

南 華 大 學

資訊管理學系碩士論文

以序列對表示法解決非切割性結構

叢聚限制的平面規劃問題

Non-Slicing Floorplan with Clustering Constraints

by the Sequence Pair



研究生：林良哲

指導教授：吳光閔

中華民國 九十一年六月

第一章 緒論

平面規劃 (floorplan) 是超大型積體電路 (very large scale integration 簡稱 VLSI) 實體設計 (physical design) 中非常重要的步驟，它是將電路分解 (circuit partition) 為模組 (module) 之後，在不違反設計的原則 (例如：要求任兩個模組不能重疊)，把所有電路元件 (circuit element) 放置於同一個晶片當中的規劃方式，其主要目的是為了將晶片的總面積、繞線長度以及電源消耗等等的目標作最佳化。近年來，隨著晶片產業進入深次微米時代，且 IC 的內容越變越複雜，再加上階層式設計 (hierarchical design) 與智慧財產權 (Intellectual Property) 重覆使用的觀念備受歡迎，使得平面規劃的地位在 IC 設計流程當中顯得格外的重要。

第一節 平面規劃的結構分類

依據平面規劃所組成的結構，可以劃分成：可分割性結構 (slicing structure) 與非分割性結構 (non-slicing structure)，如圖 1 所示，這兩種類型最大的不同在於：可分割性結構使用水平線與垂直線的切割方式，將矩形遞回的切割而成為更小的矩形，非分割性結構卻是無法如此去切割的；若我們

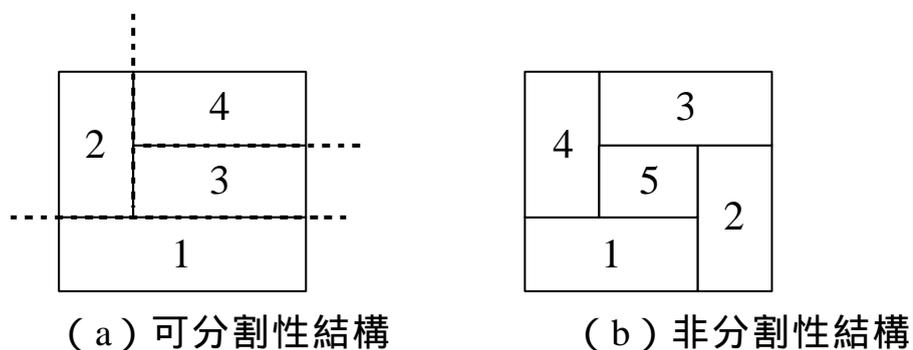


圖 1：可分割性與非分割性結構區別

將這兩類結構作整理比較，如表 1 所示，將可發現到，可分割性結構容易以樹狀的資料結構，例如：Wang-Liu[20]的分割樹 (slicing tree) 來表示，而且

它在平面規劃時的解空間較少，運算速度較快；另一方面，雖然非分割性結構解空間多，運算時間較長，但是其彈性很高，可以處理包括分割性結構的問題，並且為大多數實務上平面規劃最佳解的結構，這些優點也不容我們去忽視它，最近幾年，各種非分割性平面規劃的表示法也相繼被提出，例如：Sequence-Pair[14]、BSG[7]、O-tree[4]、B*-tree[2]、CBL[5]、TCG[12]，因此，我們規劃在表 2 當中依據這些表示法的不同，將其特徵作摘要性的介紹，並且討論其中的優點與缺點。

表 1：可分割性與非分割性結構的比較

結構性	特徵	優點 ()、缺點 ()
可分割性結構 Slicing Structure	<ol style="list-style-type: none"> 1. 可將矩形以水平與垂直的方式遞回的切割，使之變成更小的矩形。 2. 強調以單獨模組對應至整體模組的關係。 	<ul style="list-style-type: none"> ● 由於限制平面規劃的切割方式，因此，產生可行解的數目較小。 ● 運算的時間較短。 ● 表示法簡單，容易使用樹狀結構來表示。 □ 無法延伸處理非分割性結的問題。
非分割性結構 Non-Slicing Structure	<ol style="list-style-type: none"> 1. 可由任意矩形所組成模組的平面結構。 2. 強調各別模組間的相對關係。 	<ul style="list-style-type: none"> ● 為大多數實務上平面規劃最佳解的結構。 ● 可套用於可分割性結構的平面規劃問題。 □ 解空間的數目較大。 □ 需要運算的時間較長。

表 2：非分割性結構的表示法種類

表示法	內 容	優點 ()、缺點 ()
Sequence-Pair	<p>為一對記載模組在平面規劃時由不同方向搜尋的結果,以顯示兩個模組是上下或左右的相對關係。</p>	<ul style="list-style-type: none"> ● 表示法的彈性較大,能處理各種形態(shape)的問題¹。 ● 表示法編碼都可以對應至一個合理的擺置狀態,而且只有一個唯一的解。 ● 任一的擺置狀態必能對應到唯一一組表示法的編碼。 □ 解空間較大。 □ 需建立限制圖以讓表示法與擺置 (placement) 對應。
BSG	<p>BSG 為 Bounded Slice-line Grid 的縮寫,它是以三個參數來記載其空間 (room) 上下與左右模組擺置的相對位置。</p>	<ul style="list-style-type: none"> ● 表示法的彈性較大。 □ 解空間較大。 □ 需建立限制圖以讓表示法與擺置 (placement) 對應。 □ 表示法無唯一性,將會發生重覆多個表示法編碼對應至同一個擺置的情況。

¹對於模組的型態,詳細說明請參考本章的第二節。

<p>O-Tree</p>	<p>O-Tree為Ordered-Tree的縮寫，它使用樹狀的資料結構記錄，並且採取深度優先走訪（depth-first-search）的方式搜尋，利用兩個參數來記載模組位於圖形上的位置。</p>	<ul style="list-style-type: none"> ● 解空間較小。 ● 使用較少的位元編碼。 □ 某些形態下無法使用。 □ 不規則的樹狀結構，實作時困難度較高。 □ 插入節點的位置將會被限制，使得其解無繼承性，而容易錯失最佳解的機會。 □ 需建立限制圖以讓表示法與擺置（placement）對應。 □ 將會有表示法對應於不合理擺置狀態的機率發生。
<p>B*-tree</p>	<p>B*-tree使用了二元樹的資料結構，規定模組資料儲存於葉節點，以深度優先走訪的方式遞回搜尋，以找出全部模組的擺置。</p>	<ul style="list-style-type: none"> ● 解空間較小。 ● 不需額外建立限制圖以讓表示法與擺置對應，因此，運算時間可較為迅速。 ● 可以彈性處理各種不同的模組。 □ 將會有表示法對應於不合理擺置狀態的機率發生。

CBL	CBL 為 Corner Block List的縮寫，它利用每次皆從右上角插入模組的步驟，遞回記錄其與原模組之間的T形方位關係，以達成記錄整個平面規劃的圖形。	<ul style="list-style-type: none"> ● 解空間較小。 ● 可以彈性處理各種不同的模組。 □ 平面規劃在某些形態下無法使用。 □ 需建立限制圖以讓表示法與擺置 (placement) 對應。 □ 將會有表示法對應於不合理擺置狀態的機率發生。
TCG	TCG 為 Transitive Closure Graph 的縮寫，分別以兩個TCG將模組以水平和垂直方向作記錄，以顯示模組的位置。	<ul style="list-style-type: none"> ● 解空間較小。 ● 不需額外建立限制圖以讓表示法與擺置對應，因此，運算時間可較為迅速。 □ 平面規劃在某些形態下無法使用。 □ 目前能處理的限制較少。

第二節 模組的形態

在積體電路設計的初期，大部份模組 (module) 的尺寸都還尚未確定，此時的平面規劃，一般會將模組形態 (shape) 因素先列入考慮，模組的形態可區分為：(1) Hard module：長、寬固定，其作標與方向 (orientation) 為可以變動的矩形模組 (2) Soft module：面積固定，但是模組的長與寬的在某特定限制比例內可以變動，作標、方向亦可調整的矩形模組 (3) Rectilinear

module：由任意水平線與垂直線構成，其形狀固定，但其擺置的作標與方向皆可以調動的模組。至論文完稿前我們所搜集到的資料，在平面規劃表示法當中，能同時兼顧使用於不同形態的相關研究，將它分類製成表格，如表 3 所示²：

表 3：表示法與各種形態被運用的狀況

表示法 \ 型態	Hard module	Soft module	Rectilinear module
Slicing tree	[20]	[20]	[21]
Sequence-Pair	[15]	[15]	[15]
BSG	[7]	[7]	[17]
O-tree	[4]	N/A	N/A
B*-tree	[2]	[2]	[2]
CBL	[5]	[5]	[5]
TCG	[7]	N/A	N/A

由上表可以看出：Slicing tree、Sequence-Pair、BSG、B*-tree與 CBL 的表示法，在各種形態被討論的數目是平分秋色，接著，我們再考慮平面規劃在具有條件限制時，被討論的狀況。

