

以不可切割表示法處理植基於群聚策略之平面規劃問題

程仲勝 潘佳信 江昱麟 蔡宗達

大葉大學電機工程學系
彰化縣大村鄉山腳路 112 號

摘要

隨著 VLSI/SOC (very large scale integration/system-on-a-chip) 的蓬勃發展，如何有效並迅速的評估晶片面積是後端實體設計階段 (physical design phase) 中相當重要的議題。但對於許多先前研究而言，若要正確的評估具有較多模組之晶片面積，需花費大量的時間。因此在本論文中我們提出一個植基於群聚 (clustering) 策略之不可切割平面規劃演算法以能更有效且迅速的評估晶片面積。在我們的方法中，首先使用群聚技巧將模組聚集起來形成一些面積較大但個數較少的組合模組 (supermodules)，接著採用 Corner Block List 表示法來記錄組合模組間相對位置關係並在其上執行模擬退火 (simulated annealing) 程序以求得具有較小面積之平面圖 (floorplan)。由實驗結果得知，對具有較多模組電路之 GSRC (gigascale systems research center) 測試例子而言，使用群聚策略確實能快速且有效地求得較佳的平面圖。

關鍵詞：實體設計，平面規劃，群聚，模擬退火

A Clustering-Based Approach for Floorplanning by Using Non-Slicing Representation

JONG-SHENG CHERNG, CHIA-HSIN PAN, YU-LIN CHIANG and TSUNG-TA TSAI

*Department of Electrical Engineering, Da-Yeh University
112 Shan-Jiau Rd., Da-Tsuen, Changhua, Taiwan*

ABSTRACT

As VLSI/SOC (very large scale integration/system-on-a-chip) technology advances, effectively and efficiently evaluating the area of a chip is one of the key challenges in the backend physical design phase. In much previous research, the methods require spending a large amount of time to correctly evaluate the area of a chip with hundreds of modules. In this study, a non-slicing floorplan algorithm based on clustering strategy is proposed for this evaluation. In this method, clustering strategy is first applied to group modules to form fewer supermodules of a larger size; then, the Corner Block List representation for a non-slicing floorplan is used to record the relative positions among the supermodules. Finally, the simulated annealing procedure is executed according to the Corner Block List representation for obtaining an area-optimized floorplan. Experimental results on GSRC (gigascale systems research center) benchmarks demonstrate the efficiency and effectiveness of proposed clustering-based floorplan algorithm.

Key Words: physical design, floorplanning, clustering, simulated annealing

一、簡介

VLSI (very large scale integration) 後端實體設計階段 (physical design phase) 中的平面規劃 (floorplanning) 是整個階段的第一個步驟且是一個相當重要的步驟, 其影響爾後其他步驟甚鉅, 因此有許多方法被提出來解決平面規劃的問題 [1-17]。平面規劃最主要的目的是放置一組電路模組 (modules) 於晶片上並使整體晶片面積達到最小。平面規劃後所得之最終平面圖 (floorplan) 可以分成可切割 (slicing) 平面圖 [13, 16] 與不可切割 (non-slicing) 平面圖 [1-12, 14, 15] 兩大類。因此, 平面規劃演算法亦可分為處理可切割 [13, 16] 與不可切割 [1-12, 14, 15] 平面結構兩大類。在處理可切割平面結構方面可用可切割樹 (slicing tree) [13] 和波蘭表示法 (polish expression) [16] 表示模組間位置的關係。而在處理不可切割平面結構方面則可用 BSG (bounded-sliceline grid) 表示法 [12]、Sequence-Pair 表示法 [11]、O-Tree 表示法 [4]、B*-Tree 表示法 [3]、CBL (corner block list) 表示法 [5] 及 TCG (transitive closure graph) 表示法 [7] 等來表示模組間相對位置關係。

最近積體電路設計已進展到系統單晶片層級 (system on chip) 的階段, 在單一晶片中的模組個數已達到數百甚至數千, 以往解決平面規劃的方法在模組個數少的情況下較易有較佳解, 但在如系統單晶片中具有較多模組的狀況時, 卻常花費冗長的時間並較難得到較佳解。因此要如何在模組個數遽增時快速的獲得面積最佳化的平面圖, 已是目前相當重要的議題。

本論文針對具有較多模組的平面規劃問題提出一個兩階段的平面規劃演算法。首先在第一階段使用群聚 (clustering) 技巧將模組群集起來形成一些面積較大的組合模組 (supermodules), 接著再利用適當的平面規劃演算法針對較大的組合模組進行平面規劃。群聚後的模組個數小於群聚前的個數, 因此上述兩階段的方法可加速平面規劃的進行。至於平面規劃演算法, 我們採用 CBL 表示法 [5] 來記錄模組間相對位置關係並在其上執行模擬退火 (simulated annealing) 程序以求得面積最佳化的平面圖。採用 CBL 表示法的原因為其表示一個不可切割之平面圖只需要 $n(3+\lg n)$ 個位元, 且其轉換成相對應之平面圖只需要 $O(n)$ 的時間, 其中 n 為模組個數, 因此配合群聚技巧可有效解決多模

