

在無線隨意網路上能耗拓展圖之功率最小化問題的改良貪婪演算法

劉士琦，游弘毅，林添進

國立中興大學資訊科學與工程學系

g101056003@mail.nchu.edu.tw, {herbert, tienchinglin}@nchu.edu.tw

摘要

無線隨意網路是由眾多感測器節點組成，每個節點通常配有處理器、無線收發器和有限的電源，並經由廣播的方式與其他的節點交換資料。由於廣播功率的強弱直接影響節點的能量消耗，在電源受限且難以重複充電的情況下，如何指派每個節點的廣播功率使能量消耗最佳化，同時又維持某種網路所需的連通條件就成為一項重要的議題。這類問題一般統稱為能量指派問題。

在這篇論文中我們將探討的能量指派問題，是在保證網路拓撲為 t 倍能耗拓展圖的前提下，如何將所有節點被指派的功率總和降至最低，此問題已被證明為 NP-Complete。我們提供三個啟發式演算法，三者皆是基於貪婪演算法上的進一步改良，並使用不同的建構與淘汰策略來因應不同的需求。最後我們經由實驗模擬證明我們所提出的這三種演算法在能量消耗功率總和上，與此問題目前最好的演算法相比都達成了一定程度的改進。

關鍵字: 無線隨意感測網路、拓展圖、能量指派

1、緒論

無線隨意網路是由數量龐大的感測器節點所組成，而這些感測器通常配有處理器、記憶體、無線收發器和有限的電源。每個感測器可在自身感測範圍內收集所需的物理或環境狀況，例如：溫度、濕度、聲音、震動、運動等等，再經由廣播的方式與其他的感測器交換資料。最後我們可以收集所有無線隨意網路內的感測資料加以計算、分析從而達到特定的目的。無線隨意網路的應用範圍甚廣，包括環境監控[15, 21, 26]、居家看護[3, 23, 30]、軍事應用等等[19, 28, 32]。因為具有如此廣泛的應用，無線隨意網路這塊領域也迅速地吸引了大量學術研究者的關注，經過多年的研究與討論後，漸漸發展出許

多重要的議題，其中較受關注的議題如能源效率、連通性與覆蓋問題等等。

在常見的重要應用中，感測器往往無法由固定電源獲得能量補充，於是只能使用有限能源的電池作為能量供應來源，並且難以進行重複充電。這個有限能源的特性使得無線隨意網路的能源效率變得更加重要。在過去的研究文獻中，常被用來評估網路能源效率的兩個方式，是網路存活時間[4, 6]與能量指派[11, 12]。其中能量指派的意義是指派每個感測器的廣播功率以調整其廣播半徑，由於越大的廣播半徑將會消耗越多的能量，因此調降感測器的廣播功率可以達到節省能源的目的。但是為了有效收集所有感測器所收集的資料，無線隨意網路必須保持一定的連通特性，常見的連通特性包括 k 連通性與 k 邊連通性[9, 14]、拓展圖[13, 31]等。而所謂的能源指派問題，即是如何在保持適當的網路連通性時將總能量指派降至越小越好。由於對網路連通性的不同要求，以及對整個網路中電池能量耗損的不同最佳化方式，又可以進一步區分成種種不同的能量指派問題[1, 8, 10, 14, 17, 18, 27, 29]。一般而言這類問題都是屬於 NP-Complete 的問題。

這篇論文乃是研究能量指派問題中的能耗拓展圖之功率最小化問題，其要求為在進行能量指派後任兩個節點間之傳輸能耗，不可超過廣播功率不受限制時之傳輸能耗的 t 倍，並同時要求網路中所有節點的能量指派之總和最小化。此問題也被證明是 NP-Complete 問題[29]。目前對此問題之目標函數效果最好的演算法，是在建構拓展圖時以節點間之直線傳輸能耗為優先考量的啟發式貪婪演算法[29]。我們對此演算法提出了三種改良的貪婪演算法。在我們的第一個演算法中，採用了新的貪婪策略，當建構拓展圖時每次選定的邊要求讓節點能量指派增加量最小，這一策略更加符合問題要求與性

質。接下來，在對此演算法進行觀察後，我們發現有些邊被演算法排除之後就不再被考慮，以至於在後續選擇需要加入更差的邊來彌補。因此在第二與第三個演算法中，除了採用第一個演算法中的貪婪策略以外，我們又進一步提供了一個增加候選邊的策略和兩個進階刪除策略，以此刪除一些拓展圖中不需要的邊，進而減少總能量指派。

在實驗模擬方面，我們分別進行了兩種不同性質的實驗，以總節點數量為變量與以網路密度為變量。和在總能量指派這項指標中表現最好的貪婪演算法相比，實驗結果顯示我們改進的貪婪演算法在此兩種不同的實驗中，可以分別得到約 1%~3% 和 5%~8% 的改進。

本篇論文接下來的部分的整體架構如以下編排：第二節將討論過去的相關研究。第三節將會提出預備知識與問題定義，並介紹原本的貪婪演算法。第四節中我們會提供三種新的啟發式貪婪演算法及其分析。第五節將提出實驗方式與結果。第六節為本篇論文之結論與未來展望。

