

國立交通大學

應用數學系

碩士論文

以連通控制集做為無線感測網路
虛擬骨幹之研究

Using a Connected Dominating Set
as the Virtual Backbone of
a Wireless Sensor Network

研究生：羅健峰
指導教授：陳秋媛

中華民國九十九年六月

以連通控制集做為無線感測網路虛擬骨幹之研究
Using a Connected Dominating Set
as the Virtual Backbone of
a Wireless Sensor Network

研 究 生: 羅 健 峰

Student: Cheng-Feng Lo

指 導 教 授: 陳 秋 媛

Advisor: Chiuyuan Chen



Submitted to Department of Applied Mathematics
College of Science

National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master

in
Applied Mathematics

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

以連通控制集做為無線感測網路虛擬骨幹之研究

研究生：羅健峰

指導老師：陳秋媛 教授

國立交通大學

應用數學系



中文摘要

在無線感測網路中，節點以及節點之間的連接關係可以用圖來表示，如此一來，此圖的連通控制集就可以對應當成原本的無線感測網路的骨幹，此一骨幹可用以提升訊息傳遞的效率。因此找出一個給定的圖的最小連通控制集是許多學者們探討的問題，文獻中已證明了找出一個給定的圖的最小連通控制集是 NP-難的問題，於是退而求其次的有近似演算法的提出。在1999年,Wu 和 Li 提出了一個近似演算法 (為方便, 稱其為 Wu 和 Li 的演算法), 在2006年,Cokuslu 等人提出了一個近似演算法 (為方便, 稱其為 Cokuslu 的演算法)。Cokuslu等人的文章中只分析了演算法的效益、訊息複雜度、以及時間複雜度, 並未證明演算法的正確性。在這篇論文中, 我們將證明 Cokuslu 的演算法的正確性, 換句話說, 我們將證明 Cokuslu 的演算法所得的結果是一個連通控制集。此外, 我們還做了 Wu 和 Li 的演算法及 Cokuslu 的演算法的比較。

關鍵詞：無線感測網路、連通控制集、虛擬骨幹、演算法。

Using a Connected Dominating Set as the Virtual Backbone of a Wireless Sensor Network

Student: Cheng-Feng Lo

Advisor: Chiuyuan Chen

Department of Applied Mathematics

National Chiao Tung University

Hsinchu, Taiwan 30050

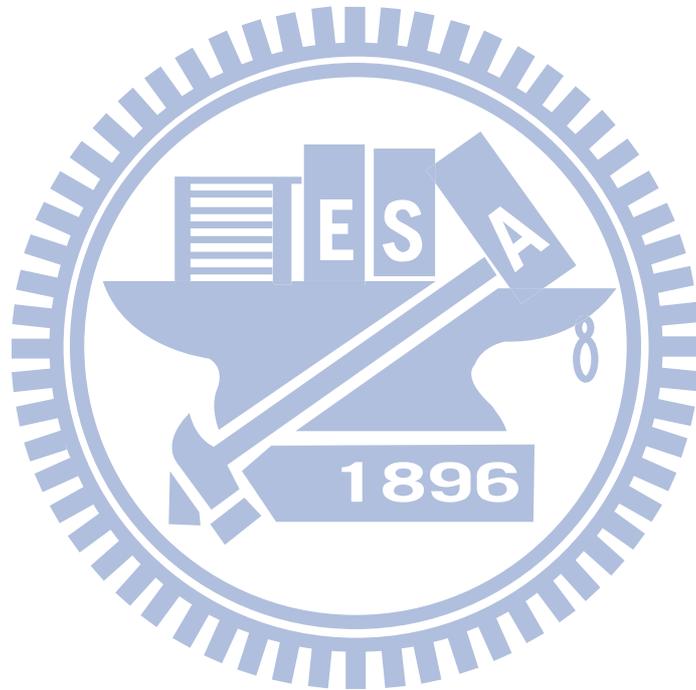
Abstract

In a wireless sensor network, the relationship between nodes can be modeled by using a graph. Consequently, a connected dominating set of such a graph is usually used to serve as a virtual backbone of the original wireless sensor network. Thus finding a minimum connected dominating set of a graph becomes a problem discussed by many researchers. It has been proven that this problem is NP-hard. Hence many researchers consider finding approximation solutions instead of the optimal solution and many approximation algorithms have been proposed. In particular, in 1991, Wu and Li proposed an approximation algorithm (call it Wu and Li's algorithm for convenience). In 2006, Cokuslu et al. proposed another approximation algorithm (call it Cokuslu's algorithm for convenience), which is an improvement of Wu and Li's algorithm. Notice that Cokuslu et al. only provided the performance, the time complexity, and the message complexity analyses; they did not prove the correctness of their algorithm. In this thesis, we will prove the correctness of Cokuslu's algorithm (i.e., we will prove that Cokuslu's algorithm obtains a connected dominating set. Also, we will give a comparison between Wu and Li's algorithm and Cokuslu's algorithm.

Keywords: wireless sensor network, connected dominated set, virtual backbone, algorithm.

誌謝

在這裡我想感謝一些老師。首先是感謝陳秋媛老師，感謝一路走來她都耐心的陪著我，當我犯錯時，老師都一一指出，很仔細地告訴我要注意出錯的問題在哪，有哪些細節要注意。感謝翁志文老師在我修習代數課時給我很多的幫助，讓我建立正確的觀念。感謝黃大原老師，在修大原老師的課時，老師都很親切的回答我所有的問題。然後，感謝藍國元和邱鈺傑學長，常常在我出現問題的時候，給我一些意見讓我在解決問題時參考。最後感謝我的同學，吳思賢、黃思綸，不厭煩的花時間陪我在研究室中一起討論我遇到的問題。



目錄

中文摘要	i
Abstract	ii
誌謝	iii
目錄	iv
圖目錄	v
表目錄	vi
1 簡介	1
2 相關定義	3
3 前人結果	5
4 Wu 和 Li 的演算法以及 Cokuslu 的演算法	8
4.1 Wu 和 Li 的演算法	8
4.2 Cokuslu 的演算法	8
5 證明 Cokuslu 的演算法的正確性	11
5.1 證明 Cokuslu 的演算法之輸出為控制集	11
5.2 證明 Cokuslu 的演算法所輸出的控制集為連通	12
6 結語	14
參考文獻	15
附錄	17

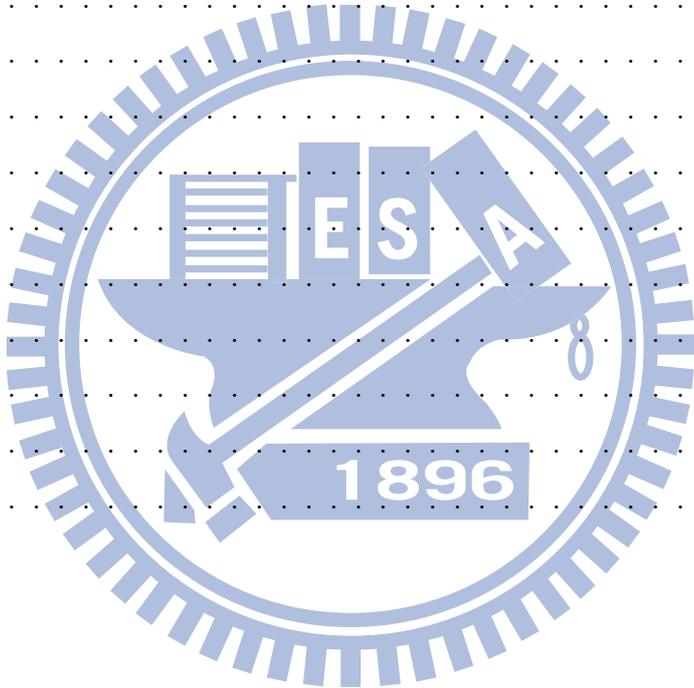
圖目錄

1	Wu和 Li 的演算法範例	9
2	Cokuslu的演算法範例	10



表目錄

1	17
2	17
3	18
4	18
5	18
6	19
7	19
8	19
9	20
10	20
11	20
12	21
13	21
14	21
15	22
16	22
17	22



