國　立　交　通　大　學

工業工程與管理學系

碩士論文

利用舒-瑞氏與秘密分享演算法進行公正的資源分配

Justified Resource Allocation through Solimosi-Raghavan and Secret-Sharing Algorithms

研究生：劉思宇

指導教授：梁高榮　博士

中華民國一百年六月

利用舒-瑞氏與秘密分享演算法進行公正的資源分配

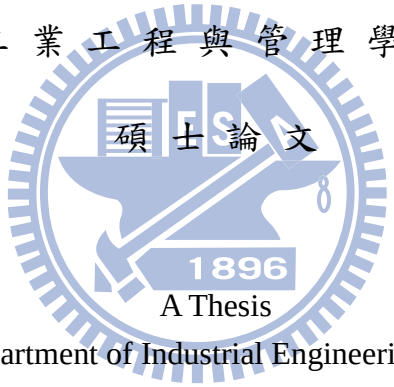
Justified Resource Allocation through Solimosi-Raghavan and Secret-Sharing Algorithms

研 究 生：劉思宇　　Student：Szu-Yu Liu

指導教授：梁高榮　　Advisor：Gau-Rong Liang

國 立 交 通 大 學

工 業 工 程 與 管 理 學 系



碩 士 論 文

A Thesis

Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Industrial Engineering and Management

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年六月

利用舒-瑞氏與秘密分享演算法進行公正的資源分配

研究生：劉思宇　　指導教授：梁高榮　博士

國立交通大學工業工程與管理學系

摘要

本文設計公正拍賣系統來作為指派賽局的公正資源分配機制。它的作業流程由前後台的競價步驟組成。在前台部份，競價者透過密碼學裡的秘密分享技術來傳遞無線電標單以避免網路收標單位的可能圍標行為。這實作的軟體模組則稱為無線競價系統。在後台部份，本文選用著名的舒-瑞氏演算法來計算指派賽局的斂核。這實作的軟體模組則稱為斂核計算系統。透過無線競價系統及斂核計算系統，競價者可以容易地透過行動裝置競價並且根據斂核計算系統所求得的公正價格支付標金。

關鍵字：
公正拍賣(Justice-based Auction)
斂核(Nucleolus)
無線競價(Wireless Bidding)
秘密分享技術(Secret-Sharing Technology)
舒-瑞氏演算法(Solymosi-Raghavan Algorithm)

Justified Resource Allocation through Solimosi-Raghavan and Secret-Sharing Algorithms

Student: Szu-Yu Liu Advisor: Dr. Gau-Rong Liang

Department of Institute of Industrial Engineering & Management
National Chiao Tung University

Abstract

A justice-based auction system has been designed as a fair resource allocation mechanism for assignment game. Its operational procedure consists of competitive steps both at front end and at back end. At the front end, the bidders submit their wireless bids through a secret-sharing procedure in cryptography for avoiding potentially corrupted behavior from their bid receiver on network. Its implemented software package is called Wireless Bidding System (WBS). At the back end, a well-known Solimosi-Raghavan algorithm is chosen to compute the nucleolus of the given assignment game. The implemented software package is named Nucleolus Computing System (NCS). Through the WBS and NCS, the bidders can easily compete with mobile phones and pay with fair prices according to the bidding results.

Keywords:
Justice-based Auction, Nucleolus, Wireless Bidding, Secret-Sharing Technology, Solymosi-Raghavan Algorithm.

誌謝

本論文的完成，首先要感謝我的指導教授梁高榮博士。無論是在學術上的教導或是人生道理上的教誨，都對於我未來的人生有著正面的影響。也感謝梁老師孜孜不倦的督促我的研究進度及論文方向，讓我能夠順利地完成此篇論文。此外，也要感謝我的口試委員周嗣文教授、唐麗英教授和陳文智教授，於口試時給予我相當多寶貴的建議，讓我的論文更臻完整。

在碩士班的兩年生活中，還要感謝實驗室的同伴，包括學長姐黃柏勳、鄭瑋廷、魏良縈、顧佳樺及何青樺；同儕鄭仲元、連惠珍、彭思瑜和朱明典；學弟妹徐宏智、林哲慧、方本欣、張舜龍及張弘。和大家一起歡笑、一起討論課業，彼此互相激勵學習，讓我的研究所生涯更加地多采多姿。也感謝交通大學提供了這麼好的求學環境，眾多的優秀人才及豐富的研究資源，都讓我能夠更紮實的學習。

最後我要感謝我的家人，無論是在精神上或是物質上給予我的支持，讓我可以無後顧之憂下完成學業。今日所有的成長，希望與你們一起分享。

本文為國科會計畫『頻譜規劃配置與管理基礎研究—子計畫二：無線頻譜執照釋出的機制設計』(編號 99-2219-E-009-002)的研究成果，特此致謝。

劉思宇

謹誌于交通大學

2011 年 6 月

目錄

圖目錄

表目錄

第一章 緒論

本章的主旨在於說明本篇論文的研究方向及目的，而在內容部份，則共分為四節。第 1.1 節「研究動機」敘述本篇論文的研究動機。第 1.2 節「問題界定及研究目的」則對於研究的問題做出界定及說明此研究的目的。第 1.3 節「研究方法」提出進行本研究所用到的方法。第 1.4 節「論文架構」對於本篇論文做完整的說明。

1.1 研究動機

在現今的社會，隨著世界人口愈來愈多，地球的資源也正一步步地減少，如何分配資源成為相當重要的課題。資源分配除了指自然資源，如：水、礦產、石油及木材等，亦包含各式的產權，如：土地權、房屋權及執照權等。這些產權雖然是無形的資產，但亦可稱之為是一種資源。在產權移轉方面，必須做到在公平、公開且公正的環境下進行。臺灣近年產權轉移中，交通部第三代行動通訊執照的發放即是相當重要的案例。目前，臺灣正進行「第 11 梯次調頻廣播電台執照開放」的產權移轉，並由國家通訊傳播委員會(National Communications Commission, NCC)[22]負責執行此計畫分配電台執照。

有關無線執照的發放可以參考美國的演進，對美國而言，其無線執照的發放共經歷過四個時期，如表 1.1 所示。

表 1.1 美國無線執照發放的重要時期[4]

制度		發展時期	法源根據
優先權制度		1912 年至 1927 年	無線電法案(1912 年)
聽證會制度		1927 年至 1984 年	無線電法案(1927 年) 通訊法案(1934 年)
抽籤制度		1981 年	預算案
拍賣制度	非廣播執照	1994 年	預算案(1993 年)
	電台及電視執照	1997 年	平衡預算案(1997 年)

若以世代而言，美國經歷了四個世代。第一代為優先權制度，在早期美國的無線執照主要考量新聞自由的因素，因此對於執照發放採用優先權制度。第二代為聽證會制度，聽證會制度希望透過聽證會來評鑑申請執照的廠商，然而卻導致聽證會召開過度頻繁，勞民傷財。第三代抽籤制度則於 1981 年雷根總統時代推行，但抽籤制度同樣面臨了審查申請書多達四十萬份之多的窘境。因此，至 1993 年柯林頓總統上台後開始推行第四代拍賣制度，其成效相當顯著，在 1994 年七月至 1998 年七月間，美國進行了 16 場拍賣並賣出 5893 張執照，獲利高達 229 億美金。

在臺灣，政府於民國 91 年進行的第三代行動通訊執照發放亦使用了拍賣制度來進行，該次執照發放透過了作業研究技術中的逆向匈牙利人法及中國郵差法(Chinese Postman Algorithm，CPA)來完成[3]。由於管理規則及制度設計得宜，不僅消除了得標者詛咒(Winner's Curse)、失標者詛咒(Loser's Curse)及拍賣者詛咒(Auctioneer's Curse)，該次執照

發放更為國庫進帳 488.99 億元台幣。

在拍賣制度上，可以分為單邊拍賣(One-sided Auction)及雙邊拍賣(Two-sided Auction)兩種。所謂單邊拍賣指的是在拍賣交易市場中，僅由買方對競價物進行競價，這種拍賣制度常見於農產品批發市場中。雙邊拍賣則是指在拍賣交易市場中，買方及賣方皆可競價，在股票、期貨及選擇權的交易所便是使用雙邊拍賣制度。在單邊拍賣制度中，可再細分為順序拍賣(Sequential Auction)、同時拍賣(Simultaneous Auction)及組合拍賣(Combinatorial Auction)[28]三種。順序拍賣指的是將拍賣物依照順序獨立拍賣；同時拍賣則是指同一時間競價者可以競價多個競價物；至於組合拍賣的主要精神在於可以將競價物以群組的方式進行競價。2010 年，顧佳樺[14]以組合拍賣方式來求解臺灣調頻廣播執照釋出的問題。同時拍賣的優點在於加快拍賣速度，其實際應用例如臺灣花卉批發市場使用的拍賣方式便是多線同時拍賣，交通部第三代行動通訊執照所使用的拍賣方式則為同時、上升及多回合競價方式[1]。在同時拍賣賽局中，可以利用核的觀念找出拍賣的合理價格，亦稱為核裡價格(The Imputation of Core)，也就是經濟學裡的柏瑞圖最佳化(Pareto-optimal)[6]結果。而本論文對於同時拍賣賽局，進一步利用斂核的觀念來令合理集合內所有利潤團體中，滿意度最小者的滿意度最大化。圖 1.1 為單邊拍賣分類示意圖。

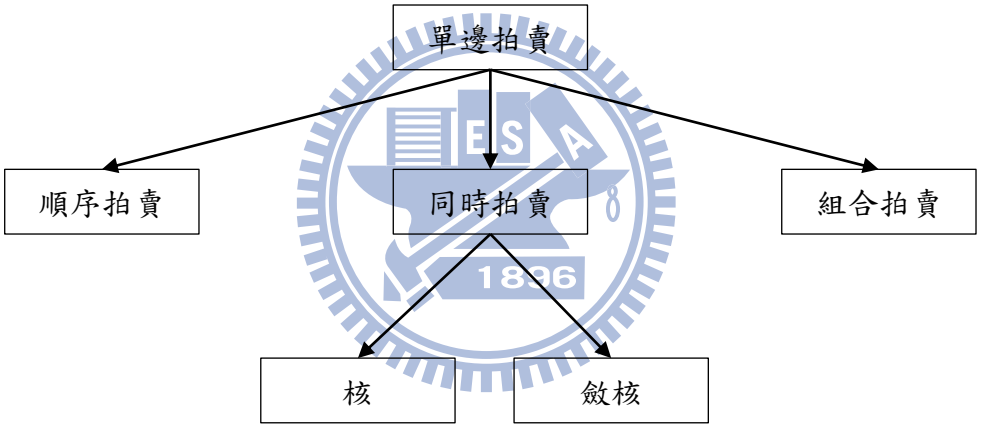


圖 1.1 單邊拍賣分類[13]

由美國的無線執照發放及交通部第三代行動通訊執照發放的經驗可以得知，拍賣制度是可行且成效顯著的一項制度。拍賣制度兼具了公平及公開的兩項優點，然而對於得標金額卻不盡然是令所有人皆滿意的結果。

因此，本論文希望透過核及斂核的觀念，計算出拍賣結果的核裡價格，並結合勞爾斯(J. Rawls)教授所提出的正義論(A Theory of Justice)[30]，即令團體中最小利潤者的利潤最大化，做出容易執行的公正資源分配。拍賣制度搭配斂核的觀念，將可成為第五代的執照發放制度。

1.2 問題界定及研究目的

臺灣自民國 82 年開始，歷經 10 梯次的調頻廣播電台執照發放，其審查制度皆採用審查制度。第 11 梯次調頻廣播電台執照開放則有鑑於第三代行動通訊執照發放的成功，因此在民國 99 年國家通訊傳播委員會通過廣電法修正草案，將廣電事業的執照發放方式增為審議制及競價制，此梯次的調頻廣播電台執照發放將以審議及拍賣制並行發放。