2、相關研究

在已知的過去文獻中，總共有三篇論文探討能耗拓展圖之功率最小化問題。第一篇研究此問題的為 Wang 和 Li [29]，他們提出了兩個啟發式演算法來建構能耗拓展圖。第一個演算法為貪婪演算法，其策略為將任兩節點之間的連結關係以通訊圖 G 表示，並將 G 中的所有邊按照由短到長的順序排列，並依照此順序檢查每條邊所連接之兩節點在拓展圖中的傳輸能耗是否滿足 t 倍之內之限制，以決定是否將此邊加入拓展圖中以連結此兩節點。此演算法的時間複雜度為 $O(n^4)$ ，在後文中我們將此演算法稱為 Greedy。第二個演算法是建構在 Yao graph[31]之上，給定一個整數 k ，對於每個節點我們皆以其為圓心畫一個包含其他所有節點的圓，並對此圓取任意的 k 等分，然後在每一等分內尋找距離此圓心最近的節點並與之相連結，按照以上步驟所建構成的圖即被稱為 Yao graph。Li [20]證明 Yao graph 中的能耗延伸因子（即拓展圖的參數） t 與 k

形成了 $\frac{1}{1-(2\sin\frac{\pi}{k})^2}$ 的比例關係，所以第二個演算法即

是利用建構 Yao graph 的方式來建構出 t 倍能耗拓展圖。此演算法的時間複雜度為 $O(n^3)$ ，在後文中我們稱此演算法為 Yao。

不同於[29]提出的啟發式演算法，Shpungin[27]從理論的角度研究此問題。他們運用 Khuller[16]所提出的 LASTs 演算法以建構 n 棵無向樹，並將這 n 棵樹聯集形成的組合作為拓展圖。給定一組參數 $\langle G, W, s \rangle$ ，LASTs 演算法可從通訊圖 G 中得出一棵無向樹 T ，其中以節點 s 作為原點並以權重函數 W 計算邊長時，此演算法可保證圖上任一點到原點 s 的距離不會超過由 G 中得出的任一最小生成樹中該兩點距離的 α 倍，並且 T 的總權重也不會超過最小生成樹的 β 倍，其中 $\alpha > 1$ 且 $\beta \geq 1 + \frac{2}{\alpha-1}$ 。單次 LASTs 演算法的時間複雜度為 $O(n^2)$ ，而該篇論文則套用了 n 次 LASTs 演算法來保證所產生的圖為 α 倍能耗拓展圖，因此 Shpungin 的演算法的其時間複雜度為 $O(n^3)$ ，在後文中我們將此演算法命名為 nLASTs。

最後一篇為 Abu-Affash [1]所提出的近似演算法，類似於前述 Greedy 演算法所用的貪婪策略，但在這之前會先將原通訊圖做一次 Delaunay 三角化(Delaunay triangulation)。Li [20]證明了對原通訊圖做 Delaunay 三角化後可保證新通訊圖為 1 倍能耗拓展圖，故先做一次 Delaunay 三角化可過濾掉大部分的邊，再依照 Greedy 的步驟建構拓展圖可以節省許多計算時間。他們證明這個演算法保證了整個網路的能量指派之總和不會超過最佳解的 $(1 + \frac{2}{t-1})$ 倍。此演算法的時間複雜度為 $O(n^2 \log n)$ ，在後文中我們將此演算法命名為 Delaunay。

3、預備知識與問題定義

在 3.1 小節中，我們將先介紹能耗拓展圖之功率最小化問題所需的預備知識，然後給出此問題的詳細定義。在 3.2 小節中，我們將介紹目前在總能量指派最小化這方面效果最好的 Greedy 演算法 [29]，以便與第四節中我們將提出的改善貪婪演算

法進行更清楚的比較。

3.1. 預備知識

令 $V = \{v_1, v_2, \dots, v_n\}$ 為 n 個散布在二維空間中的無線感測節點之集合，其中所有感測器都假設是相同型號，並且其最大廣播功率所能達到的廣播半徑都是相同長度，其值為 $R_{\max}$ 。對任兩個節點 $u, v \in V$ 而言，兩點間的直線（歐氏）距離為 $\|uv\|$ 。如果 $\|uv\| \leq R_{\max}$ ，則兩點皆在彼此的廣播範圍內，可直接互相發送訊息。在著名的功率衰減模組中[24, 25]，從節點 u 直接向節點 v 發送訊息所需消耗的傳輸能耗為兩點間之直線距離的 r 次方，寫作 $w(u, v) = \|uv\|^r$ ；其中 r 為距離功率梯度，通常被界定在 2 到 5 之間。在這篇論文中為了簡化說明，我們假設 $r = 2$ ，也就是說在後文中 $w(u, v) = \|uv\|^2$ ，但是本論文提供的所有演算法皆可支援 $r > 2$ 的情況。

以 V 中的節點為點集合，我們可以建構一個無向圖 $G = (V, E)$ ，代表節點間可直接傳送訊息的關係圖，此圖稱為原始通訊圖。對任兩個節點 $u, v \in V$ 而言，如果 $\|uv\| \leq R_{\max}$ ，則在 E 中存在一條邊 (u, v) ，其邊長為兩點間的傳輸能耗 $w(u, v)$ ，而對於圖中任何一條路徑 P 而言，我們定義其傳輸能耗為 $w(P) = \sum_{e \in P} w(e)$ ，代表此路徑途經的所有邊的傳輸能耗之總和。另外，我們還定義了一個符號 $\delta_G(u, v)$ ，代表在 G 中由節點 u 到節點 v 的最短路徑之傳輸能耗，也可以說是在 G 中由節點 u 傳輸訊息到節點 v 所需的最小傳輸能耗。現在我們可以正式定義 G 的 t 倍能耗拓展圖所需滿足的條件。