第三節 平面規劃的條件限制

一般平面規劃考慮的目標為：模組總面積的大小、模組間的線段長度以及訊號延遲等等的問題。在平面規劃設計時，為了使模組充分發揮其指定的

²其中符號“ ”表示已被學者提出討論了，符號“ [] ”表示相關文章的索引。

功能，以便產生有效的佈局，於是就需要加入特定的條件限制 (constraint)，以便輔助 IC 的設計，例如：(1) 預先固定擺放的模組限制(pre-placed module /constraint)：它是積體電路在設計之時，考量某些巨型模組 (macro cell)，像是 RAM、ROM 與 CPU core 等等，為了滿足所制定的規格，這些模組必需事先固定在特定的位置，之後其他的模組才能繼續安置 (2) 邊緣性限制 (boundary constraint)：通常是在平面規劃時，得同時考慮某些模組包含有 input/output connections 的特徵，為了減少連接線段的長度，所以特定的模組必須放置到 IC chip 的邊界旁 (3) 對稱性限制 (symmetry constraint)：當平面規劃缺乏對稱性的限制時，某些類比電路將可能因受熱不均而導致供應電壓無法達到預定的要求，更嚴重時，由於溫度的變動，間接影響電路間傳遞訊號的正確性，因此，特定的模組必須考慮對熱源裝置擺放的對稱性，以避免上述狀況發生，(4) 叢聚性限制 (clustering constraint)：通常是使用於特定模組與模組之間具有大量的連線時，若是模組能夠緊鄰著放置，便可以達到減少繞線成本的目的。在平面規劃的表示法當中，能同時兼顧在不同限制條件的研究，我們將它分類製成表格，如表 4 所示³：

³其中符號“ ”表示已被學者提出討論了，符號 “[]”表示相關文章的索引。

表 4：表示法與各種限制條件被運用的狀況

	預先固定擺放的模組	邊緣性限制	對稱性限制	叢聚性限制
Slicing tree	[22]	[9]	N/A	[19]
Sequence-Pair	[15]	[13]	[1]	N/A
BSG	[17]	[10]	N/A	N/A
O-tree	N/A	[13]	[18]	N/A
B*-tree	[6]	[23]	N/A	N/A
CBL	N/A	[16]	N/A	N/A
TCG	N/A	N/A	N/A	N/A

綜合以上的比較結果將可以察覺到，Slicing tree 與 Sequence-Pair 表示法在不同形態與具條件限制的因素下，所被提出的相關研究較多，顯示能使用的條件較廣泛，但是，在第一節的結構性分析當中，我們已發現非分割性結構其彈性很高，可以處理包含分割性結構的問題，基於此理由，所以我們著手進行在非分割性結構下，利用序列對來考慮叢聚限制的平面規劃，由這些外在條件的限制，來發展一個合理擺置（feasible placement）的演算法，以符合目前工業界在平面規劃的需求，並且提供實務上整合應用的便利性。

第四節 論文的架構

本篇論文其餘的架構如下：在第二章中，首先介紹平面規劃使用序列對（Sequence-Pair）表示法的詳細步驟，在第三章之中，我們定義了叢聚限制的問題，在第四章，探討加入序列對表示法之後，如何才能適用於叢聚特性，並且由此程序當中，我們提出將序列對的解空間轉換成滿足群聚限制的合理擺置方式，第五章是延用上一章所發展出的轉換方式，應用在模擬退火法

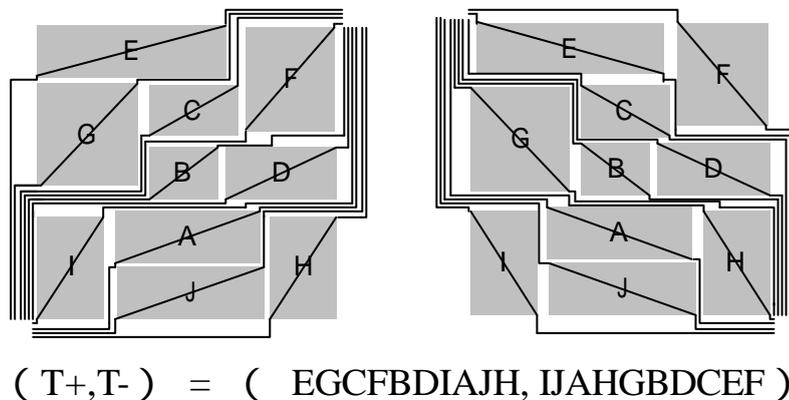
(simulated annealing) 上，來解決具叢聚限制模組擺置 (placement) 的問題，第六章為利用 MCNC benchmark circuits 做為實驗的樣本，以及套入模擬退火法後，所呈現的測試結果，第七章是結論，進而就本論文做一總結，並且提出建議做為將來研究之參考。

第二章 序列對表示法

序列對 (Sequence-Pair) 表示法是 H. Murata 等人[14], 在 1996 年所提出的, 序列對主要的精神是使用兩個序列來表示模組間的上、下或是左、右的相對關係, 它的優點為: 其所有的表示法都可以確保對應至一個合理的擺置狀態 (feasible placement), 反之亦然, 再者, 序列對可以處理 Hard module、Soft module、Rectilinear module 與多種的條件限制, 所以其表示法是非常有彈性。

第一節 序列對的編碼方式

對於多個模組的擺置 (placement), 我們可用一組序列對 (Γ^+, Γ^-) 來表示它, 其中 Γ^+ 及 Γ^- 分別是指模組位於格柵 (gridding) 上被搜尋到的結果, Γ^+ 所對應的是正階線 (positive step line), 是由上-右與下-左階線與該模組內的對角線所組成, 如圖 2 (a) 所示, 上-右階線的製作方式為: 由每個模組的右上角開始往上或是往右繪製, 在不違反: (1) 與其他的模組邊緣相交 (2)



(a) 正階線之圖示 (b) 負階線之圖示。

圖 2 : 序列對的編碼方式

與之前所劃的線段相交 (3) 與晶片的邊緣相交的條件下，它可以選擇再往右方或是往上方前進，直到達晶片的右上角才可停止；相同的，下-左階線是由每個模組的左下角往下或往左方繪製，可以選擇再往左方或是往下方前進，直達晶片的左下角才可停止，將其上-右與下-左階線與該模組內的對角線連結，即可完成對應於該模組的正階線，此外，由於正階線建構的過程當中，只要不違反上述的限制，每條正階線將不會有交叉的情況發生，因此，正階線是可以依其（左到右的）順序被編號；依此可類推， Γ^- 所對應是負階線（negative step line），如圖 2(b) 所示，乃是由左-上、右-下階線與模組內的對角線連結所組成的，因此在 Γ^- 也將可得到另一組模組之階線的排列記錄，最後整合 Γ^+ 與 Γ^- 的記錄，序列對其值即可求得。

另外，由上述階線在建構過程中觀察可得知，序列對是依據模組的位置、與格柵上不同方向搜尋方式所構成，序列對（ Γ^+ , Γ^- ）中存在的訊息是各別模組間的相對關係，而非以單獨模組對應整體模組的關係，所以在處理不同形態的模組時，會更具有彈性。

第二節 表示法對應的平面規劃

若是事先知道序列對的編碼之後，如何來判斷模組在晶片上的位置呢？甚至要如何反推回到平面規劃的佈局？首先，我們假設已知有 n 個固定方向（orientation）的矩形模組（module）需要作平面規劃，序列對中 Γ^+ 與 Γ^- 模組的編號數目必定是一致，且為 n 個，從 n 個模組當中取兩個模組 a , b ，若是在（ Γ^+ , Γ^- ）中的先後次序為：

- 左方性質：

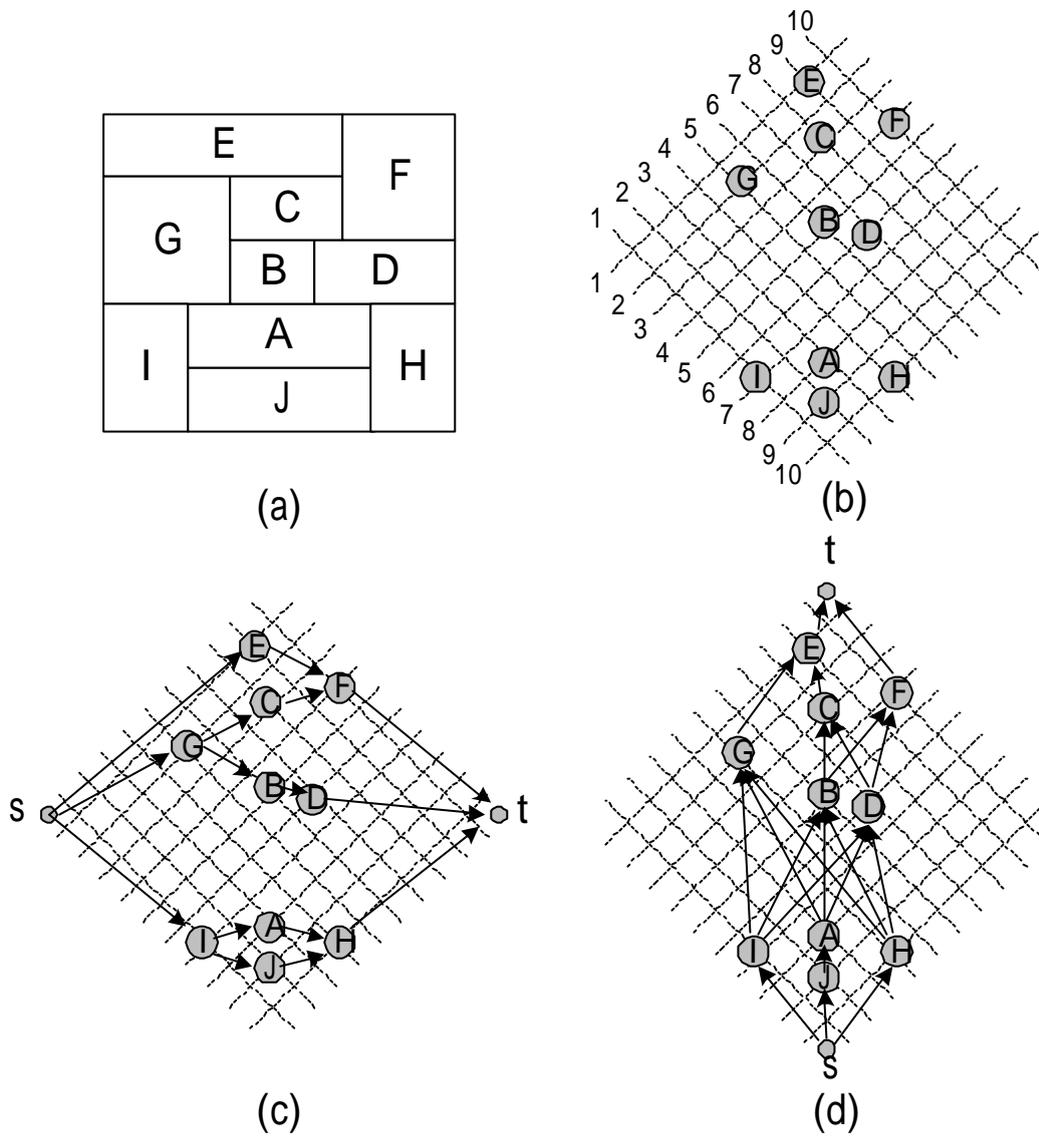
若序列對為（... a ... b ..., ... a ... b ...）表示 a 在 b 的左方。