1 簡介

無線感測網路是由一堆的感測器組成，感測器們一起執行特定任務。每個感測器可以獨立工作，感測器的功能是感測研究者有興趣的事件，可能是溫度、聲音、溼度等等。由感測器所形成的網路不同於一般的有線網路，無線感測網路沒有階層，沒有固定的網路設施，感測器都執行一樣的功能，感測器利用無線的傳輸媒介直接互相交換訊息，但是訊息傳輸距離是有限的，所以兩個不在彼此傳輸距離內的感測器要傳送訊息時，必須透過路徑上多個中間感測器轉送。因此，一個感測器除了扮演終端系統的角色，同時也要扮演轉送訊息的角色。因為要知道如何轉送訊息，每個感測器需要儲存一些路由資訊。感測器在收集路由資訊的過程，**廣播**(broadcast) 是最直接且常使用的方式。廣播的目的是讓網路中節點都收到資訊。有一種廣播的方式是：感測器收到一個廣播訊息，感測器檢查是否已經收過，若是第一次收到的廣播資訊就繼續傳送給所有傳送距離內的感測器，否則不傳送，這種方式稱為泛洪式廣播。但是利用廣播傳遞資訊的方式會造成資料重複、頻道使用的競爭、及訊息的碰撞的問題，這三個問題就是所謂的廣播風暴問題。在無線網路中感測器能源通常是有限的，廣播也可能造成能源的浪費。

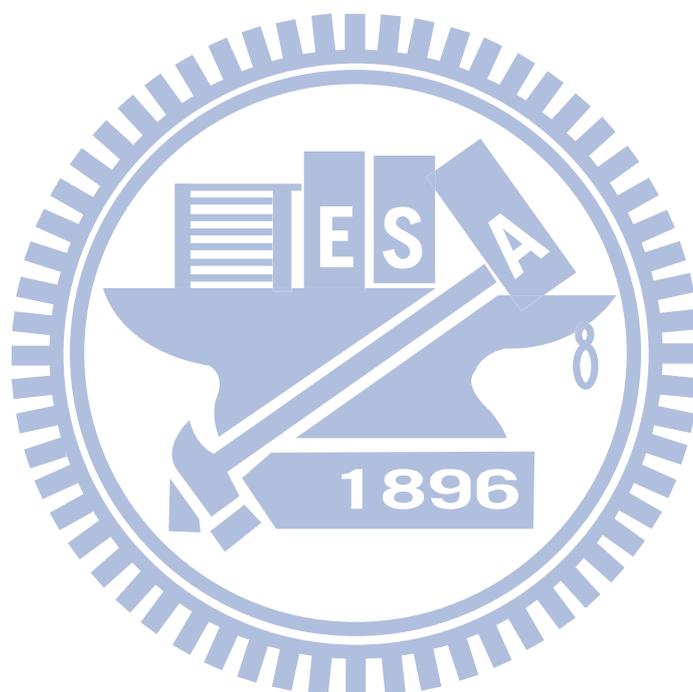
因此為了讓訊息的傳送負載在無線感測網路中降低，降低廣播風暴所帶來的影響，有人提出了**虛擬骨幹**的概念，就是找出一組感測器，讓資訊轉送的工作交給這些感測器，並且資訊的傳遞在到達目的地之前，僅在這些節點中流傳，而不是讓資料流竄在整個網路，藉以減輕整個網路的負擔。用來當做虛擬骨幹的感測器必須要滿足幾個條件：(1) 網路中不屬於骨幹的感測器傳輸半徑內至少要有一個在骨幹內的感測器；(2) 在骨幹內，感測器間的訊息要能夠不靠骨幹外的感測器就能互傳。

在無線感測網路中找出**虛擬骨幹**的問題如果用數學中的圖論來敘述的話，就是圖論中的**連通控制集**(connected dominating set) 的問題，所以將無線感測網路就對應到一個圖(graph)，在網路中找**虛擬骨幹**的問題就等同於在找一個圖中的**連通控制集**(connected dominating set) 的問題，但是這個集合內的點越少越好，因此要找的是**最小連通控制集**(minimum connected dominating set)。因為這個緣故，在無線網路蓬勃發展的同時，找出**連通控制集**的演算法陸續被提出，同時也研究**控制集**的變化及限制，證明關於**控制集**在某些特定條件下的性質等等。

在這篇論文中，我們將介紹**連通控制集**的相關研究歷程以及整理相關研究結果。在這些已知的結果之中，文獻[12]所提出的演算法是最有名的，這是一個**分散式演算法**(distributed algorithm)，我們會對這個演算法做較多的說明，以下並稱其為Wu 和 Li 的演算法。文獻[4]所提出的演算法是改進Wu 和 Li 的演算法，以下稱 Cokuslu 的演算法。文獻[4]只分析了演算法的效益、訊息複雜度、以及時間複雜度，並未證明演算法的正確性。本論文的目的在於證明Cokuslu 的演算法的正確性，換

句話說,我們將證明 Cokuslu 的演算法所得的結果是一個連通控制集;此外,我們還做了 Wu 和 Li 的演算法及 Cokuslu 的演算法的比較。

本論文的章節安排如下: 在第二小節中,我們將介紹相關定義; 在第三小節中,我們將介紹前人結果; 在第四小節中,我們將介紹 Wu 和 Li 的演算法及 Cokuslu 的演算法; 在第五小節中,我們將證明 Cokuslu 的演算法的正確性; 第六小節為結論。



2 相關定義

以下我們將介紹本論文中所要用到的名詞定義。

定義 1. 一個圖(graph) G 由點集合(vertex set) $V(G)$ 和邊集合(edge set) $E(G)$ 組成, 邊集合內的元素稱為邊, 一個邊就是一對點 (u, v) , 其中 u 和 v 都屬於 $V(G)$, 簡記作 uv 。

定義 2. 當 u 跟 v 是圖 G 某個邊上的點時, 則說 u 跟 v 相鄰(adjacent) 或者 u 跟 v 是鄰居(neighbor), 以 $u \leftrightarrow v$ 表示, 反之, 以 $u \nleftrightarrow v$ 表示。 u 在 G 中的所有鄰居以 $N(u)$ 表示; $N[u] = N(u) \cup \{u\}$ 。 $N(u)$ 的元素個數稱為 u 的分支度(degree), 以 $degree(u)$ 表示。

定義 3. 若圖 G 中的點 v , 其分支度為一, 則稱此點 v 為葉子(leaf)。

定義 4. 若一個圖 G 的任兩點都相鄰, 則 G 為一個完全圖(complete graph)。

定義 5. H 和 G 為兩個圖。 H 滿足 $V(H) \subset V(G)$ 且 $E(H) \subset E(G)$, 則 H 為一個圖 G 的子圖(subgraph), 以 $H \subseteq G$ 表示。

定義 6. 對於一個圖 G 中的兩點 u 和 v , 一個 u, v -walk 是一個序列 $u = v_0, e_1, v_1, \dots, e_k, v_k = v$, 使得 $e_i = v_{i-1}v_i$, 其中 v_i 屬於 $V(G)$, e_i 屬於 $E(G)$, 對 $1 \leq i \leq k$ 。若一個 u, v -walk 上沒有重複的點, 則稱為 u, v -path, 簡記為 $u = v_0, v_1, \dots, v_k = v$ 。若一個 u, v -walk, $u = v_0, e_1, v_1, \dots, e_k, v_k = v$, 所有的點中, 只有 u 和 v 相同, 則稱此 u, v -walk 為一個 cycle。一個 u, v -walk, u, v -path 或一個 cycle 的長度(length) 是指在序列中邊的個數, 所有 u, v -path 中邊數最少者為 u 到 v 的最短路徑。

定義 7. 一個圖 G 中, 所有點 $u, v \in V(G)$ 都存在一條 u, v -path, G 稱為連通圖(connected graph), 或說 G 是連通的(connected); 一個子圖是連通的, 則稱為連通子圖。