定義 1. 原始通訊圖 G 的 t 倍能耗拓展圖 $G' = (V, E')$ 必須是 G 的一個生成子圖，並且 G' 中任兩點 u, v 滿足 $\delta_{G'}(u, v) \leq t \times \delta_G(u, v)$ 。

接下來我們討論對 V 中的節點進行能量指派所產生的新通訊圖。令 $p: V \rightarrow \mathbb{R}^+$ 為一組能量指派函數，此函數將對每個節點 v_i 指派新的廣播功率 $p(v_i) \leq R_{\max}^2$ 以及新的傳輸半徑 $r(v_i) = \sqrt{p(v_i)}$ 。基於任一能量指派函數 p ，我們可以建構一個無向的能量指派圖 $G_p = (V, E_p)$ ，此圖上的點為原圖 G 的所有

點；而對任兩點 $u, v \in V$ 而言，邊 (u, v) 存在於邊集合 E_p 中，代表 $w(u, v) \leq p(u)$ 且 $w(u, v) \leq p(v)$ 。要注意的一點是，在 G_p 的定義中對任一點 v_i 而言， $p(v_i) \geq \max_{(v_i, v_j) \in E_p} w(v_i, v_j)$ 。最後，我們定義總能量指派 $c(p)$ 為所有節點的廣播功率之總和，寫作 $c(p) = \sum_{v_i \in V} p(v_i)$ 。接下來我們正式定義本篇論文所要探討的能耗拓展圖之功率最小化問題。

定義 2. 能耗拓展圖之功率最小化問題：給定一組 n 個點之集合 $V = \{v_1, v_2, \dots, v_n\}$ ，一個所有感測器的最大傳輸半徑 $R_{\max}$ ，與一個實數 $t \geq 1$ 。找到一組能量指派 $p = \{p(v_1), p(v_2), \dots, p(v_n)\}$ ，在滿足 G_p 為 G 的 t 倍能耗拓展圖的前提下，使總能量指派 $c(p)$ 最小化。

3.2. Greedy 演算法[29]

Wang 和 Li [29] 所提出的貪婪演算法 Greedy 其步驟如下。首先對空間中的節點建構原始通訊圖 G ，接下來我們準備另一個沒有邊的圖 G' 為建構拓展圖的基礎，並將 G 內的所有邊依照其邊長以由短到長的方式排序，依照此順序逐一檢查是否要加入 G' ：對邊 (u, v) 而言，若加入 (u, v) 之前 $\delta_{G'}(u, v) > t \times \delta_G(u, v)$ ，則將此邊加入 G' 中；反之則捨棄 (u, v) 。依照此流程將 G 中所有邊檢查完畢後演算法即結束，每個節點 v_i 的能量指派 $p(v_i)$ 就設定為 v_i 在 G' 中所有相鄰邊之邊長的最大值，也就是 $p(v_i) = \max_{(v_i, v_j) \in E_p} w(v_i, v_j)$ 。在 Greedy 中，加入邊的條件保證 G' 為 t 倍能耗拓展圖，並以短邊優先的策略對 $c(p)$ 做最小化。其時間複雜度見引理 1。

引理 1 [29]. Greedy 演算法的時間複雜度為 $O(n^4)$ 。

4、關於能耗拓展圖之功率最小化問題的改善演算法

在本節中，我們將詳細說明我們對能耗拓展圖之功率最小化問題所提出的三個改善演算法，並將每個演算法用獨立一小節來討論。

4.1. Greedy Node Power Assignment

我們提出的第一個演算法簡稱為 GN，相對於 Greedy 使用的貪婪策略，此演算法採用新的建構策略，更能符合問題性質。具體來說，Greedy 優先選擇傳輸能耗最小的邊來建構拓展圖，最後才設定能量指派值，此策略我們稱為傳輸能耗的貪婪；而 GN 會在建構過程中同時更新每個節點的能量指派值，並優先選擇對節點之能量指派增加量最小的邊來建構拓展圖，此策略我們稱為能量指派增加量的貪婪。對一條未加入的邊 (u, v) 而言，其能量指派增加量為其傳輸能耗 $w(u, v)$ 對其兩端點目前的能量指派值 $p(u), p(v)$ 所分別造成的修正值之總和，代表加入該邊所會對總能量指派增加的值，具體公式為 $(\max\{w(u, v) - p(u), 0\} + \max\{w(u, v) - p(v), 0\})$ ，參考範例如圖 1。必須注意的是，此數值必定大於等於零。

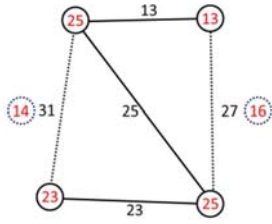


圖 1. 實線圓內的值為該節點目前必須的能量指派，實線為已經加入的邊；而虛線是未加入的邊，虛線圓形中的值代表其能量指派增加量。傳輸能耗的貪婪策略會先檢查右方的邊，而能量指派增加量的貪婪策略會先檢查左方的邊。

GN 演算法的具體步驟如下。首先建構通訊圖 $G = (V, E)$ ，再來我們準備另一個沒有邊的圖 G' ，並對 G 中所有邊計算其相對於 G' 的能量指派增加量。在每一回合中，我們從未檢查的邊中，選擇能量指派增加量最小的邊 (u, v) 進行檢查，若 $\delta_{G'}(u, v) > t \times \delta_G(u, v)$ ，則將此邊加入 G' 中；反之則捨棄 (u, v) 。必須注意的是，若 (u, v) 被加入 G' ，其端點 u 的能量指派要重新設定成其所有相鄰邊之邊長的最大值，也就是 $p(u) = \max_{(u, v_i) \in E_p} w(u, v_i)$ ，而端點 v 的能量指派 $p(v)$ 也要進行類似修正。再者， u, v 兩點的相鄰邊之能量指派增加量也會受到影響。所以在加入一條邊後，該邊兩端點之相鄰邊（少於 $2n$ 條）的能量指派增加量必須重新計算。依照此