- 下方性質：

若序列對為 $(\dots b \dots a \dots, \dots a \dots b \dots)$ 表示 a 在 b 的下方。

依據左方性質，將可以決定模組之間的左、右關係，並建構出一個水平限制圖 (horizontal constraint graph) G_H ，如圖 3(c) 所示，其建構方式如下：首先，令 $G_H = (V, E)$ ， V 是頂點 (vertex) 的集合， $V = \{s_H\} \cup \{t_H\} \cup \{v_i \mid i = 1, \dots, n\}$ ， s_H 、 t_H 、 v_i 分別表示為起點、終點和每個模組的頂點，除了 s_H 與 t_H 權重 (vertex-weight) 為 0，其餘頂點的權重則為該模組所屬的寬度， E 為限制圖 G_H 中邊線 (edge) 的集合， $E = \{s_H, v_i \mid i = 1, \dots, n\} \cup \{v_i, v_j \mid \text{模組 } j \text{ 對於模組 } i \text{ 具有左方性質}\} \cup \{v_i, t_H \mid i = 1, \dots, n\}$ ，由此法，就可以建造出一個具方向性、非環式 (acyclic) 的圖形 G_H ，接著，只要利用最長路徑演算法，計算 s_H 至 t_H 的最長路徑，即可以由 G_H 中計算出整個模組擺置狀態 (placement) 的最大寬度。

同理，也可以由下方性質的判斷方法，得到模組之間的上、下相對關係，建構垂直限制圖 (vertical constraint graph) G_V ，如圖 3(d) 所示，計算 G_V 中出整個模組擺置狀態的最大高度。將 G_H 與 G_V 的最長路徑的相乘積，即為該組序列對 (Γ^+, Γ^-) 於平面規劃的總面積。



序列對為 (EGCFBBDIAJH, IJAHGBDCEF)

圖 3: 水平與垂直限制圖的建構方式 (a) 平面規劃的圖形 (b) 將 Γ_+ 與 Γ_- 順序記錄於傾斜方格 (oblique grid) 內 (c) 水平限制圖 G_H (d) 垂直限制圖 G_V (為了簡化圖形 G_H 與 G_V , 其遞移性線段 (transitive edge) 皆已先省略了)。

第三章 叢聚限制的問題定義

平面規劃的擺置問題，已經被證明為 NP-hard Problem[14]。當模組與模組之間有大量的連線時，為了要將模組盡可能地緊鄰著放置，以達成減少繞線的成成本的目的，所以在考慮平面規劃擺置時，除了模組間不能重疊之外，將會加入叢聚限制的條件。為了之後解說上的方便，所以我們安排在本章中，先給予叢聚限制定義。

平面規劃叢聚限制的問題 (The Floorplan Problem with Clustering Constraints) 可以簡單描述為：任一參與叢聚的模組，必有另一個也是參與相同叢聚的模組，與其相鄰接 (geometrically adjacent)，為了更精確定義問題，因此我將其問題作成以下的描述：對於有 i 個長、寬固定的矩形模組集合 M ， $M = \{F, C\}$ ， F 為不參與叢聚的模組集合， C 為參與叢聚的模組集合，其中 $F = \{f_1, f_2, \dots, f_j\}$ ， $0 \leq j \leq i$ ， $C = \{C_1, C_2, \dots, C_k\}$ ， $0 \leq k \leq i - j$ ，任取其中一個相同叢聚的集合 $C_t \in C$ ，若是要達成叢聚性限制並減少繞線的長度，則需要滿足以下的定義：

- 阻礙叢聚的定義：

若是存在任意 $f_j \in F$ ，其 f_j 的上、下、左、右四個方向與 C_t 之中的模組都有相鄰，則稱 C_t 中有 f_j 的阻礙。

- 滿足叢聚限制的定義：

對於任每個 $c_p \in C_t$ 必有另一個 $c_q \in C_t$ ，與其相鄰接，且 C_t 當中不能有阻礙叢聚的狀況發生，如此，才得稱為滿足叢聚限制的條件。

如圖 4(a) 當中，假設 BDEFG 為叢聚在一起的模組，經由上述判斷，這些模組因其中有 C 的阻礙，所以會判斷屬於為不合理的解，但是在圖 4(b) 假設 ABC 為叢聚模組時，A、C 與 B、C 兩兩相鄰，且其中無阻礙，所以這樣才能稱得上為一個滿足叢聚性限制的合理解。另一方面，在考慮平面規劃時，除了要同時滿足叢聚限制的定義外，並且每個模組 $m_i \in M$ 都不可以重疊，任意模組皆可以自由地旋轉其方向 (rotated orientation)⁴，特別要提出的是，由於序列對表示法具 P-admissible 解空間的性質[14]，因此其 (1) 解空間是有限的 (2) 產生的每個解都是合理可行的 (3) 估算時間可以在 polynomial time 內完成 (4) 產生的最佳解空間符合其最佳實際的擺置方式，這些優點在找尋叢聚限制解的過程當中，可確保所有的解空間從頭至尾都是有效的。

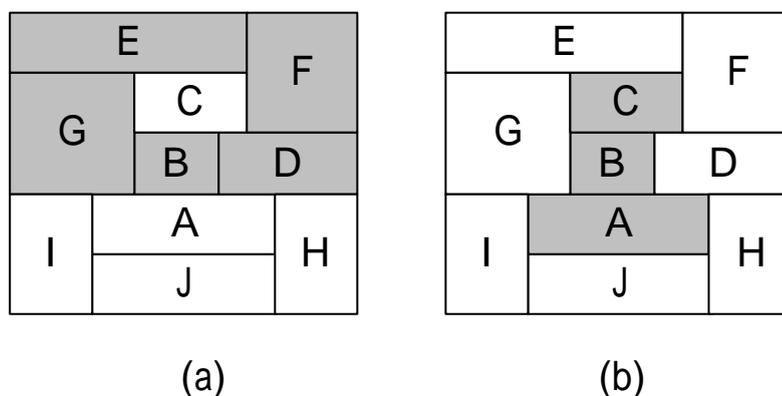


圖 4：合理的叢聚模組的擺置

由以上討論，我們可將叢聚限制的問題作一結論：在平面規劃時，若某一個序列對表示法滿足叢聚限制的定義，則稱該序列對為一組合理擺置 (feasible placement) 的解，因此最佳解即為：在此合理擺置內的所有解空間當中，其總面積為最小的。

⁴模組旋轉 90°，使其寬度與高度互換。

第四章 序列對表示法之叢聚性限制

本論文最大的貢獻在於將叢聚限制問題推廣至非分割性結構 (non-slicing structure) 的表示法上，我們選擇目前能處理最多條件限制的序列對 (Sequence-Pair) 表示法，同時其表示法也能夠處理多種不同型態的模組，最重要的是它具有 P-admissible 的解空間[14]的特徵，保證其任何解都可以對應到一組實際的擺置方式，由以上的這些優勢，促使我們在研究這個主題時，更具有實用性與說服力。

第一節 序列對表示法的叢聚限制問題

由於序列對在於顯示方向的資訊上，只有兩個模組間上、下或是左、右的相對關係，因此，並無法將所有已鄰接的叢聚模組視為一體，以利用一次上(下、左、右)方向的測試，來判斷是否是滿足叢聚的限制。依據前一章的問題定義：首先，我們假設所有的叢聚模組 C_i 已經事先叢聚在一起 (為合理的叢聚模組，因此，不能有其它 $f_j \in F$ 阻礙 C_i 叢聚，如圖 5 所示)，之後，若是能提出另一種序列對的判斷方法，檢驗 f_j 與 C_i 是否具有上、下、左、右的關係，假若檢驗結果為：在 C_i 的上、下、左、右的聯集之中沒有任何一個

