

國立成功大學
資訊管理研究所
碩士論文

以多邊定位、感測器連結性及選取種子感測器
求解感測網路定位問題

**On Localizing Sensor Networks Based on Multilateration,
Sensor Connectivity, and Seed Sensor Selection**

研究生：林子欽

指導教授：王逸琳教授

中華民國一〇二年六月

國立成功大學

碩士論文

以多邊定位、感測器連結性及選取種子感測器
求解感測網路定位問題

On Localizing Sensor Networks Based on
Multilateration, Sensor Connectivity, and Seed
Sensor Selection

研究生：林子欽

本論文業經審查及口試合格特此證明

論文考試委員：王逸琳

楊大輝

李子欣

林振榮

指導教授：王逸琳

系(所)主管：李子欣

中華民國 101 年 5 月 28 日

Abstract

A Wireless Sensor Network (WSN) consists of numerous sensors which monitor physical conditions and communicate in an ad-hoc fashion. The data collected by the sensors are usually meaningful when coupled with their locations. Since equipping GPS modules is not cost effective for localizing sensors, the Sensor Network Localization (SNL) problem is aroused to localize sensors based on few known-position anchors and internode distances, and can be formulated as an unconstrained nonlinear optimization problem. Related works show that current challenges in SNL problem are error propagation and low computational efficiency, caused by large scale network and localization failure due to noisy distances.

The localizability test, as a tool to measure the difficulty for a SNL problem, can help avoid unnecessary computation in sensor localization and improve the deployment and control of WSN. The localizability for a sensor network is closely related to its global rigidity, but is not well-defined. To test the network localizability, we propose algorithm Loc_Idx to determine the localizability of each sensor, and accordingly classify sensors into three types, depending on their difficulties to be localized. Next, we propose a novel localization algorithm named Grad_MSA to localize sensors of different types by different methods. In particular, we first apply an iterative multilateration mechanism to localize those easily localizable sensors efficiently with small errors, and then localize the rest of sensors by a gradient descent mechanism based on a nonlinear mathematical programming model. Heuristics that use shortest path and local adjustment mechanisms are also applied to refine solutions.

Moreover, an anchor-free SNL problem is studied, which is beneficial for solving WSN containing unhelpful anchors and can generalize SNL algorithms to solve related

problems such as graph realization or molecular conformation. We propose algorithm Grad_SS to localize anchor-free WSN by firstly selecting seed sensors, which are regarded as anchors in the process, localizing the rest of sensors with respect to the seed sensors, and then conducting linear coordinate transformation to calculate their original coordinates.

Numerical experiments show each heuristics in Grad_MSA takes effect to improve the solution, and among them the iterative multilateration mechanism dominates high accuracy of localization. Compared with SFSDP (Kim et al. 2009) which solves a SNL problem by Semidefinite Programming, Grad_MSA performs better in both accuracy and running time of localization. For networks comprising the same number of nodes, the accuracy and speed of localization is positively related to its localizability. For solving an anchor-free SNL problem, several protein data are tested by Grad_SS and SFSDP with the results indicating the former performs much better than the other. It also reveals that Grad_MSA is able to localize large-scale network (containing more than 10,000 nodes) in 3D space within 3 minutes, and Grad_SS manages to solve the molecular conformation problem.

Several research topics are suggested for further investigation. The localizability test remains as an open problem since there is still no efficient method to identify all localizable sensors. With a robust algorithm of localizability test, we can generate SNL benchmark test cases of different localizabilities, so that SNL algorithms can be compared on a fair basis. Localizability tests can also help select optimal seed sensors for solving anchor-free SNL problems, which may be extended for solving other Euclidean distance geometry problems.

Keywords: Sensor Network Localization Problem, Localizability, Unconstrained Nonlinear Programming, Multilateration and Gradient Descent

摘要

無線感測網路(Wireless Sensor Network, WSN)是由數個集感測、通訊與計算能力於一身的感測器組成，廣泛地應用在諸多領域中，然而大部分的應用皆需要配合感測器的位置，感測所得的資料才具有價值。本研究探討的感測網路定位(Sensor Network Localization, SNL)問題旨在利用軟體的計算能力降低實體的定位成本，亦即不在所有感測器裝備定位元件，而是僅使用少數位置已知的錨點(Anchor)感測器，透過感測器之間的距離資訊來自動定位其它位置未知的流點(Sensor)感測器，當大規模隨機佈署感測器時，如何快速且準確地定位感測器即是一大挑戰。文獻顯示當前 SNL 問題的兩大瓶頸為大規模 WSN 的定位以及使用帶誤差(Noise)的距離資訊來定位，又因感測器的訊號範圍有限且感測器間距離的量測通常不精準，在距離資訊不足且帶有誤差的情況下，當感測網路含有數百甚至數千個感測器時，其求解的困難度將大幅增加，而誤差也會在求解過程中被放大，以致求解時往往難以兼顧速度與準確度。

可定位程度(Localizability)是求解 SNL 的基礎，由於 SNL 的求解演算法通常僅能應用於可定位的 WSN，可定位程度的檢驗有助於減少無謂的計算與資源消耗，並且可加強 WSN 的控制與管理，例如我們可針對不可定位的 WSN 加強感測器的佈署，提高其可定位程度。本研究首先探討文獻中可定位網路的條件與特性，發展一個檢驗可定位程度的演算法 Loc_Idex，藉此演算法可判定一個 WSN 中可定位流點的比例，並將流點分為三類，依此分類能更有效的評比定位演算法的效能。

SNL 問題可規劃成一個無限制式的非線性最佳化問題，為能在短時間內精確定位大規模的 WSN，本研究提出的定位演算法 Grad_MSA 以梯度法(Gradient Method)為基礎，對較易定位的流點以幾何性質與感測器連結性(Connectivity)構成的多邊定位法(Multilateration)來快速地定位；而對於較難定位的流點，使用最短路徑補強未知的距離資訊，並利用感測器連結的合理性建構區域調整機制，將流點調整到合理的位置，在求解過程中將部分有可靠定位結果的流點當作錨點來輔助其它流點定位，如此重複

這些步驟來收斂各流點的定位。

本研究亦進一步探討極少錨點或無錨點(Anchor-free)感測網路的定位問題，發展無錨點網路的定位演算法不但有助於解決極少錨點的 SNL 問題，同時也可延伸 SNL 的解法到其它領域，例如分子構形(Molecular Conformation)問題。本研究提出的無錨點定位演算法 Grad_SS 係以種子感測器充當 WSN 中的錨點，而後以 Grad_MSA 求解，其中種子感測器的選擇攸關無錨點定位的成效，本研究基於貪婪法的精神，提出最大度數法與最大體積法等二種方法來選取種子感測器。

實驗結果顯示，Grad_MSA 中的迭代多邊定位法、最短路徑擴增人工節線與區域調整機制同時使用時能發揮綜效，並針對感測網路中各種不同定位難度的節點定位，其中尤以迭代多邊定位法的成效最顯著，當節線與錨點充足時能快速且準確地定位。與 Kim et al. (2009)使用半正定規劃(Semidefinite Programming)的 SFSDP 演算法比較，對於任何可定位程度的網路，本研究的 Grad_MSA 演算法均能達到較快且較準確的定位，且對於同規模的網路，隨著網路的可定位程度愈高，運算時間就愈短；而定位無錨點的網路時，Grad_SS 演算法也擁有較 SFSDP 良好的定位精準度，且運算時間隨著網路規模的增長速度也較慢，並能求解三維空間中的分子構形問題，對於含有一萬餘個節點的大規模網路，Grad_SS 能在 3 分鐘內達到相當精準的定位效果。

本研究由 SNL 問題中發現更多仍待解決的問題，其中網路的定位程度檢驗仍處於萌芽階段，發展一個能檢驗節點可否定位的演算法，除了有上述的益處以外，還能用以開發可靠的標竿測試資料，並能輔助無錨點定位演算法選擇種子感測器。此外，SNL 問題屬於歐氏距離幾何問題(Euclidean Distance Geometry Problems)中的一環，將 SNL 解法一般化將有助於應用於更多相似的問題，因此無錨點的定位演算法值得繼續發展。

關鍵字：感測網路定位問題、可定位程度、無限制非線性規劃、多邊定位法、梯度法。

誌謝

我要感謝指導教授王逸琳老師、東京工業大學的山下真教授對我無微不至的指導，以及口試委員、提案的指導教授與多次研討會中的評審對這篇論文的指教。

我要感謝同袍廖敏婷、陳昱叡、洪菁蓬、洪佳琦與李泓毅與我討論研究的一二事，在茶水間、午夜時與買宵夜的路途上歡樂地討論學術是人生越陳越美的回憶。

我要感謝大學同學陳國璋，他的筆記是我進入研究所的門票，但他一定會摸摸頭謙虛地否認。我也要感謝大學的指導教授陳育毅老師，如果我對研究有點慧根，那一定是他啟發的。

我要感謝我的 ThinkPad X60 筆電，它是陪伴我 7 年的老戰友。

除此之外，我還要感謝這一路上朋友的陪伴與家人的支持，以及老師們對我的潛移默化，讓我學到的不只是研究，更是人生。

隨著年紀的增長，要感謝的人愈來愈多，怕寫太多會被未來的自己笑，所以套摯友徐敬昕所說的：「要感謝的人太多，不如就感謝天吧！」

目錄

Abstract	I
摘要	III
表目錄/圖目錄	VIII
第一章 緒論	1
1.1 研究動機	1
1.2 研究目的	3
1.3 研究架構	5
第二章 文獻探討	7
2.1 無線感測網路定位問題	7
2.2 梯度法	9
2.3 多邊定位法	12
2.4 相關 SNL 求解方法	15
2.4.1 集中式計算演算法	15
2.4.2 分散式計算演算法	17
2.4.3 相關方法小結	18
2.5 感測器連結性	19
2.6 剛性理論	20
第三章 可定位程度檢驗	25
3.1 $D+1$ 邊定位延伸	26
3.2 D 邊定位延伸	27
3.2.1 二維空間的雙邊定位法	28
3.2.2 三維空間的雙邊定位法	29
3.3 輪狀圖延伸	31

3.4 Loc_Idx 演算法	33
3.5 流點的分類	37
第四章 定位演算法	39
4.1 Grad_MSA 演算法	39
4.1.1 迭代多邊定位法	40
4.1.2 新增人工節線	42
4.1.3 二階段梯度法	43
4.1.4 區域調整機制	44
4.1.5 停止條件	48
4.2 數值實驗	49
4.2.1 Grad_MSA 的成效	49
4.2.2 Grad_MSA 與 SFSDP 的比較	52
4.2.3 無錨點 WSN 的定位	56
4.3 小結	57
第五章 無錨點定位演算法	59
5.1 Grad_SS 演算法	59
5.1.1 種子感測器的選取	60
5.1.2 線性轉置	62
5.2 無錨點定位的優勢	62
5.3 數值實驗	64
第六章 結論與未來研究	65
6.1 結論	65
6.2 未來研究	67
參考文獻	68

表目錄

表 3-1： $D+1$ 邊定位延伸、 D 邊定位延伸與輪狀圖延伸的比較.....	34
表 5-1：Grad_SS 與 SFSDP 的比較(無錨點網路).....	64

圖目錄

圖 1-1：感測器與其硬體架構.....	1
圖 1-2：定位有助於提升感測網路的服務能力.....	2
圖 1-3：以感測器間的距離資訊計算網路的骨架.....	3
圖 1-4：以錨點的座標與距離資訊計算流點的座標位置.....	4
圖 2-1：迭代下降的收斂過程.....	9
圖 2-2：連續遞減步長的策略.....	12
圖 2-3：三邊定位法.....	13
圖 2-4：距離測量含誤差時的三邊定位法.....	13
圖 2-5：多邊定位法.....	13
圖 2-6：MDS 將距離矩陣實體化.....	16
圖 2-7：拼補策略.....	18
圖 2-8：單位圓盤圖理論.....	20
圖 2-9：感測器連結的合理性.....	20
圖 2-10：二維空間中網路圖的變形與解決方法(Yang, 2010).....	22
圖 2-11： $D+1$ 邊定位圖.....	23
圖 2-12：輪狀圖 W_6	23
圖 3-1： D 邊定位法.....	27
圖 3-2：雙重 D 邊定位法.....	28
圖 3-3：二維空間中的雙邊定位法.....	28
圖 3-4：三維空間中的雙邊定位法.....	29

圖 3-5：輪狀圖延伸.....	31
圖 3-6：輪狀圖延伸應用於邊緣節點.....	31
圖 3-7：輪狀圖的實際定位.....	32
圖 3-8：三邊定位延伸、雙邊定位延伸與輪狀圖延伸的關係.....	33
圖 3-9：Loc_Idx 實例.....	36
圖 3-10：可定位程度與通訊半徑的關係.....	36
圖 4-1：最短路徑的人工節線.....	42
圖 4-2：網路節點的分布影響到人工節線的效果.....	43
圖 4-3：推拉調整法.....	46
圖 4-4：翻轉調整法.....	47
圖 4-5：Grad_MSA 各步驟的定位精準度比較(第一類流點).....	50
圖 4-6：Grad_MSA 各步驟的定位精準度比較(第二類流點).....	51
圖 4-7：Grad_MSA 各步驟的運算時間比較.....	51
圖 4-8：Grad_MSA 與 SFSDP 的定位精準度比較.....	52
圖 4-9：Grad_MSA 與 SFSDP 的定位精準度比較(第一類流點).....	53
圖 4-10：Grad_MSA 與 SFSDP 的定位精準度比較(第二類流點).....	53
圖 4-11：Grad_MSA 與 SFSDP 的運算時間比較.....	54
圖 4-12：Grad_MSA 定位精準度的變動(第一類流點).....	55
圖 4-13：SFSDP 定位精準度的變動(第一類流點).....	55
圖 4-14：Grad_MSA 定位精準度的變動(第二類流點).....	55
圖 4-15：SFSDP 定位精準度的變動(第二類流點).....	55
圖 4-16：Grad_MSA 與 SFSDP 的定位精準度(距離資訊含誤差).....	56
圖 4-17：Grad_MSA 有/無錨點的定位精準度.....	57
圖 4-18：Grad_MSA 有/無錨點的定位結果範例.....	57
圖 5-1：種子感測器形成的暫時座標系.....	60
圖 5-2：無錨點定位演算法亦能應用錨點數極少的 WSN.....	63

第一章

緒論

1.1 研究動機

近年來無線感測網路(Wireless Sensor Network, WSN)的技術趨近成熟，其偵測與追蹤的天職被廣泛應用於環境監測、工業檢測、建設保護、軍事防護…等諸多領域，實際的應用有交通車輛或生物棲息地的追蹤、生產線的異常偵測與警報、能源的運輸與安全管理、戰場中的防禦與敵軍偵測等，甚至規模小至智能家居的多功能感測。WSN 綜合了感測、通訊與資訊處理，加上其特殊的硬體特性，帶來許多新興的研究題目，有許多 WSN 相關的議題已被明確地定義與解決，例：最小電力與最小時間的傳播(Min-energy and Min-power Broadcasting)與地理繞徑(Geographic Routing)，然而目前仍存在一些值得研究的開放問題，Locher et al. (2008)指出五大難題，包含排程(Scheduling)、拓樸控制(Topology Control)、分群(Clustering)、定位(Positioning)與時間同步(Time Synchronization)，其中定位問題即為本研究的著眼點。

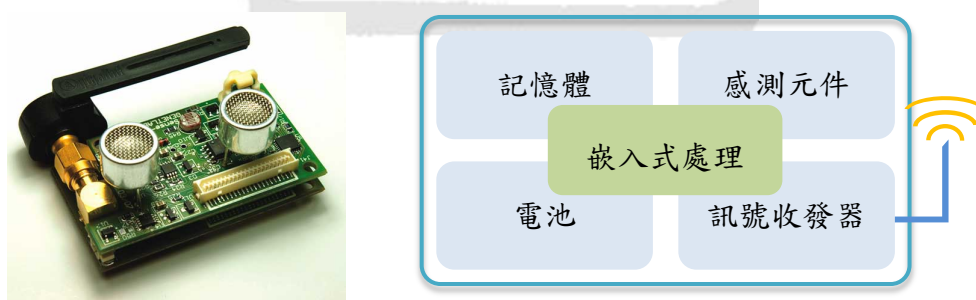


圖 1-1：感測器與其硬體架構

感測器圖片來源：Genet SenseNode

感測器是集感測、通訊與計算能力於一身的小型裝置(圖 1-1)，由於感測器通常由電池供電，電力有限，也因此通訊與計算能力都受到限制。WSN 是由諸多感測器組成，每個感測器皆只能與有限通訊範圍中的其它感測器通訊，因此當一個感測器需

傳送資料到通訊範圍以外的感測器時，則必須透過多個中繼的感測器以多重點對點跳接(Multi-hop Ad hoc)的方式來協助傳送，如此感測器之間便形成了網路架構，感測器為網路中的節點(Vertex)，而兩感測器若可以通訊則視其間存在著一條節線(Edge)。

WSN 帶來的適地性(Location-based)應用皆需要感測器的位置資訊，才能充分地發揮其感測資料的潛能，例如從高空在某地區中撒下多個偵測溫度的感測器，若所回收的感測資料不帶有位置資訊，則這些溫度資料僅能用來計算整個地區的平均溫度與整體溫度變化，但若配合了距離資訊，即可得知該地區中各處的溫度分布，並可觀察特定區域的溫度。另外，除了提升感測資料的價值，得知感測器的位置也有助於了解或提升 WSN 的服務能力(圖 1-2)，例如：涵蓋範圍(Coverage)與邊界偵測(Boundary Detection)，以及前述的地理繞徑、拓樸控制與分群(Yang, 2010)。



圖 1-2：定位有助於提升感測網路的服務能力

隨著 WSN 規模的增大，其應用又隨之增廣，但大型 WSN 的佈署存在著兩大成本，一為感測器的單位成本，另一為人力佈置與定位的成本，若個別感測器能夠自動定位，則可以省卻人工定位的麻煩。定位中常見的方法是採用全球定位系統(Global Positioning System, GPS)，但如果在每個感測器上皆裝載 GPS 衛星訊號接收器，其單位成本相當可觀，且訊號容易受到遮蔽，在氣候不佳或室內往往無法以 GPS 定位。因此，為了降低成本，如何只配置少量的 GPS 模組或甚至不使用 GPS，並配合 WSN 本身已知的資訊(如：感測器間的距離)，讓各感測器自動協調找出最佳的定位，此即為本研究所要探討的感測網路定位問題(Sensor Network Localization, SNL)。

1.2 研究目的

本研究將 WSN 中少數已知位置的感測器稱為錨點(Anchor 或 Beacon)，其可能是透過 GPS 定位或由人工擺置，而其它位置未知且待定位的感測器稱為流點(Sensor)，則定位就是指利用已知的資訊去計算流點的位置。文獻中 WSN 的定位通常包含兩個部分，一為取得有利定位的地理資訊，二為根據可用的資訊計算定位，前者涉及硬體的設置與諸多環境的變數，不在本研究的範疇中，本研究僅探討後者。

定位時可用的地理資訊包含了感測器間的距離和相鄰與否，以及感測器間相對的角度，收訊角度(Angle of Arrival, AoA)固然對定位的準確度有很大的幫助，但其需要特殊的指向性天線來測量角度；而感測器間的距離通常是透過無線射頻計算而得，例如：訊號接收強度(Received Signal Strength, RSS)或訊號傳送時間差(Time Difference of Arrival, TDoA)。為了一般化 SNL 問題，本研究不使用角度資訊，僅考慮距離資訊，而如果已知兩感測器間的距離，就意味其為相鄰。本研究不探討如何測量距離，關於定位用地理資訊的測量技術可參考 Mao et al. (2007)。

簡而言之，本研究所探討的 SNL 問題即為在給定感測器間的距離資訊與錨點的座標之下，計算流點的位置。以圖 1-3 為例，當給定流點間的距離資訊時，藉由節線與節點間的互相約束，計算流點的網路骨架，這骨架可任意平移與旋轉；但若已知該網路存在於平面空間中，並給定數個錨點(綠色方塊)，以及錨點與流點間的距離資訊，則如圖 1-4 所示，可將此網路骨架固定到平面空間中的座標位置。

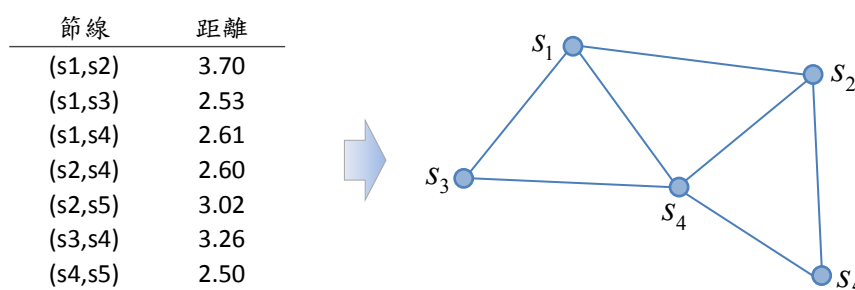


圖 1-3：以感測器間的距離資訊計算網路的骨架

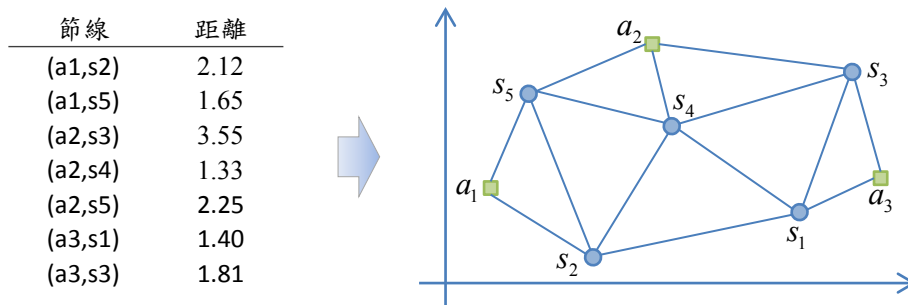


圖 1-4：以錨點的座標與距離資訊計算流點的座標位置

近年學者們針對 SNL 問題提出各式各樣的解決方法，但沒有一種方法可以適用所有的定位情境，各研究對於 SNL 問題的假設與適用範疇都有些許差異，例如定位時可使用的資源、感測器佈署的需求與限制等，集中式或分散式計算亦影響到其定位效果，並且又缺乏標竿(Benchmark)測試資料，因此各方法間難以比較優劣。儘管如此，求解 SNL 基本的目標皆是發展快速且準確定位的演算法，本研究也以此為主要目標，並期望將定位演算法一般化，使其能適用於更多的問題情境。

SNL 問題中的 WSN 通常是稀疏網路，也就是只有部分感測器間的距離資訊是已知的，定位稀疏的網路已證明是個 NP-hard 問題(Aspnnes et al., 2004)，但除此之外，SNL 還具有兩大挑戰性：處理帶有誤差(Noise)的距離資訊，以及定位大規模 WSN。測量距離時，由於電波傳遞的過程會衰落(Fading)或經過反射、折射或散射造成多徑(Multipath)傳播，產生測量的誤差，如此的誤差易在定位過程中放大與傳播，使得定位結果不準確；而對於含有數千個感測器的大規模網路，除了因為節點數增加，使得計算複雜度提高，嚴重加長求解時間，同時誤差也更容易在計算中累積放大。

本研究使用電腦模擬產生 WSN，並以集中式計算定位，雖然不考慮實際上如何測量距離，但在實驗中將使用人工產生的誤差參雜在距離資訊中，期望使用帶有誤差的距離資訊仍有穩定的定位效果；而大規模 WSN 的定位勢在必行，本研究亦試圖設計能同時兼顧效率與效能的最佳化定位演算法，使其處理大規模的網路時能保持定位準確度與計算時間的平衡。另外，為了增加定位演算法的實用性，本研究同時探討二維與三維空間中 WSN 的定位，並最少化錨點的數量，使得定位演算法在有限錨點的

環境中也能達到滿意的定位效果，甚至不需使用錨點即可定位。藉由如此對 SNL 問題限制的放鬆，有助於將定位演算法應用在其它研究領域中，例如 SNL 問題可一般化為圖實現(Graph Realization)問題，其為在給定 $G(V, E)$ 與各節線的非負權重 $\{w_{ij} : (i, j) \in E\}$ 之下，求解 G 在歐幾里得空間 \mathbb{R}^D 中的體現(Realization)，也就是放置各節點在空間中，使其各節點間的距離符合給定的節線權重。

