nd edition, 2001.

- [17] International symposium on physical design 2005 placement contest.
<http://www.sigda.org/ispd2005/contest.htm>, April 2005.



附錄 A

附錄

定理 1. *Given N_r ordered cells being consecutively overlapped in a cluster, indexed from 1 to N_r , with the ordering preserved constraint, a positive weight $e(l)$ for each cell l and a real number $p > 0$, the optimal solution for minimizing the following cost function*

$$\sum_{l=1}^{N_r} e(l) |x'(l) - x(l)|^p \quad (\text{A.1})$$

must satisfy

$$x'(l) - x'(l-1) = w(l-1), \quad l = 2, \dots, N_r$$

Here, $x(l)$ and $x'(l)$ are the bottom-left corner x -coordinates of each cell l in the global placement and in the legalization result, respectively.

Before proving Theorem 1, let's introduce a simple lemma.

引理 7. *Given N_r ordered cells being overlapped in a subrow with the ordering preserved constraint, after legalization, if the J -th cell is the first cell whose left-side x -coordinate needs to be shifted to the right direction of its original position, the left-side x -coordinate of each cell l with $l > J$ must also be shifted to the right side of its original position.*

Proof. Let $x(l)$ and $x'(l)$ be the left-side x -coordinates of each cell l in the global placement and in the legalization result, respectively. Here, we need to prove $x'(l) \geq x(l)$ for each $l > J$.

For cell $J + 1$, in the global placement, we have

$$x(J+1) - x(J) \leq w(J), \quad (\text{A.2})$$

After legalization, we have

$$x'(J+1) - x'(J) \geq w(J). \quad (\text{A.3})$$

Subtracting (A.2) from (A.3), we have

$$x'(J+1) - x(J+1) \geq x'(J) - x(J). \quad (\text{A.4})$$

Since $x'(J) \geq x(J)$, (A.4) indicates $x'(J+1) \geq x(J+1)$. By the similar way, we can conclude that $x'(l) \geq x(l)$ for each $l > J$. \square

Proof. (Proof of Theorem 1) Given the J -th cell being the first cell which satisfies $x'(J) \geq x(J)$ and according to Lemma 7, (A.1) can be rewritten as

$$\sum_{l=1}^{N_r} e(l) |x'(l) - x(l)|^p = \sum_{l=1}^{J-1} e(l) (x(l) - x'(l))^p + \sum_{l=J}^{N_r} e(l) (x'(l) - x(l))^p. \quad (\text{A.5})$$

- $\sum_{l=1}^{J-1} e(l) (x(l) - x'(l))^p$: $x'(l) \leq x(l)$, $1 \leq l \leq J-1$

After legalization, each $x'(l+1)$ can be rewritten as $x'(l+1) = x'(l) + w(l) + d_l$ for $1 \leq l \leq J-1$. The d_l is the distance between the right-side x -coordinate of cell l and the the left-side x -coordinate of cell $l+1$ after legalization. Hence,

$$\begin{cases} x'(J-1) = x'(J-2) + w(J-2) + d_{J-2}, \\ x'(J-2) = x'(J-3) + w(J-3) + d_{J-3}, \\ \vdots \\ x'(l+1) = x'(l) + w(l) + d_l \end{cases}$$

Summing up the above equations, $x'(l)$ can be represented as

$$x'(l) = x'(J-1) - \sum_{j=l}^{J-2} w(j) - \sum_{j=l}^{J-2} d_j \quad (\text{A.6})$$

To minimize $\sum_{l=1}^{J-1} e(l) (x(l) - x'(l))^p$ is equivalent to maximize each $x'(l)$ since $x'(l) \leq x(l)$. Furthermore, according to (A.6), to maximize $x'(l)$ is equivalent to simultaneously maximize $x'(J-1)$ and minimize $\sum_{j=l}^{J-2} d_j$ for each l .

Taking $d_j = 0$ for $1 \leq j \leq J - 2$, each $\sum_{j=l}^{J-2} d_j$ can be minimized to be zero which leads to

$$x'(l) - x'(l-1) = w(l-1), \quad l = 2, \dots, J-1. \quad (\text{A.7})$$

Therefore, we can conclude that to minimize $\sum_{l=1}^{J-1} e(l) (x(l) - x'(l))^p$ is equivalent to maximize $x'(J-1)$ under (A.7).

- $\sum_{l=J}^{N_r} e(l) (x'(l) - x(l))^p$: $x'(l) \geq x(l)$, $J \leq l \leq N_r$

After legalization, each $x'(l)$ can be rewritten as $x'(l) = x'(l-1) + w(l-1) + d_{l-1}$ for $J \leq l \leq N_r$. The d_{l-1} is the distance between the right-side x -coordinate of cell $l-1$ and the the left-side x -coordinate of cell l after legalization. Hence,

$$\begin{cases} x'(J+1) = x'(J) + w(J) + d_J, \\ x'(J+2) = x'(J+1) + w(J+1) + d_{J+1}, \\ \vdots \\ x'(l) = x'(l-1) + w(l-1) + d_{l-1} \end{cases}$$

Summing up the above equations, $x'(l)$ can be represented as

$$x'(l) = x'(J) + \sum_{j=J}^{l-1} w(j) + \sum_{j=J}^{l-1} d_j \quad (\text{A.8})$$

To minimize $\sum_{l=J}^{N_r} e(l) (x'(l) - x(l))^p$ is equivalent to minimize each $x'(l)$ since $x'(l) \geq x(l)$. Furthermore, according to (A.8), to minimize $x'(l)$ is equivalent to simultaneously minimize $x'(J)$ and minimize $\sum_{j=J}^{l-1} d_j$ for each l .

Taking $d_j = 0$ for $J \leq j \leq N_r - 1$, each $\sum_{j=J}^{l-1} d_j$ can be minimized to be zero which leads to

$$x'(l) - x'(l-1) = w(l-1), \quad l = J+1, \dots, N_r. \quad (\text{A.9})$$

Therefore, we can conclude that to minimize $\sum_{l=J}^{N_r} e(l) (x'(l) - x(l))^p$ is equivalent to minimize $x'(J)$ under (A.9).

- Hence, to minimize $\sum_{l=1}^{N_r} e(l) |x'(l) - x(l)|^p$ is equivalent to simultaneously maximize $x'(J-1)$ and minimize $x'(J)$ under (A.7) and (A.9). Since, after legalization, $x'(J) - x'(J-1) \geq w(J-1)$, it can be easily to conclude that to minimize $\sum_{l=1}^{N_r} e(l) |x'(l) - x(l)|^p$, $x'(J)$ and $x'(J-1)$ must satisfy

$$x'(J) - x'(J-1) = w(J-1). \quad (\text{A.10})$$

Finally, from (A.7), (A.9), and (A.10), we have

$$x'(l) - x'(l-1) = w(l-1), \quad l = 2, \dots, N_r. \quad (\text{A.11})$$

□

定理 2. *Given a discrete random variable \mathbf{X} , $E(|\mathbf{X} - \alpha|)$ is minimized by choosing α as the median of \mathbf{X} .*