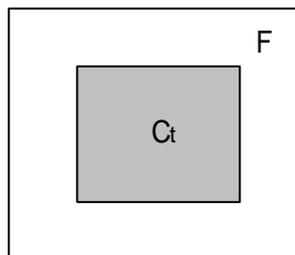


圖 5：合理叢聚模組的假設

f_j 存在，則表示 f_j 沒有阻礙 C_i 的叢聚；換句話說，即可滿足叢聚性限制，是一組合理的可行解 (feasible solution)，如此方式，便可以找出序列對中滿足叢聚限制的解空間了。

第二節 判斷方向的序列對表示法

序列對是由兩個序列 Γ_+ 及 Γ_- 所組成，如果要判斷兩個模組 a 、 b ，其操作方式如下：

- b 在 a 的上方：

$$M^T(a) = \{ b \mid \text{在 } \Gamma_+ \text{ 中 } b \text{ 排列在 } a \text{ 之前, 在 } \Gamma_- \text{ 中的次序 } b \text{ 在 } a \text{ 之後} \}$$

- b 在 a 的下方：

$$M^B(a) = \{ b \mid \text{在 } \Gamma_+ \text{ 中 } b \text{ 排列在 } a \text{ 之後, 在 } \Gamma_- \text{ 中的次序 } b \text{ 在 } a \text{ 之前} \}$$

- b 在 a 的左方：

$$M^L(a) = \{ b \mid \text{在 } \Gamma_+ \text{ 與 } \Gamma_- \text{ 中的排列次序, } b \text{ 皆在 } a \text{ 之前} \}$$

- b 在 a 的右方：

$$M^R(a) = \{ b \mid \text{在 } \Gamma_+ \text{ 與 } \Gamma_- \text{ 中的排列次序, } b \text{ 皆在 } a \text{ 之後} \}$$

序列對表示法只限於判斷兩個模組之間的相對關係，若是要運用在判斷兩個以上模組的集合 C_i 與 f_j 的方向關係，其原先的判斷方式就不可行了，因此，我們提出一種判斷叢聚模組集合與非叢聚模組的方式：首先，由第二章所介紹的序列對編碼步驟，可發現到，假使把平面規劃的模組給予迪卡兒作標，如圖 6(a) 所示，序列對中 Γ_+ 排列的次序大致會遵守由 $x+y=1$ 方向搜尋的結果（左上角的模組與右下角的模組必定分別排列在 Γ_+ 第一個與最末個）；相同的道理， Γ_- 排列的次序也大致會遵守 $x-y=0$ 方向搜尋的結果（左下角的模組與右上角的模組必定分別排列在 Γ_- 第一個與最末個），假設從同

一叢聚模組中取出左下、右上、左上與右下的四個角落，，如圖 6(b) 所示，其四個角落分別為 c_t^{LT} 、 c_t^{RB} 、 c_t^{LB} 、 c_t^{RT} 佔據著，則我們可規劃一條 c_t^{LT} 至 c_t^{RB} 方向的直線，對應至 $x+y=1$ 方向搜尋的結果，作為模擬 Γ_+ 排列的次序，同理， c_t^{LB} 至 c_t^{RT} 方向的直線，則可模擬 Γ_- 排列的次序，以判斷其方位。

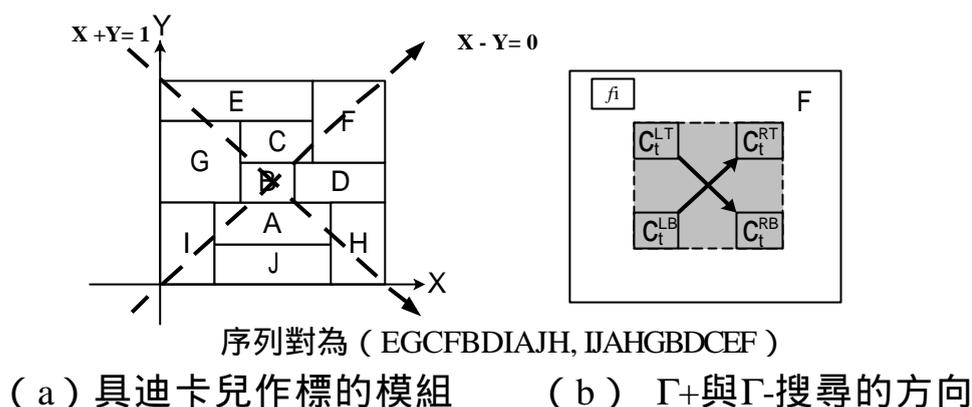


圖 6：序列對與模組的關係

假使需要判斷 f_j 是否在 C_t 的上方，其充要條件為 f_j 必須至少滿足在不在 c_t^{LB} 或是 c_t^{RB} 的下方才行（則 f_j 就會屬於在 C_t 的上、左、右方或是之內了），接著，我們利用 $M^T(a)$ 、 $M^B(a)$ 、 $M^L(a)$ 、 $M^R(a)$ 的性質，代入上述 c_t^{LT} 、 c_t^{RB} 、 c_t^{LB} 、 c_t^{RT} 的想法當中，並且依據 Γ_+ 與 Γ_- 序列對排列的次序，個別作分類，則可以得到：

● 在 Γ_+ 中：

假使 f_{j_+} 排列在 c_t^{LT} 之後：表示 f_{j_+} 可能在 c_t^{LT} 的下方或是右方 條件 1

假使 f_{j_+} 排列在 c_t^{RB} 之前：表示 f_{j_+} 可能在 c_t^{RB} 的上方或是左方 條件 2

● 在 Γ_- 中：

假使 f_j 排列在 c_t^{LB} 之後：表示 f_j 可能在 c_t^{LB} 的上方或是右方 條件 3

假使 f_j 排列在 c_t^{RT} 之前：表示 f_j 可能在 c_t^{RT} 的下方或是左方 條件 4

另一方面，我們將 F 的方向分為五類 $\{F^T, F^B, F^L, F^R, F^C\}$ ，若 f_j 在 C_t 的上方 (F^T)，則必需要滿足上述條件 2、3；若在下方 (F^B)，則需要滿足條件 1、4；若是在左方 (F^L)，則需要滿足條件 2、4；若在右方 (F^R)，則需要滿足條件 1、3，這樣的結果，再經歸納之後，就可判斷叢聚模組的集合 C_t 與非叢聚模組 $f_j \in F$ 的關係，完整的性質描述如下：

- 性質 F^T ：給予一序列對 $(\Gamma+, \Gamma-)$ ，令 $C_{RM+} \in C$ ， C_{RM+} 為在 $\Gamma+$ 中排列最右邊 (RightMost) 的叢聚模組，若是 f_j^T 在 $\Gamma+$ 的次序皆在 C_{RM+} 之前，且其 f_j^T 也符合在 $\Gamma-$ 中對應的 C_{RM+} 模組的次序之後，則我們稱這些 f_j^T 的集合是在 C_t 的上方，具有 F^T 的性質。
- 性質 F^B ：給予一序列對 $(\Gamma+, \Gamma-)$ ，令 $C_{LM+} \in C$ ， C_{LM+} 為在 $\Gamma+$ 中排列最左邊 (LeftMost) 的叢聚模組，若是 f_j^B 在 $\Gamma+$ 的次序皆在 C_{LM+} 之後，且其 f_j^B 也符合在 $\Gamma-$ 中對應的 C_{LM+} 的模組次序之前，我們稱這些 f_j^B 的集合是在 C_t 的下方，具有 F^B 的性質。
- 性質 F^L ：給予一序列對 $(\Gamma+, \Gamma-)$ ，令 $C_{RM+} \in C$ ， C_{RM+} 為在 $\Gamma+$ 中排列最右邊 (RightMost) 的叢聚模組，若是 f_j^L 在 $\Gamma+$ 以及 $\Gamma-$ 的次序分別皆在 C_{RM+} 與 $\Gamma-$ 中對應的 C_{RM+} 之前，我們稱這些 f_j^L 的集合是在 C_t 的左方，具有 F^L 的性質。
- 性質 F^R ：給予一序列對 $(\Gamma+, \Gamma-)$ ，令 $C_{LM+} \in C$ ， C_{LM+} 為在 $\Gamma+$ 中排列最左邊 (LeftMost) 的叢聚模組，若是 f_j^R 在 $\Gamma+$ 以及 $\Gamma-$ 的次序分別皆在 C_{LM+} 與 $\Gamma-$ 中對應的 C_{LM+} 之後，我們稱 f_j^R 的集合是在 C_t 的右方，具有 F^R 的性質。

- 性質 F^C : 給予一序列對 (Γ^+, Γ^-) , 若是 $f_j \in F$ 而且又同時屬於 $\{F^T\}$ $\{F^B\}$ $\{F^L\}$ $\{F^R\}$, 我們稱 f_j 在 C_i 之內, 具有 F^C 的性質。