流程在所有的邊皆被檢查過後演算法結束，同時也得到新的能量指派函數 p 。GN 演算法的時間複雜度見引理 2，由於篇幅問題，詳細證明在此省略。

引理 2. GN 演算法的時間複雜度為 $O(n^4)$ 。

4.2. Improved GN By Greedy Edge

我們對 GN 演算法進行觀察後發現了兩個現象。其一，在建構拓展圖時，早期加入的邊在我們陸續加入其他邊後，由於整個圖的連通性越來越高，可能導致該邊的兩端點不再需要該邊來維持兩點間的最小傳輸能耗，使得當初加入的這個邊變得多餘，但目前的演算法無法再對已加入的邊做更動。其二，建構拓展圖時，在檢查當下被捨棄的邊，也許對其兩端點間的最小傳輸能耗沒有直接幫助；但加入此邊對整個圖所增加的連通性可能產生一點彈性，使得我們可以去除某些對能量指派效益較差的邊，從而減少總能量指派。

基於上述的兩個觀察，我們在 GN 完成後的結果上進行延伸的優化，並提出了兩個改善方法。其一，我們提供了一個增加候選邊的策略：加入那些不會增加任何能量指派的邊。其二，我們設計了兩個進階的刪除策略：對於已經被加入的邊多加一道過濾程序，將一些不需要的邊移除。這兩個刪除策略分別延伸出我們的第二個演算法 GN-GE 與第三個演算法 GN-GN，我們會在本小節與 4.3 小節中分別進行介紹。

GN-GE 演算法是在 GN 演算法將拓展圖 G' 建構完畢後，在 G' 中執行增加候選邊策略以及第一種刪除策略的延伸，接下來我們將介紹這兩個策略。在增加候選邊方面，我們的策略是找出所有加入 G' 並不會增加總能量指派的已捨棄邊。其步驟為逐一檢查所有被 GN 演算法捨棄的邊 (u, v) ，若其兩端點 u, v 之能量指派滿足 $p(u) \geq w(u, v)$ 且 $p(v) \geq w(u, v)$ ，代表加入此邊後該邊兩端點的能量指派並不會有任何改變，可以將 (u, v) 加入 G' 做為候補邊；反之則完全捨棄。在所有 GE 之已捨棄邊被檢查完畢後，此增邊策略結束。

進階刪除這方面的目的，是測試已加入拓展圖

G' 的邊可否被移除,對象同時包括 GN 加入的邊以及新增的候選邊。由於邊的移除可能導致 G' 不再符合 G 的 t 倍能耗拓展圖的條件,因此在嘗試移除任一條邊時,必須先檢查此次移除是否會影響 t 倍能耗拓展圖的條件,如果不會影響才可移除之。必須特別注意的一點是,由於刪除某一條邊時通訊受到影響的節點很可能不只是該邊的兩端點,因此在檢查 t 倍能耗拓展圖的條件時,不能只檢查欲刪除邊之兩端點間的最小傳輸能耗,而是必須檢查任兩個端點之間的最小傳輸能耗。

在 GN-GE 演算法中所使用的第一種刪邊策略為傳輸能耗的貪婪策略,是以由大到小的順序,優先移除擁有高傳輸能耗的邊。其具體步驟如下。首先先執行一次 GN 演算法從而得出 G 的 t 倍能耗拓展圖 $G' = (V, E')$, 接下來對 G' 執行一次增加候選邊的策略,從 $E \setminus E'$ 的邊集中找出相對於 G' 之能量指派增加量為零的所有邊,並加入 E' 做為候選邊。接下來則執行一次進階刪除策略,將 E' 中的所有邊按照傳輸能耗從大到小的方式排列,然後依照此順序逐一檢查。對每一條要測試的邊 (u, v) 而言,我們會對 $G' \setminus \{(u, v)\}$ 執行全局最短路徑問題之 Floyd-Warshall 演算法,藉以測試 $G' \setminus \{(u, v)\}$ 是否為 G 的 t 倍能耗拓展圖。若是,則以 $G' \setminus \{(u, v)\}$ 取代 G' 以完成移除 (u, v) 的動作。在所有 E' 中的邊皆被測試過後,此演算法結束,然後可重新計算出新的能量指派函數。GN-GE 演算法的時間複雜度見引理 3, 由於篇幅問題,詳細證明在此省略。

引理 3. GN-GE 演算法的時間複雜度為 $O(n^5)$ 。

4.3. Improved GN By Greedy Node

GN-GN 演算法也是 GN 演算法的延伸,類似於 GN-GE, 在用 GN 將 G' 建構完畢後,同樣先增加候選邊,但是在下一步改為執行第二種刪除策略,或者稱為能量指派減少量的貪婪策略。在這個刪除策略中,我們會在刪除過程中同步更新每個節點的能量指派值,並且優先測試那些被刪除後能夠讓總能量指派值下降最多的邊,從而確保每成功刪除一條邊後都能使得拓展圖的總能量指派減少。可以注