此外，跳脫定位的演算法，對於定位目標 WSN 的分析是個較難也較少為人討論的議題，對於網路的了解如：定位該網路的難易度與定位精準度的上限，衍伸出了「可定位程度」(Localizability)的議題，可定位程度是定位的成敗關鍵，也有助於提升 WSN 的服務，因此本研究也試圖以感測器間的連結性質來分析欲定位的 WSN，發展檢驗可定位程度的演算法，將其視為定位的前置處理，並依此來檢視定位的成效，藉此也能用於產生可靠的標竿測試資料。

1.3 研究架構

在本研究中，首先融合 Yang (2010)所提出的 WHEEL 演算法，以及定位常見的三邊定位法(Trilateration)與雙邊定位法(Bilateration)，建立檢驗 WSN 可定位程度的演算法 Loc_Idx，由此可判斷給定的 WSN 中可定位流點的比例，並將流點依可定位程度分為三類，作為之後評比定位結果的分類。

接著本研究建構定位演算法 Grad_MSA，將 SNL 問題規劃成一個無限制的非線性最佳化問題，以梯度法(Gradient Method)為最佳化的基礎，在錨點充足的情況下，利用多邊定位法(Multilateration)快速定位部分錨點周圍的流點，並經過梯度法的修正後，將這些已定位的流點當作錨點來輔助其它流點定位，如此的動作經過多次迭代後，可將大部分容易定位的流點皆轉換成錨點。而後對於剩下不易定位的流點，本研究提出啟發式演算法來輔助定位，包含使用最短路徑來彌補稀疏的距離資訊，以及利用感測器連結性的合理與否來調整定位不良的流點，再配合二階段梯度法的多次迭代，逐

步收斂流點的定位結果。

為了進一步最小化錨點的成本，本研究延伸定位演算法 Grad_MSA 以求解極少錨點或無錨點 WSN 的定位，提出能選擇種子感測器的無錨點定位演算法 Grad_SS，以啟發式演算法選取適當的種子感測器作為錨點，再以 Grad_MSA 求解，並將此方法用於求解分子構形(Molecular Conformation)問題。

以下本論文的章節結構為：

第二章 首先介紹本研究中對 SNL 問題的定義與其非線性數學規劃模型，並簡單介紹求解 SNL 的基本概念，包含梯度法與多邊定位法，並探討求解同類型 SNL 問題的文獻，而後介紹在部分啟發式演算法中所利用的感測器連結性，以及可定位程度檢驗中核心的剛性理論。

第三章 發展檢驗可定位程度的演算法 Loc_Idx，闡述其構成的要素，並依可定位程度的檢驗將流點分為三類，以應用於分析第四章與第五章的實驗結果。

第四章 提出本研究的定位演算法 Grad_MSA，其中包含多邊定位法、最短路徑演算法與區域調整機制等的詳細流程，並以數值實驗探討本演算法中各步驟的成效，以及與 Kim et al. (2009)的 SFSDP 演算法比較。

第五章 基於第四章，發展無錨點定位演算法 Grad_SS，提出選取種子感測器的啟發式演算法，並以分子構形的資料組測試其定位的成效。

第六章 總結本研究，並提出未來可能的研究方向。

第二章

文獻探討

由於相關研究中對於 SNL 問題多有些微不同的定義與假設，為了不致於混淆，本章首先參考 Kim et al. (2009) 定義 SNL 問題，並建構本問題的非線性規劃模型，其後小節說明本研究中求解非線性最佳化問題的基礎—梯度法，以及幾何定位中常見的多邊定位法，之後回顧近十餘年中，求解相關 SNL 問題的文獻。而後介紹 WSN 中有利於定位的性質，包含感測器的連結性以及網路的剛性(Rigidity)理論，前者將用於發展輔助定位的啟發式演算法，後者則是探討可定位程度時的核心。

2.1 無線感測網路定位問題

在歐幾里得空間 $\mathbb{R}^D : D=2,3$ 中，感測網路 $G=(V,E)$ 包含未知位置的流點 V_s 與已知位置的錨點 V_a ，其中流點以 $1, \dots, |V_s|$ 編號，而錨點以 $|V_s|+1, \dots, |V_s|+|V_a|$ 編號，當 $V_a = \emptyset$ 時，則為無錨點(Anchor-free)的 SNL 問題。令感測器 i 的實際位置為 $\mathbf{a}_i : i \in V$ ，其在 h 維度的座標值記為 $a_{ih} : h=1, \dots, D$ ，任二感測器 i 與 j 之間的歐氏距離以 d_{ij} 表示，其展開為

$$d_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\| = \left(\sum_{h=1}^D (a_{ih} - a_{jh})^2 \right)^{1/2}, \quad i, j \in V \quad (2.1)$$

然而，感測器間距離的測量存在不可避免的誤差，定位時無法得知實際的距離，因此實作中以 \hat{d}_{ij} 表示感測器間測量的距離，且 $\hat{d}_{ij} \approx d_{ij}$ ，而本研究視距離資訊具有對稱性，故 $\hat{d}_{ij} = \hat{d}_{ji}$ 。設所有感測器的訊號傳遞半徑皆為 r ，當二感測器可相互通訊，則以一條節線表示，則全部節線的集合為 $E = E_s \cup E_a$ ，其中流點與流點間節線的集合以

$$E_s = \left\{ (i, j) \mid 1 \leq i < j \leq |V_s|, \hat{d}_{ij} \leq r \right\} \quad (2.2)$$

表示，而流點與錨點間節線的集合則為

$$E_a = \left\{ (i, j) \mid i \in V_s, j \in V_a, \hat{d}_{ij} \leq r \right\} \quad (2.3)$$

流點的實際位置 $\mathbf{a}_i : i \in V_s$ 在定位過程中是未知且待解的，故求解時的定位位置以 $\mathbf{x}_i : i \in V_s$ 表示。SNL 問題即是在給定測量的距離 $\hat{d}_{ij} : (i, j) \in E$ 與錨點的位置 $\mathbf{a}_i : i \in V_a$

之下，求解流點的位置 $\mathbf{x}_i : i \in V_s$ 。令 \bar{d}_{ij} 為定位後感測器間的距離：

$$\bar{d}_{ij} = \begin{cases} \|\mathbf{x}_i - \mathbf{x}_j\|, & i, j \in V_s, i \neq j \\ \|\mathbf{x}_i - \mathbf{a}_j\|, & i \in V_s, j \in V_a \end{cases} \quad (2.4)$$

則最佳的定位結果應是 $\bar{d}_{ij} \approx \hat{d}_{ij} : (i, j) \in E$ ，因此可將 SNL 問題可規劃成一個無限制的非線性最佳化問題(Unconstrained Nonlinear Optimization Problem)：

$$\min \sum_{(i,j) \in E} \left(\bar{d}_{ij}^2 - \hat{d}_{ij}^2 \right)^2 \quad (2.5)$$

其表示最小化定位結果的距離資訊與給定的距離資訊之間的差距，而為了便於計算，以平方來取代計算歐氏距離時的根號以及計算兩距離差距時的絕對值符號。

本研究產生虛擬 WSN，在定位過程中將流點的實際位置 $\mathbf{a}_i : i \in V_s$ 視為未知，以式 (2.5) 為目標求解流點的最佳定位 $\mathbf{x}_i : i \in V_s$ ，並在定位結束後，以流點的定位位置與實際位置的均方根誤差(Root Mean Square Deviation, RMSD)來判別定位演算法的效果，RMSD 定義為

$$RMSD(G) = \left(\frac{1}{|V_s|} \sum_{i \in V_s} \|\mathbf{x}_i - \mathbf{a}_i\|^2 \right)^{1/2} \quad (2.6)$$

當 RMSD 越低，表示定位的精準度越高；反之，當 RMSD 越高，則表示精準度越低。假設 WSN 分布於一平方公里的空間中，則當 RMSD 值低於 1e-3，就表示流點的平均誤差低於 1 公尺。

2.2 梯度法

如式(2.5)般的無限制最佳化問題可用梯度法求解，梯度法基於迭代下降(Iterative Descent)的概念，對於一個可微分的最小化目標函式 $f(x)$ ，從初始解 x_0 出發，產生一序列的 x_1, x_2, \dots ，使得每次迭代中皆能降低 $f(x)$ 函式值：

$$f(x_{p+1}) < f(x_p), \quad p = 0, 1, 2, \dots \quad (2.7)$$

如此在迭代過程中將逐漸收斂 $f(x)$ 至最小值，如圖 2-1 所示。

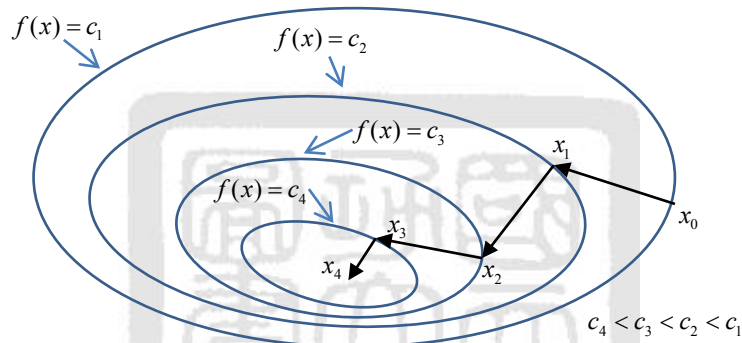


圖 2-1：迭代下降的收斂過程

每回迭代間 x_p 的變化可表示成

$$\begin{aligned} x_{p+1} &= x_p + \alpha_p d_p \\ &= x_p - \alpha_p D_p \nabla f(x_p), \quad p = 0, 1, 2, \dots \end{aligned} \quad (2.8)$$

α_p 為第 p 次迭代下降的步長(Step Size)， d_p 則是下降方向(Descent Direction)，步長與下降方向的選擇會影響到迭代收斂的效果，不同的選擇各有其優缺點與適用的問題。

其中 $d_p = -D_p \nabla f(x_p)$ ， $-\nabla f(x_p)$ 是在 $f(x_p)$ 的負梯度， D_p 為一正定對稱矩陣(Positive Definite Symmetric Matrix)，當 D_p 為單位矩陣時

$$D_p = I, \quad p = 0, 1, 2, \dots \quad (2.9)$$

此種下降方式為最陡下降法(Steepest Descent)，因其計算簡單，容易應用於非線性最

佳化問題，但卻可能因為不當設定的步長，使得迭代次數增加而收斂緩慢。

牛頓法(Newton's Method)定義

$$D_p = \left(\nabla^2 f(x_p) \right)^{-1}, \quad p = 0, 1, 2, \dots \quad (2.10)$$

其使用高階微分項做為迭代搜尋的方向，能減少迭代次數，但計算較複雜，而且如果初始解不在最佳解附近，其收斂的效率仍然有限。此外，有諸多改良的梯度法來自牛頓法，如：高斯牛頓法(Gauss-Newton Method)與類牛頓法(Quasi-Newton Method)等。

步長也有多種的選擇方式，其中最簡單的方法是固定 α_p 為一個大於零的常數，使每次迭代的步長一致，但可能造成不易收斂或收斂緩慢的情況；或者使用最小化規則(Minimization Rule)，找到符合

$$f(x_p + \alpha_p d_p) = \min_{\alpha \geq 0} f(x_p + \alpha d_p), \quad p = 0, 1, 2, \dots \quad (2.11)$$

條件的 α_p ，但通常無法確切地求得 α_p ，得依靠線搜尋(Line Search)技巧，漸進增加或減少 α ，直到 $f(x_p + \alpha d_p)$ 夠小為止；或者使用插值法(Interpolation)不斷縮減最小值所在的區間，直到區間的間距夠小為止。此外，使用了連續遞減步長(Successive Step Size Reduction)概念的 Armijo rule(Armijo, 1966)中設定 α_p 符合

$$f(x_p) - f(x_p + \alpha_p d_p) \geq -\sigma \alpha_p \nabla f(x_p)' d_p, \quad p = 0, 1, 2, \dots \quad (2.12)$$

其中 σ 為一個接近零的實數，通常 $\sigma \in [10^{-5}, 10^{-1}]$ ，設 $\alpha_p = \beta^q \gamma$ ， γ 是一個大於零的初始步長，在許多應用中大多設為1，而 β 是遞減步長的乘數，通常 $\beta \in [0.1, 0.5]$ ，視對初始步長的信心而定。依序代入 $q = 1, 2, \dots$ 後，隨著 q 愈大，步長則愈小，直到找到符合式(2.12)的 q^* ，則 $\beta^{q^*} \gamma$ 即為最適的步長。Armijo rule 直覺地表達了不僅找到的最適步長能降低函式值，也就是 $f(x_p) > f(x_p + \alpha_p d_p)$ ，更確保該步長能達到足夠的改善，至少降低函式值 $-\sigma \alpha_p \nabla f(x_p)' d_p$ 。

當 $x_p : p = 0, 1, 2, \dots$ 為無窮序列時，其極限點即是此目標函式的最佳解，然而實作中電腦必須在有限的迭代次數內計算出答案，因此必須設定迭代的終止條件，譬如設定當 $\|\nabla f(x_p)\| < \varepsilon$ 或 $\|\nabla f(x_p)\| / \|\nabla f(x_0)\| < \varepsilon$ 即停止迭代， ε 是一個相當小的門檻值，其表示當最近一次的解 x_{p+1} 對目標函式值改善程度很小時，就視同已計算得最佳解。

本研究應用 Leung & Toh (2009)設計的梯度法，其使用回溯線搜尋(Backtracking Line Search)，下降方向為最陡下降法，也就是 $d_p = -\nabla f(x_p)$ ，並使用類似 Armijo rule 的連續遞減步長策略，其演算法如下：

演算法 1 : GRADIENT DESCENT (Leung & Toh, 2009)

```

procedure GradientDescent
  for  $p = 0, 1, 2, \dots$  do
     $d_p \leftarrow -\nabla f(x_p)$ 
     $x_{p+1} \leftarrow \text{SuccessiveDescent}(x_p, \gamma, d_p)$ 
  until  $\frac{|f(x_p) - f(x_{p+1})|}{1 + |f(x_p)|} < \varepsilon$ 
end procedure

procedure SuccessiveDescent( $x, \alpha, d$ )
  while  $f(x) - f(x + \alpha d) > -\sigma \alpha \nabla f(x)'d$  do
     $x \leftarrow x + \alpha d$ 
     $\alpha \leftarrow \alpha \beta$ 
  return  $x$ 
end procedure

```

在此迭代下降的過程中，每次迭代中皆呼叫 $\text{SuccessiveDescent}(x, \alpha, d)$ ，使得 x 不斷朝著同一方向移動，但是每次移動的步長逐次遞減，如圖 2-2 所示，不同於求解式(2.12)以找到最適的步長再移動 x ，連續遞減步長策略是直接漸進地移動 x ，直到對函式值的改善太小時，再重算梯度，以新的方向繼續搜尋最佳解。

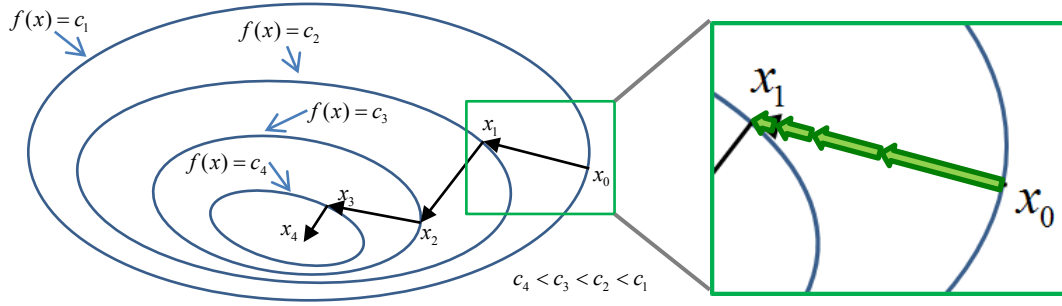


圖 2-2：連續遞減步長的策略

儘管以最陡下降法作為下降方向可能使得收斂速度過慢，但其簡化收斂的機制，快速地選擇下降方向，且配合連續遞減步長的策略，能有效率地求解式(2.5)。此外，Leung & Toh (2009)亦提及式(2.5)是一個非凸性問題(Nonconvex Problem)，其存在諸多區域最佳解(Local Minimum)，因此如果初始解離全域最佳解(Global Minimum)不夠近，使用區域搜尋的方法(如：梯度法或牛頓法)可能只能找到區域最佳解，由此可見，比起發展複雜的區域搜尋機制，找到良好的初始解更是首要之務。

2.3 多邊定位法

Savvides et al. (2001)指出了三邊定位法、三角定位法(Triangulation)與最大可能法(Maximum Likelihood)是定位問題的基礎解法，其中三角定位法需要收訊的角度資訊，僅適用於AoA的定位系統中；而三邊定位法常見用於GPS的定位，透過定位衛星與GPS裝置之間的距離形成多條距離限制式，則可侷限該裝置的可行位置。以二維空間為例，設一個流點(節點1)與其周圍三個錨點(節點2,3,4)的距離分別為 d_{12} 、 d_{13} 與 d_{14} ，若以錨點為圓心並以距離為半徑，所繪成三個圓的交點便是流點的位置(圖 2-3c)；但當流點的周圍僅有二個錨點時，則流點有一或二個可能的位置(圖 2-1b)；而當只有一個錨點時，則流點可能的位置是錨點圓周上的任意一點(圖 2-1a)。由此可知，在二維空間中，三邊定位法至少需要三個不共線的錨點才能確認一個流點的唯一位置。

然而，感測器間的距離測量通常含有不可避免的誤差，即使有三個錨點，其所繪成的圓也無法交於唯一的一點，如圖 2-4 中三圓可能交集於一範圍或甚至完全不相交，

在這樣的情況下，三邊定位法必須找出一個折衷的流點位置(如圖 2-4 中的黃色節點)，則需要用最小化平方誤差作為目標式來求解，以圖 2-4 的網路為例即是求解

$$\min \sum_{k=2}^4 \left(\hat{d}_{1k} - \|\mathbf{x}_1 - \mathbf{a}_k\| \right)^2 \quad (2.13)$$

如此以最小平方差(Least Squares)的規劃即為 Savvides et al. (2001)所述的最大可能法，可以用梯度法求解，而 SNL 問題的目標式(2.5)也是基於此規劃而成。

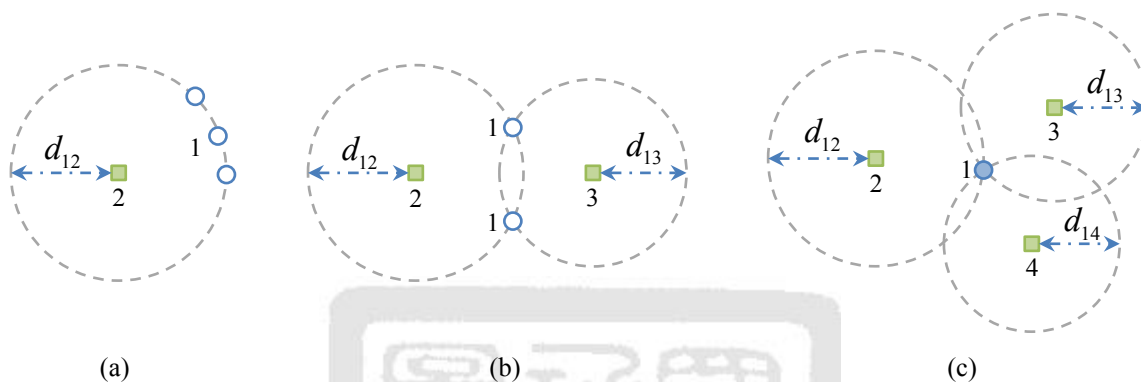


圖 2-3：三邊定位法

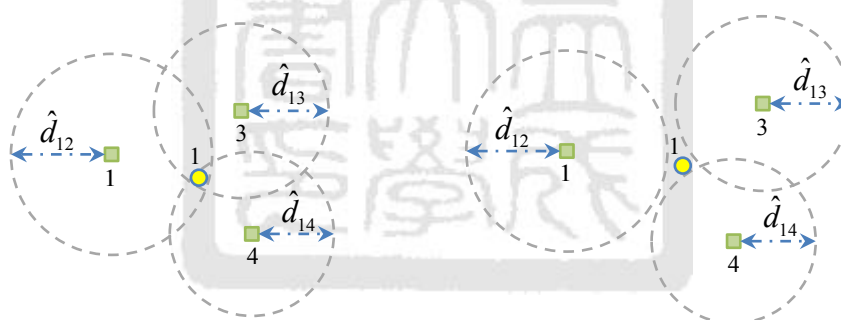


圖 2-4：距離測量含誤差時的三邊定位法

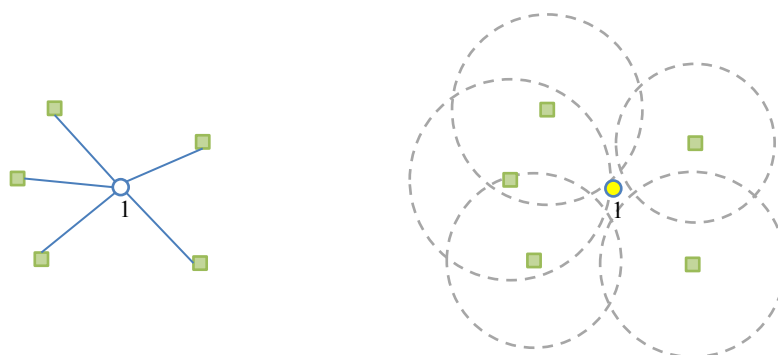


圖 2-5：多邊定位法

將三邊定位法擴展到 D 維空間中，且至少有 $D+1$ 個錨點時，稱為多邊定位法，

令一個流點(節點 1)的周圍至少有 $n-1$ 個錨點(節點 $2,3,\dots,n$)，如圖 2-5 中以綠色方塊表示錨點，當已知錨點的位置，以及流點與錨點間的量測距離，固然能以類似式(2.13)的最小平方差規劃，並以梯度法求解。然而，透過簡單的問題轉換，可以使得多邊定位法在多項式時間內快速求解，以二維空間為例，首先依式(2.1)展開每條節線的距離限制式為

$$\begin{aligned}\hat{d}_{12}^2 &= (x_{11} - a_{21})^2 + (x_{12} - a_{22})^2 \\ \hat{d}_{13}^2 &= (x_{11} - a_{31})^2 + (x_{12} - a_{32})^2 \\ &\vdots \\ \hat{d}_{1n}^2 &= (x_{11} - a_{n1})^2 + (x_{12} - a_{n2})^2\end{aligned}\quad (2.14)$$

展開式(2.14)中 $n-1$ 條等式後，將第一條等式分別與其它等式相減可得到

$$\begin{aligned}\hat{d}_{13}^2 - \hat{d}_{12}^2 &= a_{31}^2 + a_{32}^2 - a_{21}^2 - a_{22}^2 - 2(a_{31} - a_{21})x_{11} - 2(a_{32} - a_{22})x_{12} \\ \hat{d}_{14}^2 - \hat{d}_{12}^2 &= a_{41}^2 + a_{42}^2 - a_{21}^2 - a_{22}^2 - 2(a_{41} - a_{21})x_{11} - 2(a_{42} - a_{22})x_{12} \\ &\vdots \\ \hat{d}_{1n}^2 - \hat{d}_{12}^2 &= a_{n1}^2 + a_{n2}^2 - a_{21}^2 - a_{22}^2 - 2(a_{n1} - a_{21})x_{11} - 2(a_{n2} - a_{22})x_{12}\end{aligned}\quad (2.15)$$

含有 $n-2$ 條線性方程式，將其整理為

$$2 \begin{bmatrix} a_{31} - a_{21} & a_{32} - a_{22} \\ a_{41} - a_{21} & a_{42} - a_{22} \\ \vdots & \vdots \\ a_{n1} - a_{21} & a_{n2} - a_{22} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} \end{bmatrix}^T = \begin{bmatrix} a_{31}^2 + a_{32}^2 - \hat{d}_{13}^2 - a_{21}^2 - a_{22}^2 + \hat{d}_{12}^2 \\ a_{41}^2 + a_{42}^2 - \hat{d}_{14}^2 - a_{21}^2 - a_{22}^2 + \hat{d}_{12}^2 \\ \vdots \\ a_{n1}^2 + a_{n2}^2 - \hat{d}_{1n}^2 - a_{21}^2 - a_{22}^2 + \hat{d}_{12}^2 \end{bmatrix}\quad (2.16)$$