組之平面規劃問題。

二、問題描述與 CBL 表示法

平面規劃問題即是在擺置一組電路模組, 使其形成寬為 w 、高為 h 之矩形平面圖; 其中令 $B = \{b_1, b_2, \dots, b_n\}$ 為欲擺置之 n 個固定面積且固定寬高之矩形模組 (hard modules) 集合, 而第 i 個模組 b_i 之面積、寬、高分別為 a_i 、 w_i 、 h_i ; 經平面規劃處理後所輸出之平面圖須滿足模組彼此不重疊, 並使其面積 $w*h$ 為最小 (即晶片面積最小)。

由於我們所提出解決平面規劃的方法是以 CBL 表示法 [5] 為基礎, 因此以下將說明 CBL 表示法如何記錄模組間相對位置關係以及如何在其上執行模擬退火的程序。CBL 表示法是由 S 、 L 及 T 三種串列所組成; 其中 S 串列記錄各個模組的代號, 需要 n 個位元, L 串列記錄模組的位置關係, 需要 $n-1$ 個位元, T 串列紀錄每一個模組與其他相鄰模組所形成 T 字形的個數, 最多需要 $2*n-3$ 個位元, 由這三個串列, 可在 $O(n)$ 的時間內得到一個唯一的平面圖 [5]。

上述 L 串列所記錄的為目前欲擺入模組與其他已擺入模組的位置關係, 此關係可分為兩類: 一類將 L 串列相對值設為 0 (記為 $L = 0$), 而另一類則設為 1 (記為 $L = 1$); 其中 $L = 0$ 表示目前擺入模組的位置必須位於上一個擺入模組位置的上方, 而 $L = 1$ 則表示其位置在上一個擺入模組位置的右方, 如圖 1 所示。另一方面 T 串列由 0、1 所組成, 每次讀入 T 串列時以讀至 0 為停止條件, 並計算讀入 1 的個數, 若讀入 1 之個數為 N , 則表示目前欲擺入之模組與 $N+1$ 個已擺入模組相鄰。

接著我們介紹如何在 S 、 L 、 T 三個串列上執行模擬退火的程序:

步驟 1: 隨機產生一組可行的 S 、 L 、 T 串列並計算其所對應之初始平面圖面積。

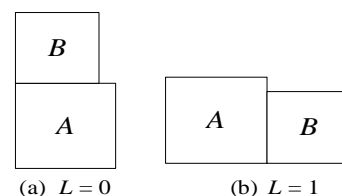


圖 1. 目前欲擺入模組 B 與上一個擺入模組 A 的位置關係

步驟 2：開始執行模擬退火程序，我們利用以下幾種改變目前平面圖結構的方式進行退火程序：

- (a) 隨機交換 S 串列中的任兩個模組。
- (b) 隨機選擇 L 串列中的一個位置，並將 1 變 0 或 0 變 1。
- (c) 隨機選擇 T 串列中的一個位置，並將 1 變 0 或 0 變 1。
- (d) 模組旋轉 90° 、 180° 或 270° 。

步驟 3：計算改變後新平面圖的面積。設定 $cost$ 為此次產生之面積減去上一次接受之面積。若 $cost$ 小於零則接受新解，若大於零則隨機產生一介於 0~1 之間的數 r 並比較 r 與 $\exp(-cost/k)$ (k 為溫度參數) 之大小以決定是否接受新解。

步驟 4：依照遞減公式逐次減小 k 值。重複步驟 2 到步驟 4 直到 k 值小於預設凝固點。

三、兩階段平面規劃

所提兩階段平面規劃演算法在第一階段使用群聚技巧將模組群集起來形成一些面積較大的組合模組，接著在第二階段採用 CBL 表示法 [5] 來記錄組合模組間相對位置關係並執行模擬退火程序以求得具有較小面積的平面圖。整個兩階段平面規劃演算法步驟可由以下之虛擬碼加以表示。在 (一)、(二) 及 (三) 子節中亦將針對演算法中重要步驟程序加以說明。

Algorithm

begin

```

Read_benchmark_file();
Cluster_original_modules();
Randomly_generate_CBL_lists();
Generate_initial_floorplan_from_CBL_lists();
Calculate_area_of_initial_floorplan();
Execute_simulated_annealing_procedure();

```

end

(一) 群聚

在處理少量模組擺放時，傳統平面規劃演算法可以在一定時限內找到近似最佳解，但當模組個數達到數百甚至數千個，若以傳統的方式直接計算則需要耗費大量的時間。為了解決這樣的問題，我們利用群聚的技巧先將模組群集起來，

以減少模組擺放演算法的複雜度，加速未來模擬退火收斂的速度。

群聚是將多個模組組合成一個較大的組合模組，而目前在本演算法中我們只考慮每兩個模組群聚成一個組合模組，如此可減少多個模組同時群聚時可能會形成較多且較分散的閒置空間 (dead space) 問題。而在挑選模組群聚時需考慮群聚後組合模組之形狀以及面積。一般而言，形狀愈趨於正方形且形成面積愈接近兩模組之面積和時愈有利於往後之平面規劃。換句話說，組合模組之長寬差值愈小且閒置空間愈少愈好。另一方面，任兩個模組群聚，依其相對位置之不同，會形成八種不同組合狀況，在不考慮連線腳位置時，其中有四種狀況是重複的，所以可以將組合的情況簡化為四種，如圖 2 所示。對於這四種組合我們將選擇長寬差值小且閒置空間少的做為兩模組群聚後之組合模組。