意到, GN-GN 演算法所用的能量指派減少量,其概念比起 GN-GE 演算法所用的傳輸能耗,還是更加貼近問題性質。

對任意一條邊 (u, v) 而言,其能量指派減少量定義如下。首先令 $s(u) = \max_{(u, v_j) \in E' \setminus \{(u, v)\}} w(u, v_j)$ 代表從 G' 中刪除邊 (u, v) 後節點 u 的新能量指派值;而 $s(v)$ 也代表類似的意義。則 (u, v) 的能量指派減少量,即為 (u, v) 的移除對其兩端點目前的能量指派值 $p(u), p(v)$ 所分別造成的修正值之總和,也就是 $(p(u) - s(u)) + (p(v) - s(v))$, 或等價於 $\max\{w(u, v) - s(u), 0\} + \max\{w(u, v) - s(v), 0\}$ 。必須注意的是,此數值必定大於等於零。

GN-GN 演算法的具體步驟如下。一開始先執行一次 GN 演算法從而得出 G 的 t 倍能耗拓展圖 $G' = (V, E')$, 接下來對圖 $G' = (V, E')$ 執行一次增加候選邊的策略,從 $E \setminus E'$ 的邊集中找出相對於 G' 之能量指派增加量為零的所有邊,並加入 E' 做為候選邊。接下來則執行一次進階刪除策略,對於屬於 E' 的所有邊,計算其相對於 G' 的能量指派減少量。在每一回合中,我們從 E' 之內未經過刪除測試的邊中,選擇目前能量指派減少量最大的邊 (u, v) 進行測試,以判斷 $G' \setminus \{(u, v)\}$ 是否為 G 的 t 倍能耗拓展圖(同樣利用全局最短路徑問題)。若是,則以 $G' \setminus \{(u, v)\}$ 取代 G' 以完成移除 (u, v) 的動作。在所有 E' 中的邊皆被測試過後,此演算法結束,同時也得到新的能量指派函數 p 。GN-GN 演算法的時間複雜度見引理 4, 由於篇幅問題,詳細證明在此省略。

引理 4. GN-GN 演算法的時間複雜度為 $O(n^5)$ 。

5、實驗結果

我們實作出所有相關的演算法,其中包含我們提出的三個演算法與先前相關文獻所提到的四個演算法,並在本節中比較所有演算法的平均節點廣播功率百分比與程式執行時間。所有的實驗皆是在一台 3.2-GHz Intel(R) 的四核心電腦上進行,其作業系統為 Windows 7, 並且配有 8G 的記憶體。

為了觀察各個演算法的優缺點,我們分別進行

兩種不同性質的實驗，其一是以總節點數量為變量，分別產生 100、200、...、1000 個在平面上亂數散佈的節點各 100 組為實驗輸入，用來觀察不同的節點總數與總能量指派的關係；其二是以網路密度為變量，分別產生可連結之邊密度最高為 10%、20%、...、100% 的節點散佈方式各 100 組，用來觀察不同的邊密度與總能量指派的關係。

在實驗模擬中，我們在亂數產生實驗的輸入資料時遇到一個問題，就是如何選取一個合理的節點散布範圍用以產生隨機的節點佈局。太大的範圍易造成隨機分配後的節點太分散，不易產生出擁有基本連通性質的節點佈局；而太小的範圍又無法產生出亂度夠高的佈局，只能產生一些密度較高的佈局。為了解決此問題，我們試圖在平面上標出一個大小適中的 bounding box，並規定節點佈局只能被限制在此 bounding box 之內。首先我們規定 bounding box 為正方形，並參考總節點數 n 、最大傳輸範圍 R_{max} 、與一個額外參數 c ，來設定此 box 的邊長。

經過多次實驗之後，我們認為此邊長的合適大小應該設定為 $c\sqrt{n} * R_{max}$ ，其中 $c = 0.8$ 。在 $c > 0.8$ 的情況下，此 box 太大，亂數散佈的感測器節點難以形成有整體連通性的無線隨機網路；而在 $c < 0.8$ 的時候，產生的節點佈局密度太高，其亂度不適合進行實驗測試。總結來說，以下實驗所用的輸入資料皆是以 $c = 0.8$ 和 $R_{max} = 10$ 的設定下所產生的隨機節點佈局。

另外，為了利於觀察出在不同輸入組別之間各演算法的優劣，我們設計了一個各組別間一致的衡量指標，稱為平均廣播功率百分比。其定義為總能量指派除以每一組輸入的總節點數 n ，再除以最大傳輸範圍的值 R_{max} 。若一個輸出結果的平均廣播功率百分比越高，代表其總能量指派越大，也就代表此演算法的輸出結果越差。

圖 3 與圖 4 分別為 $t = 2$ 條件下的平均節點廣播功率百分比與程式執行時間。為了能更清楚顯示所有演算法的實驗結果，我們將實驗數據分成兩部分呈現，第一部分只比較研究文獻中提出的既有演算法，在第二部分中我們只比較對此問題之目標函數

表現最優秀的 Greedy 與我們提出的三個演算法。如圖 3(a)所示 Greedy 與 Delaunay 的平均廣播功率百分比非常相近，其中又以 Greedy 的表現最好，而再來才是 nLASTs，而 Yao 的表現最差，至於圖 3(b)顯示出我們的三個演算法相較於 Greedy 也有不同程度的改進，其中表現最好的演算法為 GN-GN，由於 Greedy 為過去演算法中表現最好的，故我們將以 Greedy 的結果為基準，並將詳細的改進比率陳列於表 2。