這是個超定義方程組(Overdetermined System)，也就是線性系統方程式的數目多於未知數的數目，通常無解，故將其化成矩陣的形式：

$$\mathbf{A}\mathbf{x}_1 = \mathbf{B}\quad (2.17)$$

以求解最小平方差 $\|\mathbf{A}\mathbf{x}_1 - \mathbf{B}\|^2$ ，其中 \mathbf{A} 與 \mathbf{B} 分別為

$$\mathbf{A} = 2 \begin{bmatrix} a_{31} - a_{21} & a_{32} - a_{22} \\ a_{41} - a_{21} & a_{42} - a_{22} \\ \vdots & \vdots \\ a_{n1} - a_{21} & a_{n2} - a_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} a_{31}^2 + a_{32}^2 - \hat{d}_{13}^2 - a_{21}^2 - a_{22}^2 + \hat{d}_{12}^2 \\ a_{41}^2 + a_{42}^2 - \hat{d}_{14}^2 - a_{21}^2 - a_{22}^2 + \hat{d}_{12}^2 \\ \vdots \\ a_{n1}^2 + a_{n2}^2 - \hat{d}_{1n}^2 - a_{21}^2 - a_{22}^2 + \hat{d}_{12}^2 \end{bmatrix}\quad (2.18)$$

則求解此最小平方差的解為

$$\mathbf{x}_1^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (2.19)$$

且由式(2.16)中可知，欲定位 D 維空間中一個流點，其必須至少相鄰 $D+1$ 個錨點，才足以產生 D 條線性方程式，以利定位其位置。

2.4 相關 SNL 求解方法

針對求解類似 2.1 節中所定義之 SNL 問題，過去的研究提出各種解決方案，在此依其運算方式分類並簡略地探討文獻，主要分為集中式計算與分散式計算，集中式計算是將所有可用的定位資訊集中在一個核心的運算器上處理；而分散式則應用了感測網路中多重點對點跳接的特性，在個別感測器上依區域可用的資訊來計算位置。對於 SNL 問題更詳細的文獻探討可參考 Mao et al. (2007)與 Liu & Yang (2011)。

2.4.1 集中式計算演算法

使用集中式運算的文獻可主要分為兩大派，其一將 SNL 問題規劃成半正定規劃 (Semidefinite Programming, SDP)問題，再以已知的 SDP 求解演算法求解；而另一種則利用多維標度(Multidimensional Scaling, MDS)方法求解。SDP 常見的形式為

$$\begin{aligned} \min \quad & C^T X \\ \text{s.t.} \quad & AX < B \\ & X \succeq 0 \end{aligned} \quad (2.20)$$

其類似於線性規劃，唯獨其變數 X 為半正定矩陣。Doherty et al. (2001)首先將 SNL 問題以 SDP 規劃，其將感測器間的距離規劃成凸限制式(Convex Constraints)：

$$\|\mathbf{x}_i - \mathbf{x}_j\| < r, \quad (i, j) \in E \quad (2.21)$$

表示限制相鄰兩點之間的距離小於通訊半徑，然而，使用凸函數規劃必須將錨點擺放在所有流點的外圍時才能有效運作。該研究中亦同時探討了其它不能以 SDP 求解的

非凸限制式，例如

$$\begin{aligned}\|\mathbf{x}_i - \mathbf{x}_j\| &= r, & (i, j) \in E \\ \|\mathbf{x}_i - \mathbf{x}_j\| &> r, & i, j \in V, (i, j) \notin E\end{aligned}\quad (2.22)$$

前者限制兩點間的距離相當於通訊半徑，而後者對於不相鄰的兩節點，限制其距離應超過通訊半徑，此限制式提供一個「推開」不相鄰兩點的概念，除此之外，Doherty et al. (2001)亦提出混合角度限制式的概念來縮減可行解空間。

Biswas & Ye (2004)同樣運用 SDP 規劃 SNL 問題，其鬆弛了 Doherty et al. (2001)未使用的非凸函數離限制式(2.22)，並使其線性化便於求解，然而如此鬆弛的做法提升搜尋空間的大小，無法有效的應付大規模問題。在以 SDP 解 SNL 問題的後續研究中，許多學者試圖改善 SDP 來加速求解，相關文獻的探討可參考 Kim et al. (2009)，該篇亦應用了 SDP 中矩陣的稀疏性(Sparsity)加速求解，並特別著眼於解決距離資訊含有誤差時的定位，然而整體計算時間仍是依網路規模變大而遞增。此外特別一提的是，Leung & Toh (2009)以 SDP 與分治法(Divide and Conquer)求解圖實現(Graph Realization)問題，並提出一個修正定位結果的梯度法，其在實際應用中能快速收斂。

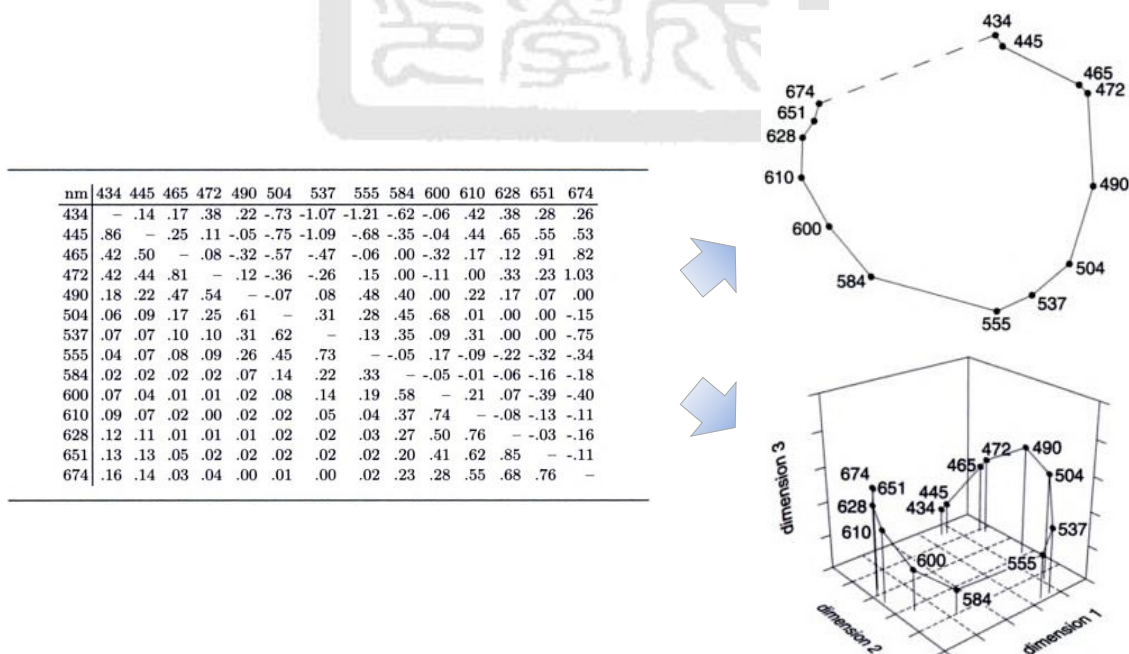


圖 2-6：MDS 將距離矩陣實體化

圖片來源：(Borg & Groenen, 2005)

另一集中式演算法的研究領域為使用 MDS 求解 SNL 問題，MDS 是一種資料簡化的技巧，其目的為挖掘資料中隱藏的結構，可將多維度資料繪製在較低維度的空間，並在降低資料複雜性的同時，保留了資料的特徵，其已被應用於諸多領域，如行銷研究中用以探討消費者對各種品牌的喜好和產品認知，或心理學中以特質的相似性找出人們認知的構面。MDS 能分析任何形式的距離矩陣或相似(Similarity)矩陣，而古典的 MDS 即是給定點與點之間的歐幾里得距離矩陣，求解各點在某個維度空間中的座標 (Borg & Groenen, 2005)，以圖 2-6 為例，給定一距離矩陣如圖左方所示，MDS 可將其點的座標繪製於二維或三維空間中(圖 2-6 右方)，若將相鄰點連線，可發現無論在二維或三維空間中，連線皆保持著其環狀的特徵。

Shang et al. (2004)首先使用 MDS 求解 SNL 問題，給定感測器間的距離，以 MDS 求出這些感測器在二維或三維空間中的網路骨架，但 MDS 運作時仰賴所有點與點間的距離資訊，而 WSN 的距離矩陣卻是稀疏的，故該研究以最短路徑補齊距離資訊，然而因為最短路徑估計的距離可能與實際距離有很大的偏差，所以計算所得的網路骨架仍需要經過特別的修補動作來改善結果，且網路骨架可能是原圖中多點位置任意翻轉後的結果，若配合錨點的座標資訊，有助於修正定位的結果。在後續以 MDS 求解 SNL 的研究中，多致力於改善稀疏的距離矩陣，以及修正 MDS 定位的結果。

2.4.2 分散式計算演算法

在分散式演算法中，個別感測器應用了與鄰近感測器的關係來定位，常利用拼補 (Patching) 的策略來達成，其為將整體網路圖分成許多細碎且重疊的子網路圖 (Local Map)，通常一個子網路只包含一個節點與其周圍通訊範圍內的其它節點，各子網路內以節點間的距離可以繪成一個子網路骨架(圖 2-7a)，再利用兩個子網路中約二或三個重疊的節點(圖 2-7 中黃色節點)將兩個子網路拼補成一個更大的子網路(圖 2-7b)，重複如此動作最終將所有子網路拼成整體網路圖。

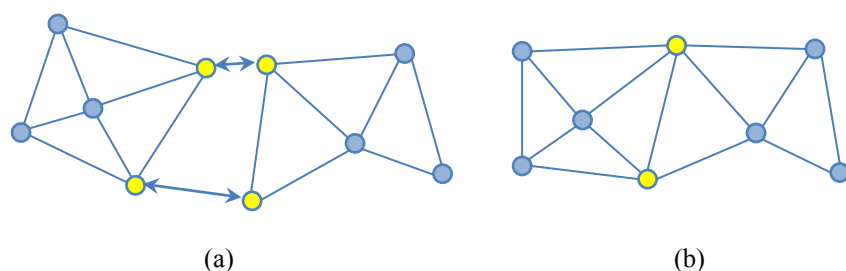


圖 2-7：拼補策略

對於二維空間的 SNL 問題，Savvides et al. (2001)與 Moore et al. (2004)都使用類似的拼組策略，前者利用三個錨點來定位一個流點後，將此流點視為錨點，再拿來定位其它未知位置的流點，稱之為迭代多邊定位法(Iterative Multilateration)；而後者證明四個節點間包含至少五條節線時，可建構成一個強韌四邊形(Robust Quadrilateral)，強韌四邊形確定此四個節點唯一的網路骨架，由此強韌四邊形中的三個節點与其它強韌四邊形重疊，則可將兩個強韌四邊形拼在一起，重複如此可拼湊出整體的網路骨架，再以周圍的錨點來修正其定位。

部分研究中在只知道感測器間相鄰與否的強況下，應用 WSN 多重點對點跳接的特性，提出另類的距離估算方法，Niculescu & Nath (2003)設計「DV-hop」演算法，由各錨點向四周鄰近感測器廣播其位置，透過類似廣度優先搜尋法(Breadth-First Search)的方式將錨點的位置資訊廣播到每個感測器，並以錨點與錨點間的距離與中繼節點(Hop)數依比例估算各流點到各錨點的距離，各流點再使用最近的三個或更多個錨點的位置以及距離來定位。Savarese et al. (2002)亦使用類似 DV-hop 的演算法，增加一個修正的步驟，將流點與周圍流點間的距離也納入考量，並使用最小平方法增加定位的精準度。

2.4.3 相關方法小結

各研究的演算法皆在定位速度與精準度間取得一個折衷，集中式計算通常能提供較高精準度的定位結果，但隨著節點數增加，求解速度也嚴重遲緩，以致無法求解大規模網路問題；而分散式演算法通常能快速求解，較不受網路規模大小影響，但其定

位的資訊僅來自鄰近的節點，結果較不精確。Mao et al. (2007)亦指出分散式演算法通常較集中式運算複雜也較難以設計，例如在分散式的定位方法總是可以應用在集中式的系統中，但分散式的定位方法不一定能應用在分散式的系統。而 Savvides et al. (2001)則指出在實際運作上，集中式運算的核心處理在取得所需的距離資訊時，必須事先知道所有感測器到集中處理器的傳訊路徑，且也存在著時間同步的問題。

距離測量的誤差在迭代計算中傳播造成錯誤的累積，以致於計算複雜度增加，無論在集中式或分散式計算中都容易出現此問題。近年有部分研究發展混合式演算法，試圖結合集中式與分散式計算的優點，在大型網路中，將網路切割成多個分群，在分群中個別施行原本集中式計算的技巧，如 Biswas & Ye (2006)發展分散式的半正定規劃來求解 SNL，Ding et al. (2009)亦發展分散式的多維度標法 MDS-DC。

2.5 感測器連結性

在任何 WSN 中，最基本可用的定位資訊莫過於感測器間的連結性(Connectivity)資訊，當二個感測器互在通訊範圍中，則稱此二感測器有相連(Connected)，或是相鄰(Adjacent)，在網路中以節線來表示，如此的連結性資訊有助於用來判斷網路的可定位程度，也對定位過程有很大的幫助。

感測器的連結性有兩大性質，首先設空間中有感測器 i 與 j ，因為 WSN 中的節線是無向的，且感測器的通訊半徑相同，其連結的關係是對稱的，因此若 i 在 j 的通訊範圍中，則 j 也必定在 i 的通訊範圍中；而當已知二個感測器間的距離或其間存在著節線，若且唯若此二個感測器間的距離小於或等於給定的通訊範圍。這些性質有助於將 WSN 轉換成單位圓盤圖(Unit Disk Graph)，該理論由 Clark et al. (1990)提出，若以感測器為圓心，用通訊半徑畫成一圓，即形成一個歐幾里得空間中的圓盤(圖 2-8a)，因通訊半徑相同，當一圓包含另一圓的圓心時，其兩圓必定互相包含圓心(圖 2-8b)，互相包含圓心的關係即對稱到 WSN 中兩節點的相連，而由多個節點相連而成的網路圖即為單位圓盤圖(圖 2-8c)，如此一來，可用單位圓盤圖的相關定理來分析 WSN。

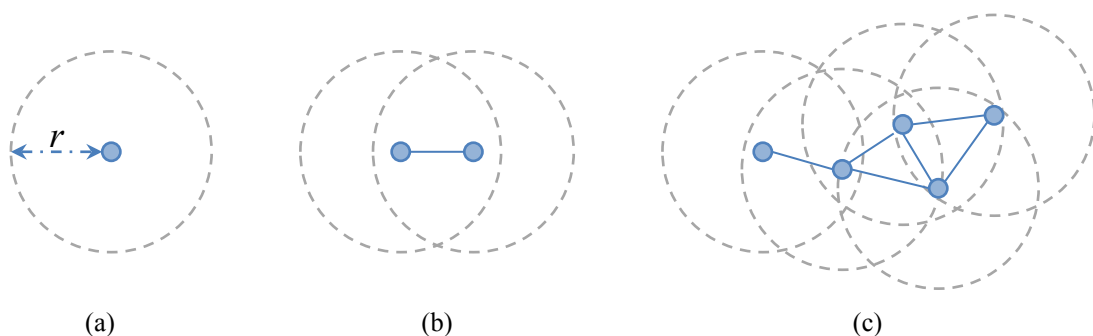


圖 2-8：單位圓盤圖理論

以上連結性的性質隱含著節點間「應不應該」相鄰的規則，有助用來判斷感測器的定位，以圖 2-9 為例，設二維空間中有一個流點(節點 1)與三個錨點(節點 2,3,4)，已知節線只有(1,2)與(1,3)，就是流點僅與錨點 2 及 3 相鄰，錨點間雖然存在節線，但不顯示在此圖中。透過此兩錨點可知流點的位置可能在 $1a$ 或 $1b$ ，若流點的位置在 $1b$ ，則因為 $1b$ 在節點 4 的通訊範圍中，必定得存在節線($1b,4$)，但這違背原本感測器連結的合理性，因此流點的位置必定不在 $1b$ 。如此運用感測器連結性所構成的邏輯規則將應用在之後的啟發式演算法中，包含多邊定位法與區域調整機制。

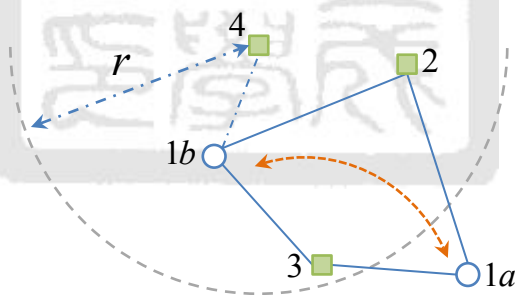


圖 2-9：感測器連結的合理性

2.6 剛性理論

求解 SNL 前往往往得先考慮一個問題：「究竟給定的問題網路可否定位？」若說一問題網路是可定位的，表示該網路有唯一的定位結果，即各節點有唯一且不模稜兩可的定位。文獻顯示網路可否定位與網路的剛性(Rigidity)息息相關，Laman (1970)提出網路的剛性，並說明其與圖實現問題的密切關係，而後 Connelly (1991, 2005)提出整

體剛性(Global Rigidity)的概念，作為圖實現的必要條件，但直至 Eren et al. (2004)與 Aspnes et al. (2004)才著重於以剛性理論來探討 SNL 是否有唯一的定位解，確認了 D 維空間中可定位的 WSN 必須含有至少 $D+1$ 個錨點，且具備整體剛性。

在介紹剛性理論前，首先定義網路骨架(Framework)為網路在歐幾里得空間中的體現，也就是一種符合距離限制式的節點位置組合，以網路骨架 (G, F^p) 表示網路圖 $G=(V, E)$ 在空間中的一種映對 $F^p: V \rightarrow \mathbb{R}^D$ ， F_i^p 為節點 $i \in V$ 在網路骨架 (G, F^p) 中的位置。則整體剛性為任二種網路骨架 F^p 與 F^q 皆符合

$$\|F_i^p - F_j^p\| = \|F_i^q - F_j^q\|, \quad \forall i, j \in V \quad (2.23)$$

表示二種網路骨架間對應的節線距離皆相同，且即使節點間不存在的節線 $(i, j) \notin E$ 也符合該限制式，即是全等(Congruence)的關係，意謂著只有唯一一種網路骨架。而整體剛性可以放鬆成剛性(Rigidity)，若存在一個足夠小的非負常數 ε ，使得滿足

$$\begin{aligned} \|F_i^p - F_i^q\| &< \varepsilon, & \forall i \in V \\ \|F_i^p - F_j^p\| &= \|F_i^q - F_j^q\|, & \forall (i, j) \in E \end{aligned} \quad (2.24)$$

表示兩種網路骨架間對應的節線距離皆相同，但對應的節點可能有些微偏移，如此的關係稱為同等(Equivalence)，如圖 2-10c 因為有多種可行的網路骨架，故不符合整體剛性，但因各種網路骨架間對應的節點距離都相符，故其具備剛性，而若節點 3 與 4 之間存在節線，則可具備整體剛性。

整體剛性的檢驗並不容易，因為當網路圖有著如 WSN 般稀疏的距離資訊，即無法完全滿足式(2.23)，為達到整體剛性的檢驗，Hendrickson (1992)提出網路圖的 $D+1$ 連通性(Vertex $(D+1)$ -Connectivity)與冗剛性(Redundant Rigidity)是整體剛性的必要條件，其中 $D+1$ 連通性表示網路中任意去除 D 個節點不會破壞網路的連通；而冗剛性表示即使在網路中去除任一條節線，該網路仍能保持剛性。Berg & Jordán (2003)與 Jackson & Jordán (2005)證明在二維空間中 Hendrickson (1992)所提的條件是整體剛性的充分條件，然而對於高維度(三維以上)空間中的網路，Connelly (1991)已發現一些

二分圖(Bipartite Graph)雖然具有 $D+1$ 連通性與冗剛性，卻不具備整體剛性，因此在高維度空間中，網路的剛性尚無明確的圖論特質，然而 Connelly (2005)認為有可能找到任何維度空間中使得網路具備整體剛性的其它充分條件，這仍是個開放問題。

若從反面的角度來探討唯一定位的條件，即是要消弭違反整體剛性的變形，一種變形(Deformation)即是一種網路骨架，Hendrickson (1992)將二維空間中網路圖的變形區分為連續型變形與非連續型變形；而二維空間中的變形雖然可以延伸到三維空間中，但目前尚無完備的探討。連續型變形又稱為彈性的(Flexible)網路骨架，網路中部分節點以其它節點為軸心而旋轉(Pivot)，如圖 2-10a 中的節點 1 與 2 以及圖 2-10b 的節點 5 可以連續性地旋轉，若增加節線連到可轉動的節點有助於使其達到剛性。剛性是整體剛性的必要條件之一，但具備剛性的網路圖仍可能有非連續型的變形，其中包含翻轉(Flip)與扭曲(Flex)，翻轉是以部分節點為對稱軸，使其它節點沿著此軸鏡射，例如圖 2-10c 中節點 4 以節點 1 與 2 為軸形成了鏡射對稱，但若網路圖具備三連通性(3-connectivity)，則可以消除翻轉的變形；而扭曲則是部分節點可連動但非連續的旋轉，如圖 2-10d 中節點 1 與 2 分別沿著節點 4 與 5 非連續的旋轉，但扭曲後的網路圖仍符合相同的距離限制式，若要避免扭曲的變形，則網路圖必須具備冗剛性。

變形	變形的圖例	解決方法	解決的圖例
連續	(a)	網路圖必須具備剛性	
	(b)		
翻轉	(c)	網路圖必須具備三連通性	
扭曲	(d)	網路圖必須具備冗剛性	

圖 2-10：二維空間中網路圖的變形與解決方法(Yang, 2010)

另外，目前已知有兩種衍伸的網路結構具備唯一的定位，分別為 $D+1$ 邊定位圖 (($D+1$)-lateration Graph)與輪狀圖(Wheel Graph)。Eren et al. (2004)提出在 D 維空間中的 $D+1$ 邊定位圖具備整體剛性，Zhu et al. (2010)亦證明之，對於 D 維空間中含有至少 $D+1$ 個節點的網路 $G=(V,E)$ ，若存在一組節點的排列方式 $\{\pi(1),\pi(2),...,\pi(|V|)\}$ ，其子網路 $\{\pi(1),\pi(2),...,\pi(D+1)\}$ 構成一個完全圖(Complete Graph)，完全圖中各節點間皆有節線相連，必具備整體剛性，而後續排列的其它節點 $\pi(i):i \geq D+2$ 連接到先前排列 $\{\pi(1),\pi(2),...,\pi(i-1)\}$ 中至少 $D+1$ 個節點，則稱此網路為 $D+1$ 邊定位圖。

以二維空間為例， $D+1$ 邊定位圖即為三邊定位圖，如圖 2-11 所示，平面上有三個節點形成一個完全圖(圖 2-11a)，其必定具備整體剛性，而後加上節點 4，其相連至本來已存在的三個節點(圖 2-11b)，則加入新節點後的網路仍保持整體剛性，其後陸續加入新節點 5 與 6(圖 2-11c,d)，只要能相連到本來平面上至少三個已存在的節點，則可以一直保持整體剛性。對於 SNL 問題，如果給定的 WSN 含有至少 $D+1$ 個錨點，且所有節點形成一個 $D+1$ 邊定位圖，則可保證此 WSN 有唯一的定位。

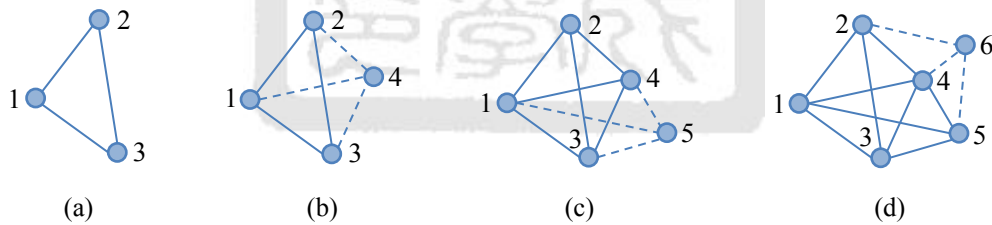


圖 2-11： $D+1$ 邊定位圖

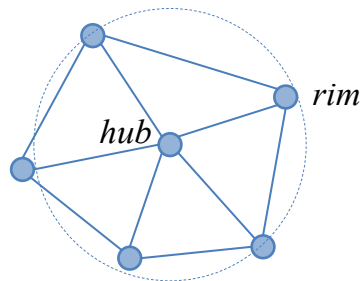


圖 2-12：輪狀圖 W_6

另一種具備整體剛性的網路結構是二維空間中的輪狀圖(Wheel Graph)，輪狀圖包含一個軸節點(Hub)與數個輪節點(Rim)，其所有輪節點都與軸節點相連，而輪節點也與其鄰近的輪節點相連成環狀，形成如輪軸般的結構(圖 2-12)，若此輪狀圖共含有 n 個節點，則稱此輪狀圖為 W_n 。Berg & Jordán (2003)觀察到輪狀圖具有冗剛性，而 Eren et al. (2004)亦證明其符合三連通性，因此輪狀圖具備整體剛性。若一個 WSN 具有輪狀圖的結構含有至少三個錨點，Yang (2010)證明無論錨點是位於軸節點或輪節點皆可唯一地定位整個輪狀圖。