接著我們說明演算法中所採用由下往上 (bottom up) 階層式 (hierarchy) 模組群聚的步驟：

步驟 1：設 $f = \alpha * W_1 + \beta * W_2$ ，其中 W_1 及 W_2 分別表示所形成組合模組之長寬差值及閒置空間大小， α 及 β 則為其權重值。

- (a) 任找一模組 A ，使其與模組 B 形成一組合模組，其中模組 B 為所有模組中與模組 A 群聚後可產生最小 f 值者。(注意：任兩模組群聚後之 f 值取其 4 種組合中 f 值最小者)。
- (b) 將已群聚之模組移除，對剩下之模組重複步驟 (a) 及 (b) 直到所有模組皆已群聚。

步驟 2：將步驟 1 完成後之組合模組重新視為一般模組並重複步驟 1 及步驟 2 直至達到預設之群聚階層數。

圖 3 中顯示所採用之群聚策略。由圖中可知組合模組數是以 2 的幕次方遞減，因此經過少許的群聚階層後即可將平面規劃問題的複雜度降低。

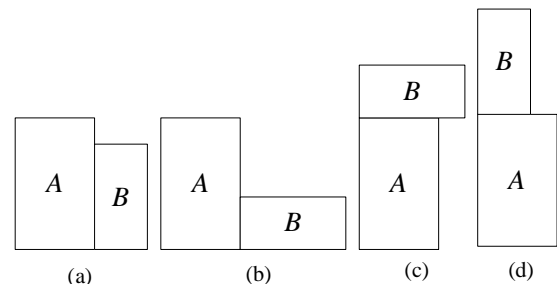


圖 2. 兩模組 A 及 B 的組合情形

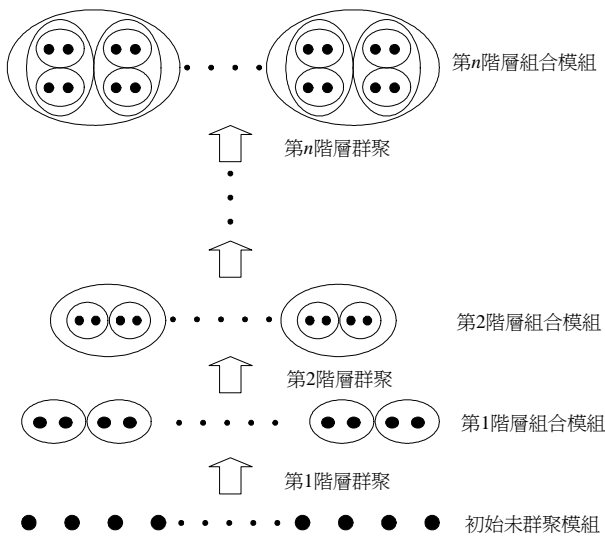


圖 3. 由下往上階層式模組群聚策略

(二) CBL 表示法之實作

經過群聚後形成一些個數較少但面積較大的組合模組，接著我們採用 CBL 表示法 [5] 來記錄組合模組相對位置關係並執行模擬退火程序以求得面積最佳化之平面圖。如何利用 CBL 表示法之三種串列對應至一個平面圖已在第二節中簡單討論過，而詳細的對應說明可參考論文 [5]。然而本子節將介紹有別於 [5] 中利用建立限制圖形來計算每一個模組實際上擺放位置座標之方法。以下即為本論文提出不需建立限制圖形即可求得各模組座標的步驟：

步驟 1：讀入串列 S 第一個元素 $S[0]$ 並將其所對應之模組左下角座標設為 $(0, 0)$ 。

步驟 2：For $i = 1$ to $n - 1$

讀入串列 L 中之元素 $L[i-1]$ 與讀入串列 T ，若讀入的元素 $L[i-1] = 0$ ，則將元素 $S[i]$ 擺在 $S[i-1]$ 的上方，並利用目前所讀入串列 T 之內容判斷與多少已擺放模組相鄰，計算 $S[i]$ 所對應模組左下角座標時，其 X 座標與相鄰模組中左下角 X 座標最小者相同，而其 Y 座標則需取相鄰各模組之左上角 Y 座標最大者為 $S[i]$ 之 Y 座標；然而若 $L[i-1] = 1$ ，則將 $S[i]$ 擺在 $S[i-1]$ 的右方，此時 $S[i]$ 所對應模組左下角 Y 座標與相鄰模組中左下角 Y 座標最小者相同，而其 X 座標則需取相鄰各模組之右下角 X 座標最大者為 $S[i]$ 之 X 座標。