定義 8. 一個圖 G 是一個樹(tree) 表示 G 是一個沒有 cycle 的連通圖。

定義 9. 考慮一個圖 G 的子圖 T , 若 $V(T) = V(G)$ 且 T 為一個樹, 則稱 T 為 G 的擴張樹(spanning tree)。

定義 10. G 是一個圖, 若 $H \subseteq G$ 是一個連通子圖, 且沒有任何連通子圖 $H' \subseteq G$ 使得 $H \subset H'$, 則 H 為一個圖 G 的連通元件(connected component), 特別的是, 一個連通圖只有一個連通元件。

定義 11. 對一個圖 G , 令 $e \in E(G)$ 、 $v \in V(G)$ 、 $M \subseteq E(G)$ 、 $S \subseteq V(G)$ 。 $G - M$ 表示 G 扣掉 M 的子圖, 其中 $V(G - M) = V(G)$ 、 $E(G - M) = E(G) - M$, 而 $G - \{e\}$ 直接表示成 $G - e$ 。 $G - S$ 表示 G 扣掉 S 的子圖, 其中 $V(G - S) = V(G) - S$ 、 $E(G - S) = E(G) - \{(x, y) \mid x \text{ 或 } y \text{ 屬於 } S\}$, 而 $G - \{v\}$ 直接表示成 $G - v$ 。 $G[S]$ 稱為 S 的生成子圖, 表示 $G - \bar{S}$, 其中 $\bar{S} = V(G) - S$ 。

定義 12. 在一個圖 G 中, 令 $S \subseteq V(G)$ 。若 $G[S]$ 沒有邊, 也就是 $E(G[S])$ 是空集合, 則稱 S 為 G 中一個獨立集(independent set)。若 S 為 G 中的一個控制集(dominating set), 則每一個不屬於 S 的點都有一個鄰居在 S 。若 S 為 G 中的一個控制集且 $G[S]$ 連通, 則 S 是 G 的一個連通控制集(connected dominating set)。



3 前人結果

連通控制集這個名詞是在七零到八零年代時被提出，在無線感測網路的發展中，繼續了這部份的研究。不幸的是，找出一個圖的最小連通控制集已被證明是 NP-難的問題[5]，換句話說，不太可能有多項式時間內找出最小連通控制集；所以，目前的研究傾向於找出近似解，也就是提出一個圖的連通控制集的近似演算法。顧名思義，近似演算法所找出來的解並不一定是最小的，只是接近最小的，所以還必須要考慮近似因子 (approximation ratio)。假設一個圖的點數為 n ，此圖的最小連通控制集的大小是 C ，由近似演算法找出來的連通控制集的大小是 C' ，如果有一函數為 $\rho(n)$ ，使得 $\rho(n) \geq \max(C/C', C'/C)$ ，則稱 $\rho(n)$ 為近似演算法的近似因子，也就是近似演算法所找出來的連通控制集的大小 C' 保證不會超過 $\rho(n)C$ 。這些演算法可以分為兩大類：分散式，集中式。

在1998年,Guha 等人[6]提出了兩個集中式的近似演算法，都是使用貪婪法則(greedy strategies)。Guha等人的第一個演算法想法如下：一開始圖中的點都是標為白色，第一步驟先挑白色鄰居最多的點，然後將這個點標為黑色，這個點的鄰居全都標為灰色，接著，重複的挑選有最多白色鄰居的灰色點，將這個灰色的點標為黑色，鄰居標為灰色；或者是挑選有最多白色鄰居的一對點（一個灰點和一個白點），將這兩個點都標為黑色，鄰居標為灰色。當圖中不再存在有白色的點的時候，演算法結束。這個演算法的近似因子是 $2(1 + H(\delta))$ ，其中 H 表示調和函數， δ 表示所有點的最大分支度。

對於無線感測網路來說，因為感測器的能源有限，若是網路感測器數量較多的時候，由一個感測器來做集中式的計算並不實用，所以就考慮分散式的演算法。在1999年,Wu 和 Li[12]提出了一個僅需要局部相鄰訊息就足夠的分散式演算法，這個演算法假設每個點都有唯一的識別碼。在整個演算法的執行過程中，每個節點只需要知道鄰居(所謂的鄰居，是指在傳輸距離內的節點)的資訊，只需要和鄰居互傳訊息就夠了。Wu和 Li 的演算法一開始所有的節點都標示為白色，然後節點和鄰居互傳彼此的鄰居清單，每個節點就可以知道鄰居之間的相鄰關係，當一個節點它有鄰居不相連時，則此節點標為黑色。當所有節點都執行結束後,Wu 和 Li 證明了所有的黑色節點所形成的集合是一個連通控制集。Wu和 Li 也觀察出兩個可以讓連通控制集的元素減少的剔除規則，如下。考慮 Wu 和 Li 的演算法所產生的連通控制集，一個黑色節點 u 可以從連通控制集中剔除，只要 u 有一個擁有較大識別碼的鄰居 v 使得 $N[u] \subseteq N[v]$ ，或者是 u 有兩個擁有較大識別碼的鄰居 v 和 w ，使得 $N(u) \subseteq N(v) \cup N(w)$ 。Wu 和 Li 針對模擬結果表示，在沒有使用剔除規則時，結果相當的不好，使用了剔除規則後，大幅改善了結果，不過 Wu 和 Li 在文章中並沒有分析其演算法的近似因子。

Guha等人的第二個演算法用了兩個階段，在2004年,Ruan 等人[9]就根據這一點做改進，提出了一個使用一個階段的集中式演算法，同樣是使用貪婪法則。演算法會將節點標為黑色，灰色或白色

其中一種，最後黑色點所形成的集合就是連通控制集。Ruan等人定義幾個函數，根據函數值來做為如何選擇節點加入連通控制集(標為黑色)的準則。考慮一個圖 $G, C \subseteq V(G)$ ，令 $p(C)$ 為 $G[C]$ 中由黑色節點所形成的連通元件 (connected components) 個數；令 $D(C) = \{uv \in E(G) \mid u \in C \text{ 或 } v \in C\}$ ； $q(C)$ 表示一個圖 $G' \subseteq G$ 的連通元件的個數，其中 $V(G') = V(G)$ ， $E(G') = D(C)$ ； $f(C) = p(C) + q(C)$ 。

Ruan等人的演算法如下：一開始所有的點都是白色，由黑色點所形成的集合 C 為空集合，接著反覆的挑點，挑加入 C 後能夠讓 f 減少最多的點，將挑出來的點標為黑色，鄰居標為灰色，當沒有點可以讓 f 減少的時候演算法結束。Ruan 等人證明由演算法標為黑色的點所形成之集合為連通控制集，及演算法的近似因子為 $(2 + \ln \delta)$ 。

在有些無線感測網路上，感測節點的傳輸距離都相同，所以就可以考慮比較特殊的圖，單位圓盤圖 (unit disk graph)。單位圓盤圖的相鄰關係是根據點在歐氏平面上的距離來定義，考慮圖中兩點，若在歐氏平面上的距離不超過一個單位長度，則此兩點有邊相連。於是就有了在單位圓盤圖上找最小連通控制集的研究，而且單位圓盤圖有一個特性[8]：每一個點最多只有五個彼此不相連的鄰居。不過，即使是單位圓盤圖，找最小連通控制集的問題被證明仍然是NP-難的問題[1]。

Guha等人和Ruan 等人的演算法都是集中式且都是針對一般圖，演算法的近似因子較高。2004年,Alzoubi 等人[10]指出Wu 的錯誤，分析了 Wu 的演算法，提出了一個針對單位圓盤圖的分散式近似演算法，以及找出獨立集和最小連通控制集的大小關係：任意一個單位圓盤圖 G 中，令 opt 為最小連通控制集的個數，則對所有的獨立集 S ， $|S| \leq 4opt + 1.2$ 。[10]的演算法要利用演算法的輸入，圖 G ，的擴張樹 T ，所以先找出一個點來當 T 的根，利用 T 替每個節點標上顏色,Alzoubi 等人證明所有被標上黑色的點會是一個獨立集 S ，滿足以下特性：