臺灣在第三代行動通訊執照發放的技術，利用了逆向匈牙利人法及中國郵差法，下方圖 1.2 為其理論與實務的應用。

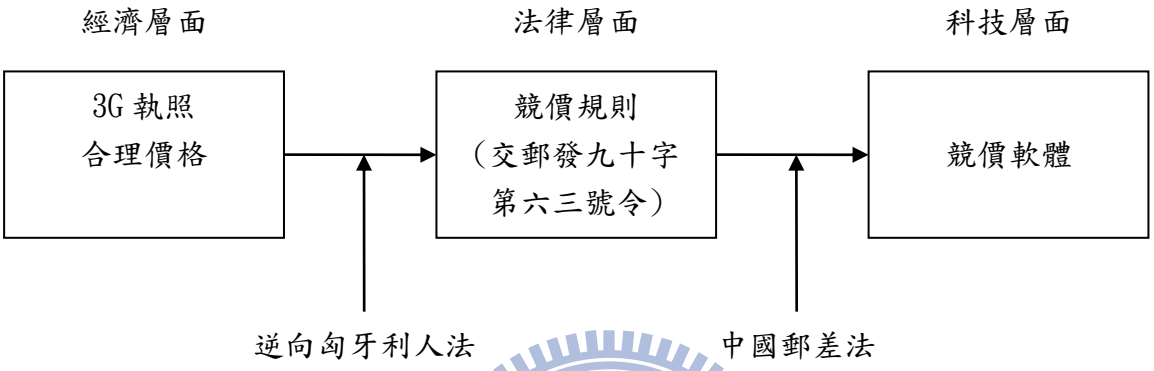


圖 1.2 從理論到實務[3]

調頻廣播電台執照發放可看成一個產權指派賽局，在以拍賣制度做為發放執照依據的前提下，本研究針對產權指派賽局的拍賣制度，求出其斂核解，使得在此賽局的所有參與者中其滿意度最小者滿意度最大化，以達到公正的資源分配。

求解斂核解的部份，利用舒里莫西(T. Solymosi)教授及瑞格曼(T. E. S. Raghavan)教授[35]所提出的指派賽局斂核演算法(以下稱舒-瑞氏演算法，Solymosi-Raghavan Algorithm)來求解，並將演算法以爪哇語言(Java Language)實作出來，以達到快速計算斂核解的目的。計算結果部份，則使用資料庫(Database)技術將所得之結果存放於資料庫中，並利用 JavaServer Pages (JSP)網頁設計技術將結果利用網頁呈現。

此外，本研究利用網際網絡方式來進行競價的動作。1999 年，李唯爾[2]與許春田[8]就曾經分別設計了雞蛋網路投標系統及雞蛋網路開標系統，以改善當時雞蛋價格訂定的弊端。李氏與許氏所設計的系統屬於雙邊拍賣中的標單制，其特色是先交易後決定價格。

2003 年，許鈞豪[9]針對本地英國式拍賣以舒-瑞氏演算法在產權指派賽局做應用，利用多回合競價的方式模擬拍賣進行，並計算拍賣結果的斂核價格。

本研究在拍賣方式部份，則以單邊拍賣的遠距投標做為拍賣進行方式，競價者在拍賣進行中所出價金額必須保密，當拍賣時間截止時，由拍賣主辦單位統一進行開標動作，並宣布得標者。圖 1.3 為許鈞豪、李唯爾、許春田及本研究拍賣方式示意圖。

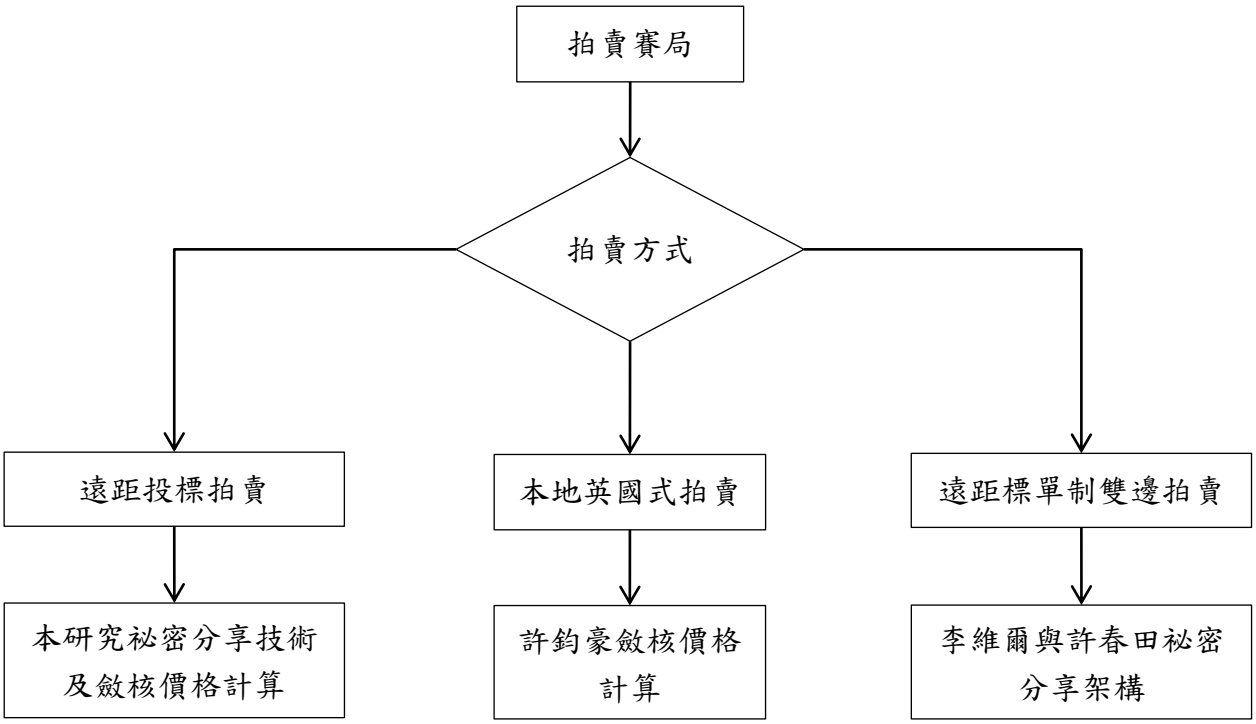


圖 1.3 拍賣方式示意圖

在現今社會，網路發展已相當普遍，網路連線速度亦大幅提升，利用網路進行物品拍賣已相當普遍。然而網際網絡最重要的一環就是傳輸資料的保密性，因此，在本研究中使用秘密分享技術(Secret-Sharing Technology)[33]來進行資料加密的動作，確保競價過程不會發生資料外洩的情況。

本研究的目的在於建構一套公正拍賣系統網絡，此系統以網際網路投標服務來進行拍賣。透過無線競價系統的輔助，參與者於網路投標後，資料將以秘密分享的方式儲存，於競價結束後，利用所開發出的斂核計算系統來快速地求得拍賣得標金額及公正價格。本研究的成果可以做為日後第 11 梯次調頻廣播電台執照開放的參考，利用本研究中建構的系統模擬計算執照拍賣的結果，並應用於更各大型的拍賣計畫中。

1.3 研究方法

本研究的主要研究方法，在於利用爪哇語言來實作出無線競價系統與敘核計算系統。利用爪哇語言可以開發出跨平台的系統程式。在網路資料儲存的部份，則利用秘密分享技術來確保資料的保密性。本研究方法的流程大致可分為五個階段，如下所示：

第一階段：『確認研究目的與範圍』進行文獻探討以決定所使用的學理與技術
第二階段：『舒-瑞氏演算法及秘密分享回顧』將演算法與秘密分享技術運用於系統
第三階段：『秘密分享技術應用』實際開發出無線競價系統
第四階段：『舒-瑞氏演算法應用』實際開發出敘核計算系統
第五階段：『系統實作及測試』實際操作並分析系統績效。