圖 4 利用圖解的方式舉例解說上述求得各模組座標的步驟。其中 CBL 三串列分別假設為 $S = \{A, B, C, D\}$ 、 $L = \{0,$

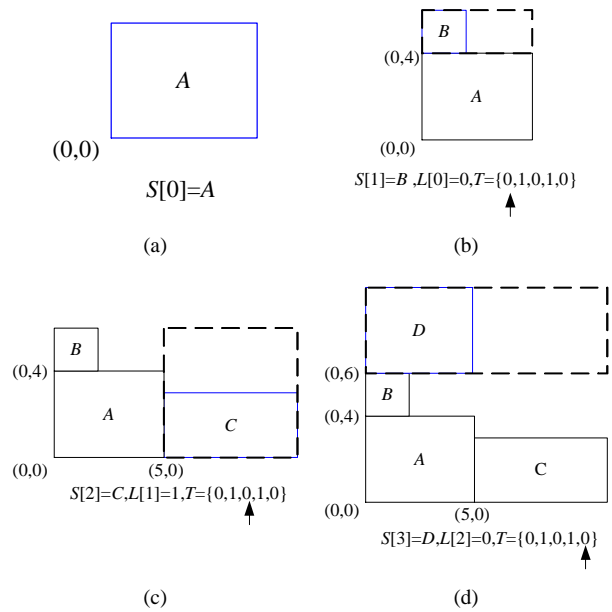


圖 4. 利用 CBL 三串列轉換求得各模組座標的過程

$1, 0\}$ 及 $T = \{0, 1, 0, 1, 0\}$ 。表 1 所列為圖 4 中各模組維度大小資料。由圖 4 可知，串列 T 讀入後即有一指標追蹤記錄每個 0 元素前 1 出現的個數，如此藉以計算目前欲擺放模組與多少已擺放模組相鄰。如圖 4(b) 中，元素 $T[0]$ 前沒有 1 出現，因此模組 B 僅與一個模組（即模組 A ）相鄰；而圖 4(c) 中，元素 $T[2]$ 前（且在 $T[0]$ 之後）1 出現 1 次，因此模組 C 與兩個模組（即模組 A 及 B ）相鄰；最後在圖 4(d) 中，元素 $T[4]$ 前（且在 $T[2]$ 之後）1 出現 1 次，因此模組 D 與兩個模組（即模組 B 及 C ）相鄰。在圖 4(b) 至 4(d) 中虛線所圍之範圍除了擺放之模組外，剩餘的空間為可能的閒置空間；然而針對此例子而言，這些剩餘的空間都成為所有模組擺放完成後之閒置空間。

(三) 模擬退火降溫方式

模擬退火在降溫的過程中，溫度高時模組的排列變化較大，隨著溫度的降低模組排列的變化越來越小，答案開始收斂。在溫度調降方式方面，不同的調降策略對結果會有一定程度的影響，所以我們在本論文中將一般傳統模擬退火的降

表 1. 各模組資料

ID	高	寬
A	4	5
B	2	2
C	6	3
D	5	4

溫方式(如圖 5 所示)改為重複升降溫的方式(如圖 6 所示),以便讓得到較佳解的機會升高。

四、實驗結果與討論

在本論文中使用 PC 為實驗平台,其 CPU 時脈為 2.2GHz,記憶體為 512MB,使用編譯軟體為 Microsoft Visual C++ 6.0。實驗所用的 benchmarks 取至 MCNC (Microelectronics Center of North Carolina) 包含 apte、xerox、hp、ami33 及 ami49 五個模組個數較少的電路與 GSRC (gigascale systems research center) 包含 n100、n200 及 n300 三個模組個數較多的電路。表 2 為每個 benchmark 電路所包含的模組個數等相關資料。

本論文將模組平面規劃實驗分為以下四種方式進行:不使用群聚技巧及利用傳統降溫方式、不使用群聚技巧及利用重複升降溫方式、使用群聚技巧及利用傳統降溫方式、使用群聚技巧及利用重複升降溫方式。其中使用群聚技巧的部分再分為群聚次數為一次與群聚次數為兩次之情形。在群聚次數為一次的部分,我們將其應用在上述的八個 benchmarks;然而在群聚次數為二次的部分,我們只應用在模組個數較多的 GSRC 三個 benchmarks。

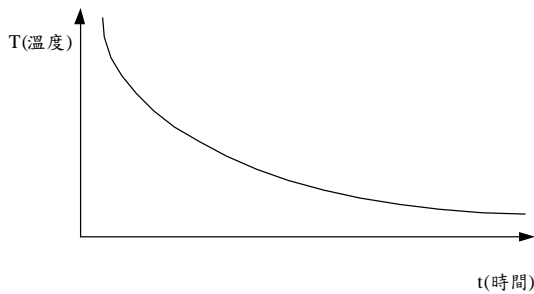


圖 5. 傳統降溫方式

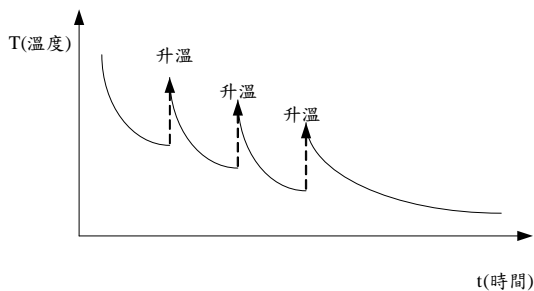


圖 6. 重複升降溫方式

表 2. 各 benchmark 電路資料

電路名稱	模組個數	模組面積總和(mm ²)
apte	9	46.561628
xerox	10	19.305296
hp	11	8.830584
ami33	33	1.156449
ami49	49	35.445424
n100	100	0.172708
n200	200	0.170129
n300	300	0.273170

(一) 不使用群聚技巧及利用傳統降溫方式

此組實驗(代號為 E1)模擬退火起始溫度設為 6000000°C,終止溫度設為 10°C,降溫比率設為 0.999998。由表 3 實驗結果可知,對 MCNC benchmarks 而言,此 E1 實驗與論文 [5] 在最終平面圖面積及所花費時間上大致相同。在 GSRC benchmarks 方面,執行時間被控制在一定的範圍內,但其閒置空間的比率相對 MCNC benchmarks 而言卻大了許多,這亦可由圖 7 及 8 觀察得知。從圖 8 中電路 n300 之平面圖可明顯的看到有許多範圍較大的閒置空間。當然若為了降低多模組電路閒置空間的比率,我們可調整模擬退火的相關參數,然而往往需付出比表 3 結果大上數十倍甚至數百倍的執行時間才能得到較合理的閒置空間比率。