由於當初我們假設所有叢聚模組 C_i 已事先叢聚在一起, 因此, 若存在 f_j 同時與 F^T 、 F^B 、 F^L 、 F^R 的模組集合有共同交集, 則可判斷 $f_j \in F$ 為阻礙 C_i 叢聚的模組, 亦顯示此序列對無法滿足叢聚性限制, 若判斷為無阻礙, 則此組序列對則稱為滿足叢聚性限制的可行解。

第三節 叢聚性限制的條件

我們使用以下的例子來介紹如何判斷叢聚限制的平面規劃, 首先, 假設有 10 個長、寬固定的矩形模組, 其編號分別為 ABCDEFGHIJ, 該序列對編碼為 $(EGCFBDIAJH, IJAHGBDCEF)$, 其中我們假設 A、B、C 為已經叢聚在一起的模組, 如圖 7 所示:

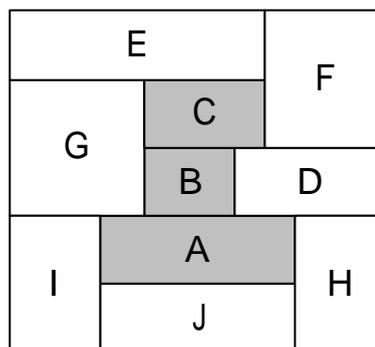


圖 7：叢聚限制的例子

- 步驟 1: 從 $(\Gamma^+, \Gamma^-) = (EGCFBDIAJH, IJAHGBDCEF)$ 找出叢聚模組 A、B、C 裡在 T^+ 中的 C_{RM^+} 與 C_{LM^+} , 所以 $C_{RM^+} = C$ 、 $C_{LM^+} = A$ 。

步驟 2：找出符合性質 F^T 、 F^B 、 F^L 、 F^R 的模組集合：

- 性質 F^T ：在 Γ_+ 的次序皆在 A 之前的 f_j 集合有 {EGFDI}，且同時符合在 Γ 的次序在 A 之後 f_j 集合有 {GDEF}，所以 {DEFG} 為滿足性質 F^T 的模組集合。
- 性質 F^B ：在 Γ_+ 的次序皆在 C 之後的 f_j 集合有 {FDIJH}，且同時符合在 Γ 的次序在 C 之前的 f_j 集合有 {IJDH}，所以 {DHIJ} 為滿足性質 F^B 的模組集合。
- 性質 F^R ：在 Γ_+ 的次序皆在 C 之後的 f_j 集合有 {FDIJH}，且同時符合在 Γ 中在 C 之後的 f_j 集合只有 {H}，所以 {H} 為滿足性質 F^R 的模組集合。
- 性質 F^L ：在 Γ_+ 的次序皆在 A 之前的 f_j 集合有 {EGFDI}，且同時符合在 Γ 中 A 之前的 f_j 集合有 {I}，所以 {I} 為滿足性質 F^L 的模組集合。

步驟 3：查看滿足 F^T 、 F^B 、 F^L 、 F^R 性質的模組是否有共同的交集， $\{DEFG\} \cap \{DHIJ\} \cap \{H\} \cap \{I\} = \emptyset$ ，表示叢聚目標的內部是無阻礙的，所以此組序列對證明是一組滿足叢聚性限制的可行解。

第四節 多組叢聚的限制

此法亦適用於擴展至判斷多組叢聚 (multi-clustering)，假設有 3 組叢聚的目標，即分別測驗 C_1 、 C_2 與 C_3 為叢聚模組時的狀況，若該組序列對能分別通過上述的條件限制，則顯示該組序列對編碼為滿足叢聚限制的可行解。

第五章 叢聚限制模組擺置的演算法

在這一章當中，我們延用上一章所提出的叢聚判斷方法，將其應用於模擬退火法(simulated annealing)當中，以解決具叢聚限制模組擺置(placement)的問題。

第一節 模擬退火法的參數設計

模擬退火法為一種機率攀登搜尋(probabilistic hill-climbing)的演算法[8]，它利用隨機過程，使其解在某些條件下，具有接受目標函數往較差條件改變的機會，這樣將可使其解跳越過局部最佳解的限制，進而更有機會求得全域的最佳解。

首先，在使用模擬退火法時，為了使序列對(Sequence-Pair)有效的產生下一個世代，因此，我們採用[11]的方法，利用三種隨機的方式擾動(Move)，分別為 $M1$ ：調換任意兩個在 Γ_+ 中的模組， $M2$ ：調換任意兩個在 Γ_- 中的模組， $M3$ ：任取一個模組旋轉其方向(rotated orientation)，使其寬度與高度互換；由於在擾動之後，並非所有序列對的解空間都能滿足叢聚限制，因此，再藉由我們所提出的判斷方式，能夠確保序列對編碼的每一個解，皆是合理可行解，又因為序列對具有 P-admissible 的性質，所以產生後的每一個合理可行解都能對應到一個唯一的擺置方式，這樣的方式，使得其平面規劃皆合理並且是有效的。

另一方面，其模擬退火法的計價函數(cost function)，依照之前的問題定義：在滿足合理的擺置方式內，求得面積需要能夠越小越好，所以其計價函數為包含所有模組的最小矩形面積。

第二節 模擬退火法的流程

模擬退火法的流程，如下頁圖 8 所示，給予起始解為所有的模組皆按照編號次序排成一行，之後再使用隨機擾動序列對的方式產生另一組新解，利用叢聚限制的條件，判斷擾動所產生新的組合是否滿足條件，若為否定，則要再進行擾動直到產生滿足叢聚限制的條件為止，若為肯定時，則需要再檢查其計價函數是否有減少，若為肯定，則擾動次數就累加一，若是計價函數沒有減少，則利用隨機過程，讓其解有機會往較差目標函數移動，遞回的一直作，直到擾動次數大於設定值為止（我們在實驗當中設定為該模組數量的 10 倍），每次到達設定的擾動次數後，溫度將依據一定的比率下降，如此一直遞回，直達溫度夠低或是面積的下降率已經收斂到一定的門閾值才可停止。

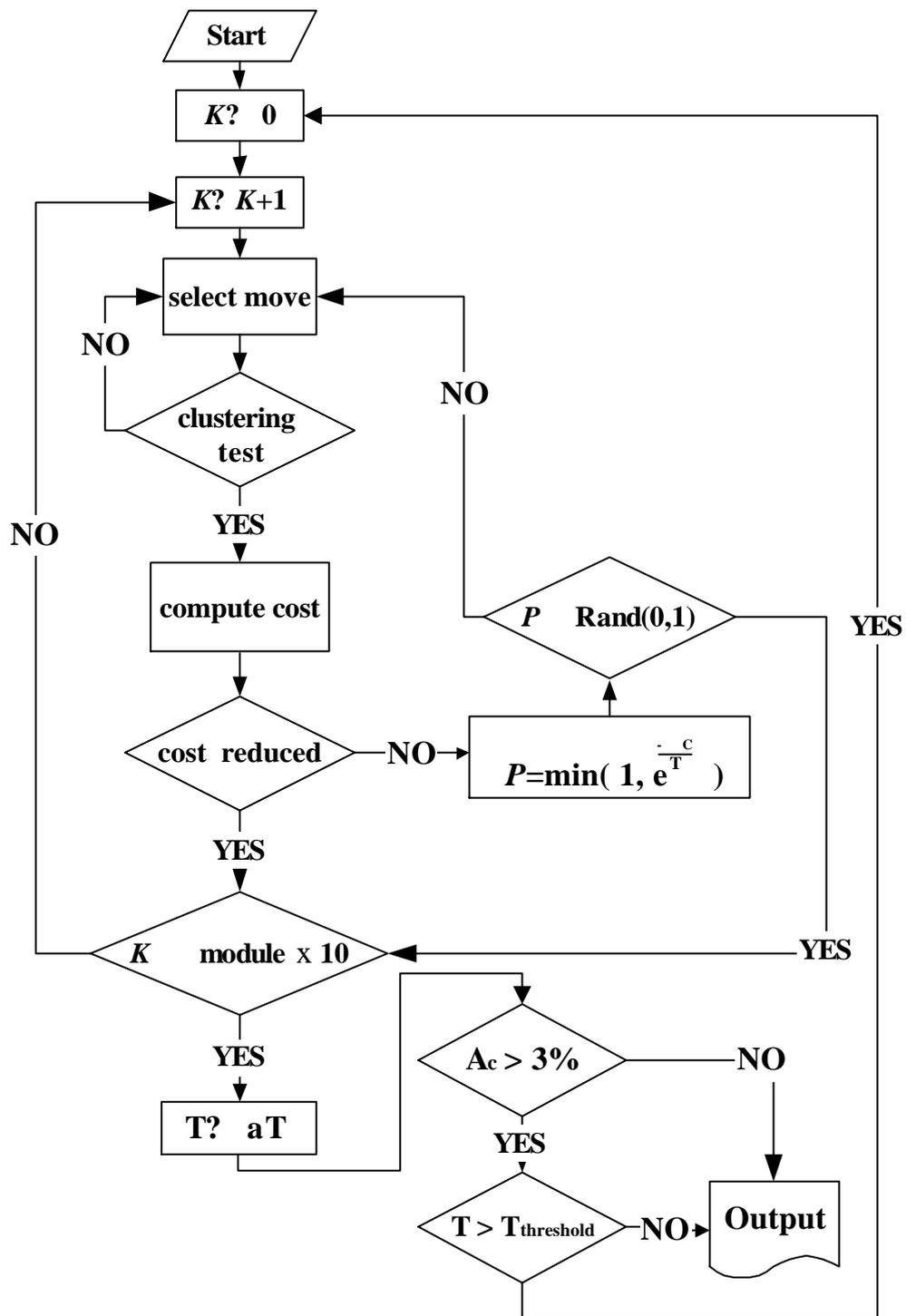


圖 8：模擬退火法的流程圖

第六章 實驗結果

第一節 測試的電路與環境設備

我們以五組 MCNC benchmark circuits 為樣本，分別是 ami33、ami49、apte、xerox 以及 hp，它們各有 33、49、9、10 與 11 個模組，並且我們以 Visual Basic 6.0 來實現測試這個演算法，實驗的環境是在 WIN98 的作業系統，CPU 為 AMD 1400 M Hz 與 256 M SDRAM 之下。