下圖 1.4 為本論文研究方法之流程。

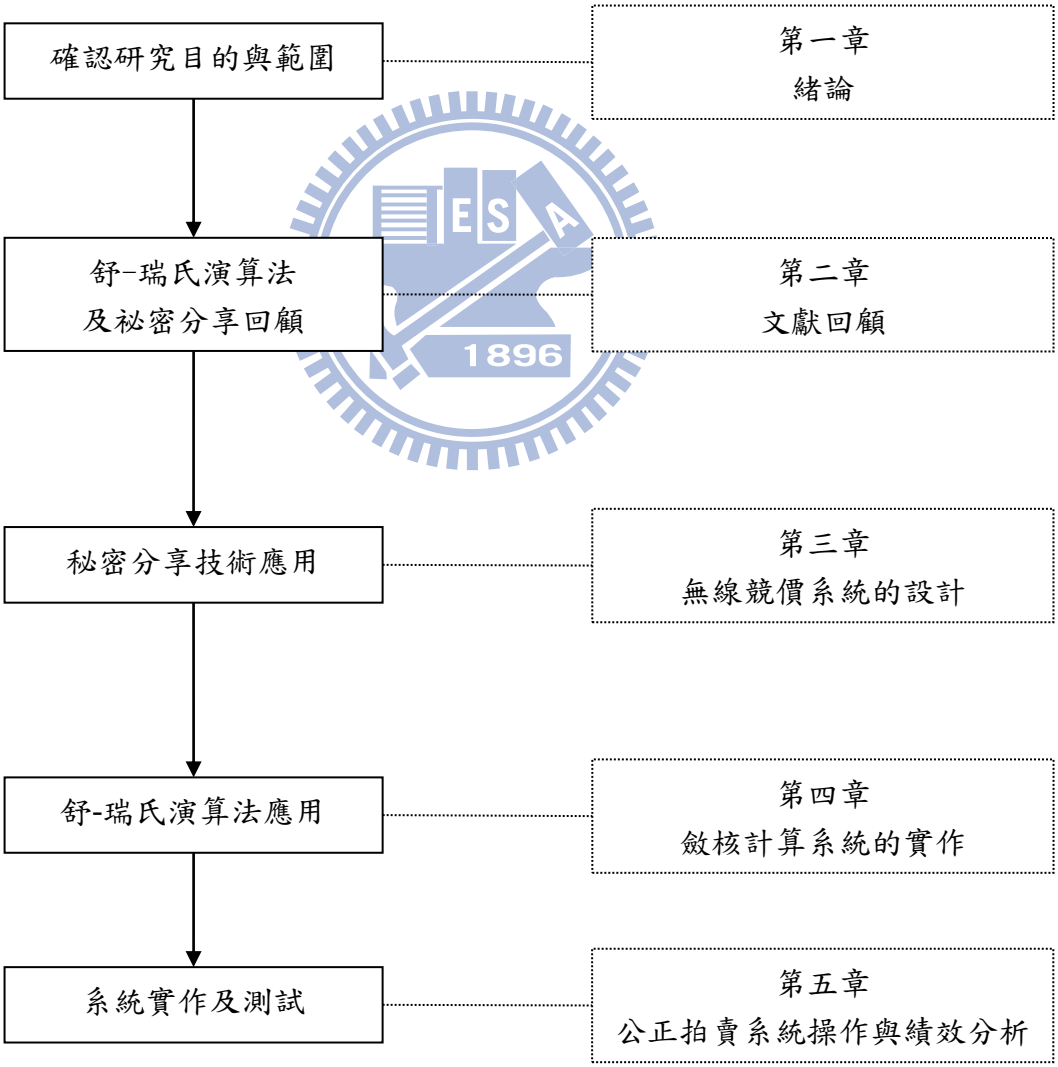


圖 1.4 研究方法流程

1.4 論文架構

本論文的內容共分為六個章節，其內容大綱如下所示。

第一章：『緒論』主要介紹本研究的研究動機、問題界定、研究目的、研究方法及論文架構的部份。

第二章：『文獻回顧』在於探討正義理論、核與敘核的觀念、舒-瑞氏演算法的內容、迴徑判斷、秘密分享的理論及爪哇語言。

第三章：『無線競價系統的設計』介紹秘密分享技術加密及解密的方式。

第四章：『敘核計算系統的實作』利用爪哇語言將舒-瑞氏演算法實作於系統中，用以快速求得敘核解。

第五章：『公正拍賣系統操作與績效分析』將實際操作無線競價系統與敘核計算系統，並分析其所呈現的績效表現。

第六章：『結論』說明本研究的研究結果及未來可以繼續研究的方向。

下方圖 1.5 為本論文中，公正拍賣系統網絡的架構圖。

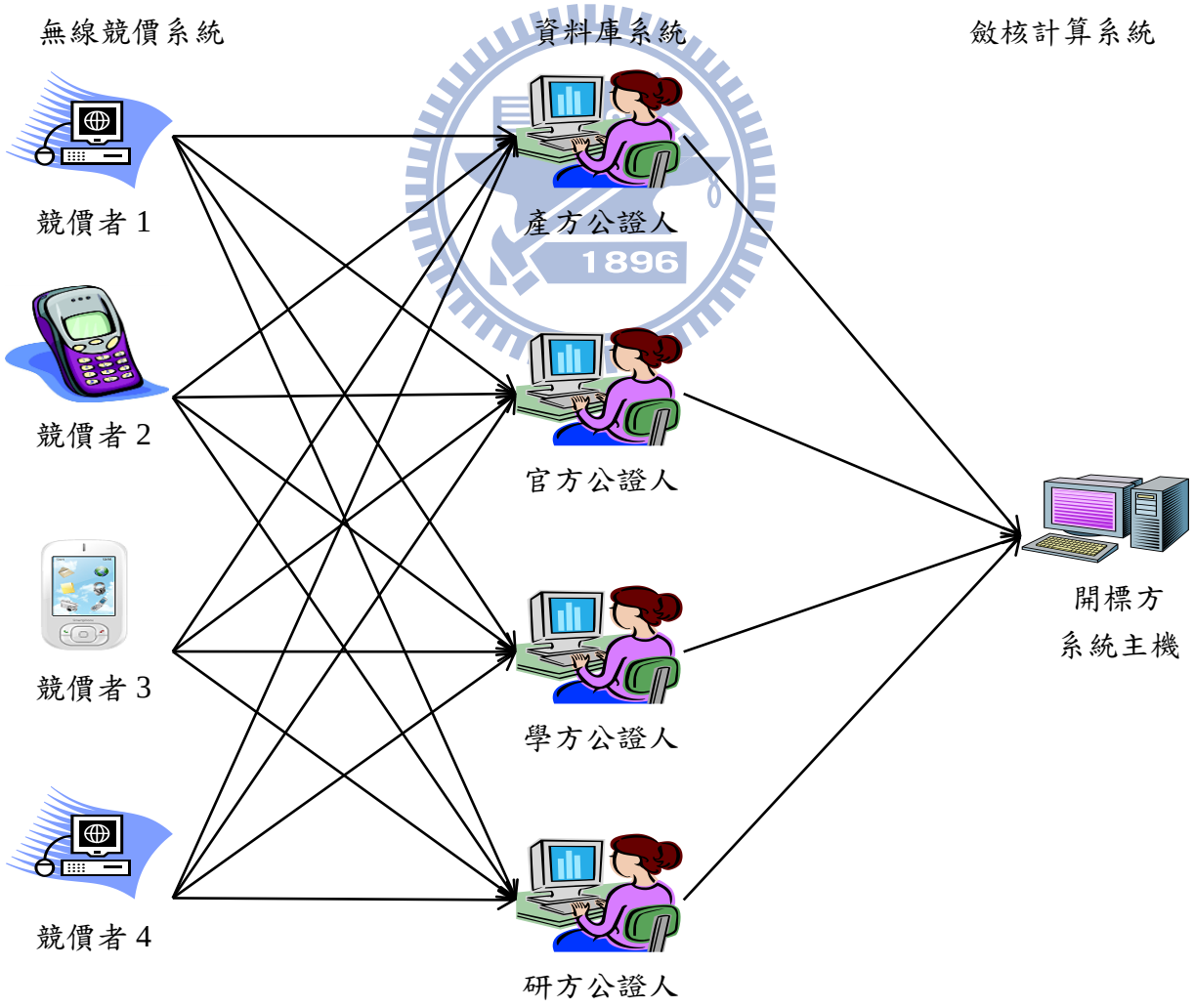


圖 1.5 公正拍賣系統網絡架構圖

第二章 文獻回顧

本章對本研究的相關文獻及背景作一回顧與介紹。第 2.1 節說明賽局理論中核與斂核的觀念，並說明公正的意義以及公正的分配。第 2.2 節說明舒-瑞氏演算法的原理。第 2.3 節說明在有向圖中找出迴徑的演算法。第 2.4 節介紹秘密分享技術。第 2.5 節則說明 J2ME 技術。

2.1 賽局的斂核與公正分配

賽局理論是馮紐曼(J. von Neumann)教授[36]於 1944 年提出。在賽局中，所有參與者所組成各種可能的配對稱為利潤團體(Coalition)，而賽局理論的主要目的在說明利潤團體的互動關係及其產生的經濟活動。

在第 2.1.1 節將針對賽局的核做說明。第 2.1.2 節則探討賽局中的斂核。第 2.1.3 節說明結合斂核及正義的想法而達到公正的分配。

2.1.1 賽局的核

賽局理論中，核(Core)是一個重要的觀念，吉利斯(D. B. Gillies)教授[18]將利潤團體的核定義為任何利潤團體的利潤分配皆亞於團體內所有成員的利潤和。在一個具有 n 個參與者的賽局中，共有2^n種可能的利潤團體組合，假設 S 為某個利潤團體，則其合作的利潤可以用操作特性函數(Characteristic Function)[32]表示為 v(S)，而x_i代表參與者 i 在此賽局中所得的利潤，所有利潤總和為υ(N)，其中 N 表示所有參與者的集合，C 表示所有合理分配所成的集合即核，吉利斯教授證明了以下定理：

定理：$C \Leftrightarrow \sum_{i\in S} x_i \geq v(S), \sum_{i\epsilon N} x_i = v(N), S \subseteq N$。

上方定理說明當賽局解進入核時，任何利潤團體 S 在此賽局所得到的個別利潤和，必定大於等於此利潤團體的單打獨鬥的利潤υ(S)；此外，所有參與者的個別利潤和必須等於總利潤和υ(N)，即表示利潤的總和是穩定的。

在一場拍賣交易中，合理分配所有賣方及買方的利潤可利用核的觀念來達成，當拍賣結果所得到的價格是所有參與者皆可接受的價格時，此時的價格稱為核裡價格，也可說明為所有參與者皆可接受的合理價格。

2.1.2 賽局的斂核

在第 2.1.1 節中可以很簡單地看出，利用核的觀念所得到的核裡價格其實是一個解集合，也就是此解可能為無解、有唯一解及有多重解等可能。因此，核並非一定存在，而這使得核的觀念在應用上並非如此便利。1969 年，修麥樂(D. Schemeidler)教授[32]提出斂核(Nucleolus)的觀念，對於一個有 n 個參與者的賽局而言，利潤向量(Payoff Vector)[15]

$$x = (x_1, x_2, \dots, x_n), x_i \geq 0, i = 1, 2, \dots, n, x(N) = v(N)$$

即對於利潤向量內的各參與者而言，其在此賽局所得到的利潤恆大於等於零，也就是沒有任何人的利潤會損失。對於任意的利潤團體 S 而言，$x(S) \equiv \sum_{i \in S} x_i$。作者定義抱怨函數(Excess Function)為

$$\theta(x) = v(S) - x(S)$$

其中，$\theta(x)$表示利潤團體 S 的抱怨值。上式代表利潤團體的抱怨值是由該團體所創造的價值減去該團體中個別的參與者在此賽局中可得到的利潤總和。當$v(S) \leq x(S)$時抱怨值為負值或零，表示該利潤團體已無人抱怨，即表示利潤團體中的所有人都接受此合理分配。令 Y 表示R^n的子集合，則斂核$\varkappa(Y)$即為 Y 集合的一點，其定義如下所示：

$$\varkappa(Y) = \{x \in Y | \theta(x) \preccurlyeq \theta(Y) \text{for all } y \in Y\}, \qquad \varkappa(v) = n(X)$$

即當求得斂核解時，抱怨值$\theta(x)$最小，換句話說，斂核解就是在找出抱怨值最大者其抱怨值最小化。此外，將抱怨函數加上負號後可得到$f(x) = -\theta(x)$，此$f(x)$稱為滿意度函數(Satisfaction Function)。則斂核解便可解釋為找出滿意度最小者其滿意度最大化，修麥樂教授提出的斂核觀念與勞爾斯教授所提出的正義論其所追求的目標是相同的。因此，斂核可以說是正義的實現，在拍賣交易中，透過求出斂核解便可找出公正的價格。

2.1.3 公正的分配

近幾年，利用拍賣的方式進行產權轉移已相當盛行，探究其原因有二。首先站在社會角度，拍賣具備良好之效率性是其盛行的原因之一。使用拍賣能夠確保拍賣品最終落入事前對其抱持有最高價值的買家手中。換言之，拍賣品將會給予「最想要」的人。再者，由於買、賣雙方之間資訊不對稱，賣方事前無從得知商品在各買方心中的價值。不確定性的存在使得賣方可能因錯估商品價值而損及利潤。藉由拍賣的方式來進行交易則可讓買方在投標過程中主動透露其心中價值給賣方。

然而，無論在何種交易制度下進行的產權轉移行為，都應符合公平、公開及公正原則始能保障買賣雙方之利潤。在拍賣制度中，公平原則指的是公平競爭，也就是每一位競價者都應有相同的得標機會；公開原則指的是交易過程透明化，競價過程中每一位競價者應掌握有相同程度的資訊。公正原則指的則是公正價格的形成。1958 年美國的勞爾斯教授提出正義(Justice)[29]想法，解釋正義即是讓最小利潤者的利潤最大化。1969 年以色列的修麥樂教授則將正義想法量化，其定義出抱怨函數再使得具有最大抱怨值之利潤團體的抱怨值最小化，並將此解稱作斂核。兩者雖然分別從利潤團體的最小利潤最大化及最大抱怨最小化兩種不同觀點解釋正義，但可視為是一體兩面之問題。由於斂核的本質即是正義觀念的實現，因此亦可將斂核解釋為公正的價格，符合斂核的分配即稱為公正的分配。

2.2 舒-瑞氏演算法

敘核的觀念由修麥樂教授於 1969 年提出，其目的在找出合作賽局中的單一解，當核存在時，敘核必定落在核內。沙普利(L. S. Shapley)教授與舒貝克(M. Shubik)教授[34]於 1972 年提出在指派賽局(Assignment Games)中永遠存在非空集合的核，即必定可以找出敘核解。

如何找出指派賽局的敘核解一直是受到關注的課題，1972 年科爾伯格(E. Kohlberg)教授[25]提出加權和法(Weighted-sum Approach)來求解敘核，然而此方法會形成一個極大的線性規劃模型，計算可行性極低；而歐文(G. Owen)教授[27]於 1974 年改進了科爾伯格教授的方法，但在 n 個參與者的賽局中，其高達$2^{n+1}+n$個變數及4^n+1個限制式，當參與者為 3 人時，即產生 19 個變數及 65 個限制式，可行性也不高；近期，桑卡蘭(J. K. Sankaran)教授[31]提出的解法對於 n 個參與者的求解，依然需要$O(2^n)$的時間複雜度進行計算。上述的方法雖可求解敘核，但所需耗費的時間過長。本篇論文選用舒-瑞氏演算法來進行敘核求解，此演算法在一個(m, n)個參與者且$m = \min(m, n)$的賽局中，僅需要$1/2 \cdot m(m+3)$個步驟，且時間複雜度為$O(m \cdot n)$，大大提升了計算敘核的效率。