表 3. 不使用群聚技巧及利用傳統降溫方式所得結果

電路名稱	E1 之最終面積 (mm ²)	E1 之閒置空間 (%)	E1 之花費時間 (s)
	[5] 之最終面積 (mm ²)	[5] 之閒置空間 (%)	[5] 之花費時間 (s)
apte	47.302240	1.5657	11
	47.270000	1.4000	25
xerox	20.315106	4.9707	12
	20.960000	7.8000	30
hp	9.208080	4.0996	13
	9.141000	3.3000	19
ami33	1.237348	6.5381	32
	1.201000	3.7000	36
ami49	38.542682	8.0359	42
	38.580000	8.2000	65
n100	0.199348	13.3636	93
	NA	NA	NA
n200	0.234548	27.4652	182
	NA	NA	NA
n300	0.417690	34.5998	282
	NA	NA	NA

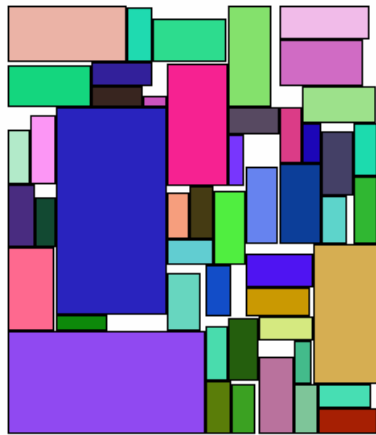


圖 7. E1 實驗所得之 ami49 電路結果

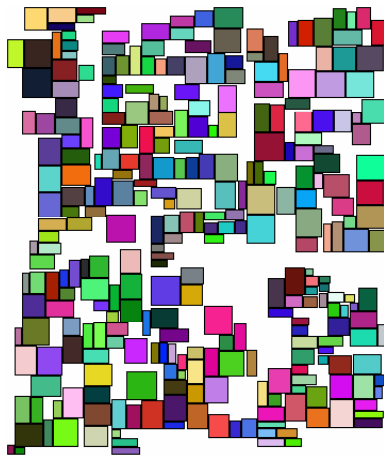


圖 8. E1 實驗所得之 n300 電路結果

(二) 不使用群聚技巧及利用重複升降溫方式

此組實驗（代號為 E2）模擬退火起始溫度設為 6000000℃，終止溫度設為 10℃，降溫比率設為 0.99998。本次實驗中模擬退火降溫方式改為重複升降溫，當溫度降至原本溫度的 1/10000 即升溫至原本溫度的 70%。由表 4 之結果可知，在執行時間方面，利用傳統降溫方式（實驗 E1）與利用重複升降溫方式（實驗 E2）大致相同。然而在最終平面圖面積 / 閒置空間方面，對每個例子而言，使用重複升降溫方式皆能得到較佳解。以 ami49 電路為例，從圖 7 與圖 9 兩圖直接比較可以看出圖 9 的閒置空間較圖 7 少，降低約 2.99%。在 n300 電路方面，圖 8 與圖 10 雖然一樣有大範圍的閒置空間，但利用重複升降溫的方式在相同的時間內將閒置空間的比率降低了 2.14%。

表 4. 不使用群聚技巧及利用重複升降溫方式所得結果

電路名稱	E2 之最終面積 (mm ²)	E2 之閒置空間 (%)	E2 之花費時間 (s)
	E1 之最終面積 (mm ²)	E1 之閒置空間 (%)	E1 之花費時間 (s)
apte	46.924848	0.7740	11
	47.302240	1.5657	11
xerox	20.017088	3.5559	12
	20.315106	4.9707	12
hp	9.031680	2.2266	13
	9.208080	4.0996	13
ami33	1.207066	4.1934	31
	1.237348	6.5381	32
ami49	37.333100	5.0500	42
	38.542682	8.0359	42
n100	0.196350	12.0407	90
	0.199348	13.3636	93
n200	0.216852	21.5460	184
	0.234548	27.4652	182
n300	0.404443	32.4577	280
	0.417690	34.5998	282