以上所述，網路的 $D+1$ 連通性與冗剛性能決定二維空間中的網路是否具備整體剛性，且由其所衍伸的 $D+1$ 邊定位圖與輪狀圖也有助於判別網路可否定位，但剛性理論仍存在著研究的空間，三維空間中要達到整體剛性的充分條件還未知，而二維空間中整體剛性的條件可能過於嚴格，以圖 2-9 為例，雖然節點 1 的位置有翻轉的變形，但以感測器的連結性仍可以確定其有唯一的定位，因此造成整體剛性的條件依然是個謎題，甚至整體剛性亦只是網路唯一定位的充分條件之一(Mao et al., 2007)，唯一定位的圖論特性仍是個開放問題。

第三章

可定位程度檢驗

可定位程度(Localizability)的檢驗是定位的基礎，對 WSN 有諸多好處，首先，由於定位的過程會消耗一定程度的資源，定位不可定位的網路只是徒勞無功，因此事先檢驗 WSN 的可定位程度能避免無意義的計算與電力消耗；其次，可定位程度的檢驗有益於 WSN 的控制與管理，例如佈署中的網路含有不可定位的流點時，可以依據可定位程度來調整該網路，像是增加通訊半徑或增設感測器來提高可定位程度；再者，對於佈署完成的 WSN，能透過可定位程度來檢視有效的定位結果，比如說部分流點已被檢驗為不可定位時，就不應該採信其定位結果，由此可見，可定位程度也能用來評估定位演算法的優劣，因為對於已被檢驗為可定位的流點，良好的演算法應該要能準確定位這些流點。

基本的可定位程度檢驗即是要判斷網路是否可定位，依 2.6 節所述，可定位的網路必須具備整體剛性，而針對二維空間中的網路，可藉由三連通性與冗剛性來判斷，此二種性質分別有成熟的檢驗方法，Hendrickson (1992)與 Berg & Jordán (2003)皆有詳細的探討。然而，除了檢驗給定的網路可否定位，可定位程度的檢驗應該還要辨別哪些節點可定位以及哪些不可定位，亦即檢驗的結果能顯示可定位節點的個數上限或比例。在這方面，Yang (2010)探討了個別節點的可定位程度，並發展 WHEEL 演算法以具備整體剛性的輪狀圖來檢驗網路的可定位程度。

WHEEL 演算法、三邊定位法與雙邊定位法皆能用來辨識可定位的節點，後兩者甚至能延伸應用於三維空間中，因此本研究將其一般化為 $D+1$ 邊定位法與 D 邊定位法，以下分別說明將這三種方法應用於可定位程度的檢驗，並融合運用於新的檢驗演算法 Loc_Idx，而最後藉由可定位程度的檢驗將流點分為三類，後續用於鑑別定位演算法的優劣。

3.1 $D+1$ 邊定位延伸

$D+1$ 邊定位延伸($(D+1)$ -lateration Extension)即是應用 $D+1$ 邊定位圖來判斷網路中哪些節點是可定位的，在二維空間中稱為三邊定位延伸，而三維空間中則是四邊定位延伸。三邊定位延伸以圖 2-11 為例，設節點 1,2,3 為錨點，由於節點 4 相鄰三個錨點，所以必定為可定位，將其視為錨點來輔助其它節點定位，以此類推，節點 5,6 皆可判斷為可定位，如此擴展到的可定位節點即稱為三邊定位延伸。 $D+1$ 邊定位延伸可以由多組錨點從多個方向開始擴展，使得擴展的速度更快或擴展的範圍更廣，其演算法可表示為：

演算法 2： $(D+1)$ -LATERATION EXTENSION

```
procedure  $(D+1)$ -laterationExt
   $S \leftarrow V_s, A \leftarrow V_a$ 
  while  $S \neq \emptyset$  do
    foreach node  $s \in S$  do
      if  $|\{(s, a) : a \in A\}| \geq D+1$  then
         $S \leftarrow S \setminus s, A \leftarrow A \cup s$ 
        apply multilateration on  $s$  (optional)
      until no node in  $S$  is  $(D+1)$ -lateration localizable
  end procedure
```

演算法 2 以各流點相鄰的錨點數來判斷該流點可否定位，然後將可定位的流點移到錨點的集合中，如此循序檢查所有流點，逐漸擴展可定位節點的範圍，直到剩下的流點最多只相鄰 D 個錨點。演算法 2 主要僅辨識網路中可定位的流點，而非實際定位出這些流點，然而當網路的距離資訊帶有測量的誤差時，為了檢驗真實的可定位程度， $D+1$ 邊定位延伸必須實際地定位流點，此即為演算法 2 中的選擇性步驟，定位的方法則是多邊定位法，如此 $D+1$ 邊定位延伸演算法就等同於迭代多邊定位法，除了用來檢驗可定位程度，亦可作為定位演算法的前置處理。

3.2 D 邊定位延伸

D 邊定位延伸類似於 $D+1$ 邊定位延伸，唯獨判斷流點可否定位的依據是 D 邊定位法(D -lateration)， D 邊定位法較少為人討論，可應用於二維與三維空間中，常見的是二維空間中的雙邊定位法，Li et al. (2007)與 Cota-Ruiz et al. (2012)對於雙邊定位法有獨立的討論，其中後者與本研究的雙邊定位法較為相近，而本研究將雙邊定位法擴充至三維空間中，因此稱為 D 邊定位法；而為了不與二維空間中的 $D+1$ 邊定位法混淆，三維空間中的 D 邊定位法仍然統稱為雙邊定位法。

在 D 維空間中，一個流點必須至少相鄰 $D+1$ 個錨點才有唯一的定位，然而當只有相鄰 D 個錨點時，則通常有二種可行的定位，雙邊定位法即是盡可能判斷這二種定位中何者正確。以圖 3-1a 為例，應用感測器連結的一致性，若已知流點 1 與錨點 2,3 相鄰，與錨點 4 不相鄰，因此流點 1 的定位為 $1a$ 或 $1b$ ，但由於 $1b$ 與錨點 4 相鄰，違反連結的合理性，故正確的定位在 $1a$ ；然而當 $1a$ 與 $1b$ 都不相鄰錨點 4 時(圖 3-1b)，就無法判斷何者才是正確的。此外，儘管一個流點相鄰 $D+1$ 個錨點，仍然可以只用 D 個錨點以 D 邊定位法找出二種可行的定位，再以剩餘的錨點判斷何者正確，因此 $D+1$ 邊定位法可視為 D 邊定位法的一種特例。

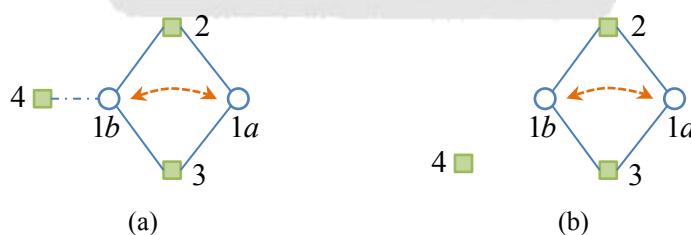


圖 3-1： D 邊定位法

嚴格來說， D 邊定位法不能保證找到唯一的定位，但其將定位問題縮小到簡單的二選一，若在網路中有 n 個流點分別都相鄰二個錨點，則各流點皆有二種可行的定位，導致所有的可行定位形成 2^n 種組合。當 n 夠小時，利用這些流點間的距離資訊可能尚有機會找出正確的定位，稱為多重 D 邊定位法，以圖 3-2 為例，若已知流點 1 與錨

點 3,4 相鄰，其可行的定位為 $1a$ 與 $1b$ ，而流點 2 相鄰錨點 5,6，可行的定位有 $2a$ 與 $2b$ ，因此所有可行定位的組合為 $(1a,2a)$ 、 $(1a,2b)$ 、 $(1b,2a)$ 、 $(1b,2b)$ 等 4 種，如果又已知流點 1,2 相鄰，則可以判斷 $(1a,2b)$ 是唯一正確的定位組合，如此二個流點一併進行 D 邊定位法即稱為雙重 D 邊定位法，將應用於在後續的定位演算法中。

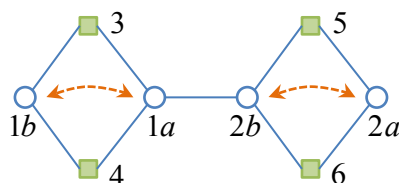


圖 3-2：雙重 D 邊定位法

D 邊定位延伸必須實際定位流點，無法直接以流點相鄰的錨點數或其它連結關係來判斷可否定位，其定位的基礎在於找出流點的二種可行定位，但這並不如多邊定位法直觀，必須以幾何的方法來求解，以下分別說明如何在二維與三維空間中找出流點的二種可行定位。

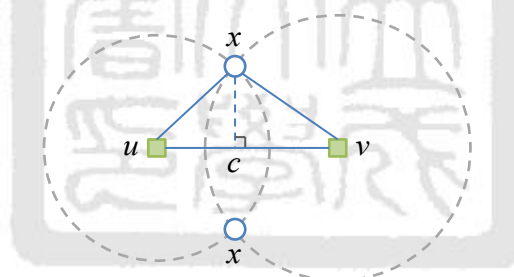


圖 3-3：二維空間中的雙邊定位法

3.2.1 二維空間的雙邊定位法

二維空間中，設已知錨點 u 與 v 的座標為 \mathbf{u} 與 \mathbf{v} ，其與流點 x 的距離為 d_{ux} 與 d_{vx} ，而錨點間的距離為 d_{uv} ，據此求解流點的座標 \mathbf{x} ，如果以 u 與 v 為圓心並以 d_{ux} 與 d_{vx} 為半徑，所繪成的圓交於兩點，即為流點的可行定位，如圖 3-3 所示。令 c 為 x 在線段 \overline{uv} 上的投影點，首先依畢氏定理可得到

$$\begin{aligned}
d_{cx}^2 &= d_{ux}^2 - d_{uc}^2 \\
&= d_{vx}^2 - d_{vc}^2 \\
&= d_{vx}^2 - (d_{uv} - d_{uc})^2 \\
d_{uc} &= \frac{d_{ux}^2 - d_{vx}^2 + d_{uv}^2}{2d_{uv}}
\end{aligned} \tag{3.1}$$

則由 u 至 c 的向量即為

$$\overrightarrow{uc} = \frac{d_{uc}}{d_{uv}}(\mathbf{v} - \mathbf{u}) \tag{3.2}$$

設 $H = (\mathbf{v} - \mathbf{u}) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ ，是與 \overrightarrow{uv} 同長度的法向量，則由 c 至 x 的向量為

$$\overrightarrow{cx} = \frac{d_{cx}}{d_{uv}} H \tag{3.3}$$

利用式(3.2)與式(3.3)即可求得流點的二個可行定位為

$$\mathbf{x} = \mathbf{u} + \overrightarrow{uc} \pm \overrightarrow{cx} \tag{3.4}$$

3.2.2 三維空間的雙邊定位法

由以上二維空間的範例擴充到三維空間，流點 x 多了一個相鄰的錨點 w ，其座標為 \mathbf{w} ，同樣已知錨點的座標以及各節點間的距離，求解流點的定位 \mathbf{x} ，如圖 3-4 所示，若分別以各錨點為球心，錨點到流點的距離為半徑，所繪成的三個球面交於兩點，即為流點可行的定位，其中三個錨點 u 、 v 與 w 構成一個平面，令 c 為 x 在該平面上的投影點，而 m 為 c 在線段 uv 上的投影點，則由於 $\overline{cm} \perp \overline{uv}$ 且 $\overline{cx} \perp \overline{cm}$ ， x 對線段 uv 的投影點亦為 m 。

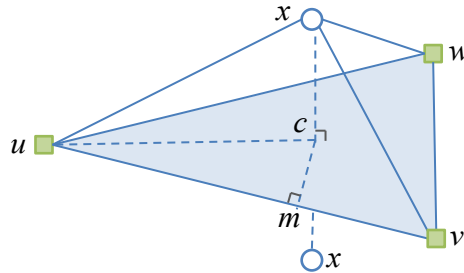


圖 3-4：三維空間中的雙邊定位法

應用以上對三個錨點形成的平面以及 x, c, m 三點間的關係，設以下聯立方程式來求解由 u 至 c 的向量

$$\begin{cases} (\overrightarrow{uv} \times \overrightarrow{uw}) \cdot \overrightarrow{uc} = 0 & (a) \\ \overrightarrow{uv} \cdot \overrightarrow{uc} = \overrightarrow{uv} \cdot \overrightarrow{ux} & (b) \\ \overrightarrow{uw} \cdot \overrightarrow{uc} = \overrightarrow{uw} \cdot \overrightarrow{ux} & (c) \end{cases} \quad (3.5)$$

其中 $\overrightarrow{uv} \times \overrightarrow{uw}$ 為三個錨點所在平面的法向量，式(3.5a)表示二個垂直向量相乘，故其內積為 0；而式(3.5b)中，等式左右邊的內積可分別展開並化簡為

$$\begin{aligned} d_{uv} d_{uc} \cos(\angle vuc) &= d_{uv} d_{ux} \cos(\angle vux) \\ d_{uv} d_{um} &= d_{uv} d_{um} \end{aligned} \quad (3.6)$$

化簡後等式的左右邊相同，因此式(3.5b)成立，以此類推，式(3.5c)亦成立。但由於 \overrightarrow{ux} 是未知的，故利用餘弦定理將式(3.5b)取代為

$$\begin{aligned} \overrightarrow{uv} \cdot \overrightarrow{uc} &= d_{uv} d_{ux} \cos(\angle vux) \\ &= \frac{d_{uv}^2 + d_{ux}^2 - d_{vx}^2}{2} \end{aligned} \quad (3.7)$$

在式(3.5c)經過類似式(3.7)的轉換後，可將式(3.5)整理為

$$\begin{cases} (\overrightarrow{uv} \times \overrightarrow{uw}) \cdot \overrightarrow{uc} = 0 \\ \overrightarrow{uv} \cdot \overrightarrow{uc} = (d_{uv}^2 + d_{ux}^2 - d_{vx}^2) / 2 \\ \overrightarrow{uw} \cdot \overrightarrow{uc} = (d_{uw}^2 + d_{ux}^2 - d_{wx}^2) / 2 \end{cases} \quad (3.8)$$

求解聯立方程式(3.8)可得 \overrightarrow{uc} ，而由 c 至 x 的向量則為

$$\begin{aligned} \overrightarrow{cx} &= \frac{\|\overrightarrow{cx}\|}{\|\overrightarrow{uv} \times \overrightarrow{uw}\|} (\overrightarrow{uv} \times \overrightarrow{uw}) \\ &= \frac{\sqrt{d_{ux}^2 - d_{uc}^2}}{\|\overrightarrow{uv} \times \overrightarrow{uw}\|} (\overrightarrow{uv} \times \overrightarrow{uw}) \end{aligned} \quad (3.9)$$

利用式(3.8)與式(3.9)的結果，帶入相同的式(3.4)即可求得 x 的二種可行定位。

3.3 輪狀圖延伸

依 2.6 節所述，二維空間中的輪狀圖若含有至少三個錨點，則輪狀圖中所有的節點皆可判斷為可定位，而輪狀圖延伸即是將輪狀圖應用於類似 $D+1$ 邊定位延伸的步驟中，以圖 3-5 為例，由三個錨點起始，則可將包含此三個錨點的輪狀圖(以粗線表示)判斷為可定位，因此圖 3-5a 中的黃色節點為可定位，然後將可定位的流點視為錨點，再繼續尋找可包含至少三個錨點的輪狀圖，圖 3-5b 中另一環狀圖涵蓋了四個可定位節點，因此又新增三個可定位的流點，以此類推地反覆交疊輪狀圖，最終擴展到所有節點(圖 3-5c)，如此以輪狀圖擴展到的可定位節點即稱為輪狀圖延伸。

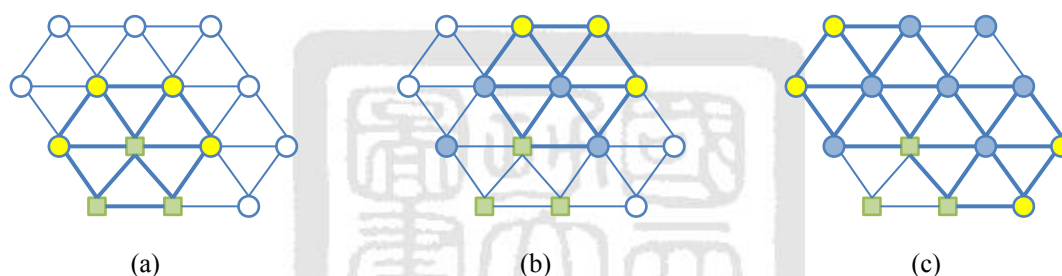


圖 3-5：輪狀圖延伸

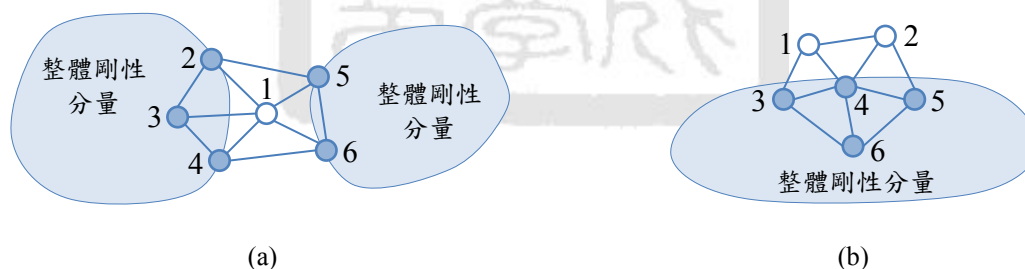


圖 3-6：輪狀圖延伸應用於邊緣節點

輪狀圖延伸往往能比三邊定位延伸辨識更多可定位的流點，以圖 3-5 為例，雖然網路中含有三個錨點，但由於各流點最多僅相鄰二個錨點，以至於如果將三邊定位延伸應於此網路，將無法辨識出任何可定位的流點。此外，Yang (2010)觀察到輪狀圖延伸有助於辨識分量(Component)邊緣的可定位流點，以圖 3-6 為例，圖 3-6a 中有二個分別具備整體剛性的分量，假設有三個錨點位於左方的分量中，如果使用三邊定位延

伸，最多只能擴展到節點 1,2,3,4，即無法繼續擴展到右方的分量，使得三邊定位延伸誤判右方分量為不可定位。然而如果使用輪狀圖延伸，就可以橋接這二個分量，並擴展到右方分量。同樣地，圖 3-6b 中若使用三邊定位延伸會誤判節點 1,2 為不可定位，但輪狀圖延伸能將其辨識為可定位。

輪狀圖延伸的基礎即是辨識網路中的輪狀圖結構，設 $N(v)$ 為所有與節點 v 相鄰的節點，則尋找以節點 v 為軸節點的輪狀圖就相當於在 $N(v)$ 中找出環(Cycle)的結構，由於環中任兩點之間都存在著二條不重複的路徑，因此環具備雙連通性(2-connectivity)，也就是去除環中任一個節點不影響到環的連通，如此 $N(v)$ 中的環能以節點 v 為軸節點形成一個輪狀圖，使得該環狀圖具備三連通性。Yang (2010)證明了在 $N(v)$ 中如果有具備雙連通性的分量，其分量必可形成環，因此尋找環又相當於辨識雙連通性的分量，而尋找具備雙連通性的分量可以藉由深度優先搜尋法(Depth First Search)達成。 $N(v)$ 中最簡單的環是由三個節點構成，與節點 v 形成 2.6 節所述的 W_4 輪狀圖，如果其中含有三個錨點，其實就等同於三邊定位法。由此可知，三邊定位延伸是輪狀圖延伸的一種特例，因此三邊定位延伸擴展到的可定位節點必定包含在輪狀圖延伸中。

輪狀圖延伸僅能用於判斷流點可否定位，無法以輪狀圖的概念實際求解流點的位置，但已經判斷為可定位的輪狀圖仍然可以透過三邊定位法與雙邊定位法達成實際定位。以圖 3-7 為例，流點 1 相鄰三個錨點，因此可以用三邊定位法來定位；而流點 2,4 能透過雙邊定位法來定位，剩下的流點 3 則能夠透過已定位的節點 2,4 以及軸心的錨點用三邊定位法來定位；或者利用多重雙邊定位法，由流點 2,3,4 的可行定位形成 2^3 種定位組合，再以節點間的距離資訊來判斷正確的定位組合。

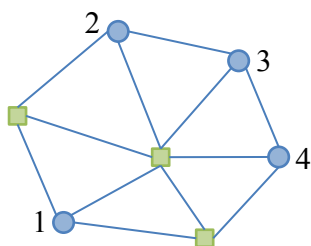


圖 3-7：輪狀圖的實際定位

3.4 Loc_Idx 演算法

$D+1$ 邊定位延伸、 D 邊定位延伸與輪狀圖延伸皆各有優缺點，其適用的空間維度、辨識可定位節點的方法與計算的複雜度都有些許不同，如表 3-1 所示， $D+1$ 邊定位延伸與 D 邊定位延伸能應用在二維與三維空間中，計算簡單並且能實際定位；而輪狀圖延伸雖然只能應用於二維空間，計算較繁雜且不能實際定位，但一次能辨識較多可定位節點；而在計算時間的方面，雖然三種方法的複雜度不同，但都能在多項式時間內快速檢驗可定位程度，其中 $D+1$ 邊定位延伸與 D 邊定位延伸在一次迭代得花 $O(|E|)$ 的時間檢查所有可定位的流點，至多得迭代 $|V|$ 次，故複雜度為 $O(|V||E|)$ ；而輪狀圖延伸則對於 $|V|$ 個節點個別以複雜度為 $O(|V|+|E|)$ 的深度優先演算法辨識其周圍具備雙連通性的分量，再以 $|V|$ 時間檢驗該分量含的錨點個數，同樣最多得迭代 $|V|$ 次（一次至少產生一個新的錨點），因此複雜度為 $O(|V|^2(|V|+|E|))$ 。

此外，此三種方法能辨識的可定位節點亦不同。以二維空間來說，三種方法涵蓋的可定位節點關係如圖 3-8 所示。由於三邊定位延伸分別是雙邊定位延伸與輪狀圖延伸的特例，因此三邊定位延伸同時包含在雙邊定位延伸與輪狀圖延伸中；而由於輪狀圖延伸能夠以三邊定位法與多重雙邊定位法達成實際定位，因此輪狀圖延伸可以包含於雙邊定位延伸中，但由於多重雙邊定位法的計算複雜度較高，在實作中不使用，僅使用雙邊定位法，因此雙邊定位延伸與輪狀圖延伸互不包含。

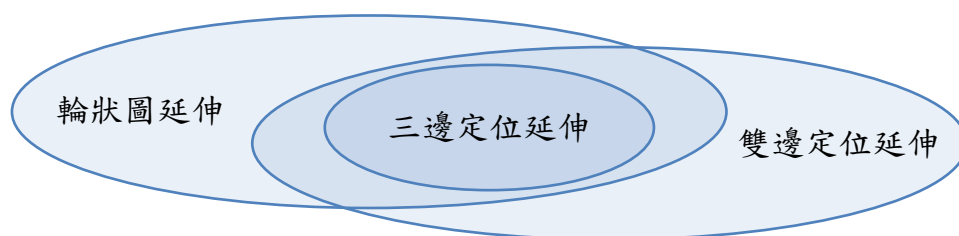


圖 3-8：三邊定位延伸、雙邊定位延伸與輪狀圖延伸的關係

表 3-1：D+1 邊定位延伸、D 邊定位延伸與輪狀圖延伸的比較

	D+1 邊定位延伸	D 邊定位延伸	輪狀圖延伸
適用空間	二維與三維空間	二維與三維空間	二維空間
辨識可定位節點	運用圖論性質辨識或實際定位以辨識	需實際定位才能辨識	運用圖論性質辨識
實際定位	使用多邊定位法或 D+1 邊定位法	使用 D 邊定位法，但不一定能成功定位	不可直接定位，需仰賴其它方法定位
擴展步伐	一次擴展一個節點	一次擴展一個節點	一次擴展一個環狀圖
計算複雜度	$O(V E)$	$O(V E)$	$O(V ^3 + V ^2 E)$