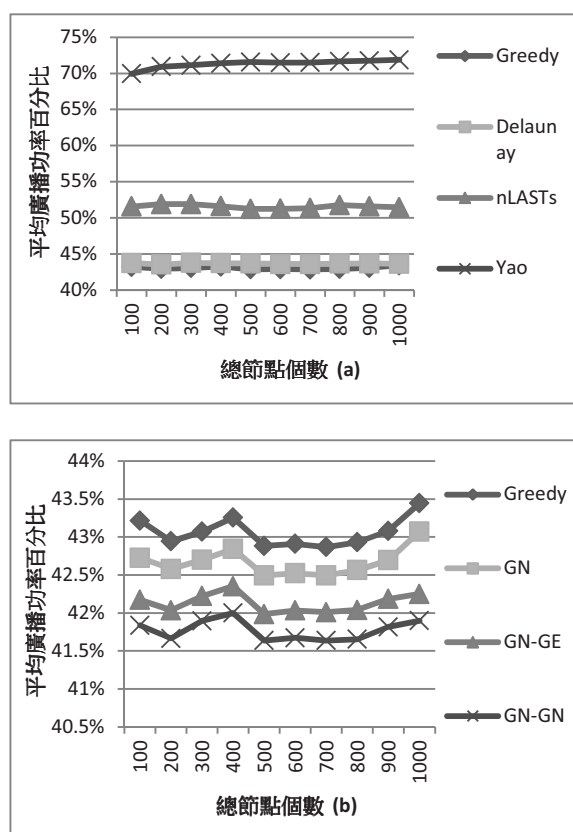


圖 3. 實驗結果 - 廣播功率。(a)過去提出的所有演算法。(b) 過去能量指派最佳的演算法與我們提出的三個演算法。

如表 2 所示，我們的演算法在平均廣播功率上大約有 1%~3% 不等的改進。而在時間方面，如圖 4 所示 Delaunay 的程式執行時間最短，是因為其有良好的過濾邊策略，而非像 Greedy 演算法的全部逐一檢查。由於我們的演算法不但要對所有邊進行逐一檢查，還要進行額外的檢查與修正，所以在執行時間方面表現比 Greedy 更差。若要縮短執行時間，需要改善現有的時間複雜度，或者是在實作層面上對程式做更進一步的優化。

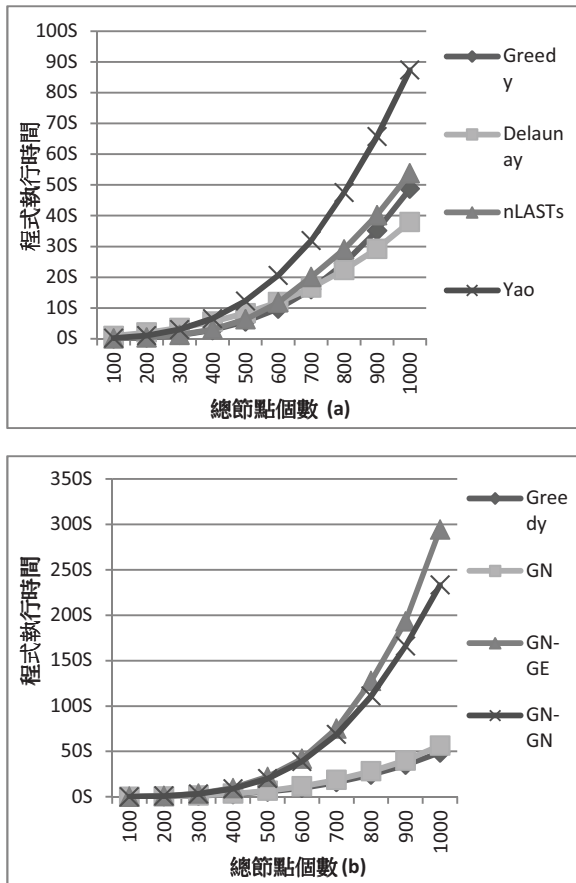


圖 4. 實驗結果 – 執行時間。(a)過去提出的所有演算法。(b) 過去能量指派最佳的演算法與我們提出的三個演算法。

表 2：當 $t = 2$ 的平均廣播功率之改進比率表。

$\frac{X - \text{Greedy}}{X} (\%)$	Greedy	GN	GN-GE	GN-GN
$n=100$	-	1.135%	2.412%	3.191%
$n=200$	-	0.851%	2.116%	2.977%
$n=300$	-	0.857%	1.965%	2.724%
$n=400$	-	0.952%	2.086%	2.904%
$n=500$	-	0.899%	2.092%	2.900%
$n=600$	-	0.905%	2.041%	2.884%
$n=700$	-	0.872%	1.999%	2.872%
$n=800$	-	0.855%	2.077%	2.975%
$n=900$	-	0.896%	2.080%	2.932%
$n=1000$	-	0.858%	2.747%	3.558%

另外我們也進行了 $t = 3$ 與 $t = 4$ 的實驗，並將其改進比率成果分別列於表 3 與表 4，因為時間方面不同的 t 並不會造成太大的改變，故在此省略其數據。由表中可見，縱使在不同的 t 的情況下，我

們的演算法依然擁有較好的表現，而且可以進一步觀察到在 t 越大的情況之下，我們的演算法之改進比率越大，或許是因為在 t 越大時， t 倍能耗拓展圖所需選取的邊越少，更顯出選邊策略的重要性。

表 3：當 $t = 3$ 的平均廣播功率之改進比率表。

$\frac{X - \text{Greedy}}{X} (\%)$	Greedy	GN	GN-GE	GN-GN
$n=100$	-	1.963%	2.915%	3.650%
$n=200$	-	1.366%	2.651%	3.393%
$n=300$	-	1.356%	2.526%	3.268%
$n=400$	-	1.424%	2.585%	3.322%
$n=500$	-	1.395%	2.554%	3.300%
$n=600$	-	1.286%	2.537%	3.284%
$n=700$	-	1.348%	2.589%	3.334%
$n=800$	-	1.391%	2.567%	3.390%
$n=900$	-	1.302%	2.342%	3.335%
$n=1000$	-	0.773%	2.466%	3.455%