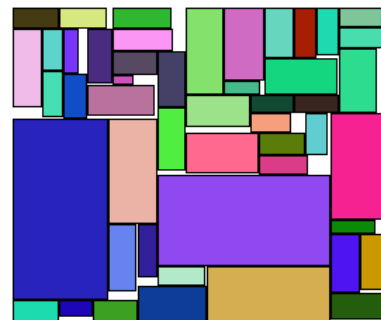


圖 9. E2 實驗所得之 ami49 電路結果

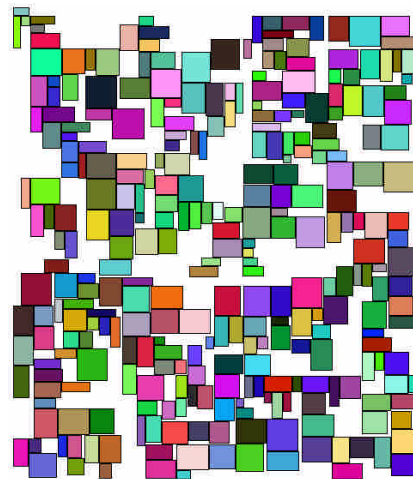


圖 10. E2 實驗所得之 n300 電路結果

(三) 使用群聚技巧及利用傳統降溫方式

此組實驗（代號為 E3-1）模擬退火起始溫度設為 6000000°C，終止溫度設為 10°C，降溫比率設為 0.99999，群聚次數設為 1。由表 5 實驗結果可以看出群聚策略對模組數少的電路面積最佳化沒有太大的幫助。模組個數少的電路在模擬退火的階段幾乎可以將每一種狀況組合一次，因此若將模組個數少的電路做群聚的動作，反而會侷限其解空間，如此要得到較佳解比較困難。由圖 11 中可看見還有一些閒置空間可將鄰近的模組移動擺入，但基於群聚的因素已不可再移動。然而群聚策略卻對模組個數多的電路有較好的功效，如 n100、n200 及 n300 電路。尤其對具有 300 個模組的 n300 電路，閒置空間相較 E1 實驗足足降低了 17%，其間的差異可由觀察比較圖 12 與圖 8 得知。在執行時間方面，對所有的例子而言，群聚策略確實能大幅加速模擬退火收斂速度，由表 5 實驗結果可以看出每個電路執行時間約只為 E1 實驗的十分之一。

針對 GSRC 之 benchmarks，我們將群聚次數增為 2 次，降溫比率改為 0.999993，其餘條件不變重做實驗（代號為 E3-2）。當群聚的次數增加，由表 6 得知執行時間可以更加減少。另外在最終平面圖面積及閒置空間方面，由表 6 亦可得知對所有 GSRC benchmarks 電路而言，實驗 E3-2 較實驗

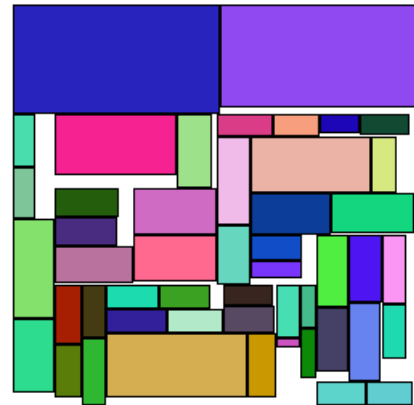


圖 11. E3-1 實驗所得之 ami49 電路結果

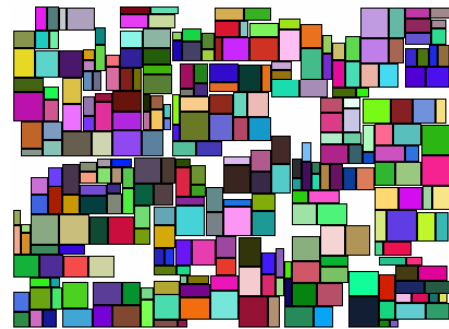


圖 12. E3-1 實驗所得之 n300 電路結果

表 5. 使用群聚技巧及利用傳統降溫方式所得結果

電路名稱	E3-1 之最終面積 (mm ²)	E3-1 之閒置空間 (%)	E3-1 之花費時間 (s)
	E1 之最終面積 (mm ²)	E1 之閒置空間 (%)	E1 之花費時間 (s)
apte	47.448800	1.8697	1
	47.302240	1.5657	11
xerox	20.622875	6.3889	1
	20.315106	4.9707	12
hp	9.779616	9.7041	2
	9.208080	4.0996	13
ami33	1.286593	10.1154	3
	1.237348	6.5381	32
ami49	39.424616	10.0931	5
	38.542682	8.0359	42
n100	0.197415	12.5152	9
	0.199348	13.3636	93
n200	0.217000	21.5995	18
	0.234548	27.4652	182
n300	0.331474	17.5893	25
	0.417690	34.5998	282

表 6. 使用群聚技巧及利用傳統降溫方式所得結果

電路名稱	E3-2 之最終面積 (mm ²)	E3-2 之閒置空間 (%)	E3-2 之花費時間 (s)
	E3-1 之最終面積 (mm ²)	E3-1 之閒置空間 (%)	E3-1 之花費時間 (s)
n100	0.189856	9.0321	7
	0.197415	12.5152	9
n200	0.188802	9.8902	13
	0.217000	21.5995	18
n300	0.310389	11.9910	20
	0.331474	17.5893	25

E3-1 更為有效。比較圖 12 及圖 13 可知，圖 13 中模組與模組間的閒置空間更加減少且集中。

(四) 使用群聚技巧及利用重複升降溫方式

此組實驗（代號為 E4-1）模擬退火起始溫度設為 6000000°C，終止溫度設為 10°C，降溫比率設為 0.9999，群聚次數設為 1。本次實驗中模擬退火降溫方式改為重複升降溫，當溫度降至原本溫度的 1/10000 即升溫至原本溫度的 70%。由表 7 實驗結果可以看出與表 5 的情形大致相同。