另一方面，設定模擬退火法的起始的溫度為 1000 度，溫度下降比率為 0.95，擾動方式 $M1 : M2 : M3$ 的比率為 40% : 40% : 20%，每一溫度內設定擾動成功的次數，要達該模組數量的 10 倍，才可依比率再下降其溫度，最佳解停止的條件為溫度小於 1 或是面積的下降率到達 3% 以下。

第二節 實驗結果

我們的實驗設計規劃如下：首先取無叢聚限制下平面規劃的數據，如表 5 所示，表 6~8 為取模組總數目的 20% 為叢聚的對象，並且使用不同編號的模組來測試其叢聚的效果，由於 ami33 與 ami49，模組數量較大，所以在表 6 與表 7 當中我們加入考量多組叢聚的情形，表 8 為 apte、xerox 與 hp 只考慮單組叢聚的狀況；表 5~8 當中每一項條件下皆實驗 10 次，取其平均值與最小值；表 9 為取表 6~8 之中各平均值與無叢聚限制的數據作比較，這樣便可以了解其演算法在有、無叢聚限制之間的差異了。

實驗結果顯示，使用我們叢聚限制的判斷方法，在考慮面積的最佳化的條件下，其 Dead Space 平均值的範圍在 2.82~12.% 之間，其所花費時間的平均值約在 7 秒至 120 分鐘之內；在有、無叢聚限制的比較上，其 Hp 的 Dead Space 平均值差異最大為 4.03 %，xerox 差異最小為 0%。

表 5：無叢聚限制狀況下的數據

Circuits	# of modules	Total area of Modules (m ²)	Average		Minimum	
			Dead Space (%)	Time (sec)	Dead Space (%)	Time (sec)
ami33	33	1.16	7.73	415.37	5.68	408.7
ami49	49	35.43	7.35	2101.12	6.15	1965.73
apte	9	46.56	2.4	15.56	1.84	15.05
xerox	10	19.32	8.95	21.16	6.42	17.22
hp	11	8.92	7.5	25.81	3.16	21.91

表 6：ami33 模組在叢聚限制狀況下的數據

Circuits	N	# of Clusters (Cluster Size)	Number of clustering modules	Average		Minimum	
				Dead Space (%)	Time (sec)	Dead Space (%)	Time (sec)
Ami33-1a	33	1 (7)	(6 7 9 18 20 22 24)	9.84	1699.38	9.88	1653.26
Ami33-1b	33	1 (7)	(5 7 11 13 14 27 29)	9.46	1739.13	9.07	1473.27
Ami33-1c	33	1 (7)	(19 20 21 23 24 28 29)	10.63	1769.54	9.65	1733.74
Ami33-3a	33	3 (4,4,3)	(2 6 9 7, 24 5 22 3, 20 1 18)	10.22	1633.2	9.56	1599.32
Ami33-3b	33	3 (4,4,3)	(5 7 11 13, 14 27 30 19, 22 25 29)	10.48	1532.82	9.8	1443.48
Ami33-3c	33	3 (4,4,3)	(1 2 3 18, 19 20 21 23, 24 28 29)	11.23	1601.56	9.88	1562.11
Ami33-4a	33	4 (3,3,3,2)	(2 6 9, 7 24 5, 22 3 20, 1 18)	10.58	1681.47	10.1	1622.52
Ami33-4b	33	4 (3,3,3,2)	(5 7 11, 13 14 27, 30 19 22, 25 29)	9.66	1683.26	10.24	1543.48
Ami33-4c	33	4 (3,3,3,2)	(1 2 3, 18 19 20, 21 23 24, 28 29)	10.56	1675.3	10.7	1623.21
Ami33-5a	33	5 (3,2,2,2,2)	(2 6 9, 7 24, 5 22, 3 20, 1 18)	9.88	1785.52	10.2	1743.5
Ami33-5b	33	5 (3,2,2,2,2)	(5 7 11, 13 14, 27 30, 19 22, 25 29)	9.07	1818.14	9.07	1791.14
Ami33-5c	33	5 (3,2,2,2,2)	(1 2 3, 18 19, 20 21, 23 24, 28 29)	10.56	1766.1	10.03	1722.02