表 4：當 $t = 4$ 的平均廣播功率之改進比率表。

$\frac{X - \text{Greedy}}{X} (\%)$	Greedy	GN	GN-GE	GN-GN
$n=100$	-	2.167%	2.653%	3.640%
$n=200$	-	1.844%	2.525%	3.567%
$n=300$	-	1.598%	2.274%	3.345%
$n=400$	-	1.623%	2.310%	3.371%
$n=500$	-	1.671%	2.443%	3.544%
$n=600$	-	1.574%	2.317%	3.420%
$n=700$	-	1.703%	2.429%	3.579%
$n=800$	-	1.632%	2.354%	3.488%
$n=900$	-	1.573%	2.363%	3.486%
$n=1000$	-	0.806%	2.381%	3.715%

在第二種實驗模擬中，我們的目標是產生各種不同密度的隨機節點佈局，而我們使用的方法如下。首先我們先在任意範圍內隨機產生一張完全圖，該圖有 n 個點與 $\frac{n(n-1)}{2}$ 條邊，此時若我們想產生網路密度為 d 的圖，其中 $0 < d \leq 1$ ，只需要將所有邊照長度排序好，並挑選出第 $d * \frac{n(n-1)}{2}$ 小的邊，並以其長度當作此組輸入資料的最大感測範圍，並測試此時節點佈局的連通性。若其通訊圖為滿足連通性

則予以保留，反之則捨棄。按照上述步驟我們即可得到一張其網路密度大於等於 d 的圖，而我們只選定那些密度剛好為 10%、20%、30%...的圖來執行我們第二部分的實驗。在以下實驗中，我們設定總節點數為 $n=300$ 且 $t=2$ ，而為了清楚顯示我們提出的演算法與 Greedy 相比的成果，我們將其平均節點廣播功率百分比與其改進比率分別列於表 5 與表 6。由於篇幅問題，執行時間的數據在此省略。

表 5: 在不同網路密度下的平均廣播功率百分比。

	Greedy	GN	GN-GE	GN-GN
$d=10\%$	10.10%	9.537%	9.370%	9.281%
$d=20\%$	4.618%	4.372%	4.288%	4.249%
$d=30\%$	2.765%	2.693%	2.637%	2.612%
$d=40\%$	1.922%	1.861%	1.828%	1.810%
$d=50\%$	1.430%	1.388%	1.361%	1.349%
$d=60\%$	1.085%	1.057%	1.038%	1.029%
$d=70\%$	0.847%	0.828%	0.813%	0.805%
$d=80\%$	0.662%	0.641%	0.628%	0.622%
$d=90\%$	0.502%	0.492%	0.483%	0.478%
$d=100\%$	0.207%	0.202%	0.199%	0.197%

表 6: 當 $t=2$ 的平均廣播功率百分比之改進比率表。

$\frac{x-\text{Greedy}}{x}(\%)$	Greedy	GN	GN-GE	GN-GN
$d=10\%$	-	5.627%	7.279%	8.156%
$d=20\%$	-	5.326%	7.144%	7.994%
$d=30\%$	-	2.612%	4.634%	5.523%
$d=40\%$	-	3.192%	4.913%	5.845%
$d=50\%$	-	2.889%	4.788%	5.651%
$d=60\%$	-	2.562%	4.276%	5.165%
$d=70\%$	-	2.212%	4.010%	4.921%
$d=80\%$	-	3.190%	5.131%	5.966%
$d=90\%$	-	1.983%	3.849%	4.761%
$d=100\%$	-	2.370%	3.989%	4.909%

網路密度越密的通訊圖，代表其越多的節點互相在彼此的感測範圍內，也就是說感測節點有越大的機會花費較少的傳輸能耗，即可傳送封包給鄰近節點，如此一來平均廣播功率就越小。如表 5 所示，

對於各種不同的密度我們的演算法依然有所改進，並且由表 6 可看出我們的演算法擁有 5%~8% 的改進。

6、結論與未來展望

在本篇論文中我們探討了能耗拓展圖之功率最小化問題，並提出三個啟發式演算法。經由實驗模擬可看出，與先前最好的演算法相比，我們的演算法能夠取得更好的總能量指派，在以總節點數為變量的實驗中有著約 1%~3% 的改善，而在以網路密度為變量的實驗中也有約 5%~8% 的改善。

在時間複雜度以及執行時間方面，我們提出的演算法表現並不突出，GN 演算法的效率略差於 Greedy，而 GN-GE 和 GN-GN 兩個演算法的耗時還要高出許多。實際上如果套用更複雜的動態全局最短路徑演算法，例如[2]，這兩個演算法的 $O(n^5)$ 時間複雜度可以降到和 Greedy 持平的 $O(n^4)$ 。不過由於這類動態演算法的內容都相當複雜，因此與其相關的實作都還在進行中。

另一條值得研究的方向，是進一步將環境干擾的因素列入考量。在現實生活中，節點的廣播常常會受到自然環境的干擾，例如沙塵、降雨、甚至是高聳的山或龐大的建築，故添加干擾的變數進去可使得研究更具有實際價值。