- (A1) S 加入任何點 $v, v \in V(G) - S$ 都不是獨立集；
- (A2) 若 $S = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$ ，則存在 $u \in S_1, v \in S_2$

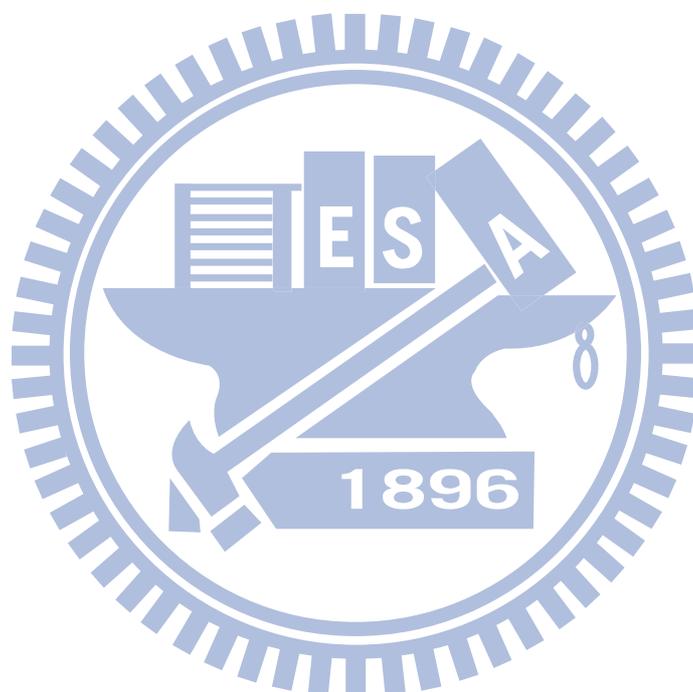
使得 u 到 v 在 G 中的最短路徑長度為2。再根據這個特性，產生另一個擴張樹 T^* ，而 T^* 所有不是葉子的點就會是一個連通控制集。最後,Alzoubi 等人證明了演算法的近似因子為常數:8。

2003年 Cheng 等人[3]針對單位圓盤圖，提出一個可以得出任意近似因子的多項式演算法的策略。一個多項式演算法的策略可以給定一個圖和指定的值, ϵ ，產生一個近似因子為 $(1 + \epsilon)$ 的近似演算法。若要得到較小的近似因子，這個演算法將會執行較多的時間。

2005年,Yingshu 等人[7]針對單位圓盤圖提出了一個近似演算法。這個演算法會將圖中的點標為黑色，灰色，或藍色其中一種顏色，另外還要考慮藍-黑元件，藍-黑元件就是由黑點和藍點所形成的子圖的連通元件，但是不考慮藍點間的相鄰關係。演算法的想法如下：令 G 為輸入的圖，利用Alzoubi

等人[10]的演算法先找出一個滿足條件A1 和 A2的獨立集 S , 將所有在 S 內的點標為黑色, 所有 $G - S$ 的點標為灰色, 反覆的挑出連接最多不同藍-黑元件內黑點的灰點標為藍色, 因為[8], 一個灰點最多只有五個黑色鄰居, 所以從連接在不同藍-黑元件的五個黑點開始找, 直到沒有灰點連接至少兩個不同藍-黑元件內的黑點時結束。Yingshu 等人證明所有藍色和黑色的點就是一個連通控制集, 演算法的近似因子是 $4.8 + \ln 5$ 。

在2006年 Cokuslu 等人[4]提出了一個改進Wu 和 Li 的分散式演算法, 但是在文章中並未證明演算法所找出來的集合是一個連通控制集, 以下證明其結果是正確的。



4 Wu 和 Li 的演算法以及 Cokuslu 的演算法

這兩個演算法都分兩個階段。這兩個演算法都假設網路對應的圖 G 是連通圖，但不是完全圖， G 中每一個點 v 都有一個唯一的識別碼，記成 $\text{id}(v)$ ；每一個點都知道自己的鄰居成員，可以分享彼此的鄰居成員資訊，也就是可以知道每個鄰居的鄰居成員。

4.1 Wu 和 Li 的演算法

現在介紹由 Wu 和 Li 在 1999 年所提出的演算法[12]。目標是要找任一給定圖 G 的連通控制集，演算法會為節點標上一種顏色，黑色或白色，最後所有黑色的點所成的集合就會是連通控制集。這個演算法的基本想法就是將所有“可能需要的點”都挑出來。也就是說，若圖 G 中一個點 u 有兩個鄰居不相連，則 u 就是“可能需要的點”，因為只要 u 有鄰居 v, w 不相連， v, w 就可能需要透過 u 來傳遞訊息。但是將所有“可能需要的點”都挑出來，有些會是多餘的，多餘的點就必須要透過一些規則剔除。

以下是 Wu 和 Li 的演算法的敘述。在演算法第一階段的開始，每個節點都是標為白色，然後檢查有沒有不相連的鄰居，若有，則節點標為黑色。第二階段的目的是在不破壞連通的性質下，減少第一階段選出來的黑色節點。Wu 和 Li 證明經過演算法第一階段，所有黑色的節點所形成的集合就是連通控制集。第二階段的剔除規則就是，若一個黑點 v 滿足以下兩個條件之一，則 v 改標為白色：

1. v 存在一個標為黑色的鄰居 u ，使得 $\text{id}(v) < \text{id}(u)$ 且 $N[v] \subseteq N[u]$;
2. v 存在兩個標為黑色的鄰居 u 和 w ，使得 $\text{id}(v) = \min\{\text{id}(v), \text{id}(u), \text{id}(w)\}$ 且 $N(v) \subseteq N(u) \cup N(w)$ 。

以圖 1 為例子，圖 1 (a) 中黑色節點是根據 Wu 和 Li 的演算法第一階段的規則挑出來“可能需要的點”，圖 1 (b) 是根據第二階段的剔除規則，減去不必要的點之後的情形。

4.2 Cokuslu 的演算法

現在介紹由 Cokuslu 等人所提出的演算法[4]。這個演算法是以 Wu 和 Li 的演算法為基礎，一樣是以顏色來區分節點，但是比 Wu 和 Li 的演算法多使用了一種顏色：灰色。Cokuslu 等人觀察到在 Wu 和 Li 的演算法中，一定是標為白色的點有：樹葉節點以及所有鄰居都兩兩相連的點；此外，Cokuslu 等人也觀察到樹葉節點的鄰居一定是標為黑色，因為樹葉節點沒有辦法不透過這個唯

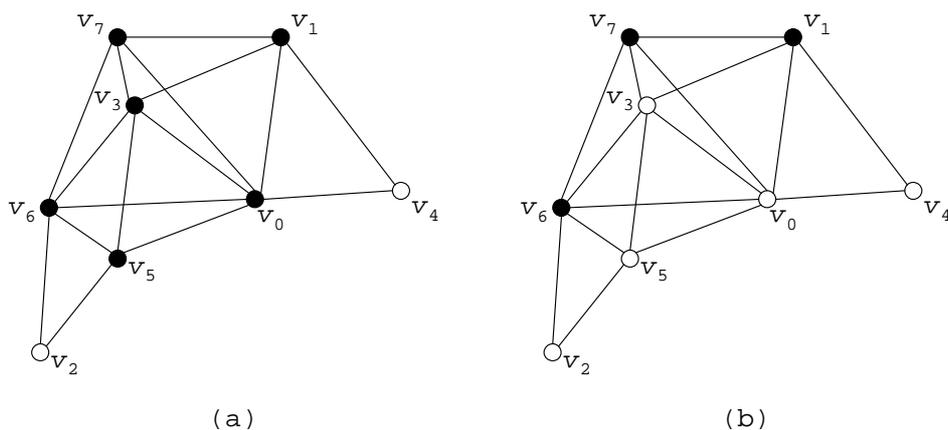


圖 1. Wu 和 Li 的演算法範例
(a) 第一階段執行後, (b) 第二階段執行後

一的鄰居和其他節點傳遞訊息。上述節點只要一決定顏色, 就不需要再改變顏色了。除了樹葉節點、樹葉節點的鄰居、所有鄰居都兩兩相連的節點, 其他節點有可能在 Wu 和 Li 的演算法第二階段中改變顏色。Cokuslu 等人利用這個性質, 設計出和 Wu 和 Li 的演算法一樣是兩個階段的演算法。Cokuslu 的演算法第一階段先找出所有在 Wu 和 Li 的演算法第一階段決定後就不會變更顏色的節點, 這些不會變更顏色的節點會被標為黑色或白色, 其餘則會被標為灰色。在第二階段, Cokuslu 等人想出了四個判斷條件, 灰色節點根據這四個判斷條件來決定最終顏色是黑色或白色。這四個判斷條件使用了分支度的資訊, 灰色節點決定改標為黑色或白色的時候, 除了 id 識別碼以外還以分支度大的節點優先考慮。