利用舒-瑞氏演算法來求得敘核必須給定一個指派賽局的初始解，接著計算賽局的滿意度矩陣，反覆更新滿意度矩陣來使得最小滿意度最大化。在第 2.2.1 節舒-瑞氏演算法架構中，首先介紹舒-瑞氏演算法的架構。第 2.2.2 節匈牙利人法，探討匈牙利人法求解最佳指派，以做為舒-瑞氏演算法的初始解。在第 2.2.3 節產權指派賽局應用，則說明舒-瑞氏演算法的應用。

2.2.1 舒-瑞氏演算法架構

舒-瑞氏演算法目的為求解指派賽局的敘核解。對於一個包含至少一位買方及一位賣方的指派賽局，首先找出其最佳指派。根據最佳指派的結果將交易成功後的初始利潤全部歸於賣方，得到一組買賣雙方的利潤向量初始解。由於舒-瑞氏演算法利用利潤向量的改變來使得滿意度最小者其滿意度最大化，透過將最小滿意度最大化的過程來找出此初始解的移動方向及移動距離，以提升最小滿意度並移動至下一個最小滿意度水準。當最小滿意度無法再提升時，則最終的利潤向量就稱為敘核，也就是公正價格。圖 2.1 為滿意度水準更新的示意圖。

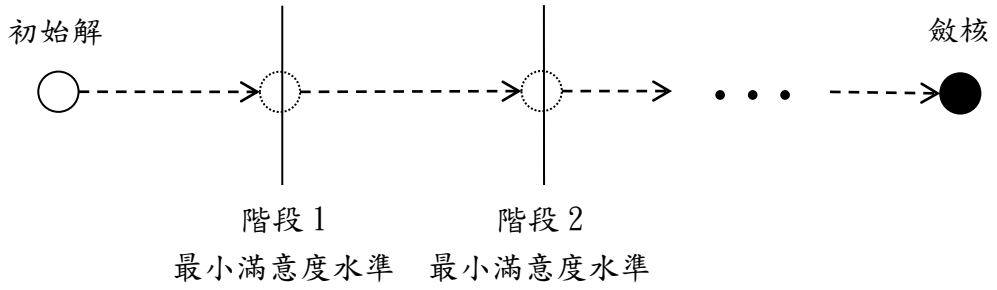


圖 2.1 滿意度水準更新示意圖

在圖 2.1 中，由最佳指派得到一組利潤向量的初始解，透過分配利潤來提升最小滿意度，並移動到階段 1 的最小滿意度水準。此時再次分配利潤以提升最小滿意度至階段 2 的最小滿意度水準。反覆此過程直到無法更新最小滿意度至下一階段最小滿意度水準後，表示已移動到最終斂核解。

舒-瑞氏演算法用於指派賽局求解斂核的流程可分為五個步驟。第一步驟，將各賣方出價的標單資料整理為標單矩陣(Bid Matrix)。第二步驟，將標單矩陣轉換為利潤矩陣(Profit Matirx)，以簡化運算。第三步驟，利用匈牙利人法由第二步驟中得到的利潤矩陣，求出最佳指派。第四步驟，求出最佳指派後，計算出由各個買方及賣方所組成的利潤團體，其利潤團體滿意度，並且整理為滿意度矩陣(Satisfaction Matrix)，滿意度矩陣的建立在第 4.3 節做說明。第五步驟，檢查滿意度是否有增加空間，當滿意度仍可增加時，表示最小滿意度尚未最大化，此時找出滿意度增量，並且再次回到第四步驟重新計算滿意度矩陣；而當滿意度已無法再增加時，此時最小滿意度以達到最大化，即求得斂核解。舒-瑞氏演算法流程如所示。

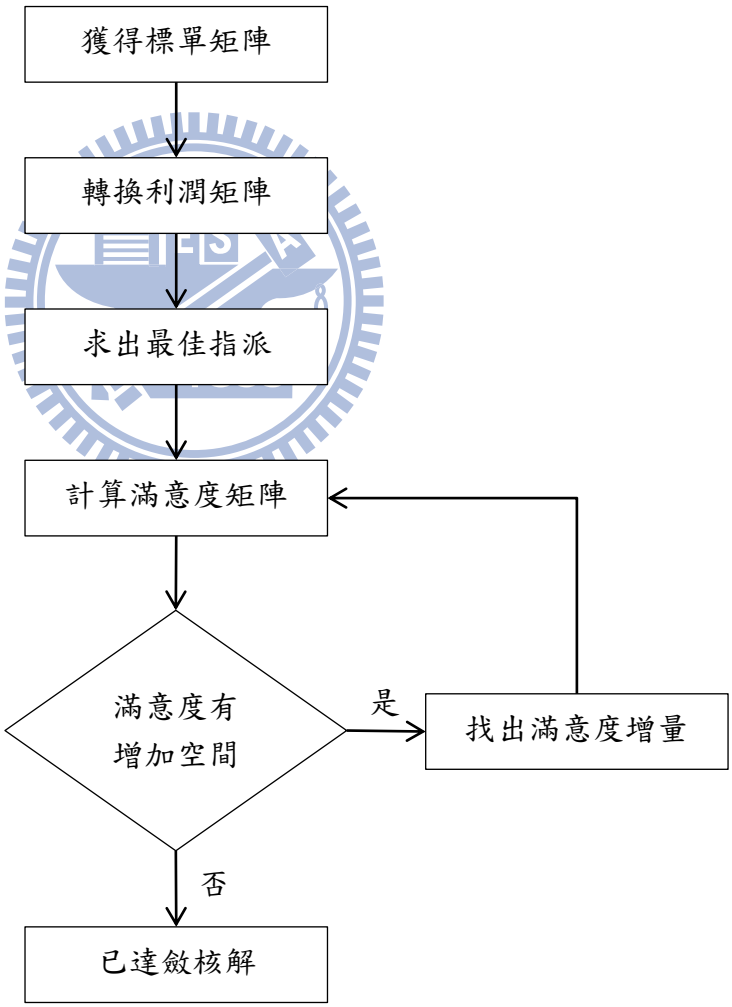


圖 2.2 舒-瑞氏演算法流程

2.2.2 匈牙利人法

在本研究中，利用匈牙利人法(Hungarian Method)[26]來求解指派問題中的最佳指派，匈牙利人法的名稱是為了紀念匈牙利的兩位數學家 J. Egervàry 及 D. König 而命名，由於匈牙利人法多以計算最小成本的問題呈現，因此當原始問題為求最大利潤時，則可轉為極小值問題再行計算。匈牙利人法的計算流程如圖 2.3 所示。

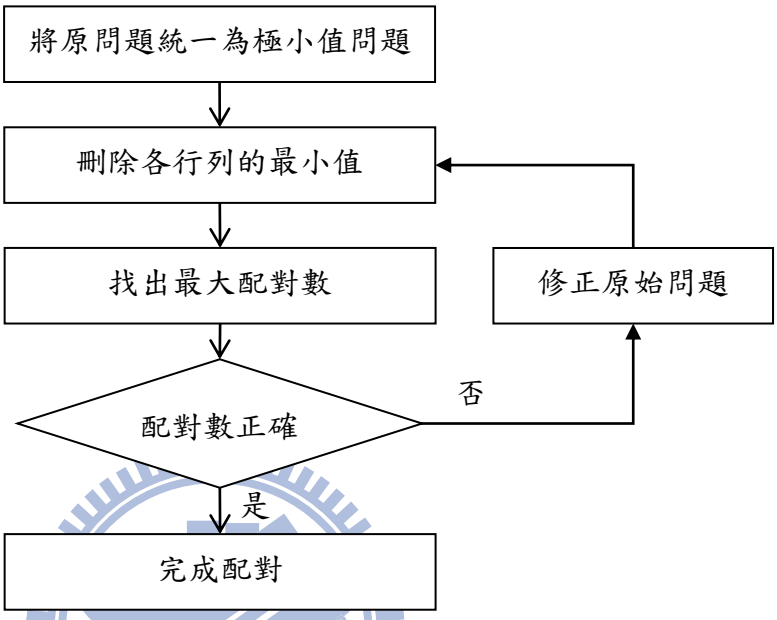


圖 2.3 匈牙利人法計算流程

步驟 1 轉為極小值問題目的在於方便匈牙利人法的進行，當原始問題為求解極大值問題時，則可找出矩陣中的最大值，並利用此最大值個別減去矩陣中所有元素形成新矩陣，此時極大值問題便可轉為極小值問題。步驟 2 先減去各列最小值，接著再減去各欄最小值以形成新的矩陣，此時便可利用矩陣中元素值為零的部份進行配對。步驟 3 為找出最大配對數，在一個指派問題中，買賣雙方的人數必須一致，當某一方人數不足時，則需添加虛擬方來達成方陣的形式。步驟 4 判斷最大配對數是否等於買方等於賣方，若成立時，此時即找出此指派問題的一組最佳指派解，若最大配對數小於目標值時，則刪除零值的行及列並再次重複步驟 2 直至找到目標值為止。

表 2.1 利潤矩陣(單位：萬元)

產品編號	買方 A	買方 B	買方 C	買方 D
S1	270	210	40	280
S2	290	20	60	80
S3	200	160	0	20

表 2.1 的利潤矩陣是一個三物四人的指派賽局。以表 2.1 來說明匈牙利人法求解最佳指派的過程。由於利用匈牙利人法求解指派賽局必須是一個方陣，因此，新增一個虛擬的拍賣物 S4 來形成一個四乘四的方陣，虛擬拍賣物不會和任何買家交易並產生利潤，因此其利益為零。增加虛擬拍賣物後的擴充利潤矩陣如表 2.2 所示。在得到擴充利潤矩陣後，首先，利用機會成本的概念，將利潤矩陣中最大數字 290 減去各欄位的數字，轉換為成本矩陣，此時指派賽局總利潤最大化解就變成總成本最小化的最佳解，如表 2.3 所示。

表 2.2 擴充利潤矩陣(單位：萬元)

產品編號	買方 A	買方 B	買方 C	買方 D
S1	270	210	40	280
S2	290	20	60	80
S3	200	160	0	20
S4(dummy)	0	0	0	0

從表 2.3 的成本矩陣中，將每一列減去該列最小值，再將每一行減去該行最小值，形成新的成本矩陣，修改成本矩陣後，利用最少的直線數來覆蓋所有為零的數字，當最少直線數等於列數時，便可找出最大配對數及最佳配對。

表 2.3 成本矩陣(單位：萬元)

產品編號	買方 A	買方 B	買方 C	買方 D
S1	20	80	250	10
S2	0	270	230	210
S3	90	130	290	270
S4(dummy)	290	290	290	290

由表 2.4 修改後成本矩陣中可以看出，其覆蓋元素值為零的最少直線數為 3，小於總列數 4，因此必須再次修改成本矩陣，增加矩陣零值個數以提高最少直線數。在表 2.4 中，找出未被直線覆蓋元素的最小值，並且將這些元素同減去最小值，而直線交叉部份的元素，則加上此最小值。

表 2.4 修改後成本矩陣(單位：萬元)

產品編號	買方 A	買方 B	買方 C	買方 D
S1	10	70	240	0
S2	0	270	230	210
S3	0	40	200	180
S4(dummy)	0	0	0	0

再次修改成本矩陣後如表 2.5 所示。表 2.5 中，其最少直線數等於列數 4，此時便可利用最大配對數計算法來得到目標為 4 的最大配對數並找出成本矩陣的一組最佳指派。最大配對數計算法分為三個步驟，首先任選一組配對，接著，若初始配對未達最大配對數則修正配對，找出新改進點，最後，利用間隔路徑(Alternating Path)形成新的一組配對，反覆此過程至最大配對數無法再改進。