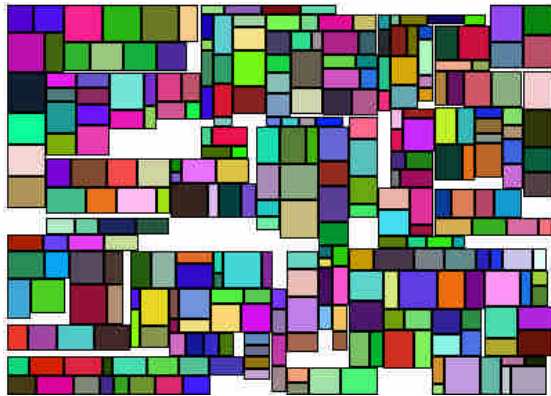


圖 13. E3-2 實驗所得之 n300 電路結果

表 7. 使用群聚技巧及利用重複升降溫方式所得結果

電路名稱	E4-1 之最終面積 (mm ²)	E4-1 之閒置空間 (%)	E4-1 之花費時間 (s)
	E1 之最終面積 (mm ²)	E1 之閒置空間 (%)	E1 之花費時間 (s)
apte	47.448800	1.8697	2
	47.302240	1.5657	11
xerox	20.622875	6.3889	2
	20.315106	4.9707	12
hp	9.779616	9.7041	2
	9.208080	4.0996	13
ami33	1.245580	7.1557	5
	1.237348	6.5381	32
ami49	38.748024	8.5232	5
	38.542682	8.0359	42
n100	0.188097	8.1814	10
	0.199348	13.3636	93
n200	0.205884	17.3665	20
	0.234548	27.4652	182
n300	0.323300	15.5057	29
	0.417690	34.5998	282

圖 14 與圖 15 分別繪出本實驗所得 ami49 及 n300 電路之最終平面圖，其平面圖面積及閒置空間皆較實驗 E3-1 少。

針對 GSRC 之 benchmarks，我們將群聚次數增為 2 次，降溫比率改為 0.99995，其餘條件不變重做實驗（代號為 E4-2）。由表 8 可知，實驗 E4-2 較實驗 E4-1 有效。圖 16 中 n300 電路閒置空間已減少至 9.4459%，至於 n100 與 n200 電路也降低至 7.2041% 與 8.5445%。

圖 17 及圖 18 以長條圖的方式顯示 MCNC 及 GSRC benchmarks 在本論文所做不同實驗中（即 E1~E4）所得平面圖閒置空間比率之比較圖。從圖 17 中可知，群聚策略對 MCNC 模組數較少的電路例子無法發揮其功能，主要原因

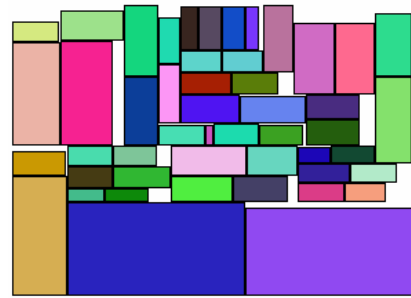


圖 14. E4-1 實驗所得之 ami49 電路結果

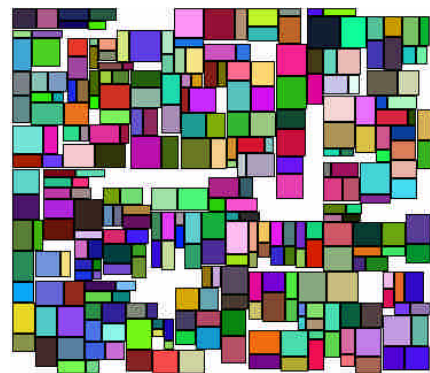


圖 15. E4-1 實驗所得之 n300 電路結果

表 8. 使用群聚技巧及利用傳統降溫方式所得結果

電路名稱	E4-2 之最終面積 (mm ²)	E4-2 之閒置空間 (%)	E4-2 之花費時間 (s)
	E4-1 之最終面積 (mm ²)	E4-1 之閒置空間 (%)	E4-1 之花費時間 (s)
n100	0.186116	7.2041	9
	0.188097	8.1814	10
n200	0.186024	8.5445	17
	0.205884	17.3665	20
n300	0.301665	9.4459	25
	0.323300	15.5057	29

為 MCNC 的 benchmark 中模組大小維度差異性大且模組數目不多，因此不易群聚形成閒置空間較小的組合模組，進而無法發揮群聚的效能。然而從圖 18 中可知，對 GSRC 模組數較多的電路例子而言，使用群聚策略較不使用群聚策略更能獲得低的閒置空間比率，因為 GSRC 模組之維度差異較小且模組數目多，所以在使用群聚策略後所產生組合模組之閒置空間較小，因此可發揮群聚的效能。在現今單一晶片中的模組個數已達到數百甚至數千的情況下，使用群聚策略確實能在合理的時間內求得較佳的平面圖。