下面是 Cokuslu 等人的演算法的敘述。一開始, 每一個點都沒有標上顏色。在第一階段, 每個點 v 會先收集所有鄰居的資訊, 等到收集完後就根據兩個條件來決定自己的顏色:

1. 如果 v 存在鄰居 u , 且 u 為葉子, 則 v 標為黑色;
2. 如果 v 的鄰居皆彼此相鄰, 或者 v 為葉子, 則 v 標為白色。

若 v 不滿足第一階段的兩個條件, 則 v 標為灰色。只有標上灰色的點才會再依據第二階段的條件改變顏色, 否則不再改變顏色; 灰色的點將會被改標為黑色或白色。在第二階段中, 每一個灰色的點 v 會收集所有鄰居的顏色資訊, 收集完後就根據四個條件來決定最終的顏色:

1. v 存在一個標為黑色的鄰居 u , 使得 $N[v] \subseteq N[u]$;
2. v 存在兩個標為黑色的鄰居 u 和 w , 使得 $N(v) \subseteq N(u) \cup N(w)$;

3. v 存在一個標為灰色的鄰居 u , 使得 $N[v] \subseteq N[u]$ 且 $\text{degree}(v) < \text{degree}(u)$, 或者是 $\text{degree}(v) = \text{degree}(u)$ 但 $\text{id}(v) < \text{id}(u)$;
4. v 存在兩個鄰居 u 和 w , 顏色標為黑色或灰色, 使得 $N(v) \subseteq N(u) \cup N(w)$ 且 $\text{degree}(v) < \min\{\text{degree}(u), \text{degree}(w)\}$, 或者是 $\text{degree}(v) = \min\{\text{degree}(u), \text{degree}(w)\}$ 但 $\text{id}(v) < \min\{\text{id}(u), \text{id}(w)\}$ 。

只要 v 滿足其中一個條件, 則 v 標為白色, 否則 v 就標為黑色。

以圖 2 為例子, 圖 2 (a) 中白色節點是根據 Cokuslu 的演算法第一階段的規則挑出來不會再變更顏色的節點, 這個圖沒有樹葉節點, 所以沒有任何的點在第一階段被標為黑色, 灰色節點 (在圖中用斜線表示) 則是要再依據第二階段的四個判斷條件決定最終顏色。圖 2 (b) 是根據第二階段的四個判斷條件, 將所有灰色節點改標為黑色或白色。 v_0, v_6 在這個例子中並不滿足第二階段中所有的判斷條件, 所以被標為黑色。 v_1, v_3, v_5, v_7 都至少滿足第二階段四個判斷條件之一, 所以標為白色。

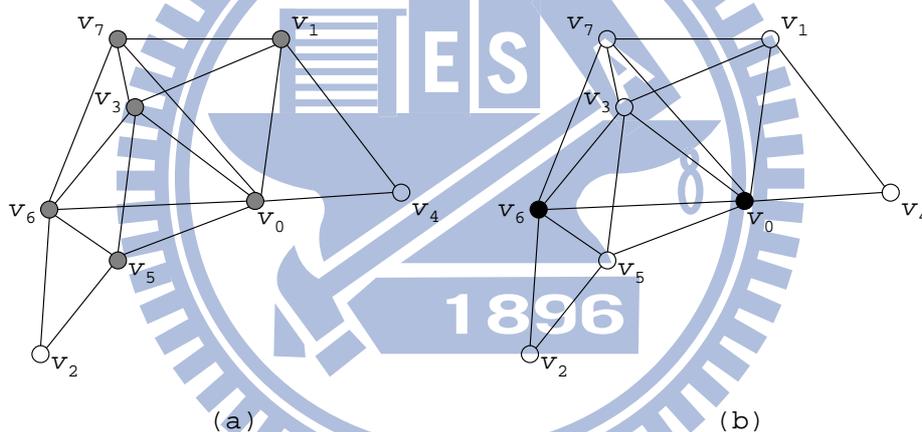


圖 2. Cokuslu 的演算法範例

(a) 第一階段執行後, (b) 第二階段執行後。

5 證明 Cokuslu 的演算法的正確性

根據 Cokuslu 的演算法假設, 輸入的圖 G 是一個連通圖, 但不是完全圖。Cokuslu 的演算法執行完後, G 中的每一點只有可能標為黑色和白色兩色的其中一色。令 D 為所有標為黑色的點所成之集合。以下證明分兩個部分, 先證明 Cokuslu 的演算法之輸出, 即 D , 在 G 中具有控制集的性质, 再證明這個 D 在 G 中的生成圖 $G[D]$ 是連通的。

5.1 證明 Cokuslu 的演算法之輸出為控制集

定理 1. D 是一個控制集。

證明. 欲證 D 是控制集, 相當於要證: 演算法結束後 G 中任一點的顏色是被標為黑色, 或者有一個鄰居是被標為黑色。由於 Cokuslu 的演算法執行完後, G 中的每一點只有可能標為黑色或白色, 因此, 證明以下敘述 (*) 成立就足夠了。

(*) G 中標為白色的點有一個鄰居是黑色。

令 v 是 G 中任意一個標為白色的點。 v 被標為白色的只有兩可能: 根據第一階段的條件、或根據第二階段的條件。以下證明: 不論是根據第一階段或第二階段的條件, v 都有一個鄰居是黑色。

Case 1: 假設 v 是根據第一階段的條件被標為白色。

此時還有兩種可能: v 是葉子、或者 v 的鄰居彼此互相直接連接。

Case 1-a: 假設 v 是葉子。

此時, v 只有一個鄰居 (設為 x), 因為圖 G 不是完全圖, 所以 x 不是葉子; 根據 Cokuslu 的演算法, x 會標為黑色; 所以 (*) 成立。

Case 1-b: 假設 v 的鄰居彼此互相直接連接。

此時, 因為圖 G 是連通圖但不是完全圖, 所以 v 至少有一個鄰居其 degree 比 v 大, 令 u 為這樣的鄰居中 degree 最大且 id 最大者。因為 $\text{degree}(u) > \text{degree}(v)$, u 不會是葉子, u 的鄰居也沒有彼此互相直接相連; 所以根據演算法, u 在第一階段不會被標為白色, u 在第一階段只會是被標為黑色或灰色。若 u 在第一階段被標為黑色, 則 (*) 成立。

若 u 在第一階段被標為灰色, 表示 u 是在第二階段決定顏色白或黑。如果 u 在第二階段被標為黑色的話, 則 v 有一個黑色鄰居, (*) 成立。

若 u 是在第二階段被標為白色, 則 u 滿足第二階段的四個條件之一。

如果 u 在第二階段被標為白色是因為條件1或條件2, 而不論條件1或條件2都表示 u 有一個黑色鄰居, 因為 $N(v) \subseteq N(u)$, 所以這個黑色鄰居也同時是 v 的鄰居, 也就是 v 有一個黑色鄰居, (*) 成立。