由於 D+1 邊定位延伸、D 邊定位延伸與輪狀圖延伸各有專長，且由圖 3-8 可知綜合使用三種方法才能辨識最多的可定位流點，因此本研究混合應用此三種方法，發展新的可定位程度檢驗演算法 Loc_Idx：

演算法 3：Localizability_Index

procedure Loc_Idx

- step1: Apply (D+1)-lateration on sensors incident to D+1 anchors or more
- step2: Apply D-lateration on sensors incident to D anchors
- step3: Update sensors localized by step1 and step2 as anchors
- step4: Repeat step1~step3 until no more sensors are updated
- step5: Find wheels containing D+1 anchors
- step6: Update sensors within wheels found by step5 as anchors
- step7: Repeat step5~step6 until no more sensors are updated

end procedure

為了使用輪狀圖延伸，Loc_Idx 僅適用於二維空間中，其應用三邊定位延伸與雙邊定位延伸反覆地定位所有可定位的流點，並將已定位的流點視為錨點來輔助其它流點定位，直到所剩的流點都無法使用三邊定位延伸或雙邊定位延伸來定位為止，才開始應用輪狀圖延伸，判斷所剩的流點是否包含於任何可定位的輪狀圖中。由於雙邊定位延伸需要實際定位流點才能判斷其是否可定位，因此三邊定位延伸也必須實際定位流點，才能與雙邊定位延伸穿插使用，而輪狀圖延伸不能直接定位流點，因此不能與前二者混合使用。若捨棄雙邊定位延伸，則可穿插使用三邊定位延伸與輪狀圖延伸，

且不需要實際定位；而如果要將 Loc_Idx 應於三維空間中，則必須捨棄輪狀圖延伸，並使用三維空間中雙邊定位延伸與四邊定位延伸。

Loc_Idx 的步驟五是找出網路中的輪狀圖結構，但實際上尋找輪狀圖的動作相當繁瑣，而且只有少數的輪狀圖有助於擴展輪狀圖延伸，因此為了省卻不必要的計算，首先找出周圍含有流點的錨點，再分別尋找以這些錨點為中心的輪狀圖，並判斷這些輪狀圖是否含有三個錨點，若有的話，則該輪狀圖中的所有節點皆能判斷為可定位。

Loc_Idx 演算法輸出的結果即是網路中可定位的節點，也就是輪狀圖延伸、三邊定位延伸與雙邊定位延伸的聯集，扣除給定的錨點後即是可定位的流點，則該網路的可定位程度可定義為

$$localizability(G) = \frac{\text{number of localizable sensors}}{\text{number of all sensors}} \quad (3.10)$$

以一個含有 100 個流點與 10 個錨點的 WSN 為例，其分布於平面座標 $[0,1] \times [0,1]$ 之中，圖 3-9 分別顯示(a)三邊定位延伸、(b)三邊定位延伸與輪狀圖延伸、(c)三邊定位延伸與雙邊定位延伸以及(d)Loc_Idx 演算法應用於該網路的結果，圖中綠色方塊表示錨點，而紅色節點表示判斷為可定位的流點，四種方法檢驗結果的可定位程度分別為 14%、48%、73%與 73%。圖 3-9a,b 顯示輪狀圖延伸的確包含三邊定位延伸，並能比三邊定位延伸辨識更多可定位的流點；而圖 3-9d 顯示 Loc_Idx 綜合了三邊定位延伸、雙邊定位延伸與輪狀圖延伸，擁有最佳的判斷能力，而在此例中，由於三邊定位延伸與雙邊定位延伸已經涵蓋了所有輪狀圖延伸的節點，因此方法(c)(d)判斷的結果相同，如此也顯示了輪狀圖延伸的實際定位可由三邊定位延伸與雙邊定位延伸達成。

對於已佈署完成的 WSN，可定位程度的檢驗有助於用來預測實際定位該網路時的成效，以圖 3-9d 為例，Loc_Idx 辨識網路中只有 73%的流點可定位，因此如果對該網路進行定位可能失敗；舉例來說，該網路左上方的一群節點已被判斷為不可定位，因此在該網路的實際定位結果中，其左上方這群流點的定位結果應不可採信。此外，對於佈署中的 WSN，可定位程度的檢驗有助於調校該網路，以一個含有 1000 個流點

與 100 個錨點的 WSN 為例，圖 3-10 顯示該網路在設定不同通訊半徑時，可定位程度隨之改變的情況。由折線的趨勢可知 $r \geq 0.07$ 時，該網路的可定位程度才可達到 100%，因此為了達到該網路的最佳服務能力，可將通訊半徑設為 0.07 以上，或是當通訊半徑受限於 0.07 以下時，可增設感測器到網路中以提高可定位程度。

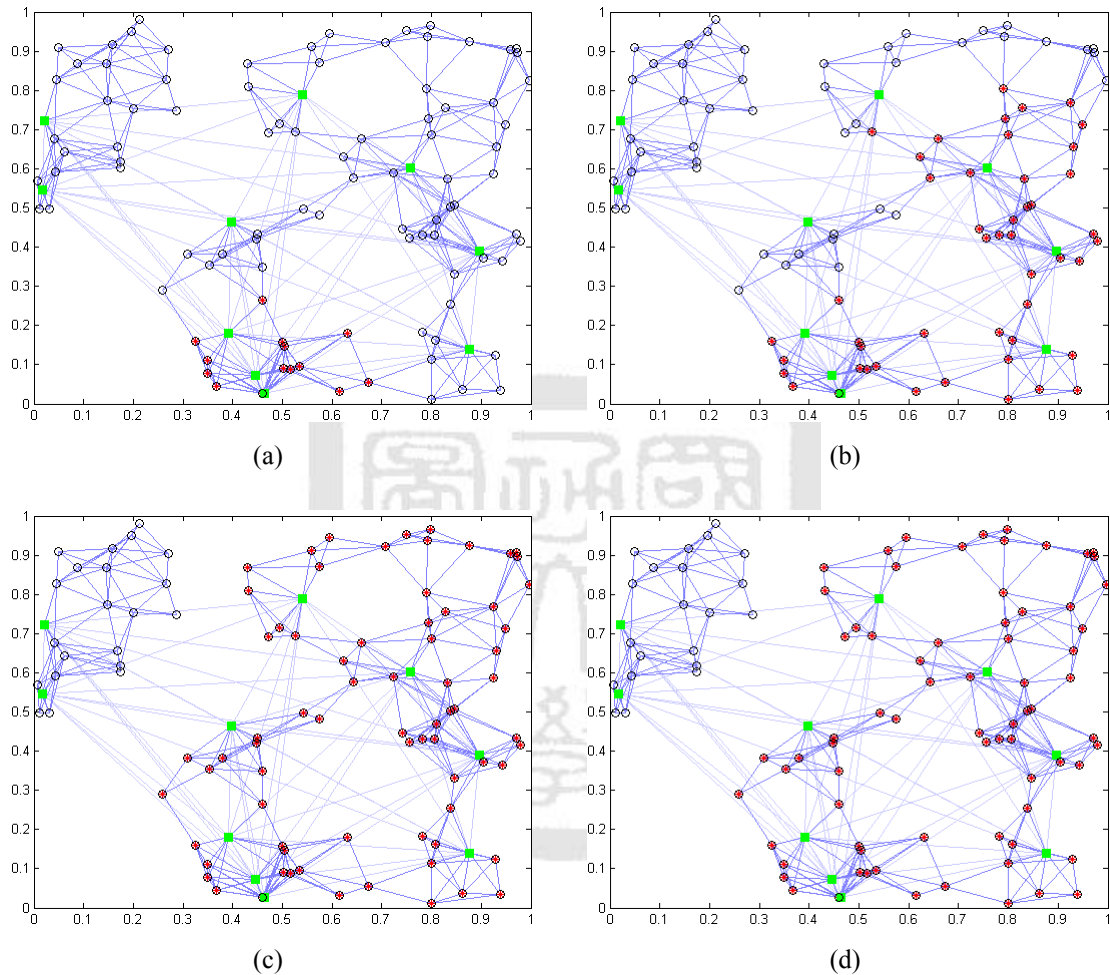


圖 3-9：Loc_Idx 實例

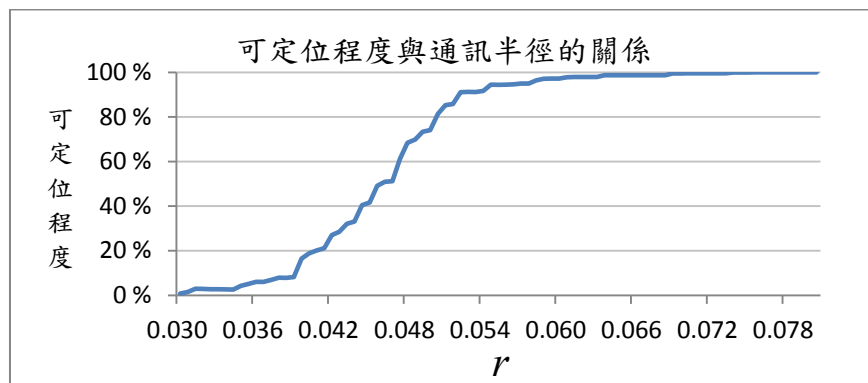


圖 3-10：可定位程度與通訊半徑的關係

3.5 流點的分類

文獻中比較定位演算法的成效時，大多以整體網路的定位誤差(如：RMSD)來比較，然而此類誤差的計算方式可能因極大的個別誤差而影響整體定位表現，因此可能無法完整呈現演算法的優劣。舉例來說，假設我們分別以二種演算法定位一個含有 100 個流點的 WSN，甲演算法的定位結果中只有一個流點有誤差，其單獨 RMSD 為 1，而其它流點都是零誤差；乙演算法的定位結果則是個別流點的 RMSD 皆為 0.1。若依式(2.6)計算，則此二種演算法的整體網路 RMSD 同為 0.1，如此看似二種演算法的定位能力相當，但實質上其個別的定位結果卻相差甚多。因此，為了能更深入地檢視定位的結果，本研究依據可定位程度檢驗的結果，將流點分為三類：

第一類：Loc_Idx 判斷為可定位的流點；

第二類：相鄰至少 D 個節點(錨點或流點)、但不包含在第一類中的流點；

第三類：相鄰不足 D 個節點的流點。

第一類與第三類流點都有明確的圖論特性，第一類流點即為 $D+1$ 邊定位延伸、 D 邊定位延伸與輪狀圖延伸的聯集，而且必定符合整體剛性，對於這種已判斷為可定位的流點，良好的定位演算法應該要能準確地定位。反之，第三類流點相鄰的節點太少，容易產生連續型變形(圖 2-10a,b)，因此無論用任何演算法都不可能定位，而且會產生相當大的定位誤差，即使其它流點都有相當好的定位結果，仍會因為少數第三類流點而大幅地提高整體的 RMSD 值，如此可能誤判了演算法的成效。雖然在實際定位中，網路難免存在第三類流點，但在研究中，為了要正確地比較定位演算法的優劣並深入了解造成誤差的原因，定位過程中應該忽略第三類流點。

第二類流點是介於第一類與第三類流點之間的灰色地帶，包含部分可定位的流點或是會發生變形的流點，如 2.6 節所述，以整體剛性作為判斷可否定位的準則可能過於嚴格，因此流點可能被誤判為不可定位；而雖然有些流點相鄰超過 D 個節點，但仍然可能發生翻轉或扭曲的變形(圖 2-10c,d)，甚至會產生連續型變形。雖然理論上第

二類流點是不可定位的，但對於已經佈署完成的 WSN，良好的定位演算法仍應致力於猜測第二類流點的合理定位，並且不應該因為第二類流點的存在，影響到本該準確定位的第一類流點。

在後續第四章與第五章的實驗當中，將在定位前先將第三類流點移除，以免影響其它流點的定位，再以可定位程度的檢驗來辨識第一類與第二類流點，並在定位後，分別計算第一類與第二類流點的 RMSD，依此來檢視定位的成效。



第四章

定位演算法

本研究的定位演算法基於梯度法求解 SNL，並使用若干技巧來加強定位效果，以下首先介紹本演算法與其細節，而後以模擬的網路測試演算法中各種啟發式演算法的定位效果，並與文獻中應用半正定規劃的定位演算法比較定位成效。

4.1 Grad_MSA 演算法

本研究的定位演算法簡稱為 Grad_MSA，其中「MSA」分別代表了多邊定位法、最短路徑擴增人工節線以及區域調整機制，此三者是輔助梯度法定位的啟發式演算法，則整體 Grad_MSA 演算法的流程為：

演算法 4：LOCALIZATION ALGORITHM FOR ANCHOR-BASED WSN

procedure *Grad_MSA*

- step1: Apply *IterativeMultilateration*
- step2: Update sensors localized by step1 as anchors
- step3: Set artificial edges between non-adjacent nodes
- step4: Obtain initial solution \mathbf{x}_i by *GradientDescent*
- step5: Approximate optimal solution by *GradientDescent*
- step6: Adjust solution according to node connectivity
- step7: Repeat step5~step6 until solution is converged

end procedure

Grad_MSA 演算法將流點依其相鄰的錨點數分成較易定位與較不易定位的流點來處理，其中步驟 1 與 2 負責處理前者，較易定位的流點由於相鄰的錨點數足夠，因此使用幾何的多邊定位法求解，而部分已定位的流點能反覆用來協助定位其它流點，形成迭代的多邊定位法，直到剩下的流點都無法以多邊定位法求解時，將這些已定位的流點標記為錨點，而剩餘的流點則以後續的步驟 3 至 7 處理。

對於剩下較不易定位的流點，則以梯度法求解式(2.5)，逐步地逼近到最佳解，尋找最符合距離限制式的位置。由於梯度法的收斂結果容易受到初始解的影響，而決定初始解必須仰賴各流點與錨點間的距離資訊，距離資訊如果太稀疏將難以產生好的初始解，因此本研究首先在步驟 3 中以最短路徑來擴增距離資訊，再以擴增後的距離資訊決定初始解，而後步驟 5 再依此初始解以梯度法求解定位。另一方面，梯度法的收斂仍可能因為不足的距離資訊而陷於區域最佳解，為了跳脫區域最佳解並提供梯度法新的搜尋起點，本研究提出區域調整機制，在步驟 6 依節點之間連結性資訊，將流點調整到合理的位置，再重新以梯度法逼近最佳解。如此反覆地調整與逼近，直到解已經足夠收斂。

Grad_MSA 演算法將流點依定位的容易與否分開來處理，即是對應到 3.5 節中對流點的分類，第一類的流點可以直接使用多邊定位法來求解，而對於第二類的流點，雖然理論上不可定位，但本演算法仍試圖以梯度法與啟發式演算法找出可行的定位。本演算法除了能處理不同可定位程度的 WSN，也能作用於二維空間或三維空間中的網路，以下小節將分別敘述 Grad_MSA 各步驟的作法。

4.1.1 迭代多邊定位法

在步驟 1 所使用的多邊定位法並非如 2.3 節所述的多邊定位法，而是指混合應用 D 邊定位法與 $D+1$ 邊定位法，以二維空間為例，即是混合雙邊定位法與三邊定位法。 D 邊定位法即如 3.2 節所述，當一個流點只有相鄰 D 個錨點時，可以依此計算二個可行的定位，再以周圍的連結性資訊來判斷何者為正確的定位，計算的過程如式(3.1)至(3.9)所示。而 $D+1$ 邊定位法是多邊定位法中最簡單的應用，只要一個流點相鄰至少 $D+1$ 個錨點，就能使用 $D+1$ 邊定位法，因此如果周圍有超過 $D+1$ 個錨點時，則只從中隨機選出 $D+1$ 個錨點來進行定位，而求解的過程如式(2.14)至(2.19)所示，如此以 $D+1$ 邊定位法來取代多邊定位法的作法並不會影響定位結果，因為在距離資訊無測量誤差的情況下，二種方法的定位結果相同；而如果距離資訊含有測量誤差，則二

種方法都不能保證求得較好的定位，但因為 $D+1$ 邊定位法只用了最少需求的錨點，定位速度比多邊定位法來得快。

步驟 1 所使用的迭代多邊定位法可表示為：

演算法 5 : ITERATIVE MULTILATERATION

procedure *IterativeMultilateration*

$S \leftarrow V_s, A \leftarrow V_a$

while $S \neq \emptyset$ **do**

foreach node $s \in S$ **do**

if $|\{(s, a) : a \in A\}| \geq D+1$ **then** apply $(D+1)$ -lateration on s

else if $|\{(s, a) : a \in A\}| = D$ **then** apply D -lateration on s

if s is localized **then** $S \leftarrow S \setminus s, A \leftarrow A \cup s$

 apply *GradientDescent* on $A \setminus V_a$

until no node in S is D -lateration or $(D+1)$ -lateration localizable

end procedure

Grad_MSA 先盡可能對所有流點執行 D 邊定位法或 $D+1$ 邊定位法，並將已定位的流點當成錨點來協助其它流點定位，迭代直到剩下的流點都無法定位為止。然而，如此的迭代過程容易形成誤差的累積與傳播(Propagation)，由於給定的距離資訊通常帶有測量誤差，流點的定位必然存在著定位誤差，而迭代間將已定位的流點作為錨點，誤差即會傳播與累積到其它流點，使得後續定位的流點帶有更大的定位誤差，最終可能造成錯誤的定位；為了解決這問題，本演算法在每回迭代後應用梯度法修正當前已定位的流點，如此能有效地降低累積的誤差。

D 邊定位法與 $D+1$ 邊定位法的優點即是運算簡單又快速，雖然定位精準度可能不及梯度法，但若將這二種方法的定位結果作為梯度法的初始解，即能快速又準確地定位。由於演算法 5 的定位精準度相當高，即使迭代地定位了數千個流點後，其累積的誤差仍保持在可接受的範圍之內，因此 Grad_MSA 演算法在步驟 1 結束後，將這些已定位的流點設為錨點，使後續的步驟不再修改其位置。

4.1.2 新增人工節線

錨點的位置與節點間的距離資訊是定位時唯一的依據，而充足的距離資訊能夠限制定位的可行解空間，然而通常給定的距離資訊是稀疏的，也就是只有部分節點間的距離是已知的，如此使得梯度法的求解不易收斂，或是僅搜尋到不佳的區域最佳解。因此，為了增強梯度法的求解，Grad_MSA 演算法在步驟 3 先以節點間的最短路徑作為人工節線，擴增原本缺乏的距離資訊，而後才以梯度法定位。

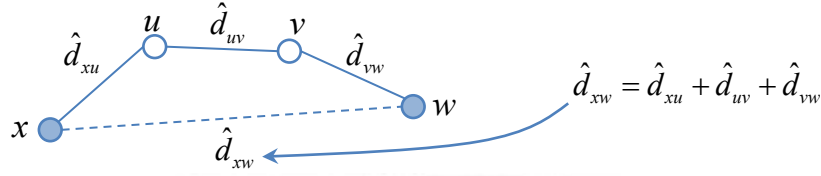


圖 4-1：最短路徑的人工節線

當不相鄰的二個節點間存在著一條或多條路徑(Path)時，則可以在此二個節點間新增一條虛擬的人工節線(Artificial Edge)，將其長度設為所有路徑中的最短距離，以圖 4-1 為例，節點 x 能透過 u 與 v 連結到節點 w ，且此條路徑是節點 x 與 w 間的最短路徑，則能定義人工節線 $\hat{d}_{xw} = \hat{d}_{xu} + \hat{d}_{uv} + \hat{d}_{vw}$ 。設 d_{xw} 為節點 x 與 w 的實際距離，則依人工節線可建立 d_{xw} 的上、下界為

$$r < d_{xw} \leq (\hat{d}_{xu} + \hat{d}_{uv} + \hat{d}_{vw}) \quad (4.1)$$

由此可知，隨著中繼節點數的增加，最短路徑 \hat{d}_{xw} 與實際距離 d_{xw} 的落差可能就越大。

當網路是連通圖(Connected Graph)時，任二個節點間至少存在一條路徑，則能透過新增人工節線補齊任意節點間的距離資訊，而求取所有節點間的最短路徑可藉由任何現存的最短路徑演算法達成，包含著名的 Dijkstra 演算法或 Floyd-Warshall 演算法，本研究以 Johnson 演算法(Johnson, 1977)計算最短路徑，其以 Dijkstra 演算法為基礎，能有效地計算稀疏網路的最短路徑，複雜度為 $O(|V||E|\log|V|)$ ，當節線數足夠少時，計算速度比複雜度為 $O(|V|^3)$ 的 Floyd-Warshall 演算法快。

儘管使用人工節線有助於縮小可行解空間，但節點間的最短路徑長與實際距離可能存在相當大的誤差，而且當中繼節點愈多時，誤差就愈大，雖然能藉由類似 DV-hop 演算法(Niculescu & Nath, 2003)，依中繼節點數縮放人工節線來減輕這個問題，然而這種做法的效果有限，且在節點非均勻分布的網路中，節點間的最短路徑長可能難以比擬實際距離(Liu & Yang, 2011)。以圖 4-2 為例，紅色虛線為二個節點間的實際距離，而黑色折線表示最短路徑，圖 4-2a 的節點為均勻分布，因此最短路徑長接近於實際距離，但圖 4-2b 中的節點非均勻分布，最短路徑即無法比擬實際距離。

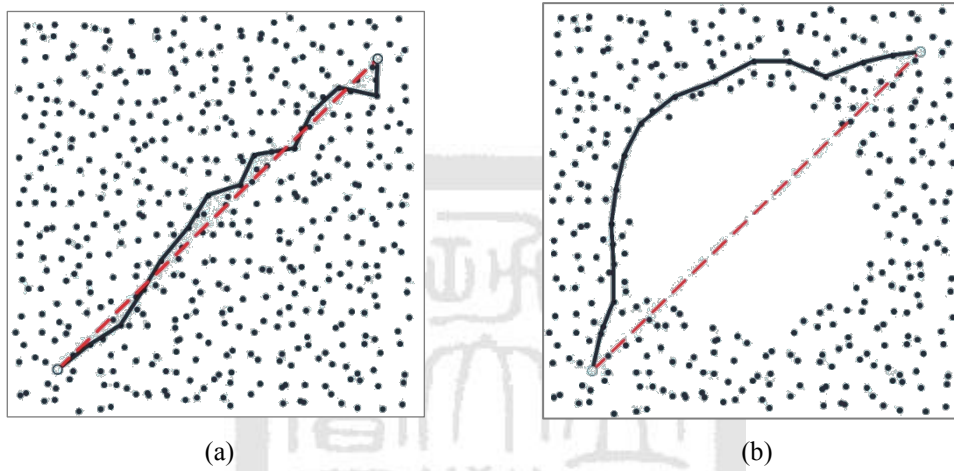


圖 4-2：網路節點的分布影響到人工節線的效果

圖片來源：(Liu & Yang, 2011)

4.1.3 二階段梯度法

由於梯度法的求解效果易受到初始解的影響，因此 Grad_MSA 演算法在步驟 4 先盡可能求取良好的初始解，首先以錨點的位置給予流點一個初估的位置：

$$\mathbf{x}_i = \frac{\sum_{(i,j) \in E_a} \mathbf{a}_j \exp(-\hat{d}_{ij})}{\sum_{(i,j) \in E_a} \exp(-\hat{d}_{ij})} \quad \forall i \in V_s \quad (4.2)$$

對於每一個流點，依它與鄰近各錨點的距離計算折衷的位置，其概念是當該流點愈接近某錨點時，則該流點的座標值理應愈近似於這個錨點的座標，以此形成對周圍錨點的權重，而權重乘以周圍錨點座標的總和即為初估的流點位置。然而，並非所有流點

的周圍都有錨點，式(4.2)不是對所有流點都有效，因此為了初估所有流點的位置，我們先隨機指派座標給流點，再運用人工節線擴增後的距離資訊以梯度法定位，而如此初估的定位即是步驟 5 的初始解。