7、參考文獻

1. Abu-Affash, A.K., et al., *Minimum power energy spanners in wireless ad hoc networks*. Wireless Networks, 2011. **17**(5): p. 1251-1258.
2. Baswana, S., R. Hariharan, and S. Sen. *Improved decremental algorithms for maintaining transitive closure and all-pairs shortest paths*. in *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*. 2002. ACM.
3. Bulusu, N., et al. *Scalable coordination for wireless sensor networks: self-configuring localization systems*. in *International Symposium on Communication Theory and Applications*

- (ISCTA 2001), Ambleside, UK. 2001.
4. Cardei, M. and D.-Z. Du, *Improving wireless sensor network lifetime through power aware organization*. Wireless Networks, 2005. **11**(3): p. 333-340.
 5. Casey, K., A. Lim, and G. Dozier, *A sensor network architecture for tsunami detection and response*. International Journal of Distributed Sensor Networks, 2008. **4**(1): p. 28-43.
 6. Chang, J.-H. and L. Tassiulas, *Maximum lifetime routing in wireless sensor networks*. IEEE/ACM Transactions on Networking (TON), 2004. **12**(4): p. 609-619.
 7. Chekuri, C. and A. Kumar, *Maximum coverage problem with group budget constraints and applications*, in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. 2004, Springer. p. 72-83.
 8. Chen, W.-T. and N.-F. Huang, *The strongly connecting problem on multihop packet radio networks*. Communications, IEEE Transactions on, 1989. **37**(3): p. 293-295.
 9. Cheriyan, J., S. Vempala, and A. Vetta. *Approximation algorithms for minimum-cost k-vertex connected subgraphs*. in *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*. 2002. ACM.
 10. Clementi, A.E., P. Penna, and R. Silvestri, *Hardness results for the power range assignment problem in packet radio networks*, in *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*. 1999, Springer. p. 197-208.
 11. Clementi, A.E., P. Penna, and R. Silvestri. *The power range assignment problem in radio networks on the plane*. in *STACS 2000*. 2000. Springer.
 12. Clementi, A.E., P. Penna, and R. Silvestri, *On the power assignment problem in radio networks*. Mobile Networks and Applications, 2004. **9**(2): p. 125-140.
 13. Eppstein, D., *Spanning trees and spanners*. Handbook of computational geometry, 1999: p. 425-461.
 14. Hajiaghayi, M., N. Immerlica, and V.S. Mirrokni. *Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks*. in *Proceedings of the 9th annual international conference on Mobile computing and networking*. 2003. ACM.
 15. Intanagonwiwat, C., R. Govindan, and D. Estrin. *Directed diffusion: a scalable and robust communication paradigm for sensor networks*. in *Proceedings of the 6th annual international conference on Mobile computing and networking*. 2000. ACM.
 16. Khuller, S., B. Raghavachari, and N. Young, *Balancing minimum spanning trees and shortest-path trees*. Algorithmica, 1995. **14**(4): p. 305-321.
 17. Kirousis, L.M., et al. *Power consumption in packet radio networks*. in *STACS 97*. 1997. Springer.
 18. Krumke, S.O., et al., *Topology control problems under symmetric and asymmetric power thresholds*, in *Ad-Hoc, Mobile, and Wireless Networks*. 2003, Springer. p. 187-198.
 19. Le Martret, C.J. and G.B. Giannakis. *All-digital impulse radio for MUI/ISI-resilient multiuser communications over frequency-selective multipath channels*. in *MILCOM 2000. 21st Century Military Communications Conference Proceedings*. 2000. IEEE.
 20. Li, M., P.-J. Wan, and Y. Wang. *Power efficient and sparse spanner for wireless ad hoc networks*. in *Computer Communications and Networks*,

2001. *Proceedings. Tenth International Conference on*. 2001. IEEE.
21. Mainwaring, A., et al. *Wireless sensor networks for habitat monitoring*. in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. 2002. ACM.
22. Megiddo, N., E. Zemel, and S.L. Hakimi, *The maximum coverage location problem*. SIAM Journal on Algebraic Discrete Methods, 1983. 4(2): p. 253-261.
23. Noury, N., et al. *Monitoring behavior in home using a smart fall sensor and position sensors*. in *Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000*. 2000. IEEE.
24. Pahlavan, K. and A.H. Levesque, *Wireless information networks*. Vol. 95. 1995: Wiley New York.
25. Rappaport, T.S., *Wireless communications: principles and practice*. Vol. 2. 1996: prentice hall PTR New Jersey.
26. Romer, K. and F. Mattern, *The design space of wireless sensor networks*. Wireless Communications, IEEE, 2004. 11(6): p. 54-61.
27. Shpungin, H. and M. Segal, *Near-optimal multicriteria spanner constructions in wireless ad hoc networks*. IEEE/ACM Transactions on Networking (TON), 2010. 18(6): p. 1963-1976.
28. Simon, G., et al. *Sensor network-based countersniper system*. in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 2004. ACM.
29. Wang, Y. and X.-Y. Li, *Minimum power assignment in wireless ad hoc networks with spanner property*. Journal of combinatorial optimization, 2006. 11(1): p. 99-112.
30. Warneke, B., et al., *Smart dust: Communicating with a cubic-millimeter computer*. Computer, 2001. 34(1): p. 44-51.
31. Yao, A.C.-C., *On constructing minimum spanning trees in k-dimensional spaces and related problems*. SIAM Journal on Computing, 1982. 11(4): p. 721-736.
32. Yick, J., B. Mukherjee, and D. Ghosal. *Analysis of a prediction-based mobility adaptive tracking algorithm*. in *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*. 2005. IEEE.