考慮第二階段中的條件3和條件4。如果 u 滿足這兩個條件, 表示 u 存在一個鄰居 x , 使得 x 也是 v 的鄰居且 x 的分支度大於 u 的分支度, 或者是 x 的分支度等於 u 的分支度但 x 的 id 大於 u 的 id 。但是 v 是 u 的鄰居中分支度最大且 id 最大者, 所以不存在這種鄰居, 也就是 u 不會滿足第二階段的條件3或條件4。

Case 2: 現在假設 v 是根據第二階段的條件被標為白色。

考慮 v 在第二階段被標為白色的情況, 亦即 v 滿足第二階段的四個條件之一。若 v 在第二階段被標為白色是因為條件1或條件2, 表示 v 有一個黑色鄰居。若 v 在第二階段被標為白色是因為條件3或條件4, 則觀察其中分支度最大且 id 最大的鄰居, 稱 w 。由條件3或條件4都可推知 $\text{degree}(v) < \text{degree}(w)$, 或者是 $\text{degree}(v) = \text{degree}(w)$ 但 $\text{id}(v) < \text{id}(w)$ 。因此, 如同說明 v 在第一階段的情況中的鄰居 u , w 也不會在第一階段中被標為白色, 只會在第一階段中決定為黑色或灰色、或在第二階段中決定為黑色或白色, w 也不滿足第二階段中的條件3和條件4。 w 不論在哪一個階段被標為黑色, 都表示了 (*) 成立, 所以只剩下要考慮 w 在第二階段中決定為白色的情況, 但這個情形也跟討論 v 在第一階段的鄰居 u 一樣, w 只有可能滿足第二階段的條件1或條件2, 但這些條件都代表 (*) 成立。 \square

引理 2. [11] 如果 G 為一個圖, $u, v \in V(G)$, 則每一個 u, v -walk 中包含一個 u, v -path。

5.2 證明Cokuslu 的演算法所輸出的控制集為連通

定理 3. $G[D]$ 是連通的。

證明. 將所有在第一階段標為黑色或灰色的點所形成的集合稱為 C , 因為 D 是 C 扣掉所有在第二階段改標為白色的灰色點, 所以要證明 $G[D]$ 為連通, 只要證明下面兩點就足夠:

(1) $G[C]$ 會是連通的。

(2) 令 S 為 G 中由標為黑色和標為灰色的點所形成的集合, $G[S]$ 為連通。若 v 為 $G[S]$ 中一個滿足第二階段其中一個條件的點, 且讓 v 滿足第二階段的條件的點也屬於 S , 則 $G[S] - v$ 仍然為連通。

要證明 (1) 就是要證明任取 $G[C]$ 中沒有邊相連的兩點都存在至少一條路徑在 $G[C]$ 。

令 v, u 為 $G[C]$ 中沒有邊相連的兩點。因為 G 是連通, G 中存在一條 v, u -path, 令 $P = (v = v_0, v_1, \dots, v_k = u)$, 為 G 中 v 到 u 最短路徑的其中一條。因為 v, u 沒有邊相連, 所以 $k > 1$ 。任取 P 上一點 $v_i, k > i \geq 1$ 。 v_{i-1} 和 v_{i+1} 沒有邊相連, 否則就和 P 是最短路徑矛盾, 所以 v_i 有不相連的鄰居。根據第一階段的條件, 在 G 中, 有鄰居不相連的節點就只可能標為黑色或灰色, 因此 v_i 在第一階段只會標為黑色或灰色, 所以 v_i 屬於 $G[C]$ 。因為 v_i 是 P 上任意不等於 v, u 的點, 所以 P 是由標為黑色或灰色的點所形成, 也就是 P 屬於 $G[C]$ 。因為 v, u 為 $G[C]$ 中任意沒有邊相連的兩點, 所以 $G[C]$ 是連通。

要證明 (2), 只要證明 $G[S]$ 中任意兩點 x 和 y , 如果 x, y 在 $G[S]$ 有路徑經過 v , 則 x, y 在 $G[S]$ 也會有路徑不經過 v 即可。以下假設 $P = (x = v_0, v_1, \dots, v_i = v, \dots, v_k, v_{k+1} = y)$ 為 $G[S]$ 中一條 x 到 y 經過 v 的路徑。

(2-a): 若 v 滿足第一個或第三個條件, 則 $G[S]$ 中存在點 u , 使得 $N[v] \subseteq N[u]$, 因此可知 v_{i-1} 和 v_{i+1} 是 u 的鄰居。考慮 u 的可能。 u 可能在 P 上, 也可能不在 P 上。如果 u 在 P 上, 則 $u = v_m, m \neq i$ 。

如果 $m < i$, 則 $(x = v_0, v_1, \dots, v_m, v_{i+1}, \dots, v_k, v_{k+1} = y)$ 為 $G[S]$ 中一條 x 到 y 的路徑; 如果 $m > i$, 則 $(x = v_0, v_1, \dots, v_m, v_{i-1}, v_m, \dots, v_k, v_{k+1} = y)$ 為 $G[S]$ 中一條 x 到 y 的路徑。如果 u 不在 P 上, 則 $(x = v_0, v_1, \dots, v_{i-1}, u, v_{i+1}, \dots, v_k, v_{k+1} = y)$ 為 $G[S]$ 中一條 x 到 y 的路徑。以上情況表示 x, y 在 $G[S]$ 有路徑不經過 v 。

(2-b): 若 v 滿足條件2或條件4, 則 $G[S]$ 中存在點 $u, w \in N(v)$, 使得 $N(v) \subseteq N(u) \cup N(w)$, 因此 P 中 v_{i-1}, v_{i+1} 是 u 或 w 的鄰居, 不失一般性, 可以假設 v_{i-1} 是 u 的鄰居。若 v_{i+1} 也是 u 的鄰居, 則 $(x = v_0, v_1, \dots, v_{i-1}, u, v_{i+1}, \dots, v_k, v_{k+1} = y)$ 是一個 x, y -walk, 根據引理2, 存在一條 x 到 y 的路徑; 若 v_{i+1} 是 w 的鄰居, 則 $(x = v_0, v_1, \dots, v_{i-1}, u, w, v_{i+1}, \dots, v_k, v_{k+1} = y)$ 是一個 x, y -walk, 根據引理2, 存在一條 x 到 y 的路徑。因此不論 v 是滿足第二階段的哪一個條件, x, y 在 $G[S]$ 都有路徑不經過 v 。所以 $G[D]$ 為連通。 \square

6 結語

在1999年,Wu 和 Li 提出了一個近似演算法,在2006年,Cokuslu 等人改進了 Wu 和 Li 的演算法而提出了一個近似演算法。由於 Cokuslu 等人的文章中只分析了演算法的效益、訊息複雜度、以及時間複雜度,並未證明演算法的正確性,在這篇論文中,我們證明了 Cokuslu 的演算法的正確性。爲了知道 Cokuslu 的演算法到底比 Wu 和 Li 的演算法好多少,我們還實際跑程式做兩者的比較。在模擬的結果內,N 代表點數,trans. range 代表傳輸半徑, W-Algo代表 Wu 和 Li 的演算法產生的點數,C-Algo 代表 Cokuslu 的演算法產生的點數, difference表示 C-Algo 減 W-Algo。以下是模擬的方式。在一個 $1000 \times 1000\text{m}^2$ 的範圍中,我們利用隨機的方式產生點的座標,再根據給定的 trans. range 來決定圖中點的相鄰關係,每造一個圖,就對這個圖執行兩個演算法。在附錄中我們列出了執行的結果,大多數的情形下,C-Algo 優於 W-Algo,出乎意料的是 C-Algo 可能比 W-Algo 來的差,當 $N=140$,trans. range=450時,以及 $N=200$,trans. range=600時,都有這樣的情形。