Grad_MSA 演算法中步驟 4 與 5 都應用梯度法定位，其差別在於步驟 4 使用了含人工節線的距離資訊，因此較能限制定位的可行空間，但由於人工節線畢竟不能比擬實際距離，步驟 4 的定位必定不精準，因此步驟 5 接著以不含人工節線的距離資訊定位，目的是基於步驟 4 的結果，細微地改善定位。二階段的梯度法皆為 2.2 節介紹的演算法 1(Leung & Toh, 2009)，但因為預期步驟 4 的定位不精準，所以步驟 4 中的梯度法不需要迭代太多次，而且迭代的停止門檻也設定為較寬鬆；反之，步驟 5 為了修正定位，則梯度法需要設定較多的迭代次數與較小的門檻值。總括而言，步驟 4 與 5 是相輔相成，如果只有步驟 5，容易因為不足的距離資訊而收斂到區域最佳解，因此配合步驟 4 產生的初始解，定位得以更準確。

4.1.4 區域調整機制

二階段梯度法是用於定位較不易定位的流點，但由於這些流點相鄰的錨點數相當少，不足的距離資訊與不理想的初始解可能使得梯度法定位錯誤，收斂到不理想的區域解，在測試中發現這些不理想的定位通常是翻轉或扭曲的變形，因此本研究提出二種區域調整機制，分別為「推拉調整法」與「翻轉調整法」，目的在於利用感測器間的連結性資訊，將定位誤差較大的流點調整到合理的位置，提供下一回合梯度法新的出發點，使其有機會收斂到全域最佳解。

Grad_MSA 演算法的步驟 6 可應用推拉調整法或翻轉調整法，每次進行調整前必須先辨別哪些節點有可靠的定位能做作為調整的參考，以及哪些流點的定位不可靠且需要調整，令流點 i 在實際位置相鄰的節點集合為

$$\hat{N}_i = \{j | (i, j) \in E\} \quad (4.3)$$

而其以梯度法定位後相鄰的節點集合為

$$\bar{N}_i = \{j | \bar{d}_{ij} \leq r, j \in V\} \quad (4.4)$$

並令

$$\tilde{N}_i = \left\{ j : \left| \bar{d}_{ij} - \hat{d}_{ij} \right| \leq \lambda, j \in \bar{N}_i \right\} \quad (4.5)$$

表示流點 i 相鄰的節點中，定位後與流點 i 的距離跟定位前給定的距離相似的節點，門檻值設定 $\lambda = [0.01r, 0.1r]$ 為最佳。

本研究設所有可作為調整時的參考節點為

$$V_R = V_a \cup \left\{ i : \left| \tilde{N}_i \right| / \left| \bar{N}_i \right| > \rho \right\} \quad (4.6)$$

包含錨點以及部分定位後的流點，當一個流點周圍的節線中，有百分之 ρ 的節線其長度相似於給定的距離長時，則視此流點有可靠的定位，通常 $\rho = [0.4, 0.6]$ 。再者，設 V_A 為需要調整的流點，最簡易的判斷方法是除了 V_R 以外皆為需要調整的流點，也就是

$$V_A = V_s / V_R \quad (4.7)$$

此外，本研究亦提出另一個啟發式演算法來找出需要調整的流點以及調整的優先順序。首先找出所有定位後長度與給定距離較不相似的節線 $\left\{ (i, j) \mid i \in V_s, j \in (\bar{N}_i / \tilde{N}_i) \right\}$ ，而後加總這些節線兩端各流點重複的次數，並依次數來排序流點，重複次數越高的流點表示定位誤差較大，從它開始調整有機會一次改進較多定位誤差。在辨別調整的參考節點與需要調整的節點後，以下分別介紹推拉調整法與翻轉調整法的做法。

4.1.4.1 推拉調整法

推拉調整法目的為將不良定位的流點推出不適當的位置，並且拉向適當的位置，以流點 $i \in V_A$ 與另一節點 $j \in V_R$ 來探討， $j \in \hat{N}_i$ 表示 i 與 j 在實際位置中是相鄰的，而 $j \in \bar{N}_i$ 表示 i 與 j 在定位結果中是相鄰的，如果定位是正確的，則 i 與 j 相鄰與否的關係應該在實際位置中與定位結果中皆相符，因此當 $j \in \hat{N}_i$ 卻 $j \notin \bar{N}_i$ ，或者 $j \notin \hat{N}_i$ 卻

$j \in \bar{N}_i$ 時，表示 i 的位置應該被調整，以下分別探討在此二種狀況中如何調整。

在 $j \in \hat{N}_i$ 卻 $j \notin \bar{N}_i$ 的情況中，二個感測器實際上應該相鄰，但是在定位結果中卻不相鄰，因此應該將 i 拉到 j 的通訊範圍之內，其拉力位移的向量為

$$\Delta_i^{pull} = \frac{\bar{d}_{ij} - r}{\bar{d}_{ij}} (\mathbf{x}_j - \mathbf{x}_i), \quad j \in \hat{N}_i, j \notin \bar{N}_i, j \in V_R \quad (4.8)$$

而在 $j \notin \hat{N}_i$ 卻 $j \in \bar{N}_i$ 的情況中，二個感測器實際上不應該相鄰，但是在定位結果中卻相鄰，因此應該將 i 推出 j 的通訊範圍之外，其推力位移的向量為

$$\Delta_i^{push} = \frac{r - \bar{d}_{ij}}{\bar{d}_{ij}} (\mathbf{x}_i - \mathbf{x}_j), \quad j \notin \hat{N}_i, j \in \bar{N}_i, j \in V_R \quad (4.9)$$

最終，流點 i 調整後的新位置為

$$\mathbf{x}'_i = \mathbf{x}_i + \Delta_i^{pull} + \Delta_i^{push} \quad (4.10)$$

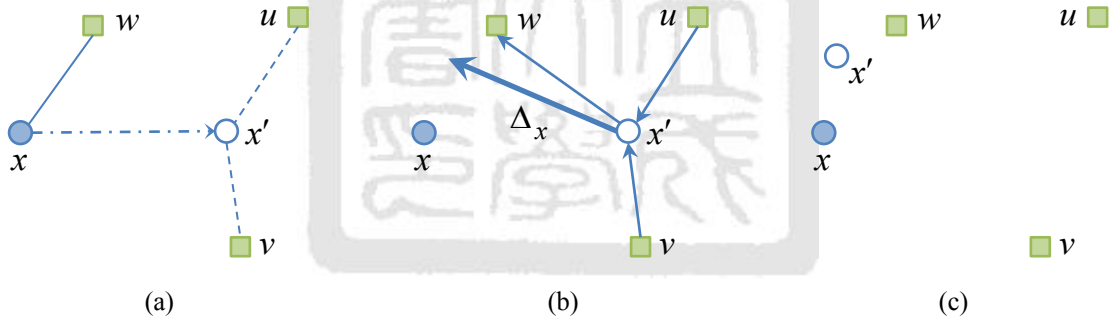


圖 4-3：推拉調整法

以圖 4-3 為例，在實際位置中流點 x 只與節點 w 相鄰，不與節點 u 與 v 相鄰，但在定位結果中，流點定位於 x' ，不與 w 相鄰，卻與 u 與 v 相鄰(圖 4-3a)，即是 $w \in \hat{N}_x$ 卻 $w \notin \bar{N}_x$ ，而 $u, v \notin \hat{N}_x$ 卻 $u, v \in \bar{N}_x$ ，若 $u, v, w \in V_R$ ，則在推拉調整法中， w 要拉近 x' ，而 u 與 v 要推開 x' ，其拉力與推力的向量分別如圖 4-3b 中細箭號所示，而所有的加總 Δ_x 即為粗箭號所示，將 x' 依向量 Δ_x 平移即為調整的結果(圖 4-3c)，雖然透過推與拉的調整不一定能將流點移到最佳的位置，但調整後的位置符合其與周圍節點相鄰與

否的一致性，在爾後的梯度法中較有機會收斂到最佳解。

4.1.4.2 翻轉調整法

定位的結果容易因不足的距離資訊產生 2.6 節所述的翻轉變形，通常經過推拉調整法後仍無法矯正的定位即是翻轉變形，因此本研究在 Grad_MSA 演算法多次迭代的步驟 6 中使用一次性的翻轉調整法，目的為矯正網路中所有可能的翻轉變形。翻轉的變形往往是一群定位不佳的流點同時沿著某一對稱軸翻轉，也就是這一群流點間的相對距離雖然與給定的距離資訊相符，但這群流點與其它節點的距離卻不符合距離資訊，因此翻轉調整法必須先辨識翻轉變形的群組，再求解其鏡射的對稱位置。

首先辨識需要調整的流點中有哪些連通分量(Connected Component)，以深度優先搜尋法或廣度優先搜尋法能夠在線性時間內辨識所有的連通分量，而每個連通分量即是一個翻轉變形的群組。假設有一個連通分量 C ，則與 C 相鄰的錨點便是翻轉的依據，在 D 維空間中至少需要相鄰 D 個錨點才能計算對稱軸或對稱面，以二維空間為例，若分量 C 恰好相鄰二個錨點，則通過此二個錨點的直線即是分量 C 翻轉的對稱軸，如果相鄰超過二個錨點，則取這些錨點的線性回歸做為對稱軸，而後依此對稱軸計算分量 C 在另一端的鏡射位置，即為調整後的位置。

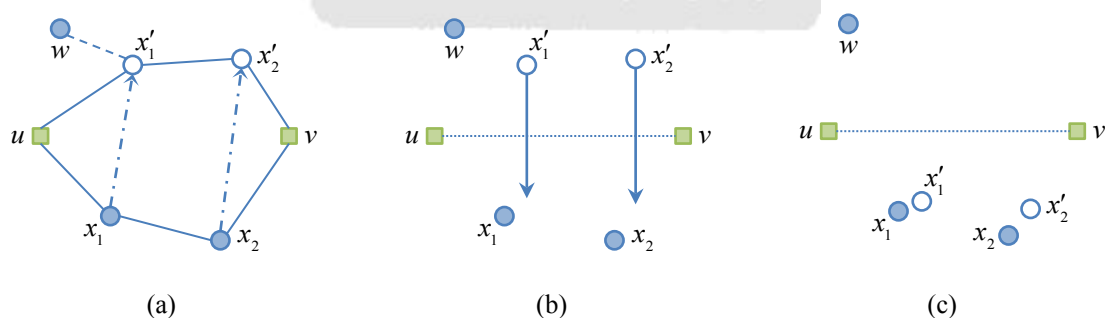


圖 4-4：翻轉調整法

以圖 4-4 為例，流點 x_1 與 x_2 定位的結果在 x'_1 與 x'_2 ，與另一節點 w 相鄰(圖 4-4a)，但藉由推拉調整法無法矯正，因此觀察其是否能以翻轉調整法矯正。 $x_1, x_2 \in V_A$ 且 x_1 與

x_2 屬於同一個連通分量，這個分量恰好相鄰錨點 u 與 v ，因此由通過錨點 u 與 v 的直線作為對稱軸，再求解 x'_1 與 x'_2 於對稱軸另一端的鏡射位置(圖 4-4b)，調整後的位置即如圖 4-4c 所示。由於這個分量本不應該與節點 w 相鄰，因此流點 x'_1 與 x'_2 定位後的位置偏右，以至於翻轉到對稱軸另一端後，與實際位置 x_1 與 x_2 仍有偏差，但透過後續的梯度法有機會收斂到最佳解。

4.1.5 停止條件

在錨點與距離資訊充足的情況下，Grad_MSA 演算法在步驟 1 至 3 就能快速且準確地定位好所有流點，即可終止演算法。然而當距離資訊較稀疏時，則必須藉由步驟 4 與 5 的二階段梯度法定位，梯度法定位後以 4.1.4 小節的方法判斷哪些流點的定位不可靠，形成流點定位不佳的清單 V_A ，如果 $V_A = \emptyset$ 就能終止演算法；但如果仍存在定位不佳的流點，則需要透過步驟 6 的調整機制來矯正這些流點的位置，調整後再次進行梯度法來定位，並重新判斷定位不可靠的流點清單，如此反覆以梯度法與區域調整機制修正流點的定位，直到 $V_A = \emptyset$ 即可終止演算法。

在步驟 5 與 6 的反覆迭代中，有可靠定位的流點會逐漸增加，這些流點能扮演如錨點般的角色來輔助定位，使得 V_A 中的流點個數的逐漸減少，最終使得所有流點都有可靠的定位結果；然而，缺乏有利定位的資訊時，部分流點可能遲遲無法有良好的定位，因此演算法在步驟 5 與 6 的反覆迭代後，如果 V_A 內容一直沒有變化，則終止演算法。測試中步驟 5 與 6 迭代約三至六次後， V_A 內容即不再變化，因此本研究也依此設定最高迭代次數，限制演算法在有限迭代次數後終止。

4.2 數值實驗

本節測試 Grad_MSA 其中各啟發式演算法對定位的幫助，包含迭代多邊定位法、最短路徑組成的人工節線以及區域調整機制，並與 Kim et al.(2009)的 SFSDP 演算法比較。以下實驗隨機產生 50 個 WSN，皆均勻分布於 $[0,1]^D$ 空間中，透過控制通訊半徑由每個 WSN 分別衍生 10 種不同可定位程度的 WSN，其分別介於 1%至 100%分成的 10 個區間中。以上共 500 種 WSN 首先去除第三類流點，再進行定位，而後定位結果分成第一類與第二類流點分別探討其定位精準度。所有實驗皆在一台處理器為 Intel Core i7-3770 與 4GB 記憶體的电腦中運行，作業系統為 Ubuntu 10.04，定位演算法均在 Matlab R2008a 環境中運作。

4.2.1 Grad_MSA 的成效

「Grad_MSA」命名中 M、S 與 A 分別代表迭代多邊定位法、最短路徑擴增人工節線與區域調整機制，以下首先測試這三種啟發式演算法對定位的影響，分別實驗僅使用梯度法(Grad)、梯度法附加迭代多邊定位法(Grad_M)、梯度法附加最短路徑擴增節線(Grad_S)、梯度法附加區域調整機制(Grad_A)與 Grad_MSA 完整演算法共五種方法的定位成效，實驗的 WSN 含有 1000 個流點與 100 個錨點，其錨點與流點數量的比例參考自 Kim et al.(2009)的設定。實驗結果分成第一類與第二類流點討論定位的誤差，列於圖 4-5 與圖 4-6，數列資料為 50 組 WSN 的平均，其橫軸為可定位程度的區間，縱軸為定位誤差 RMSD 值，其以對數刻度顯示，RMSD 值愈低表示定位愈精準。

由圖 4-5 與圖 4-6 中可歸納以下觀察：

- (1) 從第一類流點的定位結果可明顯看出各方法的定位成效，而對於第二類流點的定位效果不顯著，其中 Grad 的定位誤差均為最高，而 Grad_MSA 的定位誤差最低，表示 Grad_MSA 發揮三種方法的綜效，尤其在圖 4-5 中，可定位程度達 50%時，RMSD 下降至令人滿意的定位精準度，而在可定位程度達

70%以上時，其定位第一類流點的精準度相當高。

- (2) 迭代多邊定位法是專為定位第一類流點所設計，因此 Grad_M 在圖 4-5 中表現相當優異，而 Grad_M 的數列貼近於 Grad_MSA，表示迭代多邊定位法是 Grad_MSA 對於第一類流點定位成功的關鍵方法，尤其對可定位程度達 70% 的網路中，迭代多邊定位法能成功定位所有第一類流點；而可定位程度低於 50% 時，表示距離資訊相當稀疏，部分流點也無法以迭代多邊定位法定位。
- (3) Grad_S 一直都有比 Grad 稍好的表現，尤其在圖 4-5 中可定位程度達 90% 以上時能有效降低定位誤差，而人工節線主要是設計來彌補第二類流點不足的距離資訊，因此圖 4-6 中 Grad_S 的數列接近於 Grad_MSA，顯示對於第二類流點的定位，以最短路徑新增人工節線的確有助於加強梯度法的定位。
- (4) 對於第二類流點的定位，Grad_A 的數列僅些微低於 Grad，表示單獨附加區域調整機制對定位並沒有顯著的影響，由於定位過程中，僅能透過相當有限的資訊來辨別當下各流點的定位是否可靠，因此區域調整機制無法保證其調整了所有需要調整的流點，或是錯誤調整了不需要調整的流點。
- (5) 理論上，第二類流點是難以定位的，但由圖 4-6 顯示這三種輔助的啟發式演算法皆有助於定位難以定位的第二類流點，值得注意的是，可定位程度達到 100% 的網路中將沒有第二類流點，因此圖 4-6 的數列缺了一個區間。

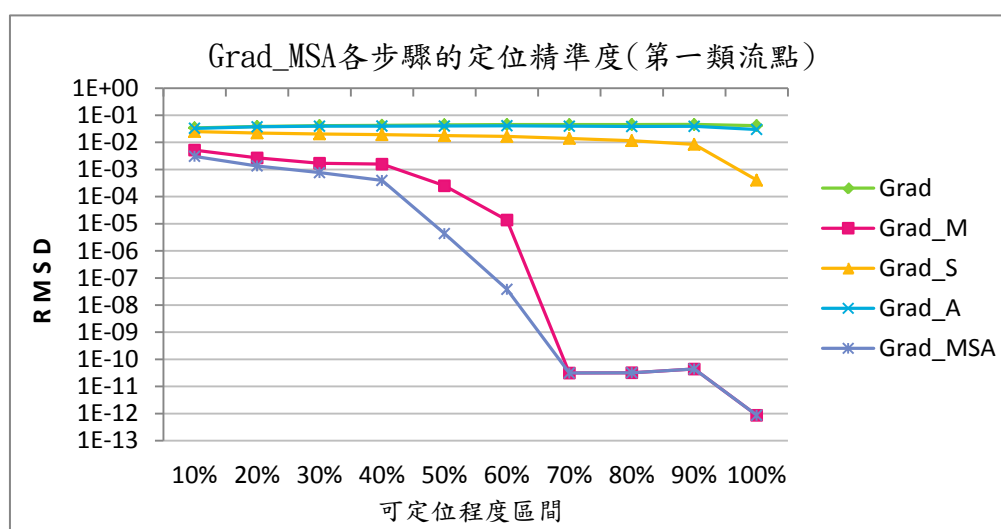


圖 4-5：Grad_MSA 各步驟的定位精準度比較(第一類流點)

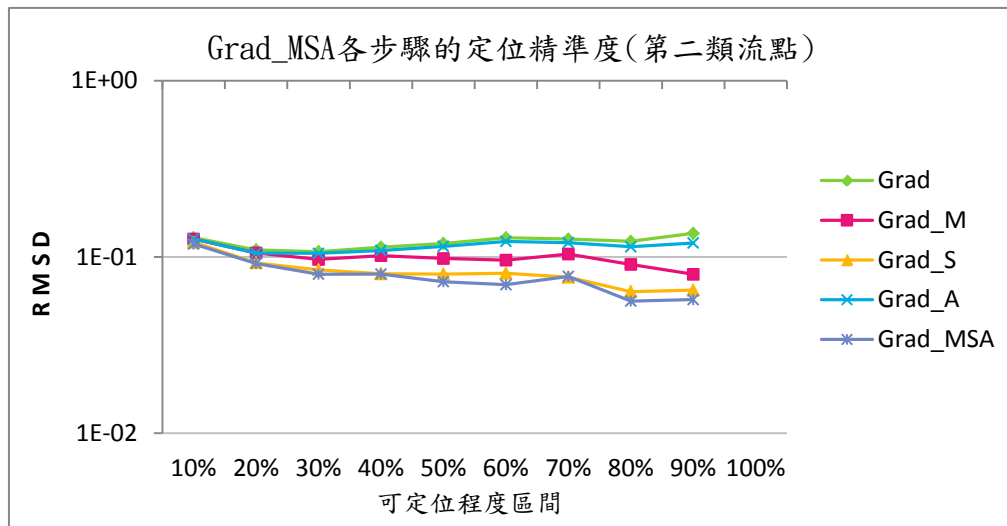


圖 4-6：Grad_MSA 各步驟的定位精準度比較(第二類流點)

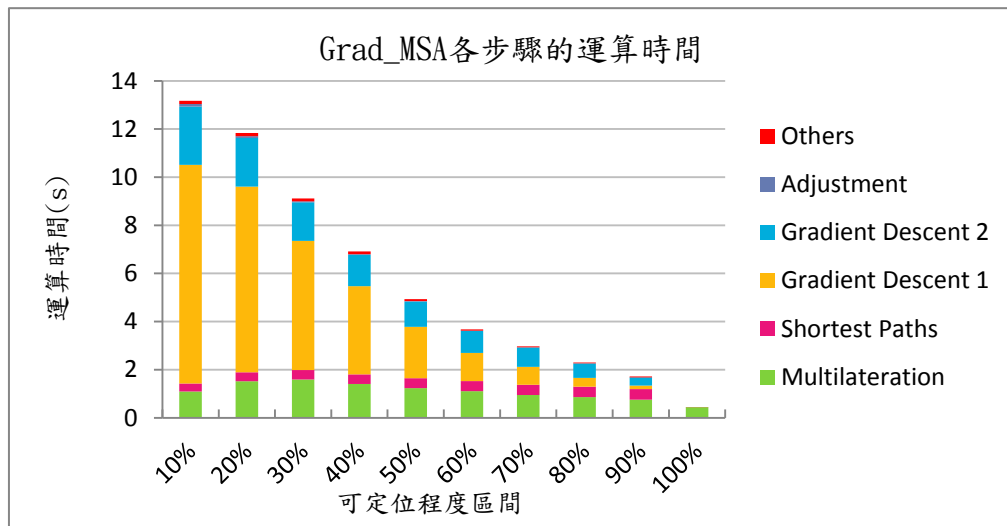


圖 4-7：Grad_MSA 各步驟的運算時間比較

圖 4-7 堆疊直條圖列出 Grad_MSA 定位時各步驟的運算時間，其橫軸同樣為可定位程度的區間，縱軸則是運算時間。結果顯示隨著可定位程度的提高，定位時間大幅降低，其中變動最大的是兩階段梯度法的運算時間，尤其以第一階段梯度法花費的時間最長，由於可定位程度愈低時，距離資訊愈稀疏，第一階段梯度法難以收斂，因此會迭代搜尋較多次。而第二階段梯度法又由不佳的初始解開始搜尋，因此也會花費較長的計算時間。而事實上可定位程度低時，表示第二類流點數遠多於第一類流點，由圖 4-5 與 4-6 已知定位難以精準，因此若限制兩階段梯度法的運算時間，並不會對定位精準度有顯著的影響。

隨著可定位程度的提高，迭代多邊定位法的計算時間有降低的趨勢，且迭代多邊定位法所占的時間比例也就愈高，由於可定位程度愈高即距離資訊愈充足，因此迭代多邊定位法能夠快速地定位，而且因為第一類流點的比例提高，使得迭代多邊定位法機能定位大多數的流點，就可以減少二階段梯度法的時間。而最短路徑演算法的計算時間稍微隨著可定位程度提高，呼應到 Johnson 演算法專長於稀疏的網路，因此對於節線愈多的網路，計算時間就愈長。

4.2.2 Grad_MSA 與 SFSDP 的比較

本研究探討的問題相近於 Kim et al.(2009)，因此以下首先以該文獻提出的 SFSDP 演算法定位 4.2.1 小節中相同的 50 組 WSN，與本研究 Grad_MSA 演算法的定位結果比較，同樣將流點分類來觀察定位精準度。此外，我們亦在網路的距離資訊加入人工的誤差，比較二種演算法在距離測量含有誤差時的定位成效。

實驗結果同樣以可定位程度的區間為橫軸，縱軸以對數刻度顯示定位誤差值，圖 4-8 比較 Grad_MSA 與 SFSDP 的定位精準度，顯示二種演算法的 RMSD 數列相近，並隨著可定位程度的提高而降低定位誤差，尤其在可定位程度達 90% 後，Grad_MSA 的定位精準度遠超越 SFSDP。然而，如此的比較未將流點分類觀察，難以看出二種演算法的特性，因此以下分成第一類與第二類流點討論定位精準度。