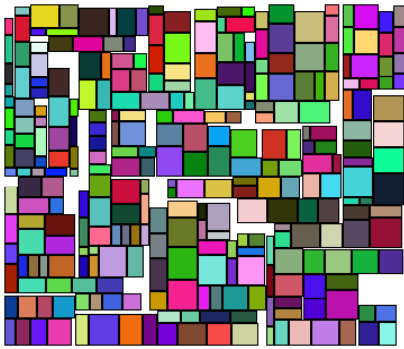


圖 16. E4-2 實驗所得之 n300 電路結果

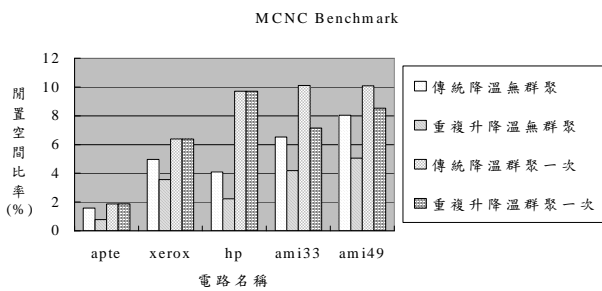


圖 17. MCNC benchmarks 閒置空間比率比較圖

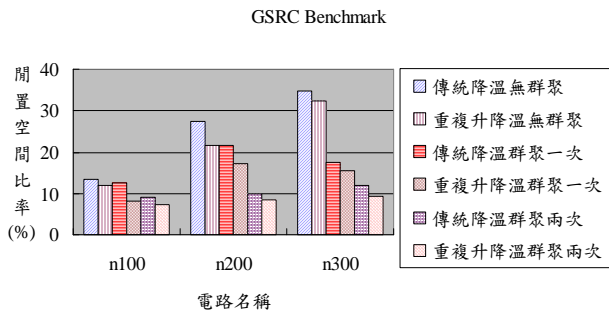


圖 18. GSRC benchmarks 閒置空間比率比較圖

五、結論

在本論文中我們提出一個植基於群聚策略之不可切割平面規劃演算法以便能有效且迅速的評估具有較多模組電路所形成平面圖之面積。所提出的方法中，首先使用群聚技巧將模組兩兩聚集起來形成一些面積較大但個數較少的組合模組，接著採用 Corner Block List 表示法來記錄組合模組間相對位置關係並在其上執行模擬退火程序以求得面積最佳化的平面圖。由我們的實驗結果可知，對具有較多模組電路而言，使用群聚策略確實能在合理的時間內求得較佳的平面圖。

除了面積之外，未來我們亦將於模組群聚時考慮網列連線長度及其分佈狀況，藉以求得具有較短網列連線長度及較低繞線擁擠程度的平面圖。

參考文獻

1. 吳彬玄、程仲勝 (民 91)，降低電磁干擾之後置平面規劃器，2002 台灣電磁相容研討會，台北。
2. 吳彬玄、習存榮、程仲勝 (民 92)，考慮電磁相容之超大型積體電路平面規劃之研究，2003 台灣電磁相容研討會，台北。
3. Chang, Y. C., Y. W. Chang, G. M. Wu and S. W. Wu (2000) B*-trees: A new representation for non-slicing floorplans. Proceedings of the 37th ACM/IEEE Design Automation Conference, Los Angeles, CA.
4. Guo, P. N., C. K. Cheng and T. Yoshimura (1999) An O-tree representation of non-slicing floorplan and its applications. Proceedings of the 36th ACM/IEEE Design Automation Conference, New Orleans, LA.
5. Hong, X., G. Huang, Y. Cai, J. Gu, S. Dong, C. K. Cheng and J. Gu (2000) Corner Block List: An effective and efficient topological representation of non-slicing floorplan. Proceedings of the 2000 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA.
6. Li, J., T. Yan, B. Yang, J. Yu and C. Li (2004) A packing algorithm for non-manhattan hexagon/triangle placement design by using an adaptive O-tree representation. Proceeding of the 41th ACM/IEEE Design Automation Conference, San Diego, CA.
7. Lin, J. M. and Y. W. Chang (2001) TCG: A transitive closure graph-based representation for non-slicing floorplans. Proceeding of the 38th ACM/IEEE Design Automation Conference, Las Vegas, NV.
8. Long, C., L. J. Simonson, W. Liao and L. He (2004) Floorplanning optimization with trajectory piecewise-linear model for pipelined interconnects. Proceedings of the 41th ACM/IEEE Design Automation Conference, San Diego, CA.
9. Ma, Y., S. Dong, X. Hong, Y. Cai, C. K. Cheng and J. Gu (2001) VLSI floorplanning with boundary constraints based on corner block list. Proceedings of Asia and South Pacific Design Automation Conference, Yokohama.
10. Ma, Y., X. Hong, S. Dong, Y. Cai, C. K. Cheng and J. Gu

-
- (2001) A compact algorithm for placement design using corner block list representation. Proceedings of the 4th ASIC Conference, Shanghai.
11. Murata, H., K. Fujiyoshi, S. Nakatake and Y. Kajitani (1996) VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12), 1518-1524.
12. Nakata, S., K. Fujiyoshi, H. I. Murata and Y. Kajitani (1996) Module placement based on BSG-structure and IC layout applications. Proceedings of the 1996 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA.
13. Otten, R. H. J. M. (1982) Automatic floorplan design. Proceedings of the 19th ACM/IEEE Design Automation Conference, Miami Beach, FL.
14. Sassone, P. G. and S. K. Lim (2003) A novel geometric algorithm for fast wire-optimized floorplanning. Proceedings of the 2003 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA.
15. Tang, X. and D. F. Wong (2002) Floorplanning with alignment and performance constraints. Proceedings of the 39th ACM/IEEE Design Automation Conference, New Orleans, LA.
16. Wong, D. F. and C. L. Liu (1986) A new algorithm for floorplan designs. Proceedings of the 23th ACM/IEEE Design Automation Conference, Las Vegas, NV.
17. Xiang, H., X. Tang and D. F. Wong (2003) Bus-driven floorplanning. Proceedings of the 2003 IEEE/ACM International Conference on Computer- Aided Design, San Jose, CA.

收件：94.04.13 修正：94.05.30 接受：94.08.04