表 2.5 再次修改後成本矩陣

產品編號	買方 A	買方 B	買方 C	買方 D
S1	10	30	200	0
S2	0	230	190	210
S3	0	0	160	180
S4(dummy)	40	0	0	40

表 2.5 中，成本為零的配對共有六種，根據這六種可能配對，繪製配對圖來找出成本矩陣的一組最佳指派，圖 2.4 配對圖呈現出表 2.5 的六種可能配對。首先，任選一組配對並檢查最大配對數是否為 4 如圖 2.4 粗線部份所示。由於初始配對數即為 4，因此完成最大配對數計算，且其最佳指派為買方 D 得標拍賣物 S1、買方 A 得標拍賣物 S2、買方 B 得標拍賣物 S3，而買方 C 得標虛擬拍賣物 S4，即表示買方 C 沒有完成任何交易。

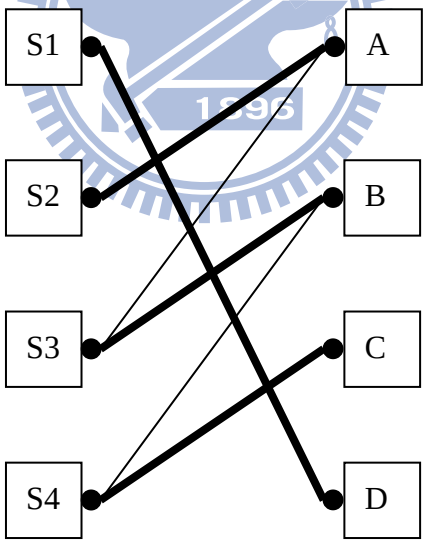


圖 2.4 配對圖

對於匈牙利人法的程式實作部份，則根據[20]中的 Another Java implementation with JUnit tests (Apache 2.0)程式碼修改而來。

2.2.3 產權指派賽局應用

在第 2.2 節中，說明舒-瑞氏演算法可用於求解指派賽局的斂核解。2002 年，許鈞豪、梁高榮及陳宗沂[10]利用舒-瑞氏演算法開發斂核計算軟體。2003 年，許鈞豪[9]設計 NICE 軟體進行模擬來求出產權指派賽局中的斂核解，將舒-瑞氏演算法做實際應用。

本論文同樣以舒-瑞氏演算法求解斂核解，但再以秘密分享理論建構公正拍賣系統網絡。在公正拍賣系統中，透過網際網路的資訊傳輸，以爪哇程式語言建立無線競價系統及斂核計算系統。利用秘密分享技術來加密傳輸的資訊，再使用 JDBC(Java Database Connectivity)技術連結資料庫來儲存所有投標資訊、開標資訊及求解斂核過程的計算數據，以驗證資料正確性。表 2.6 為本論文與許鈞豪的異同處比較。

表 2.6 論文異同比較

相同處	相異處
指派賽局觀念	秘密分享理論
舒-瑞氏演算法運用	網際網路運用
爪哇程式語言撰寫	人機介面無線競價系統
	人機介面斂核計算系統
	資料庫技術運用

2.3 迴徑判斷

在一個有向圖中，判斷圖形是否有迴徑的產生是常常面對的問題。在舒-瑞氏演算法中，必須透過有向圖來找出每回合買賣方利潤的流向，因此必須找出有向圖中是否有存在迴徑的情況，以確保利潤流向不會落入迴徑而造成無止盡的流動。

在第 2.3.1 節及第 2.3.2 節中，分別介紹佛氏演算法及深度優先搜尋演算法，此二種演算法皆可用於判斷圖形內所包含的迴徑。第 2.3.3 節則說明佛氏演算法及深度優先搜尋演算法所適用的圖形。

2.3.1 佛氏迴徑發現演算法

佛洛伊德(R. W. Floyd)教授[17]於 1967 年提出了找出有向圖(Directed Graph)內迴徑的方法，稱為佛氏迴徑發現演算法(以下簡稱佛氏演算法，Floyd's Cycle-Finding Algorithm)。其原理在於利用兩個指標來找尋有向圖中的迴徑，指標 i 每次前進一步，而指標 j 每次前進兩步；該方法又名龜兔演算法(Tortoise and Hare Algorithm)。當指標 i 與指標 j 所到達的節點重疊時，即表示此有向圖有迴徑產生。當指標 i 移動 k 步，指標 j 便移動 2k 步，指標 i 及指標 j 有如龜兔賽跑中的烏龜與兔子般。佛氏演算法的流程如圖 2.5 所示。

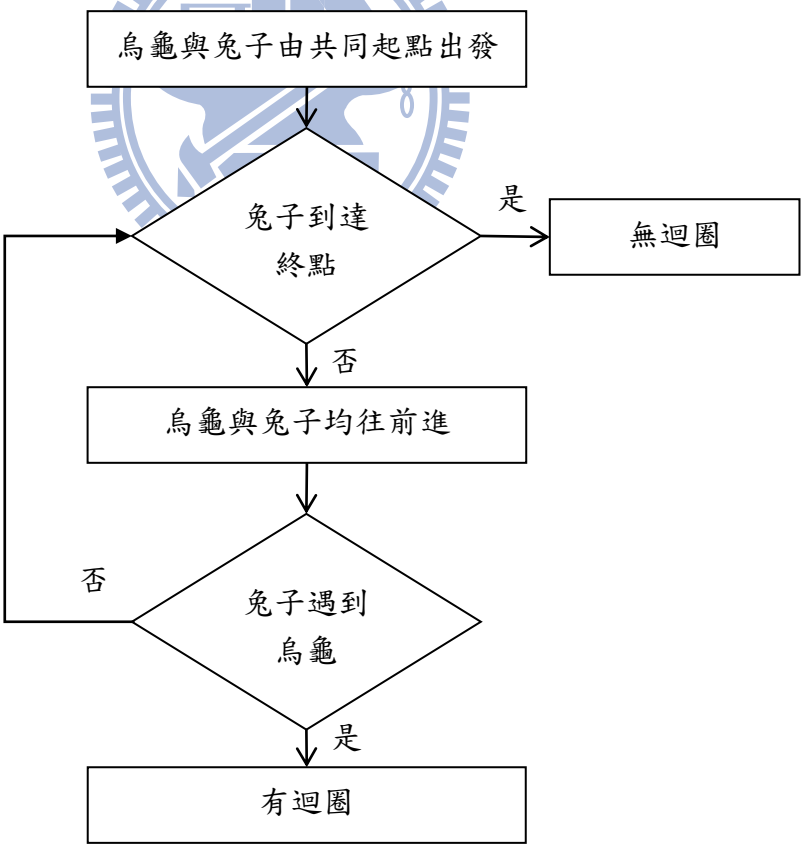


圖 2.5 佛氏演算法流程

為了更詳細說明佛氏演算法的運用，以下以兩個例子說明佛氏演算法。圖 2.6 為一個由三個節點及三個邊組成的奇數節點有向圖，在有向圖中分別有 A、B 及 C 三個節點，此三個節點形成一個迴徑 ABC。

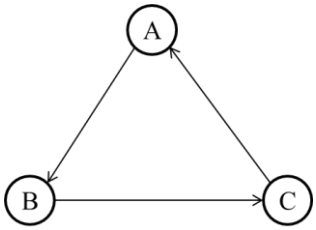


圖 2.6 奇數節點有向圖

假設節點 A 為起點，烏龜 T 及兔子 H 皆從 A 點出發，烏龜 T 一次僅能前進一步，而兔子 H 一次可前進兩步。第一回合，烏龜 T 到達 B 點，兔子 H 到達 C 點，雙方尚未碰面；第二回合，烏龜 T 到達 C 點，而兔子 H 則繞至 B 點，雙方尚未碰面；第三回合，烏龜 T 到達 A 點，兔子 H 經過兩步後，正好也到達 A 點，此時雙方於 A 點同時碰面，表示此有向圖包含迴徑。若繼續進行三回合後，烏龜 T 及兔子 H 會再次於 A 點碰面，因此可知迴徑由節點 ABC 形成。圖 2.7 為奇數節點有向圖迴徑發現示意圖，圖中虛線部份及實線部份分別表示下一回合烏龜 T 及兔子 H 的行走路徑。

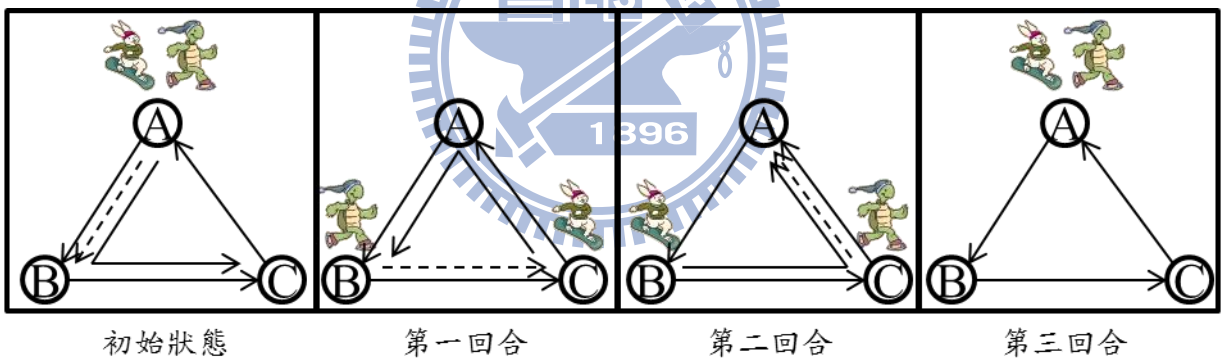


圖 2.7 佛氏演算法於奇數節點有向圖

由圖 2.7 中，可知佛氏演算法可用於判斷由奇數節點組成的迴徑。圖 2.8 為一個由四個節點及四個邊組成的偶數節點有向圖，在有向圖中分別有 A、B、C 及 D 四個節點，此四個節點形成一個迴徑 ABCD。

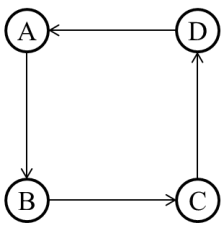


圖 2.8 偶數節點有向圖

一樣假設節點 A 為起點，烏龜 T 及兔子 H 皆從 A 點出發，烏龜 T 一次僅能前進一步，而兔子 H 一次可前進兩步。第一回合，烏龜 T 到達 B 點，兔子 H 到達 C 點，雙方尚未碰面；第二回合，烏龜 T 到達 C 點，兔子 H 到達 A 點，雙方尚未碰面；第三回合，烏龜 T 到達 D 點，兔子 H 則繞至 C 點；第四回合，烏龜 T 繞回到 A 點，而兔子 H 正好也由 C 點前進兩步回到 A 點，此時雙方於 A 點同時碰面，表示此有向圖包含迴徑。若繼續進行四回合後，烏龜 T 及兔子 H 會再次於 A 點碰面，因此可知迴徑由節點 ABCD 形成。圖 2.9 為偶數節點有向圖迴徑發現示意圖，圖中虛線部份及實線部份分別表示下一回合烏龜 T 及兔子 H 的行走路徑。

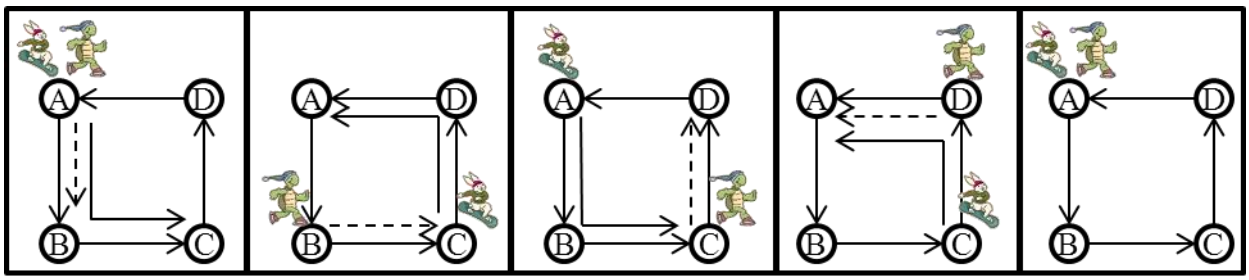


圖 2.9 佛氏演算法於偶數節點有向圖

圖 2.9 的結果顯示佛氏演算法可用於判斷由偶數節點組成的迴徑。由圖 2.6 及圖 2.8 的例子可以得知，無論奇數節點或偶數節點所形成的迴徑皆可使用佛氏演算法來找出迴徑。