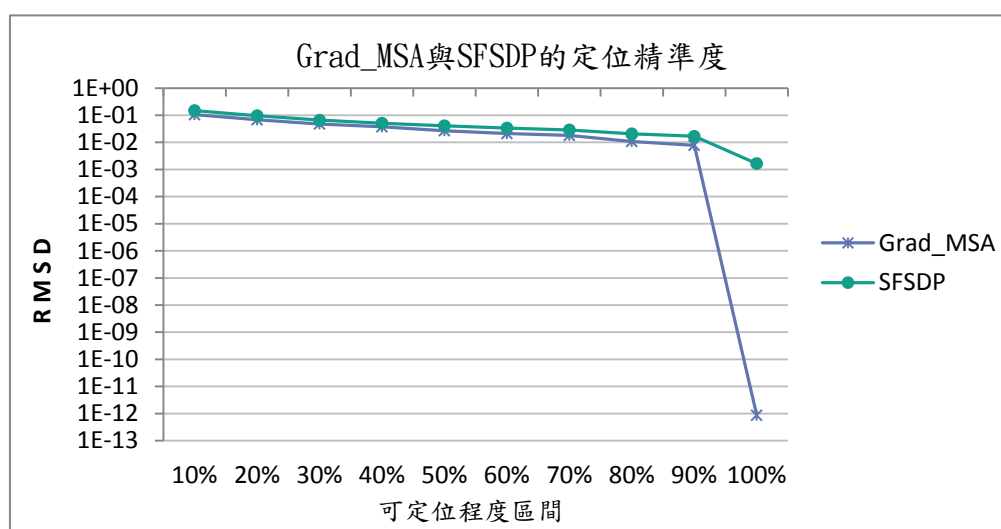


圖 4-8：Grad_MSA 與 SFSDP 的定位精準度比較

圖 4-9 與圖 4-10 將流點分類顯示二種定位演算法的定位誤差，從中可觀察到無論是定位哪一類流點，Grad_MSA 的表現皆較好，因為 Grad_MSA 對不同類的流點分別以不同方法處理，對第一類流點特使用迭代多邊定位法，在可定位程度達 50% 以上時，定位精準度明顯優於 SFSDP，而第二類的流點難以定位，二種演算法的定位結果皆不佳，但 Grad_MSA 仍略勝一籌。SFSDP 的定位誤差都維持地差不多，因為它將整體網路以半正定規劃求解，因此第一類流點的定位可能會受第二類流點的影響而使得定位誤差增加，唯獨在可定位程度達 100% 時，定位誤差稍微降低。

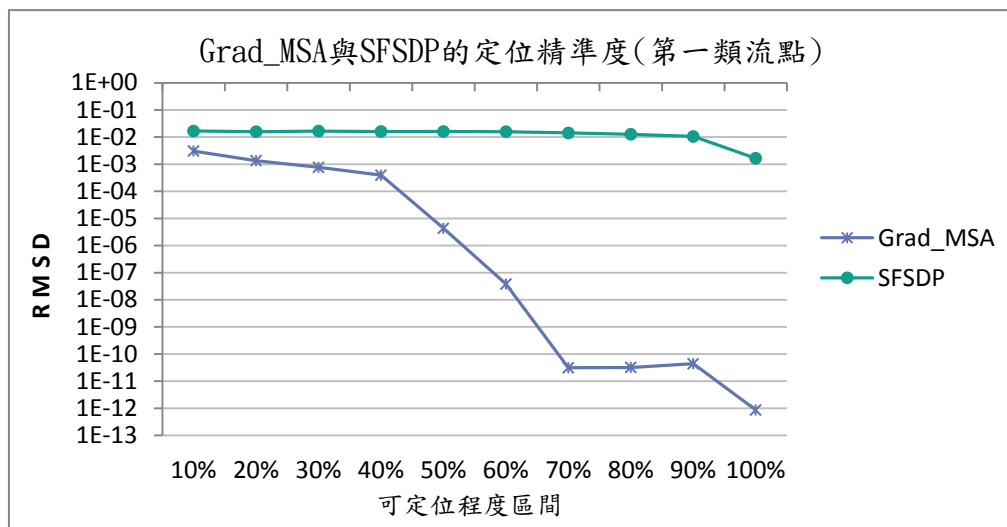


圖 4-9：Grad_MSA 與 SFSDP 的定位精準度比較(第一類流點)

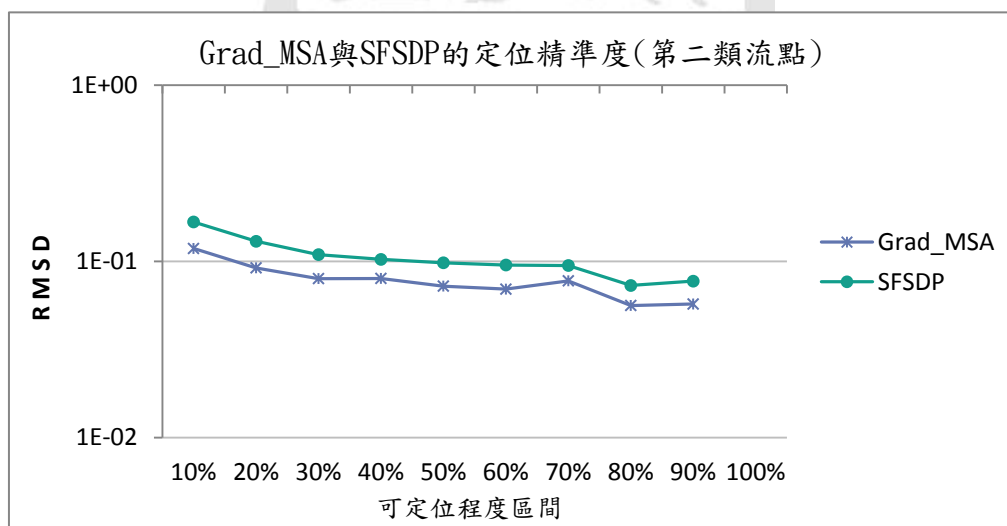


圖 4-10：Grad_MSA 與 SFSDP 的定位精準度比較(第二類流點)

圖 4-11 為 Grad_MSA 與 SFSDP 的運算時間，對於定位 1000 個流點的 WSN 頂多花費 13 秒，而隨著可定位程度的增加，可明顯看出 Grad_MSA 的運算時間遞減，但 SFSDP 的時間卻遞增，這是因為 Grad_MSA 在可定位程度較低時，需要花費較長的時間於二階段梯度法，而可定位程度高時，就可以靠迭代多邊定位法處理大部分的流點，在可定位程度達 100% 時，甚至能夠在 1 秒內定位完成；但 SFSDP 的運算時間卻容易受網路的規模影響，因此當可定位程度愈高時，網路的節線數愈多，SFSDP 的計算時間就愈長。若配合定位精準度的結果，則當可定位程度在 40% 以上時，使用 Grad_MSA 得以較快速且較精準地定位。

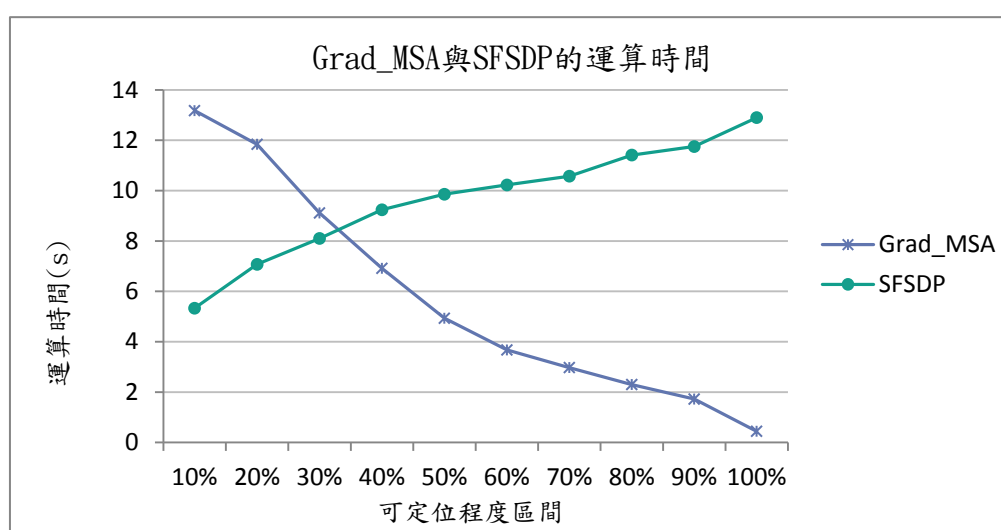


圖 4-11：Grad_MSA 與 SFSDP 的運算時間比較

圖 4-12 至圖 4-15 以盒鬚圖顯示 50 組 WSN 分別以 Grad_MSA 與 SFSDP 定位後 RMSD 值的變動範圍，同樣將流點分類觀察，縱軸以對數刻度顯示定位誤差值，數列中盒狀的中線為中位數，盒狀上限為第一分位數，下限為第三分位數，鬚線延伸盒狀的 1.5 倍高。在可定位程度 50% 以上時，Grad_MSA 對於第一類流點有較穩定的定位精準度，但在可定位程度不足 50% 時，因為距離資訊的不足，定位精準度的變動幅度較大；而 SFSDP 則一樣在可定位程度為 100% 時，定位精準度才有明顯的變動，表示當可定位程度不滿 100% 時，SFSDP 的定位表現均不佳。而對於定位第二類的流點，二種方法的定位結果變動差不多，但都沒有盒狀區間都處於較高的 RMSD 值。

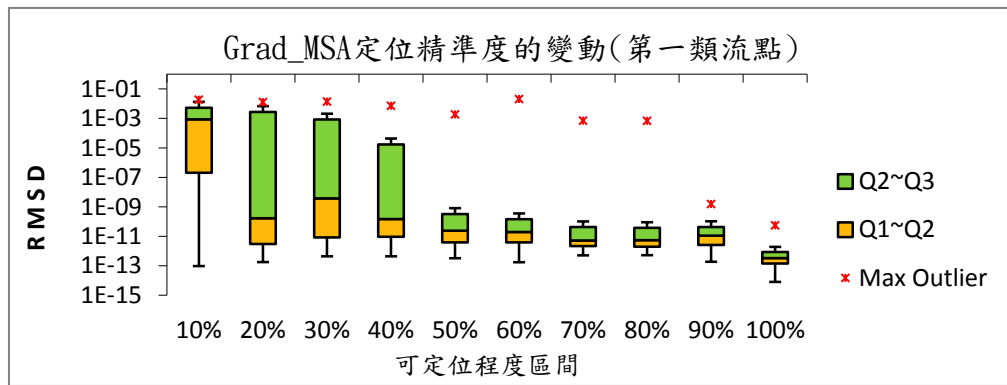


圖 4-12：Grad_MSA 定位精準度的變動(第一類流點)

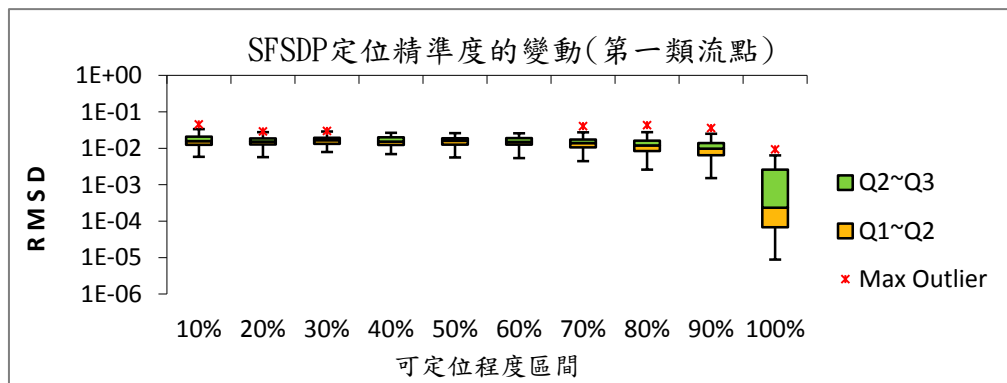


圖 4-13：SFSDP 定位精準度的變動(第一類流點)

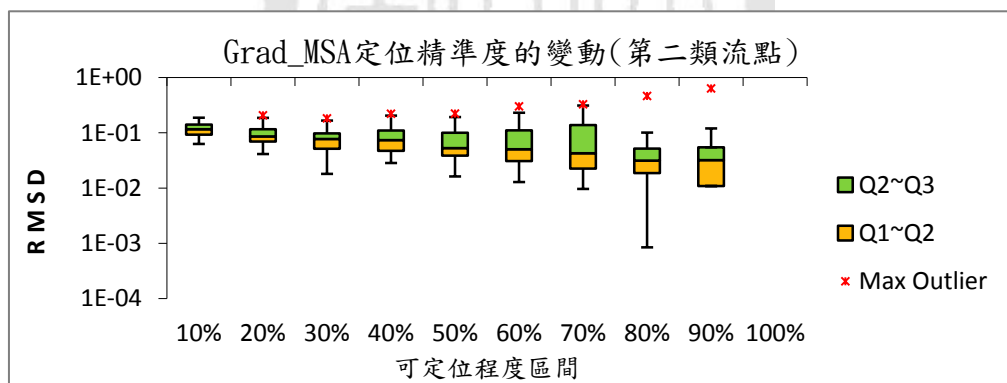


圖 4-14：Grad_MSA 定位精準度的變動(第二類流點)

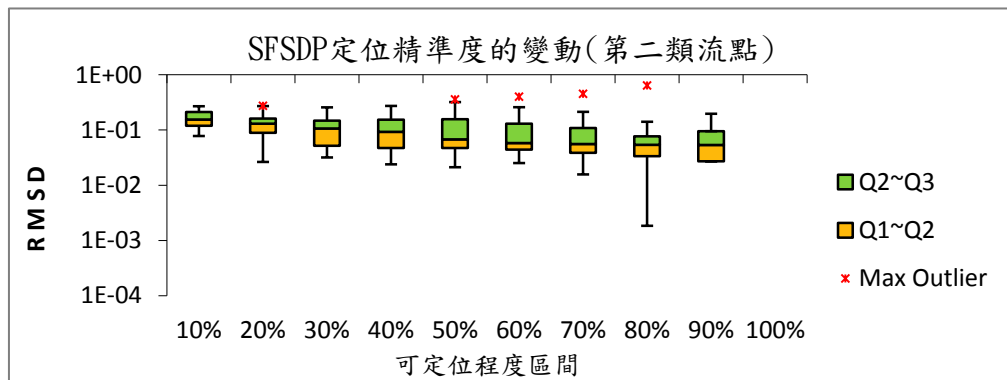


圖 4-15：SFSDP 定位精準度的變動(第二類流點)

SNL 問題的挑戰性之一即是當距離資訊含有誤差時，是否還能得到令人滿意的定位成效，將以上 50 組 WSN 的距離資訊加入人工誤差後，定位結果列於圖 4-16，其顯示 Grad_MSA 與 SFSDP 的平均定位誤差相近，且只有在可定位程度達 100%時，才有令人滿意的定位精準度，由於 Grad_MSA 並沒有特別處理距離資訊的誤差，因此除非網路的可定位程度夠高，否則定位成效與 SFSDP 差不多。

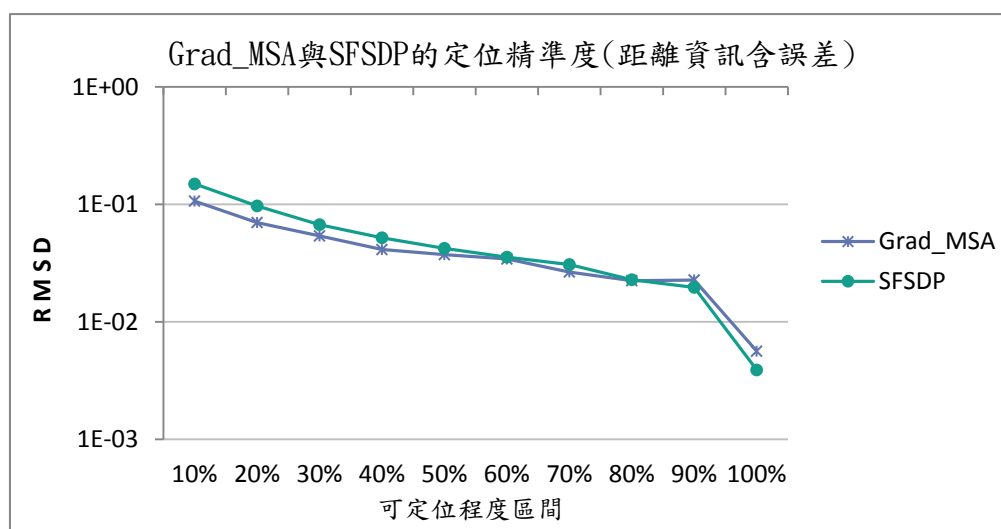


圖 4-16：Grad_MSA 與 SFSDP 的定位精準度(距離資訊含誤差)

4.2.3 無錨點 WSN 的定位

以上實驗的 WSN 中，錨點數皆為流點數的 1/10，然而在實際的定位情況中網路的錨點數可能更少，甚至沒有錨點。如果以上的測試例中不含有錨點，則定位後最佳的定位精準度如圖 4-17 中紅色數列所示，顯示當缺乏錨點時，僅依靠流點間的距離資訊不足以精準地定位。以一個含有 100 個流點的 WSN 為例，若給定 10 個錨點，則 Grad_MSA 能夠精確地定位，如圖 4-18a 所示。然而如果網路不含錨點，則定位結果如圖 4-18b 所示，其中圓圈為流點的真实位置，而紅色節點為定位的結果，若定位很精確，紅色節點位於圓圈當中，若定位不精確，則圓圈與紅色節點間以粉紅色線表示定位誤差。為了降低定位時對錨點的依賴，並一般化 SNL 問題，我們在下一章亦將發展無錨點的定位演算法，使得 Grad_MSA 即使在處理錨點數極少甚至不存在錨點的網路時，也能夠準確且快速地定位。

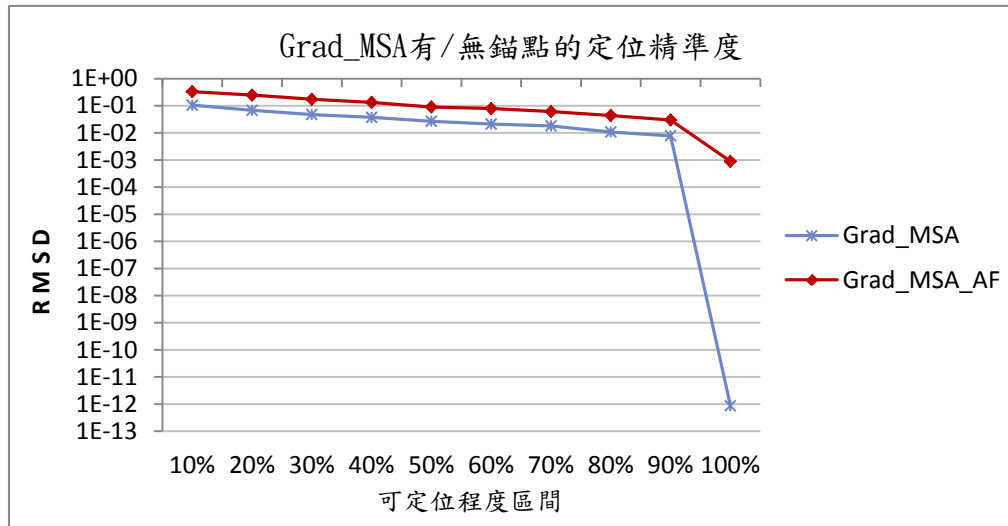


圖 4-17：Grad_MSA 有/無錨點的定位精準度

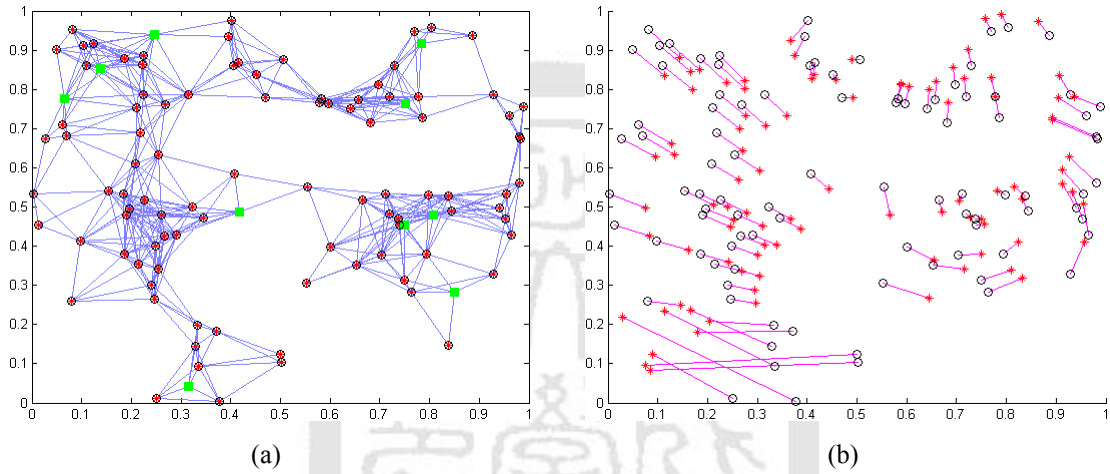


圖 4-18：Grad_MSA 有/無錨點的定位結果範例

4.3 小結

本章基於梯度法發展 Grad_MSA 演算法求解 SNL 問題，對於較易定位的流點以迭代多邊定位法快速準確地定位，包含 D 邊定位法與 $D+1$ 邊定位法，而對於較難定位的流點以最短路徑組成的人工節線輔助梯度法，並利用節點間的連結性質以推拉調整法與翻轉調整法調整定位到合理的位置，重複迭代這些動作使定位更精確。

實驗顯示 Grad_MSA 中各啟發式演算法都有助於增進定位的成效，並且與 SFSDP 比較，無論是運算時間或定位精準度，Grad_MSA 都有較佳的表現，且 Grad_MSA 能分別針對第一類流點與第二類流點定位，且第一類流點的定位精準度不受第二類流點

影響。Grad_MSA 對於可定位程度達 70%以上的 WSN 能有較佳的定位成效，而對於距離資訊含有誤差的網路，則可定位程度達 90%以上時定位效果較佳。

SFSDP 對於可定位程度達 90%以上的 WSN 才能有較佳的定位效果，但第一類流點的定位精準度會受到第二類流點影響，使得所有流點的誤差都維持在同一水平。在 Kim et al.(2009)文獻的實驗數據中雖然宣稱定位誤差能達到 $2e-8$ 或更低，然而該文獻測試含 1000 個流點的 WSN 時，所使用的最小通訊半徑為 0.1，遠大於本研究最大的通訊半徑(約為[0.06,0.08])，因此該文獻測試的 WSN 都達到 100%的可定位程度，是相對簡單易解的測試例，難以看出定位演算法的優劣。



第五章

無錨點定位演算法

錨點是影響定位成效的一大因素，然而錨點感測器的成本較高，在實際定位中必須盡可能降低錨點數，甚至在部分情況中 WSN 不含錨點，或是所佈署的錨點不足以用來定位，使得 Grad_MSA 演算法難以準確地定位，因此本研究亦發展無錨點定位演算法，以下首先介紹該演算法與其用處，再以三維空間的網路測試，並與類似文獻比較定位成效。

5.1 Grad_SS 演算法

本研究發展的無錨點定位演算法稱為 Grad_SS，其中「SS」表示選取種子感測器 (Seed Selection)，可以視為 Grad_MSA 的前置處理，而主要定位仍透過 Grad_MSA 達成，整體 Grad_SS 演算法的流程為：

演算法 6：LOCALIZATION ALGORITHM FOR ANCHOR-FREE WSN

procedure Grad_SS

step1: Select seed sensors

step2: Set seed sensors as anchors

step3: Apply Grad_MSA

step4: Apply linear transformation using original anchors(optional)

end procedure

Grad_SS 首先從流點中找出數個種子感測器，將其當作錨點，由這些錨點形成暫時的座標系，在該座標系中運用 Grad_MSA 展開定位，若本來的 WSN 中含有錨點，則定位後將結果依原本的錨點轉置到實際的座標系中。Grad_SS 的成敗在於選取適當的種子感測器，以下分別介紹種子感測器的選取與線性轉置。

5.1.1 種子感測器的選取

由於種子感測器將作為錨點來幫助定位，因此在 D 維空間中至少需要用 $D+1$ 個種子感測器，而種子感測器與錨點的差異在於，種子感測器與其它流點一樣，其位置都是未知的，因此種子感測器在作為臨時錨點之前，必須先對其賦予一個暫時的座標。方法則是先將其中一個種子感測器當作原點，再取原點與另一個種子感測器間的節線設為座標軸(如： x 軸)，而後利用種子感測器間的距離就可以算出其它種子感測器在此座標系中的位置；如此的方法可以簡單的轉換到三維空間中，同樣以一個種子感測器為原點，與另外二個種子感測器形成平面，之後即計算其它種子感測器的位置。