參考文獻

- [1] Brent N. Clark, Charles J. Colbourn, David S. Johnson, “Unit Disk Graphs,” *Discrete Mathematics*, vol. 86, 1990, pp. 165–177.
- [2] Xiuzhen Cheng, Ding-Zhu Du, “Virtual Backbone-based Routing in Ad Hoc Wireless Networks,” Technical report, Department of Computer Science and Engineering, University of Minnesota, 2002.
- [3] Xiuzhen Cheng, Xiao Huang, Deying Li, Weili Wu, Ding-Zhu Du, “A Polynomial Time Approximation Scheme for the Minimum Connected Dominating Set in Ad Hoc Wireless Networks,” *Networks*, vol. 42, 2003, pp. 202–208.
- [4] Deniz Cokuslu, Kayhan Erciyes, Orhan Dagdeviren “A Dominating Set Based Clustering Algorithm for Mobile Ad Hoc Networks,” in *in Proc. ICCS 2006*, LNCS 3991, 2006, pp. 571-578.
- [5] Michael R. Garey, David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1978.
- [6] S. Guha, S. Khuller, “Approximation Algorithms for Connected Dominating Sets,” *Algorithmica*, vol. 20, 1998, pp. 374–387.
- [7] Yingshu Li, My T. Thai, Feng Wang, Chih-Wei Yi, Peng-Jun Wan, Ding-Zhu Du, “On Greedy Construction of Connected Dominating Sets in Wireless Networks,” *Wireless Communications and Mobile Computing*, vol. 5, 2005, pp. 927–932.
- [8] Madhav V. Marathe, Heinz Breu, Harry B. Hunt III, S. S. Ravi, Daniel J. Rosenkrantz, “Simple Heuristics for Unit Disk Graphs,” *Networks*, vol. 25, 1995, pp. 56–89.
- [9] Lu Ruan, Hongwei Du, Xiaohua Jia, Weili Wu, Yingshu Li, Ker-I Ko, “A Greedy Approximation for Minimum Connected Dominating Sets,” *Theoretical Computer Science*, vol. 329, 2004, pp.325–220.

- [10] Peng-Jun Wan, Khaled M. Alzoubi, Ophir Frieder, “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks,” *Mobile Networks and Applications*, vol 9, 2004, pp. 141–149.
- [11] Douglas B. West, *Introduction to Graph Theory 2E*, Prentice Hall, 2001.
- [12] Jie Wu and Hailan Li, “On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks,” *Proceedings of the 3rd International Workshop On Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 7–14, Seattle, USA, August 1999.



附錄

N	trans. range	W-Algo	C-Algo	difference
40	250	19	19	0
40	300	22	14	-8
40	350	13	9	-4
40	400	11	10	-1
40	450	9	7	-2
40	500	8	4	-4
40	550	6	3	-3
40	600	9	1	-8

表 1.

N	trans. range	W-Algo	C-Algo	difference
50	250	26	21	-5
50	300	18	14	-4
50	350	17	10	-7
50	400	15	10	-5
50	450	14	8	-6
50	500	11	5	-6
50	550	4	2	-2
50	600	9	1	-8

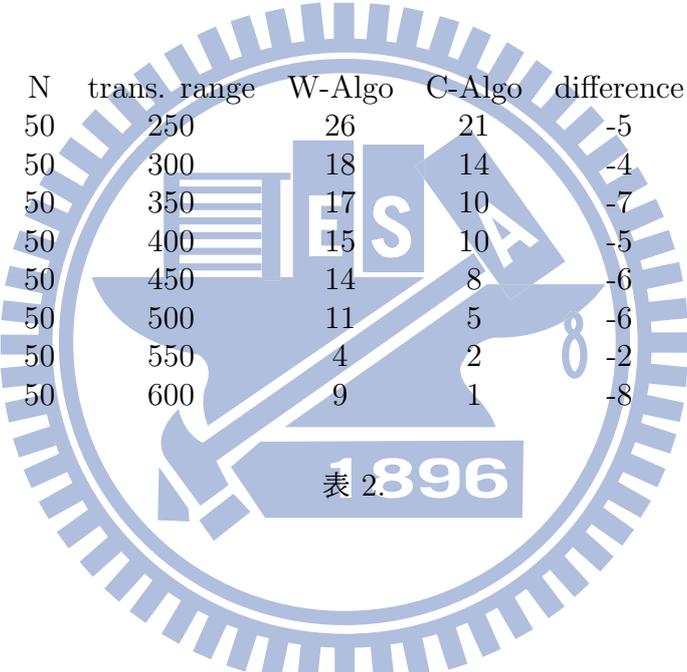
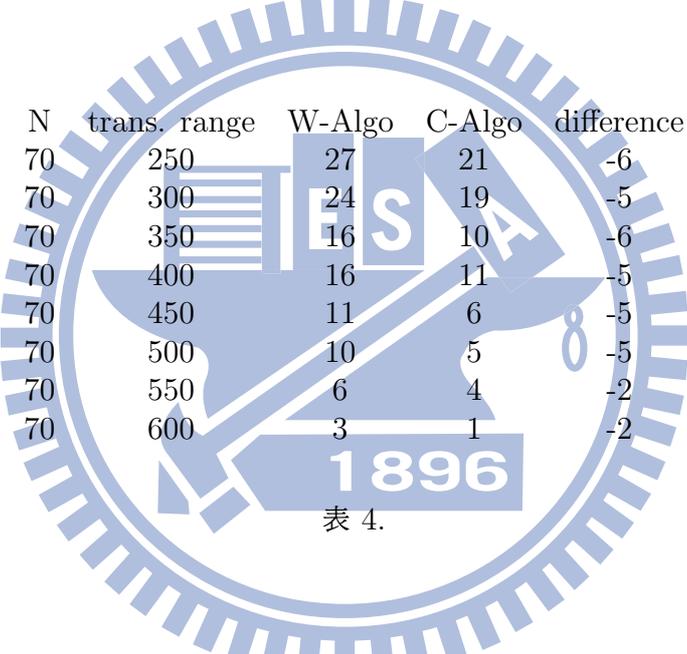


表 2.

N	trans. range	W-Algo	C-Algo	difference
60	250	27	21	-6
60	300	22	16	-6
60	350	16	9	-7
60	400	12	11	-1
60	450	9	5	-4
60	500	8	6	-2
60	550	5	5	0
60	600	5	2	-3

表 3.



N	trans. range	W-Algo	C-Algo	difference
70	250	27	21	-6
70	300	24	19	-5
70	350	16	10	-6
70	400	16	11	-5
70	450	11	6	-5
70	500	10	5	-5
70	550	6	4	-2
70	600	3	1	-2

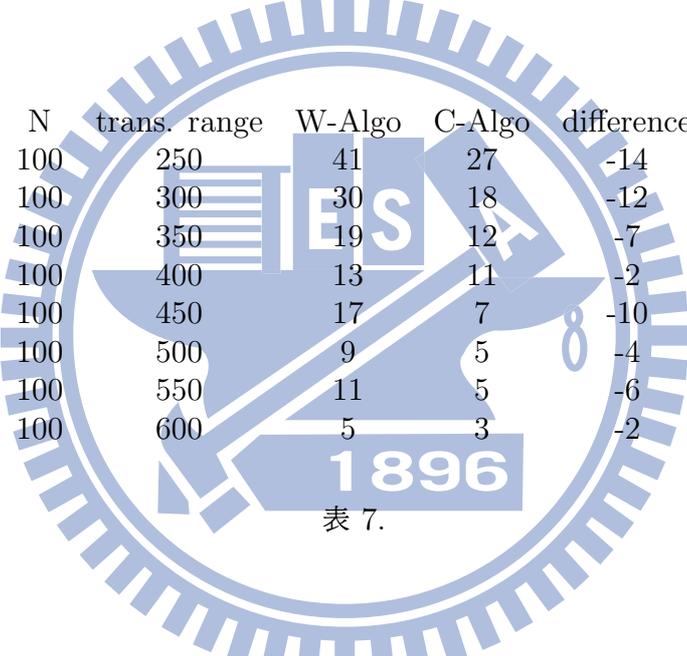
表 4.

N	trans. range	W-Algo	C-Algo	difference
80	250	35	24	-11
80	300	26	17	-9
80	350	19	12	-7
80	400	11	10	-1
80	450	15	8	-7
80	500	12	7	-5
80	550	8	3	-5
80	600	7	4	-3

表 5.