2.3.2 深度優先搜尋

深度優先搜尋(Depth First Search, DFS)為圖形理論中，用來遍歷節點的一種方法。一個圖形中，包含節點(Node)及邊(Edge)，而深度優先搜尋可用來走訪圖形中所有的節點，是一種路徑搜尋完整性佳的演算法。

在舒-瑞氏演算法中，利用有向圖來找出利潤的流向及大小，並且判斷當前利潤流向是否正確。在上一節提到，當有向圖出現迴徑時，此時利潤流向會形成一個無窮迴徑，使得利潤無法被計算出來。因此，必須先檢查有向圖是否有迴徑產生，才能找出正確的利潤流向及大小。本論文利用深度優先搜尋演算法來找出有向圖中所包含的迴徑。深度優先搜尋的運算流程，如圖 2.10 所示。

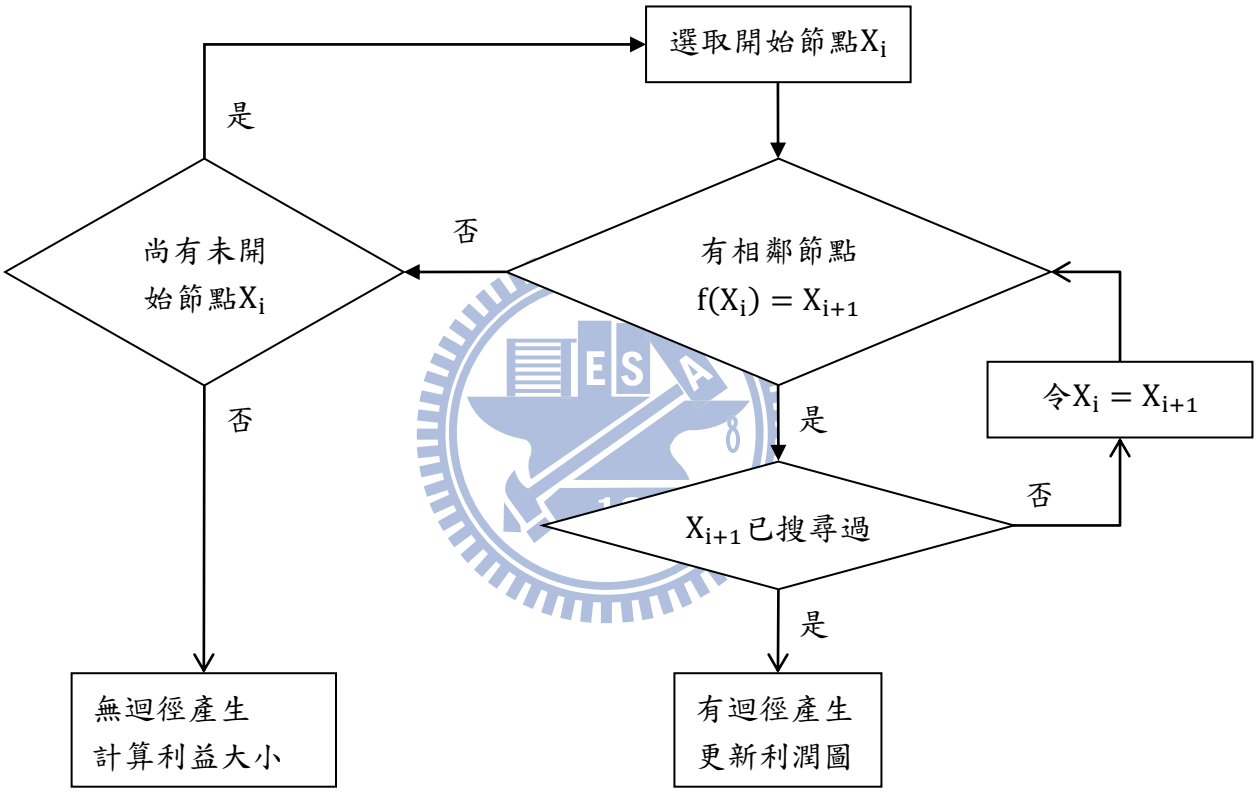


圖 2.10 深度優先搜尋流程

在舒-瑞氏演算法中的有向圖，其不具有展開樹(Spanning Tree)的特性，即某些節點沒有任何進入邊。因此，必須令每一個節點為開始節點，分別進行深度優先搜尋。當搜尋過程中，出現重複節點時，則表示有迴徑產生；此時停止向下搜尋並更新有向圖，圖 2.11 為一個有向圖的範例。

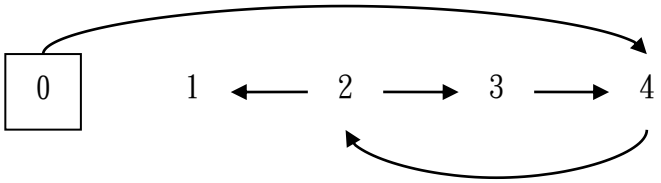


圖 2.11 有向圖範例

在圖 2.11 中，可以很明顯看出節點 2、3 及 4 形成了一個迴徑，而在舒-瑞氏演算法程式中，則利用深度優先搜尋來找出迴徑。首先，將有向圖以樹狀圖方式呈現，圖 2.12 為此有向圖轉成樹狀圖的圖形。

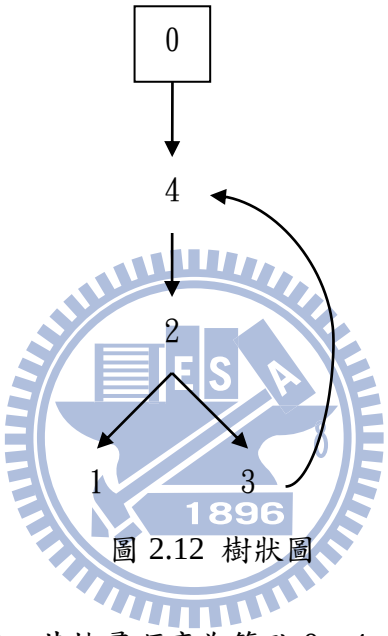


圖 2.12 樹狀圖

此時，進行深度優先搜尋，其搜尋順序為節點 0、4、2、1、3 及 4，當由節點 3 到達節點 4 時，發現節點 4 已經搜尋過，表示有迴徑產生，必須更新有向圖將節點 2、3 及 4 融合為單一節點 2，以消除迴徑；更新後的有向圖如圖 2.13 所示。

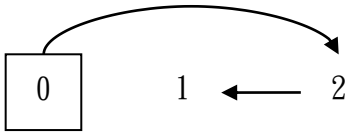


圖 2.13 更新後有向圖

更新後的有向圖已無迴徑產生，此時，便可計算有向圖中的利潤大小。

2.3.3 有向圖迴徑判斷

第 2.3.1 節及第 2.3.2 節分別介紹了佛氏演算法及深度優先搜尋演算法。在迴徑判斷的部份，佛氏演算法在每一步驟進行時，僅需判斷烏龜及兔子所在的節點是否一致，而深度優先搜尋演算法在每一步驟進行時卻必須與先前所搜尋過的所有節點比較。因此，佛氏演算法在判斷迴徑的速度上比深度優先搜尋演算法的速度要來得快。而在路徑的窮舉部份，則以深度優先搜尋來得仔細。然而，佛氏演算法所能面對的圖形必須為功能性圖形(Functional Graph)，即有向圖中，任意一個節點僅能有一個輸出的邊，如圖 2.14 所示。

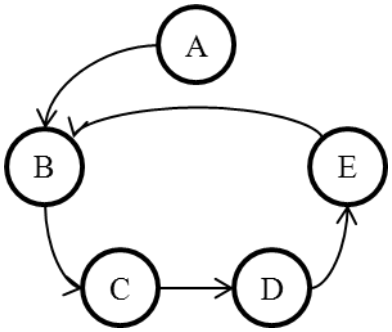


圖 2.14 功能性圖形

在圖 2.14 中，A、B、C、D 及 E 五個節點都只有一個輸出邊，節點 B 雖然有 A 點及 E 點連結來的輸入邊，但仍然保持單一輸出邊的特性。然而，對於舒-瑞氏演算法所使用的有向圖而言，卻可能出現多個輸入邊或多個輸出邊的情形，圖 2.15 為一個舒-瑞氏演算法的典型多輸出邊有向圖。

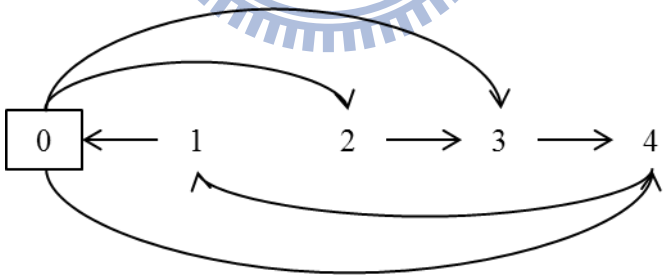


圖 2.15 多輸出邊有向圖

由圖 2.15 可以看出，節點 0、節點 2、節點 3、節點 4 及節點 1 可以形成一個迴徑；節點 0、節點 3、節點 4 及節點 1 也可形成一個迴徑。在節點 0 的部份，擁有兩個輸出邊分別連結至節點 2 及節點 3，此種圖形不符合佛氏演算法的圖形假設，因此無法使用佛氏演算法來判斷迴徑。在本論文中，選擇以深度優先搜尋演算法窮舉出所有路徑來做為有向圖迴徑判斷的方式，對於舒-瑞氏演算法中的有向圖而言，只須找出任意一個迴徑便可以進行更新滿意度的動作，因此即便使用窮舉方式來判斷迴徑，卻可以巧妙地在程式設計時給予條件限制當發現第一個迴徑時，即停止搜尋其他迴徑。如此一來，便可以兼顧佛氏演算法的效率及深度優先搜尋演算法的窮舉特色。

2.4 秘密分享

無線執照的拍賣所影響的金額往往相當龐大，且與國家通訊產業的應用與發展息息相關，因此必須抱持小心謹慎的態度去面對。在網路發達的今天，利用網路進行無線執照的拍賣是絕對可行的事。而在網路傳遞訊息時，其資料的安全性便成為了相當重要的課題。

所謂的秘密分享[33]，是基於金鑰安全管理所發展出來的密碼技術。在現實生活中，保險箱是存放重要物品時常常使用的保存方式，若保險箱僅存在一份金鑰時，一旦金鑰遺失或毀損就會造成保險箱無法開啟，為了防止發生此種情況，最直覺的做法便是備份，也就是重複複製多副金鑰來避免鑰匙遺失或毀損的情況。然而，當同時存在多副金鑰時，資料遭到竊取的風險也會大大的提高。此外，當金鑰保管人欲私吞保險箱內物品時，若金鑰僅為保管人一人擁有，則將有監守自盜的情況發生。因此，能夠避免僅保存一把鑰匙所帶來的風險，又能提供鑰匙備份的功能，便是秘密分享技術所要達成的目的。