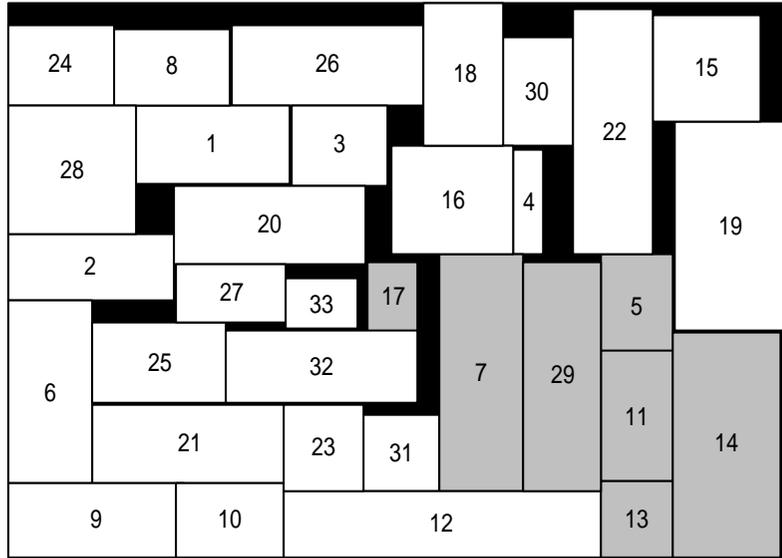


圖 9：Ami33-1b 最佳的擺置圖形

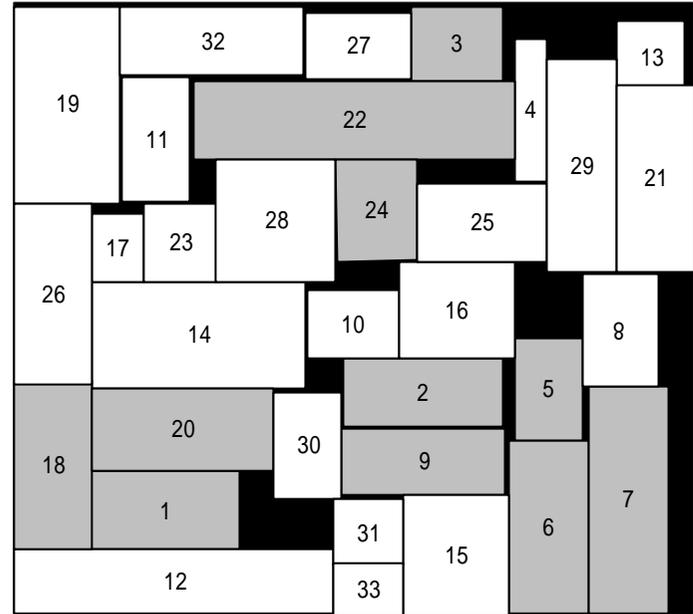


圖 10：Ami33-3a 最佳的擺置圖形

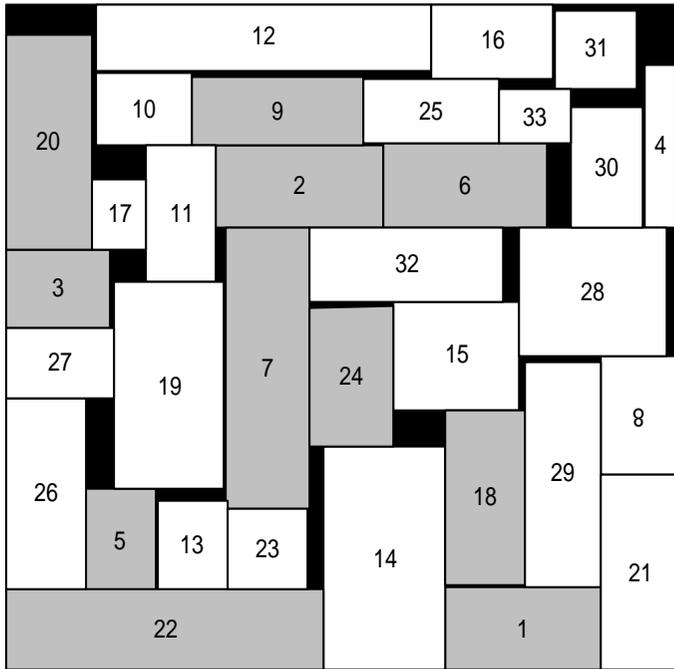


圖 11：ami33-4a 最佳的擺置圖形

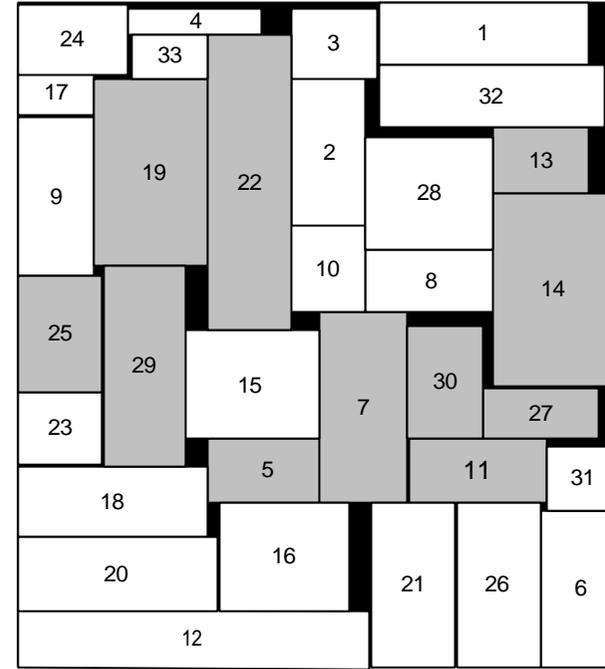


圖 12：ami33-5b 最佳的擺置圖形

表 7：ami49 在叢聚限制的狀況下的數據

Circuits	N	# of Clusters (Cluster Size)	Number of clustering modules	Average		Minimum	
				Dead Space (%)	Time (sec)	Dead Space (%)	Time (sec)
ami49-1a	49	1 (10)	(6 7 8 9 10 11 12 13 15 16)	8.54	4093.32	8.02	3968.57
ami49-1b	49	1 (10)	(13 14 27 30 29 32 35 39 40 41)	10.84	4122.51	10.6	4052.47
ami49-1c	49	1 (10)	(19 16 21 26 22 27 32 33 38 42)	11.42	4371.85	9.88	4125.2
ami49-3a	49	3 (6, 5, 5)	(6 7 8 9 10 11, 12 13 15 16 17, 18 19 20 21 22)	10.5	5632.12	9.6	5400.74
ami49-3b	49	3 (6, 5, 5)	(5 7 11 13 14 27, 30 29 32 35 39, 40 41 43 44 45)	10.6	6222.52	10.3	6039.45
ami49-3c	49	3 (6, 5, 5)	(2 5 6 7 9 14, 19 16 21 26 22, 27 32 33 38 42)	12.2	6543.48	10.52	6443.59
ami49-4a	49	4 (4, 4, 4, 4)	(6 7 8 9, 10 11 12 13, 15 16 17 18, 19 20 21 22)	11.63	6723.21	10.3	6296.21
ami49-4b	49	4 (4, 4, 4, 4)	(5 7 11 13, 14 27 30 29, 32 35 39 40, 41 43 44 45)	10.28	6743.5	10.96	6565.95
ami49-4c	49	4 (4, 4, 4, 4)	(2 5 6 7, 9 14 19 16, 21 26 22 27, 32 33 38 42)	11.97	6791.14	9.83	6702.91
ami49-5a	49	5(4, 3, 3, 3, 3)	(6 7 8 9, 10 11 12, 13 15 16, 17 18 19, 20 21 22)	12.16	7308.19	9.52	7258.37
ami49-5b	49	5(4, 3, 3, 3, 3)	(5 7 11 13, 14 27 30, 29 32 35, 39 40 41, 43 44 45)	10.78	7459.56	10.22	7119.23
ami49-5c	49	5(4, 3, 3, 3, 3)	(2 5 6 7, 9 14 19, 16 21 26, 22 27 32, 33 38 42)	10.82	7289.89	10.76	7145.46

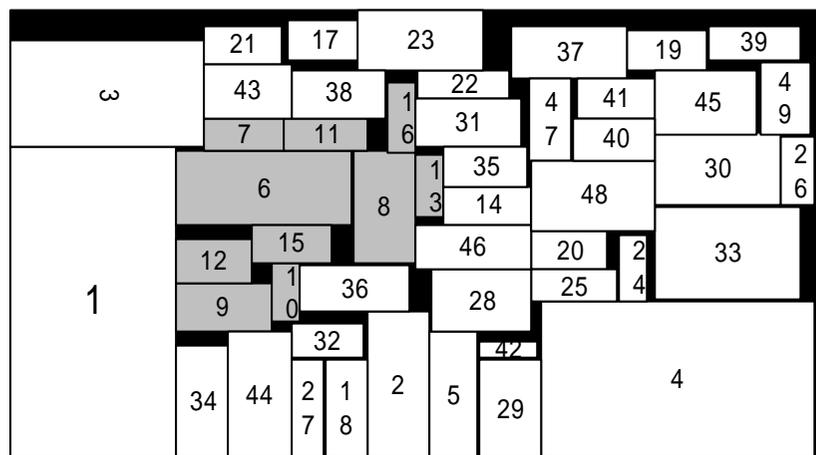


圖 13：ami49-1a 最佳的擺置圖形

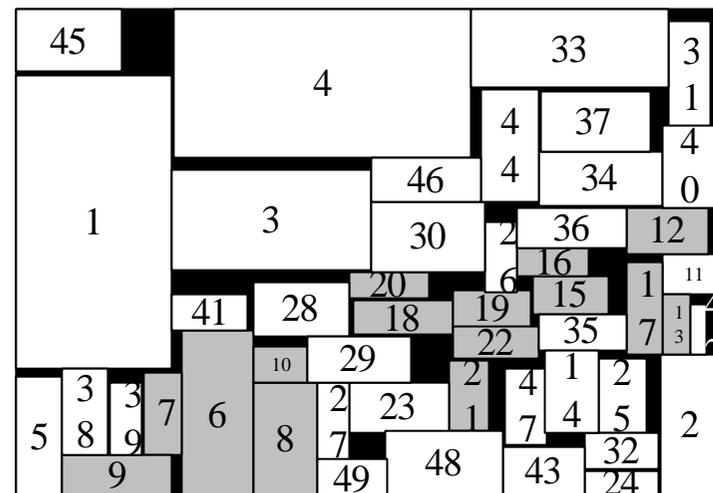


圖 14：ami49-3a 最佳的擺置圖形

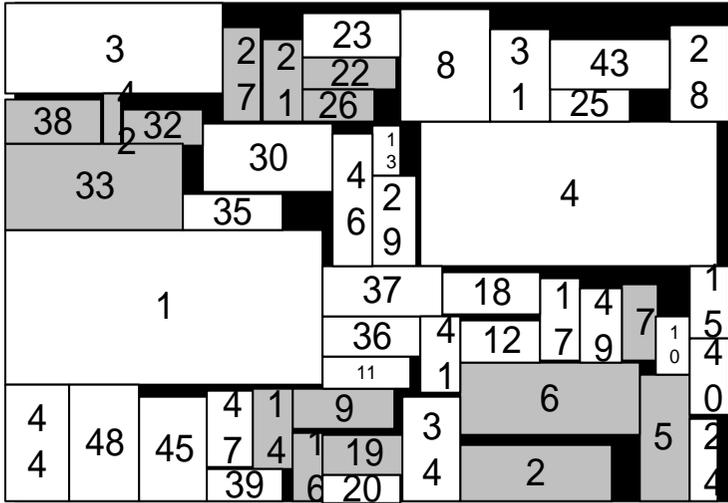


圖 15 : ami49-4c 最佳的擺置圖形

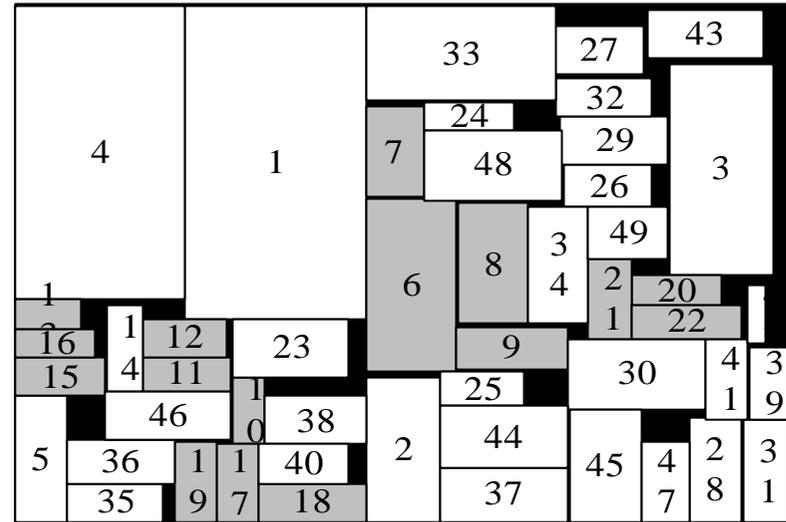


圖 16 : ami49-5a 最佳的擺置圖形

表 8：其他在叢聚限制的狀況下的數據

Circuits	n	# of Clusters (Cluster Size)	Number of clustering modules	Average		Minimum	
				Dead Space (%)	Time (sec)	Dead Space (%)	Time (sec)
apte-1a	9	1 (2)	(1,2)	2.82	17.09	2.82	16.82
apte-1b	9	1 (2)	(3,5)	3.55	18.33	2.96	17.59
apte-1c	9	1 (2)	(5,8)	3.92	17.52	2.82	17.14
xerox-1a	10	1 (2)	(1,2)	8.38	24.23	8.37	18.34
xerox-1b	10	1 (2)	(3,5)	9.58	24.59	7.31	23.02
xerox-1c	10	1 (2)	(5,8)	8.89	25.89	7.55	21..89
hp-1a	11	1 (2)	(1,2)	10.83	28.18	10.41	25.03
hp-1b	11	1 (2)	(3,5)	11.02	27.03	10.18	27.84
hp-1c	11	1 (2)	(5,8)	11.33	28.66	9.4	27.12