以圖 5-1 為例，假設在一個二維空間的網路中選取了 u 、 v 和 w 為種子感測器，則將 u 作為原點， v 作為 x 軸上的一點，而 x 軸上通過原點法向量為 y 軸，則可計算出 w 的位置，形成的暫時座標系是實際座標系平移與旋轉後的結果，之後即可將這些種子感測器當作錨點開始定位整個網路。由圖例可知，種子感測器必須相鄰才能賦予暫時的座標，最佳的情況即是所有種子感測器組成一個分團(Clique)，分團為一個完全圖，其中各節點都互相相鄰，因此符合整體剛性。

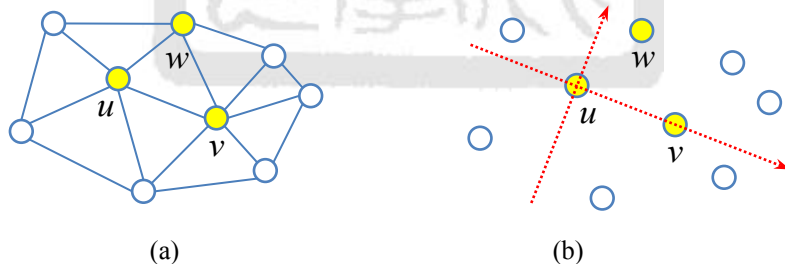


圖 5-1：種子感測器形成的暫時座標系

Grad_MSA 演算法中定位效果最好的步驟即是迭代多邊定位法，因此種子感測器的選取應該要有利於應用多邊定位法，也就是說如果種子感測器座落於在網路中節點較密集的部分，有較高的機會能定位較多流點，以下依此概念發展二種方法來選取種子感測器，且設計為只取最少需求的種子感測器數。

5.1.1.1 最大度數法

度數(degree)是一個節點相鄰的節點數，當一個節點的度數愈高時，除了代表這個節點愈容易被定位，同時也表示這個節點如果已定位，可以幫助定位愈多周圍的其它節點，因此在選擇種子感測器時，優先選取度數較高的流點。當要選擇 $D+1$ 個種子感測器時，則必須先列舉網路中所有含 $D+1$ 個節點的分團，再統計各分團相鄰的節點數，取出度數最高的分團即為種子感測器，由於尋找分團尚未有更有效的方法，因此列舉所有分團的複雜度為 $O(|V|^{D+1})$ 。

為了降低複雜度，另一種加速的作法是先找出度數最高的流點，再從包含此流點的分團中，選取相鄰節點數最多的分團為種子感測器，原因是高度數節點周圍的節點通常度數也較高，因為這些節點可能都相鄰同樣的節點，如此複雜度可降至 $O(|V|^D)$ 。最大度數法是以貪婪法的精神找出有希望定位最多節點的種子感測器，然而無法保證找到最佳的種子感測器，其用於定位的成效仍然視網路的拓撲而定。

5.1.1.2 最大體積法

最大體積法是選擇所有含 $D+1$ 個節點的分團中體積或面積最大的團作為種子感測器，當網路是均勻分布時，如果種子感測器涵蓋的體積愈大，則有機會定位較多的流點，而且最大體積法選出的分團不易共線或共面，因此定位時可避免發生錨點共線或共面產生的翻轉變形。同樣地，列舉所有分團的複雜度是 $O(|V|^{D+1})$ ，但也可以透過類似上一個小節所述的加速方法，使得複雜度降至 $O(|V|^D)$ ，且最大體積法亦無法保證找到最佳的種子感測器。

5.1.1.3 種子感測器選取方法的比較

最大度數法與最大面積法皆是採用貪婪法的精神，求取最有可能定位最多流點的種子感測器，但都無法保證可以依此種子感測器的定位成效最佳。計算複雜度方面，

兩者的計算複雜度相當，而最大度數法的計算尤其簡單。若由定位結果而論，以最大度數法所選取的種子感測器來定位網路可能會產生整體網路翻轉的變形，原因在於種子感測器若呈共線或共面時，則定位結果可能以種子感測器為對稱軸翻轉；而最大面積法能避免這種情況發生，因為在求取最大面積的同時，也確保種子感測器之間的距離最大化，有助於避免共線或共面。

5.1.2 線性轉置

對於無錨點的網路，節點並沒有明確的位置，只有節點間的相對位置才有意義，因此只需要執行 Grad_SS 演算法到步驟 3 即可得到定位結果。然而，如果網路本來含有錨點，但卻因為這些錨點無法應用於定位才使用 Grad_SS 定位的話，則定位的結果仍是建立於種子感測器形成的暫時座標系，必須藉由本來的錨點，將結果線性轉置 (Linear Transformation) 到真實的座標系。

線性轉置並不會改變節點間的相對位置，只包含了數個動作：平移(Translation)、對映(Reflection)、旋轉(Rotation)與縮放(Scaling)，這些動作必須基於二個座標系之間的關聯性，也就是欲轉置的網路圖與真實的座標系之間必須有數個參考的標的，在此即是網路中原本的錨點。透過對 Grad_SS 演算法步驟 3 的定位結果進行平移、對映、旋轉與縮放後，找到二種座標系之間錨點的最佳對應方式，再將所有節點都依此轉置到真實的座標軸。為了達到正確的轉置效果，在 D 維空間中至少需要用 $D+1$ 個參考標的。

5.2 無錨點定位的優勢

無錨點定位的優點除了可以大幅降低錨點的需求，也有助於一般化 SNL 定位問題為圖實現問題，圖實現的應用相當廣，分子構形(Molecular Conformation)問題即是其中之一，隨著核磁共振光譜學(Nuclear Magnetic Resonance Spectrometers)的興起，得以測量亞原子的距離或角度等資訊，透過這些資訊能夠繪製分子在三維空間中的結

構，而如何利用原子間的距離就能視覺化分子的結構，此問題與 SNL 相像，因此部分研究已將 SNL 的無錨點定位方法應用於分子構形問題，如 Biswas et al. (2008)與 Leung & Toh (2009)將其以 SDP 發展的演算法應用於求解蛋白質的構形，說明了 SNL 的解法可以應用到分子構形問題中。

此外，無錨點定位也有助於定位錨點數極少的 WSN，當網路中的錨點數相當低時，可能沒有任何流點跟 $D+1$ 個以上的錨點相鄰，因此如果直接以 Grad_MSA 演算法定位此網路，則 Grad_MSA 步驟 1 的迭代多邊定位法完全無法生效，整體定位結果可能很差。對於這類錨點稀少的網路，如果採用無錨點定位的技巧，首先選取種子感測器作為臨時的錨點，以這些錨點進行 Grad_MSA 的定位，有機會能成功定位更多流點，而定位後的結果再以本來給定的錨點線性轉置到實際的座標上。

以圖 5-2 為例，圖 5-2a 中網路的原始錨點位於綠色方塊，由於錨點不集中，使得多邊定位法無法定位任何流點。但若以 Grad_SS 演算法首先選擇數個種子感測器，如圖 5-2b 中的紅色方塊處，再以種子感測器形成暫時的座標系，如圖 5-2c 所示，而後依這些種子感測器使用迭代多邊定位法，則能成功定位所有流點，再對應原始錨點的位置找出線性轉置的方式，如圖 5-2c 旋轉 15 度再平移後即可重合到原始的座標系中，將所有節點轉置後，即可以得到相當優異的定位結果。

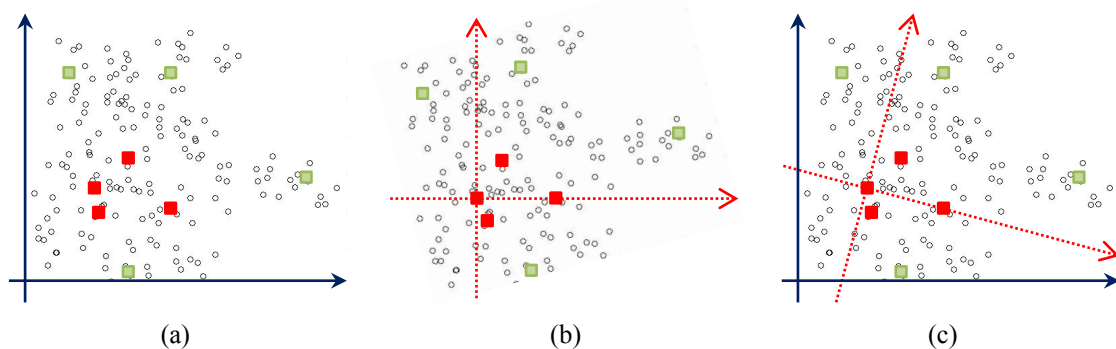


圖 5-2：無錨點定位演算法亦能應用錨點數極少的 WSN

5.3 數值實驗

本節測試 Grad_SS 與 SFSDP 定位無錨點網路的效能，測試例(Leung & Toh, 2009)為 13 組來自 Protein Data Bank 的蛋白質構形問題，目的為藉由原子間的距離重建蛋白質的結構，有助於了解蛋白質的功能，蛋白質構形問題與 SNL 問題相似，唯一差異是前者不含有錨點。測試例的規模從 166 個至 13488 個節點，節線長度不超過 6Å，且網路拓撲並非均勻分布。經 Loc_Idx 判斷，所有測試例的可定位程度皆達 100%。實驗中以最大面積法從各測試例選出 10 組 4 個種子感測器，分別以二種方法定位，再取平均的 RMSD 與運算時間，實驗結果如表 5-1 所示。

表 5-1：Grad_SS 與 SFSDP 的比較(無錨點網路)

測試例	節點數	種子 感測器數	Grad_MSA		SFSDP	
			RMSD	耗時(s)	RMSD	耗時(s)
1GM2	166	4	4.83E-11	0.22	4.06E-04	4.04
1PTQ	402	4	9.69E-08	0.27	3.83E-05	25.87
1HOE	558	4	1.02E-10	0.27	2.68E-05	53.26
1PHT	814	4	2.57E-06	1.61	4.46E-04	85.57
1AX8	1003	4	2.06E-08	0.82	1.05E-04	198.82
1F39	1534	4	2.17E-06	2.57	1.18E-03	667.26
1RGS	2015	4	8.63E-07	1.75	1.68E-04	276.47
1KDH	2923	4	4.18E-05	7.38	N/A	>1800
1BPM	3672	4	1.78E-07	6.87	N/A	>1800
1TOA	4292	4	5.70E-05	11.93	N/A	>1800
1MQQ	5681	4	1.30E-06	24.99	N/A	>1800
1I7W	8629	4	2.66E-04	79.54	N/A	>1800
1YGP	13488	4	1.62E-05	169.05	N/A	>1800

實驗結果中，當節點數達 3000 時，SFSDP 無法在 30 分鐘內完成定位，因此截斷實驗，顯示在定位資源有限的情況下，Grad_SS 較能成功完成定位。在定位成功的測試例中，Grad_SS 皆有較好且令人滿意的定位精準度，而結果顯示定位精準度不一定跟網路規模相關，表示定位精準度除了受到節點數與種子感測器的影響，網路的拓撲(Topology)也是一大因素，SFSDP 的表現則維持一致。在運算時間方面，二種方法的耗時都隨著網路的規模而遞增，但 Grad_SS 計算時間的增長較 SFSDP 慢，對於最大含有約 13000 個節點的網路，Grad_SS 耗費約略為 SFSDP 定位 1000 個節點的時間。

第六章

結論與未來研究

6.1 結論

無線感測網路(WSN)的應用往往需要事先知道各感測器的位置，且得知感測器的位置也有助於提升 WSN 的服務能力，而感測網路定位(SNL)問題是使用少數已知的錨點感測器位置與感測器間的距離資訊來定位出所有感測器的位置，藉以降低以人力配置或使用定位硬體的成本。SNL 是個 NP-hard 問題，並存在兩大挑戰性：首先，因為感測器間測量到的距離資訊通常與實際距離稍有誤差，使得在定位的過程中容易累積誤差，造成不精準的定位；再者，WSN 通常是大規模的佈署，當網路中節點數增加，計算時間也會隨著指數成長。

SNL 問題能以最小平方方法規劃成一個無限制式的非線性數學模式，在文獻中求解 SNL 問題的方法大致可分為集中式與分散式運算。集中式運算常見的作法是將本來非線性的數學模型加以線性化之，如半正定規劃，或運用線性代數的多維標度法求解；而分散式運算則自各個節點應用其與相鄰節點的關係來計算區域的相對位置，再以拼補策略將所有區域的定位結果組合成整體網路。分散式網路將運算分散在各個感測器上，其計算時間較短，但因只應用區域的資訊，所以定位較不準確；而集中式計算直接考慮整體網路的定位，可以得到較精確的定位結果，但也因此消耗較多計算時間。定位速度與精準度的權衡一直是文獻中定位演算法的挑戰，本研究以電腦模擬集中式定位為主，目的為開發能同時兼顧速度與精準度的演算法。

本研究首先建立檢驗 WSN 可定位程度的演算法 Loc_Idx，以多邊定位法與輪狀圖判斷網路的剛性，辨別給定的網路中哪些流點是可定位的，可定位程度有助於避免無謂的定位且可用以加強 WSN 的管理；此外，我們將流點依可定位程度分為三類，

其中第一類流點是判別為可定位的流點，第三類為不可定位的流點，而第二類則為雖然理論上不容易但有機會定位的流點。本研究認為定位時應忽略第三類流點以避免其所造成的定位誤差大幅影響到其它流點的定位，而定位結果中將第一類與第二類流點分開來檢視能更有效地辨別演算法的成效。

再者，本研究發展定位演算法 Grad_MSA，對不同定位難度的流點以不同的方法定位：對於較易定位的流點，以迭代多邊定位法快速地定位，並將已定位的流點作為錨點來幫助其它流點定位；而對於較難定位的流點，以二階段的梯度法直接求解 SNL 的數學模型，配合最短路徑擴增的人工節線來獲得好的初始解，並以推拉調整法與翻轉調整法等區域調整機制校正定位到合理的位置，反覆地調整與梯度法收斂以達到最佳的定位精準度。

實驗結果顯示 Grad_MSA 中各啟發式演算法都有助於增進定位的成效，且能分別針對第一類流點與第二類流點定位，使得第一類流點的定位精準度不受第二類流點影響。其中尤以迭代多邊定位法的效果最突出，能在短時間內準確定位第一類流點。而與 Kim et al.(2009)的 SFSDP 演算法比較，Grad_MSA 對於任何可定位程度的網路均有較佳的表現，且當網路的可定位程度愈高時，Grad_MSA 的計算速度與定位精準度都隨著提高，對於可定位程度達 70% 的網路，能有相當令人滿意的定位成效。然而，對於距離資訊含有誤差的網路，二種的方法的定位效果僅算差強人意。

實驗顯示當 WSN 中的錨點數極少或無錨點時，Grad_MSA 的定位準確度下降，計算時間也較長，因此本研究也發展了無錨點定位演算法 Grad_SS，其以最大度數法或最大體積法選擇種子感測器，將其作為錨點以 Grad_MSA 定位，而後再將定位結果線性轉置到真實座標系中。無錨點定位的發展不但有助於將 SNL 問題一般化，使 SNL 解法延伸到其它諸如分子構形問題等領域，並且也能用來定位錨點極少的網路。當給定的錨點對定位的幫助不大時，則能利用種子感測器定位，再利用本來的錨點進行線性轉置。實驗結果顯示，以蛋白質構形作為測試例，Grad_MSA 擁有比 SFSDP 更佳的定位精準度，且隨著網路規模愈大，Grad_MSA 的運算時間增長地較慢，能在

3 分鐘內求解高達 13000 個節點的大規模網路。

6.2 未來研究

網路的剛性理論雖然已經存在數十年，然而對於網路的可定位程度檢驗，目前仍無相當有效的方法。尤其在三維空間中，符合整體剛性的網路具備何種幾何性質仍是未知，本研究以輪狀圖延伸、D 邊定位延伸與 D+1 邊定位延伸發展 Loc_Idx 演算法，雖能辨識大部分的可定位節點，但由於其中仍涉及實際的定位動作，因此運作效能不佳，且難以保證是否能辨識出所有可定位的節點。在理想的情況中，可定位程度應能僅透過網路的連結性質而快速判斷出來，並應用於各維度空間的網路。

可定位程度的應用很多，首先其影響到流點的分類，本研究中對流點分為三類，其中第二類流點是第一類與第三類之間的灰色地帶，雖然理論上很難定位，但部分的第二類流點卻仍可能藉由其它資訊的輔助來定位。在未來的研究中，應該釐清這些可定位程度模糊的流點。其次，由於目前並沒有任何 WSN 的標竿測試資料，因此文獻中定位演算法的成效難以比較，部分文獻中雖然以無錨點的蛋白質構形來測試定位演算法，但此作法仍無法與 WSN 相提並論，畢竟錨點是影響定位成效的一大因素，這問題將可透過可定位程度的檢驗解決。可定位程度的檢驗使得不同的演算法在測試不同的資料例時，可依其資料例的可定位程度來比較出演算法的優劣。再者，Grad_SS 選擇種子感測器的方法尚不能保證找到最有益定位的種子感測器，若應用可定位程度的檢驗判斷種子感測器對定位的影響，將有助於找出最佳的種子感測器。

此外，SNL 問題的一般化已是個趨勢，本研究中將定位演算法 Grad_MSA 改進為 Grad_SS，使其能求解無錨點的 WSN，如此能將 SNL 問題的解法用於圖實現問題，分子構形問題即為其中一例。而未來經過適當的修改，SNL 問題的解法與可定位程度檢驗甚至能應用在 Alfakih et al. (1999) 與 Ding et al. (2010) 所討論的歐氏距離矩陣完成 (Euclidean Distance Matrix Completion) 問題，或是處理 Kim et al., (2012) 所探討的歐式距離幾何問題 (Euclidean Distance Geometry Problems)，具有不錯的延伸與應用性。

參考文獻

- Alfakih, A. Y., A. Khandani & H. Wolkowicz (1999), "Solving Euclidean Distance Matrix Completion Problems Via Semidefinite Programming," *Computational Optimization and Applications*, 12 (1-3), pp.13-30.
- Armijo, L. (1966), "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives," *Pacific Journal of Mathematics*, 16 (1), pp.1-3.
- Aspnes, J., D. Goldenberg & Y. R. Yang (2004), "On the Computational Complexity of Sensor Network Localization," *Algorithmic Aspects of Wireless Sensor Networks*, Berlin Heidelberg: Springer, 3121, pp.32-44.
- Berg, A. R. & T. Jordán (2003), "Algorithms for Graph Rigidity and Scene Analysis," *Algorithms - Esa 2003*, Berlin Heidelberg: Springer, 2832, pp.78-89.
- Berg, A. R. & T. Jordán (2003), "A Proof of Connelly's Conjecture on 3-Connected Circuits of the Rigidity Matroid," *Journal of Combinatorial Theory Series B*, 88 (1), pp.77-97.
- Biswas, P., K.-C. Toh & Y. Ye (2008), "A Distributed Sdp Approach for Large-Scale Noisy Anchor-Free Graph Realization with Applications to Molecular Conformation," *SIAM Journal on Scientific Computing*, 30 (3), pp.1251-1277.
- Biswas, P. & Y. Ye (2004), "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, ACM, pp.46-54.
- Biswas, P. & Y. Ye (2006), "A Distributed Method for Solving Semidefinite Programs Arising from Ad Hoc Wireless Sensor Network Localization," *Multiscale Optimization Methods and Applications*, US: Springer, 82, pp.69-84.
- Borg, I. & P. J. F. Groenen (2005), *Modern Multidimensional Scaling*, New York: Springer.
- Clark, B. N., C. J. Colbourn & D. S. Johnson (1990), "Unit Disk Graphs," *Discrete Mathematics*, 86 (1-3), pp.165-177.
- Connelly, R. (1991), "On Generic Global Rigidity," *Applied Geometry and Discrete Mathematics*, Providence, RI: American Mathematical Society, 4, pp.147-155.
- Connelly, R. (2005), "Generic Global Rigidity," *Discrete and Computational Geometry*, 33 (4), pp.549-563.

- Cota-Ruiz, J., J.-G. Rosiles, E. Sifuentes & P. Rivas-Perea (2012), "A Low-Complexity Geometric Bilateralization Method for Localization in Wireless Sensor Networks and Its Comparison with Least-Squares Methods," *Sensors*, 12 (1), pp.839-862.
- Ding, Y., L. Du, T. Yang & Y. Sun (2009), "Distributed Localization Algorithm for Wireless Sensor Network Based on Multidimensional Scaling and the Shortest Path Distance Correction," *Transactions of Tianjin University*, 15 (4), pp.237-244.
- Ding, Y., N. Krislock, J. Qian & H. Wolkowicz (2010), "Sensor Network Localization, Euclidean Distance Matrix Completions, and Graph Realization," *Optimization and Engineering*, 11 (1), pp.45-66.
- Doherty, L., K. S. J. Pister & L. El Ghaoui (2001), "Convex Position Estimation in Wireless Sensor Networks," *INFOCOM 2001 Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, 3, Anchorage, AK, April 2001, pp.1655-1663.
- Eren, T., D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson & P. N. Belhumeur (2004), "Rigidity, Computation, and Randomization in Network Localization," *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 4, March 2004, pp.2673-2684.
- Hendrickson, B. (1992), "Conditions for Unique Graph Realizations," *SIAM Journal on Computing*, 21 (1), pp.65-84.
- Jackson, B. & T. Jordán (2005), "Connected Rigidity Matroids and Unique Realizations of Graphs," *Journal of Combinatorial Theory, Series B*, 94 (1), pp.1-29.
- Johnson, D. B. (1977), "Efficient Algorithms for Shortest Paths in Sparse Networks," *Journal of the ACM*, 24 (1), pp.1-13.
- Kim, S., M. Kojima & H. Waki (2009), "Exploiting Sparsity in Sdp Relaxation for Sensor Network Localization," *SIAM Journal on Optimization*, 20 (1), pp.192-215.
- Kim, S., M. Kojima & M. Yamashita (2012), "Parallel Implementation of Successive Sparse Sdp Relaxations for Large-Scale Euclidean Distance Geometry Problems," Department of Mathematical and Computing Sciences Tokyo Institute of Technology, B-470.
- Laman, G. (1970), "On Graphs and Rigidity of Plane Skeletal Structures," *Journal of Engineering Mathematics*, 4 (4), pp.331-340.
- Leung, N.-H. Z. & K.-C. Toh (2009), "An Sdp-Based Divide-and-Conquer Algorithm for Large-Scale Noisy Anchor-Free Graph Realization," *SIAM Journal on Scientific*

- Computing*, 31 (6), pp.4351-4372.
- Li, X., B. Hua, Y. Shang, Y. Guo & L. Yue (2007), "Bilateration: An Attack-Resistant Localization Algorithm of Wireless Sensor Network," *Embedded and Ubiquitous Computing*, Berlin Heidelberg: Springer, 4808, pp.321-332.
- Liu, Y. & Z. Yang (2011), *Location, Localization, and Localizability: Location-Awareness Technology for Wireless Networks*, New York: Springer.
- Locher, T., P. Rickenbach & R. Wattenhofer (2008), "Sensor Networks Continue to Puzzle: Selected Open Problems," *Proceedings of the 9th International Conference on Distributed Computing and Networking*, Kolkata, India, Springer-Verlag, pp.25-38.
- Mao, G., B. Fidan & B. D. O. Anderson (2007), "Wireless Sensor Network Localization Techniques," *Computer Networks*, 51 (10), pp.2529-2553.
- Moore, D., J. Leonard, D. Rus & S. Teller (2004), "Robust Distributed Network Localization with Noisy Range Measurements," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, ACM, pp.50-61.
- Niculescu, D. & B. Nath (2003), "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, 22 (1), pp.267-280.
- Savarese, C., J. M. Rabaey & K. Langendoen (2002), "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," *Proceedings of the General Track of the Annual Conference on USENIX Annual Technical Conference*, USENIX Association, pp.317-327.
- Savvides, A., C.-C. Han & M. B. Strivastava (2001), "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, ACM, pp.166-179.
- Shang, Y., W. Ruml, Y. Zhang & M. Fromherz (2004), "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, 15 (11), pp.961-974.
- Yang, Z. (2010), *Localization and Localizability in Sensor and Ad-Hoc Networks*, Published Doctoral Dissertation, The Hong Kong University of Science and Technology, Computer Science and Engineering.
- Zhu, Z., A. M.-C. So & Y. Ye (2010), "Universal Rigidity: Towards Accurate and Efficient Localization of Wireless Networks," *INFOCOM 2010 Proceedings of the 29th Conference on Information Communications*, San Diego, CA, March 2010, pp.1-9.