秘密分享技術的概念就在於秘密擁有者希望將秘密分享給其他參與者，但每一位參與者皆無法獨自獲得這個秘密，而是當部份或全部參與者聚集時才能解開這個秘密。在一個國家級的博物館中，收藏著許多珍貴的物品，為了避免珍品遭到盜竊，珍品在非展覽時必須放置於保險庫中並上鎖，當能夠為保險庫解鎖的人僅有館長一人，保險庫所需承擔的風險相當高，不僅需擔心館長監守自盜，亦需擔心館長所擁有的金鑰遭到竊取。若利用秘密分享機制來打造金鑰，便可將解鎖金鑰分給數個博物館核心管理階層，開啟保險庫時，必須所有金鑰一起使用，才能打開，如此便可避免上述的風險發生。

隨著網際網路的快速發展，秘密分享有衍伸了相當多的應用問題，秘密分享的應用牽扯到多人共管的運作協調，在本篇論文中，利用秘密分享技術作為整個公正拍賣系統網絡的加密機制。在整個公正拍賣系統網絡中，包含了相當多資訊的傳遞，在主辦單位與公證方間，必須傳遞各競價者的標單拼圖，而公證方與競價者間則須傳遞競價者所分割出的標單拼圖，而主辦單位與競價者間則必須傳遞競價者個人資料，這些資訊的傳遞，都必須使用秘密分享來做加密的動作。下方圖 2.16 為秘密分享的示意圖。

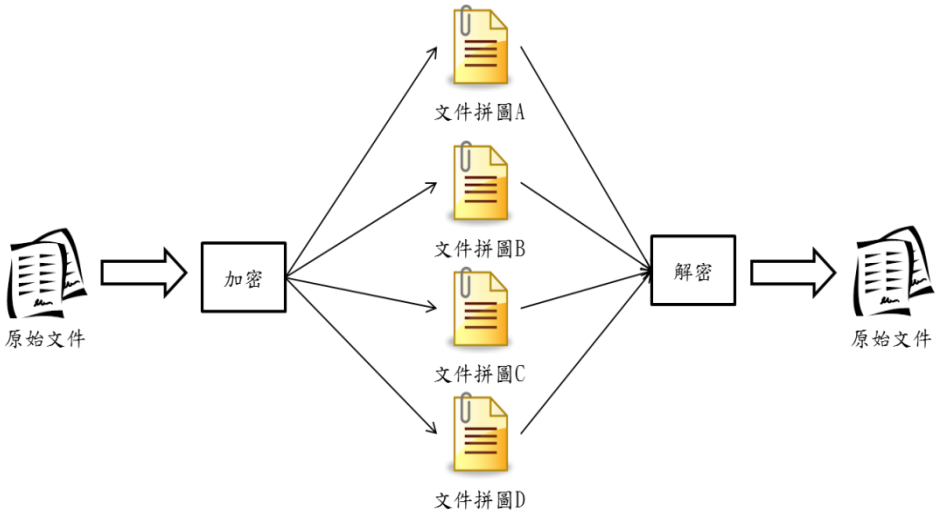


圖 2.16 秘密分享示意圖

秘密分享的技術可擴展到相當廣泛的領域與實務系統，如影像、語音等領域，是理論與實務結合的技術。在第 2.4.1 節將介紹沙氏秘密分享法。第 2.4.2 節是沙氏秘密分享法的應用。

2.4.1 沙氏秘密分享法

在網路進行資訊傳輸時，必須考慮資訊加密的問題，1979 年，沙米爾(A. Shamir)教授於提出一種數學模式來應用秘密分享，稱為沙氏秘密分享法(Shamir's Secret Sharing Method)[33]。沙氏秘密分享法的主要目的，在於利用一個 k-1 次的多項式將原始資料 D 分為 n 個部份，分別為D_1，D_2，…，D_n，而資料 D 可以相當容易地利用其中 k 筆部份資料來還原，此方案必須符合兩個基本條件分別為：

(1) 當具有 k 個以上的部份資料時，便可輕易拼湊出原始資料 D。
(2) 當擁有的資料少於 k 個時，永遠無法將原始資料 D 還原。

滿足上述條件的方案稱為(k, n)門檻方案((k, n) Threshold Scheme)，門檻方案主要建立在內插多項式(Interpolating Polynomial)的觀念上。其原理利用二維座標裡，線上二點可唯一決定一直線，曲線上三點可唯一決定一條二次曲線，曲線上四點則可唯一決定一條三次曲線。由此類推，故僅須獲得原曲線上的部份點集合即可重造原曲線。

假設原始資料 D 為一個數字，今欲將資料 D 分割為 n 個部份，則隨機選定一個 k-1 次多項式

$$q(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1}，令a_0 = \mathrm{D}$$

且計算出

$$D_1 = q(1), D_2 = q(2), \ldots, D_i = q(i), \ldots, D_n = q(n)$$

利用方程式$q(x)$即可將資料 D 分割成 n 個部份。今給定 k 個部份資料D_i值，則可利用內插法找出$q(x)$的係數，並計算出原始資料$\mathrm{D} = q(0)$。

拉氏內插多項式(Lagrange Interpolating Polynomial)的發展可以分為三個階段[24]。1779 年韋靈(E. Waring)首次發表，並在 1783 年由歐拉(L. Euler)重新發現此方法，最終於 1795 年由拉氏(J. L. Lagrange)正式發表，因此將此方法稱為拉氏內插多項式。沙米爾教授便是利用拉氏內插多項式來達到秘密分享的效果。

拉氏內插多項式可於二維座標上，利用 n 個點$(x_i, f(x_i))$值唯一決定 n-1 階多項式。今有 n 個點$(x_1, y_1 = f(x_1))$、$(x_2, y_2 = f(x_2))$、…、$(x_n, y_n = f(x_n))$，則拉氏內插多項式為

$$P(x) = \sum_{j=1}^{n} P_j(x)，且P_j(x) = y_j \prod_{\substack{k=1 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k}$$

將上式展開後可得

$$P(x)=\frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)}y_1+\frac{(x-x_1)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)}y_2+\cdots+\frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})}y_n$$

今在二維座標上選擇五點分別為$(-2,9)$、$(-1,0)$、$(0,1)$、$(1,0)$及$(2,9)$，根據拉氏內插多項式將此五點代入，以求得其所決定的曲線函數

$$\begin{aligned}f(x)&=\frac{[x-(-1)](x-0)(x-1)(x-2)}{[-2-(-1)](-2-0)(-2-1)(-2-2)}\times 9+\frac{[x-(-2)](x-0)(x-1)(x-2)}{[-1-(-2)](-1-0)(-1-1)(-1-2)}\times 0\\&\quad+\frac{[x-(-2)][x-(-1)](x-1)(x-2)}{[0-(-2)][0-(-1)](0-1)(0-2)}\times 1+\frac{[x-(-2)][x-(-1)](x-0)(x-2)}{[1-(-2)][1-(-1)](1-0)(1-2)}\times 0\\&\quad+\frac{[x-(-2)][x-(-1)](x-0)(x-1)}{[2-(-2)][2-(-1)](2-0)(2-1)}\times 9\\&=\frac{(x+1)x(x-1)(x-2)}{24}\times 9+\frac{(x+2)(x+1)(x-1)(x-2)}{4}\times 1+\frac{(x+2)(x+1)x(x-1)}{24}\times 9\\&=\frac{24x^4-48x^2+24}{24}=x^4-2x^2+1\end{aligned}$$

由五點所決定的曲線函數為一個四階多項式，其曲線圖形如圖 2.17 所示。

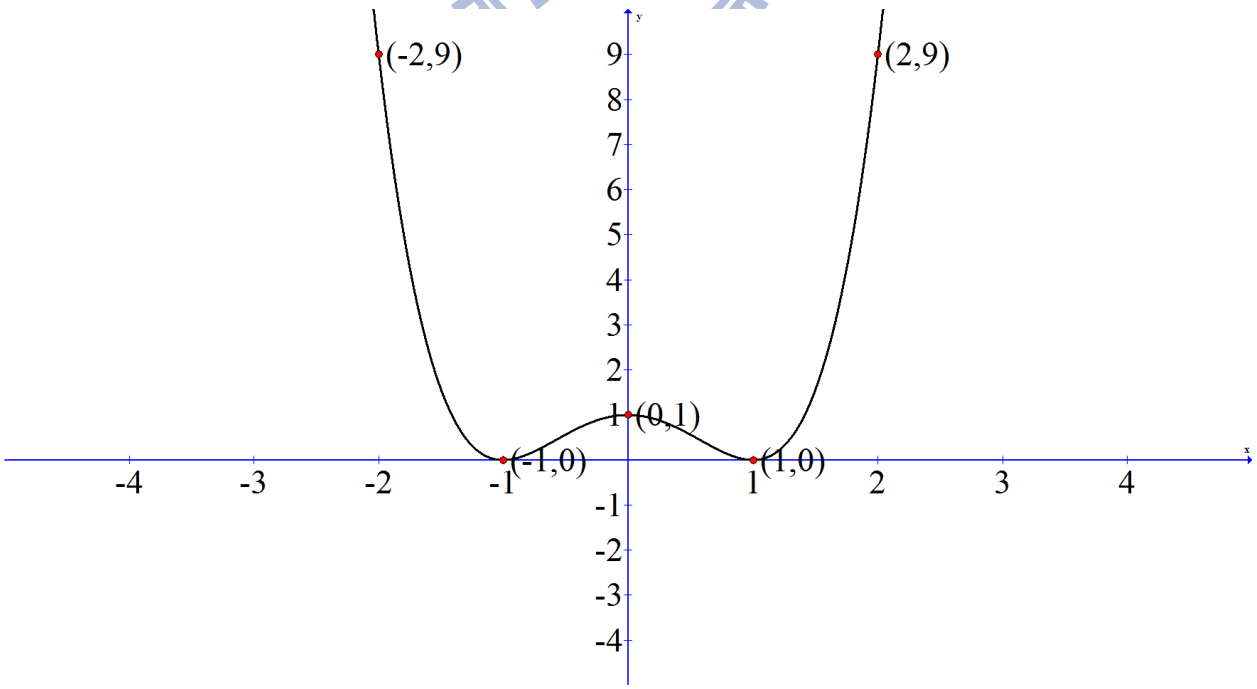


圖 2.17 四階多項式曲線

本篇論文在公正拍賣系統網絡設計上，以秘密分享技術建立其架構。在公正拍賣系統網絡中，不同參與者間彼此傳送資訊時的加密方式，便可透過沙氏秘密分享法來進行。其計算流程如圖 2.18 所示。