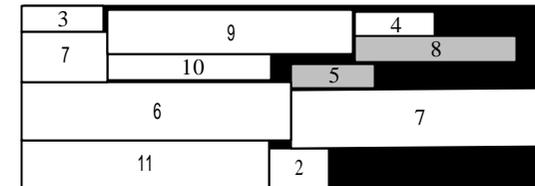
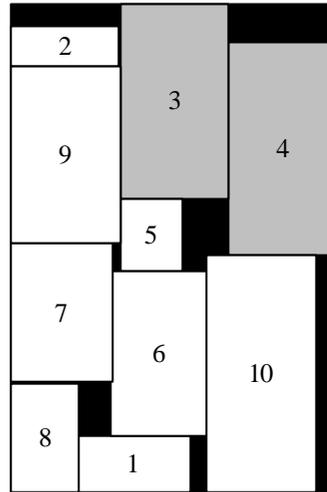
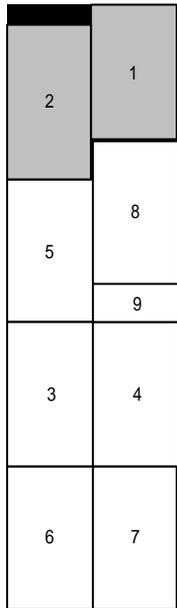


圖 18 : xerox-1b 最佳的
擺置圖形

圖 19:hp-1c 最佳的
擺置圖形

表 9：叢聚限制與無叢聚限制相差的比率⁵

Circuits	Average		Minimum	
	Dead Space (%)	Time (sec)	Dead Space (%)	Time (sec)
ami33	2.45	1283.42	4.17	1217.22
ami49	3.36	4007.32	3.89	5926.18
apte	1.03	2.09	1.03	2.13
xerox	0.00	3.74	1.32	3.46
Hp	4.03	2.15	6.84	4.75

⁵ 為表 6-8 當中各組叢聚限制的平均值減去表 5 中無叢聚限制數據的結果。

第七章 結論與建議

本篇論文最大的貢獻為利用非分割性結構的序列對表示法，討論具有叢聚限制的平面規劃，我們發展出一套判斷是否能符合叢聚限制的演算法，以正確地篩除序列對當中非叢聚的解空間，同時也很容易的來延伸處理多組叢聚的狀況。雖然非分割性結構，其解空間較分割性結構多，運算處理時間也較長，但是，其彈性很高，我們將演算法套至模擬退火法之後，利用 MCNC benchmark Circuits 來作測試，實驗顯示，使用我們叢聚限制的判斷方法，在考慮面積的最佳化的條件下，其 Dead Space 平均差異值在 0~4.03 % 之間，其所花費時間也在可接受範圍之內，其結果算是蠻不錯的。

另一方面，由於當初模擬退火法時使用的計價函數只考慮到面積的最佳化，沒有考慮繞線因素來驗證叢聚的結果，所以本論文嚴格的來說只能算是提出判斷叢聚限制的演算法，如果同時兼顧面積與繞線長度的最佳化，那將會使問題變得十分複雜，因此至目前為止，具叢聚限制的相關文獻在這一方面的討論也都相當缺乏，但是反觀在真實的 IC 設計的流程當中，使用平面規劃之後，往往會再要求的繞線長度越短越好，因此如何能夠同時達成面積和繞線長度的最佳化，這樣的研究是我們非常引領期待的。

第八章 參考文獻

- [1] F. Balasa and K.V. Lampaert, "Symmetry within the Sequence-Pair Representation in the Context of Placement for Analog design," *IEEE Trans. on CAD of IC's and Systems*, Vol. 19, No. 7, pp. 721-731, July 2000.
- [2] Yun-Chih Chang, Yao-Wen Chang, Guang-Ming Wu and Shu-Wei Wu, "B*-Trees: A New Representation for Non-Slicing Floorplans," *Proc. of ACM/IEEE Design Automation Conference (DAC-2000)* , pp. 458-463, LA, CA, June 2000.
- [3] K. Fujiyoshi and H. Murata, "Arbitrary Convex and Concave Rectilinear Block Packing Using Sequence-Pair," *Proc. ISPD*, pp. 103-110, 1999.
- [4] P. N. Guo, C. K. Cheng and T. Yoshimura, "An O-Tree Representation of Non-Slicing Floorplan and Its Applications," *Proc. on IEEE/ACM Design Automation Conference*, pp. 268-273, 1999.
- [5] Xianlong Hong, Sheqin Dong, Gang Huang, Yuchun Ma, Yici Cai, Chung-Kuan Cheng and Jun Gu, "A Non-Slicing Floorplanning Algorithm Using Corner Block List Topological Representation," *IEEE APCCAS*, pp. 833 -836, 2000.
- [6] Y.-H. Jiang, J. Lai and T.-C. Wang, "Module Placement with Pre-Placed Modules Using the B*-Tree Representation," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)* , pp. 347-350, 2001.
- [7] M. Z. Kang and W. Dai, "General Floorplanning with L-shaped, T-shaped and Soft Blocks Based on Bounded Slicing Grid Structure," *Proc. Asia and South Pacific Physical Design Automation Conference (ASP-DAC)* , pp. 265-270, 1997.

- [8] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [9] E.-C. Liu, T.-H. Lin and T.-C. Wang, "On Accelerating Slicing Floorplan Design with Boundary Constraints," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. III399-III402, 2000.
- [10] J. Lai and T.-C. Wang, "Module Placement with Boundary Constraint Based on BSG-Structure," *Proc. of the 10th VLSI Design/CAD Symposium*, pp. 43-46, 1999.
- [11] J. Lai, M.-S. Lin, T.-C. Wang and L.-C. Wang, "Module Placement with Boundary Constraints Using the Sequence-Pair Representation," *Proc. Asia and South Pacific Physical Design Automation Conference (ASP-DAC)*, pp. 515-520, 2001.
- [12] J.-M. Lin and Y.-W. Chang, "TCG : A Transitive Closure Graph Based Representation for Non-Slicing Floorplans," *Proc. of ACM/IEEE Design Automation Conference (DAC-2001)*, pp. 764-769, Las Vegas, NV, June 2001.
- [13] Rui Liu, Xianlong Hong, Sheqin Dong, Jun Gu, Chung-Kuan Cheng and Yici Cai, "Module Placement with Boundary Constraints Using O-Tree Representation," *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2002.
- [14] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair," *IEEE Trans. on CAD*, Vol. 15, No. 12, pp.1518-1524, Dec. 1996.
- [15] H. Murata and Ernest S. Kuh, "Sequence Pair Based Placement Method for Hard/ Soft/Pre-placed Modules," *Proc. Internal Symposium on Physical Design*, pp. 167-172, 1998.

- [16] Yuchun Ma, Sheqin Dong, Xianlong Hong, Yici Cai, C.K. Cheng and Jun Gu, "VLSI Floorplanning with Boundary Constraints Based on Corner Block List," *Proc. Asia and South Pacific Physical Design Automation Conference (ASP-DAC)*, pp. 509-514, 2001.
- [17] S. Nakatake, M. Furuya and Y. Kajitani, "Module Placement on BSG-Structure with Pre-Placed Modules and Rectilinear Modules," *Proc. Asia and South Pacific Physical Design Automation Conference (ASP-DAC)*, pp. 571-576, 1998.
- [18] Y.-X. Pang, F. Balasa, K.V. Lampaert and C.-K. Cheng, "Block Placement with Symmetry Constraints Based on the O-Tree Non-Slicing Representation," *Proc. of the 37th ACM/IEEE Design Automation Conference (DAC)*, pp. 464-467, Los Angeles CA, June 2000.
- [19] W. S. Yuen and F. Y. Young, "Slicing Floorplans with Clustering Constraints," *IEEE Asia South Pacific Design Automation Conference*, pp. 503-508, 2001.
- [20] D. F. Wong and C. L. Liu, "A New Algorithm for Floorplan Design," *Proc. of ACM/IEEE Design Automation Conference*, pp. 101-107, 1986.
- [21] D. F. Wong and C. L. Liu, "Floorplan Design for Rectangular and L-shaped Modules," *Proc. ICCAD*, pp. 520-523, 1987.
- [22] F. Y. Young and D. F. Wong, "Slicing Floorplans with Pre-placed Modules," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 252-258, 1998.
- [23] 衣懷恩, "以 B*樹處理邊界條件限制下的擺置問題之研究," 交通大學資訊科學研究所, 碩士論文, 2001 年 6 月。