N	trans. range	W-Algo	C-Algo	difference
90	250	31	21	-10
90	300	26	19	-7
90	350	20	15	-5
90	400	15	10	-5
90	450	10	9	-1
90	500	14	7	-7
90	550	5	5	0
90	600	5	3	-2

表 6.



N	trans. range	W-Algo	C-Algo	difference
100	250	41	27	-14
100	300	30	18	-12
100	350	19	12	-7
100	400	13	11	-2
100	450	17	7	-10
100	500	9	5	-4
100	550	11	5	-6
100	600	5	3	-2

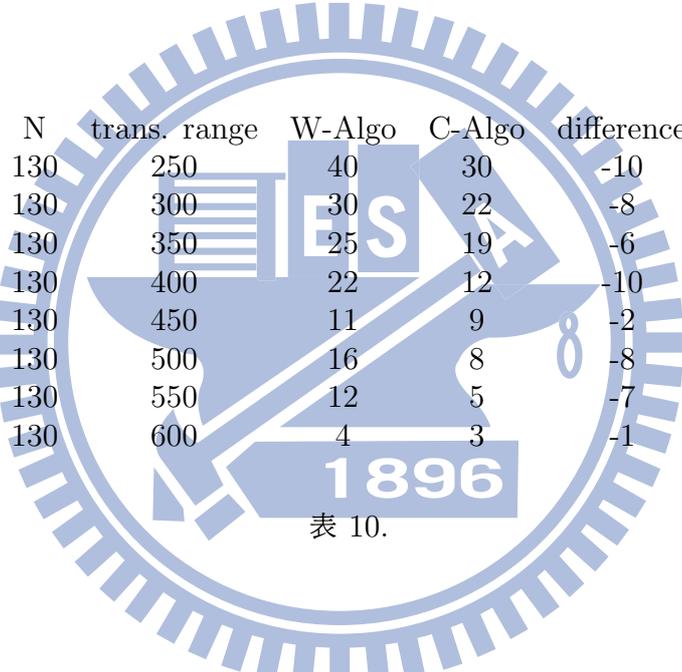
表 7.

N	trans. range	W-Algo	C-Algo	difference
110	250	32	27	-5
110	300	32	22	-10
110	350	22	14	-8
110	400	16	10	-6
110	450	13	7	-6
110	500	8	6	-2
110	550	8	7	-1
110	600	11	4	-7

表 8.

N	trans. range	W-Algo	C-Algo	difference
120	250	34	25	-9
120	300	31	18	-13
120	350	20	15	-5
120	400	12	11	-1
120	450	11	8	-3
120	500	11	8	-3
120	550	9	5	-4
120	600	7	4	-3

表 9.



N	trans. range	W-Algo	C-Algo	difference
130	250	40	30	-10
130	300	30	22	-8
130	350	25	19	-6
130	400	22	12	-10
130	450	11	9	-2
130	500	16	8	-8
130	550	12	5	-7
130	600	4	3	-1

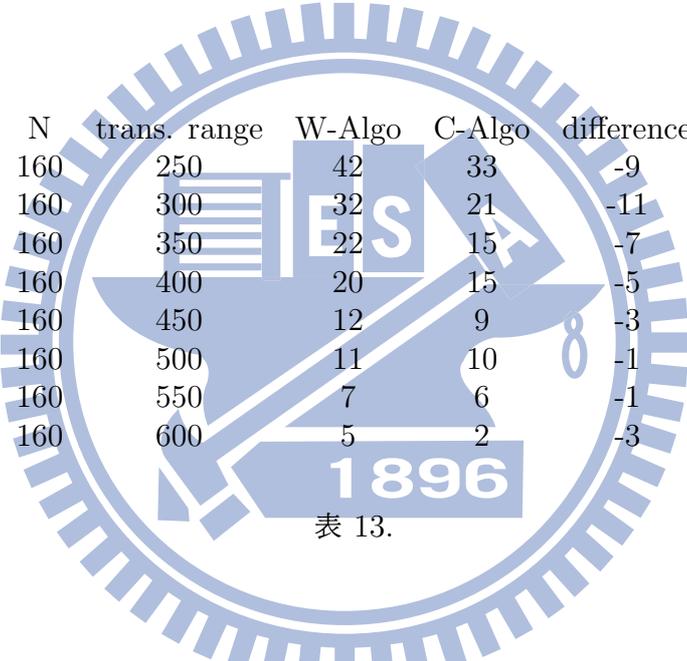
表 10.

N	trans. range	W-Algo	C-Algo	difference
140	250	39	25	-14
140	300	34	17	-17
140	350	23	12	-11
140	400	16	10	-6
140	450	12	13	1
140	500	9	6	-3
140	550	7	5	-2
140	600	5	2	-3

表 11.

N	trans. range	W-Algo	C-Algo	difference
150	250	46	27	-19
150	300	31	19	-12
150	350	20	18	-2
150	400	16	13	-3
150	450	12	9	-3
150	500	11	6	-5
150	550	5	5	0
150	600	8	4	-4

表 12.



N	trans. range	W-Algo	C-Algo	difference
160	250	42	33	-9
160	300	32	21	-11
160	350	22	15	-7
160	400	20	15	-5
160	450	12	9	-3
160	500	11	10	-1
160	550	7	6	-1
160	600	5	2	-3

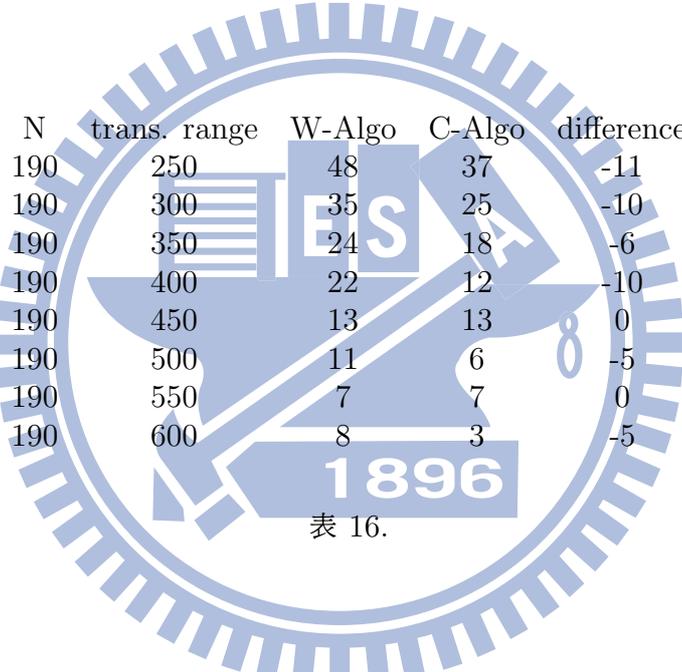
表 13.

N	trans. range	W-Algo	C-Algo	difference
170	250	47	28	-19
170	300	30	25	-5
170	350	24	20	-4
170	400	20	17	-3
170	450	13	10	-3
170	500	15	6	-9
170	550	7	7	0
170	600	6	4	-2

表 14.

N	trans. range	W-Algo	C-Algo	difference
180	250	45	36	-9
180	300	28	22	-6
180	350	27	17	-10
180	400	15	13	-2
180	450	18	11	-7
180	500	10	10	0
180	550	6	4	-2
180	600	8	4	-4

表 15.



N	trans. range	W-Algo	C-Algo	difference
190	250	48	37	-11
190	300	35	25	-10
190	350	24	18	-6
190	400	22	12	-10
190	450	13	13	0
190	500	11	6	-5
190	550	7	7	0
190	600	8	3	-5

表 16.

N	trans. range	W-Algo	C-Algo	difference
200	250	41	34	-7
200	300	36	28	-8
200	350	25	20	-5
200	400	19	14	-5
200	450	17	10	-7
200	500	10	7	-3
200	550	12	7	-5
200	600	3	5	2

表 17.