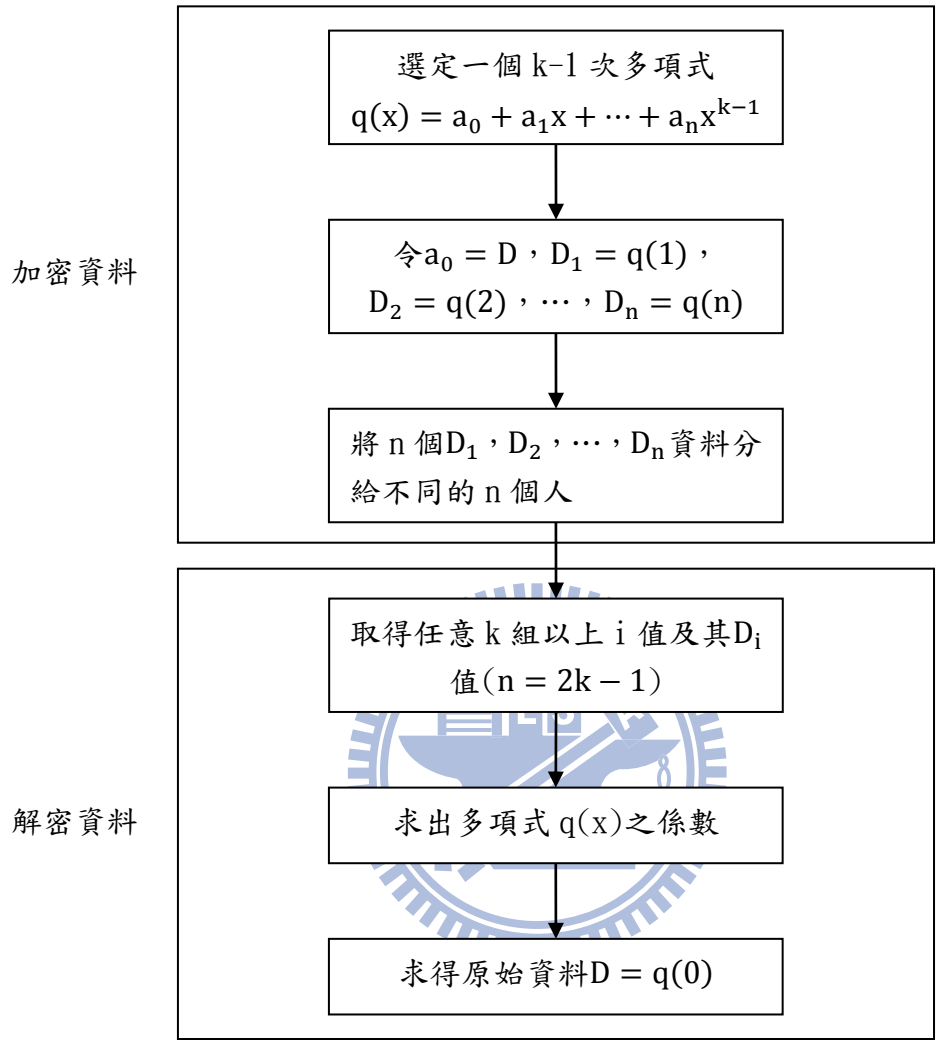


圖 2.18 拉氏內插多項式計算流程

在沙氏秘密分享法中，(k, n)門檻方案的$n = 2k - 1$，今假設欲加密的資料為 D，將此資料 D 拆為 n 個部份D_1，D_2，…，D_n時，只要取得其中 k 個部份以上的資料便可還原完整的原始資料 D。沙氏秘密分享法重建資料的主要方式在於利用一個 k-1 次的多項式來加密及解密資料，在 k-1 次多項式中，連同常數項共有 k 個係數，也就是未知數，當取得 k 個部份D_i資料時，可得到 k 個方程式，根據 k 個方程式求解出 k 個未知數，將加密過的資料重新解密出來。沙氏秘密分享法共可分為兩個部份，第一部份為加密部份，而第二部份則為解密的部份。

2.4.2 沙氏秘密分享法應用

對於沙氏秘密分享法的應用可以用一個簡單的例子做為說明。假設欲加密的資料 D 為 500，利用沙氏秘密分享法的概念選定(3, 5)門檻方案，並將 D 分為D_1、D_2、D_3、D_4及D_5共五個部份，則可以寫出一多項式

$$q(x) = 500 + x + x^2$$

且 $D = q(0) = 500$ ， $D_1 = q(1) = 502$ ， $D_2 = q(2) = 506$ ， $D_3 = q(3) = 512$ ，$D_4 = q(4) = 520$，$D_5 = q(5) = 530$。利用多項式求得D_1至D_5五個部份後，便可將原始多項式的係數刪除，此時加密動作便完成了。

利用拉氏內插多項式進行解密動作的部份，以上方相同例子做說明。在上例中其 k 值為 3，因此只要取得任意 3 組以上D_i值便可進行解密的動作，假設取得之值分別為$D_2 = 506$、$D_3 = 512$及$D_5 = q(5) = 530$三個部份，此時可利用$q(x) = a_0 + a_1x + a_2x^2$列出三條方程式如下所示：

$$q(2) = a_0 + 2a_1 + 4a_2 = 506$$
$$q(3) = a_0 + 3a_1 + 9a_2 = 512$$
$$q(5) = a_0 + 5a_1 + 25a_2 = 530$$

利用上方三個聯立方程式便可利用反矩陣求得三個未知係數a_0、a_1及a_2，將上方三個聯立方程式以矩陣表示如下：

$$\begin{bmatrix} q(2) \\ q(3) \\ q(5) \end{bmatrix} = \begin{bmatrix} 506 \\ 512 \\ 530 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

則此時可利用反矩陣求出a_0、a_1及a_2如下所示：

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \end{bmatrix}^{-1} \begin{bmatrix} 506 \\ 512 \\ 530 \end{bmatrix} = \begin{bmatrix} 500 \\ 1 \\ 1 \end{bmatrix}$$

透過反矩陣求得$a_0 = 500$，$a_1 = 1$，$a_2 = 1$，且由於原始資料$D = a_0 = 500$，透過反矩陣可以得到原始資料為 500。

2.5 J2ME 技術

爪哇語言一個相當重要的特性就是其可以跨平台的特性，透過爪哇虛擬機器(Java Virtual Machine，JVM)使得程式設計師僅需在個人電腦上撰寫程式碼，便可在不同的作業平台上運作，這樣的特性帶給程式設計師相當大的便利性。而爪哇語言的跨平台特性在現今手機產業如此發達的今日，其重要性更加顯著了。隨著科技日新月異，手機功能自然也愈來愈強大，然而，每一個廠牌的手機其作業平台並不一定相同，若是想要開發手機執行的應用程式時，必須配合不同的作業平台撰寫不同的程式碼，其工程之浩大可見一斑。此時，具備跨平台的 Java 2 Platform, Micro Edition(J2ME)[21]技術便應運而生。

爪哇語言的特性包含了易用性、平台無關性及易移植性，爪哇程式的易移植性主要仰賴於其虛擬機器的優點，而由於手機平台推陳出新的速度遠快於個人電腦，為每一個新款手機重新開發專屬的應用程式是相當耗費成本的舉動，因此，不同作業平台的手機只要在其作業平台上提供可執行 J2ME 的爪哇虛擬機器，便可執行以 J2ME 技術所開發出來的應用程式，此優點使得現今手機功能幾乎都內建爪哇虛擬機器以執行 J2ME 的應用程式。

J2ME 技術是由美國昇陽公司(SUN Microsystems, Inc.)針對一般的消費型小型設備進行程式開發的工具，內容包含了虛擬機器及一系列標準化的應用程式介面(Application Programming Interface，API)，J2ME 可以在個人電腦上進行開發及模擬運行，並且相當容易安裝至移動裝置上，開發、發布及測試的便易性也令 J2ME 有廣泛的用途及願景。

在爪哇語言的發展上，至今可分為三個版本[23]，分別為 Java 2 Platform, Enterprise Edition(J2EE)、Java 2 Platform, Standard Edition(J2SE)及 J2ME。不同的版本皆有其適用的對象，J2EE 為 Java 2 平台企業版，提供大型、可升級並且可靠的商業服務，主要針對伺服器端程式進行開發及應用；J2SE 為 Java 2 平台標準版，為一般個人電腦最常使用的版本；J2ME 則為 Java 2 微型版，專門用於嵌入式設備或移動式消費性電子產品。此三種版本間以 J2EE 功能最為強大，J2SE 次之，而本節所介紹的 J2ME 技術則為功能較少的版本，Java 2 平台的關係如圖 2.19 所示。

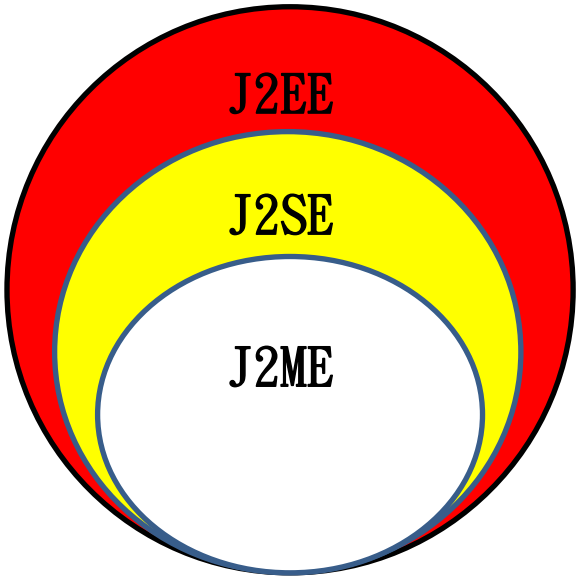


圖 2.19 Java 2 平台關係圖

J2ME 在規格上可以分為組態(Configuration)及範本(Profile)。在第 2.5.1 節中，介紹 J2ME 的組態規格。在第 2.5.2 節中，則說明在 J2ME 的組態中，再細分出連結設備組態及有限連結設備組態。而第 2.5.3 節則說明 J2ME 的範本規格。

2.5.1 J2ME 組態

由於嵌入式消費電子設備種類相當多，作業平台也不盡相同，針對此問題，J2ME 利用了兩個新的概念來建立這些規格，此兩個概念分別為組態及範本。組態和範本最大的用途就是針對爪哇虛擬機器給予更精確的規格定義，而爪哇虛擬機器則是在各種不同的作業系統上，對爪哇語言的運作環境規格給予規範，如此一層層堆疊而成的規格使得 J2ME 能夠更清楚地辨認各種不同的設備，J2ME 的規格堆疊示意圖如圖 2.20 所示。

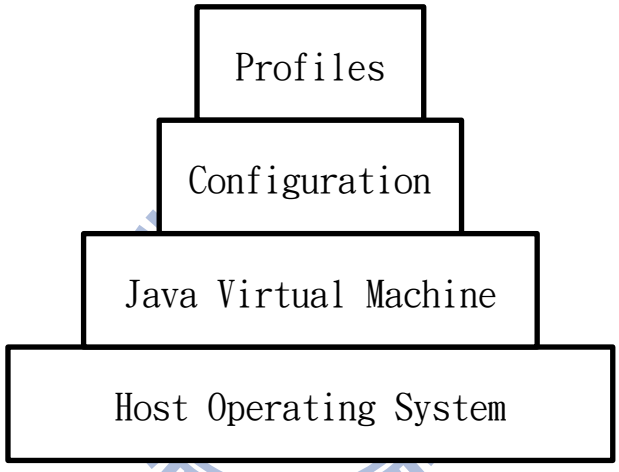


圖 2.20 J2ME 規格堆疊式意圖

組態的主要目的是針對各種消費性電子設備的網路連結能力、處理器速度及記憶體容量等特性分類，其針對所使用的裝置定義了核心類別函式庫，凡是屬於組態的設備，其內部所內建的爪哇虛擬機器都必須支援組態中所定義的函式庫，換言之，組態的用意就是對於特定的設備規定了爪哇虛擬機器及其對應的應用程式介面規格。

2.5.2 連結設備組態及有限連結設備組態

在目前的 J2ME 技術中，對組態定義了兩種不同規格，讓組態能夠更符合不同的消費電子產品，兩種組態分別為連結設備組態(Connected Device Configuration，CDC)及有限連結設備組態(Connected Limited Device Configuration，CLDC)兩種。連結設備組態適用於性能相對較好且資源較不受限的設備，如：電視機、冰箱、互動式電視盒及網路電話等；而有限連結設備組態則適用於性能相對較差且資源受限的設備，這些設備其運算能力、記憶體容量及電力供應等資源較為有限，例如：手機及個人數位助理(Personal Digital Assistant，PDA)等。兩者差異在於其所面對的設備硬體不同，連結設備組態所針對的設備則可具備

32 位元或 64 位元處理器，記憶體容量也大於 512KB 以上，而有限連結設備組態所針對的設備處理器能力較為有限，其記憶體容量往往只有 128KB 至 512KB 之間[11]。

在類別函式庫部份，連結設備組態及有限連結設備組態皆使用了一部份 J2SE 的類別函式，針對 J2SE 類別函式進行更新並增加一部份專為移動設備而開發的類別函式，然而，有限連結設備組態在類別函式的更新上完全是以連結設備組態為基礎進行，並無增加本身專用的類別函式，J2SE、連結設備組態及有限連結設備組態的關係圖如圖 2.21 所示。

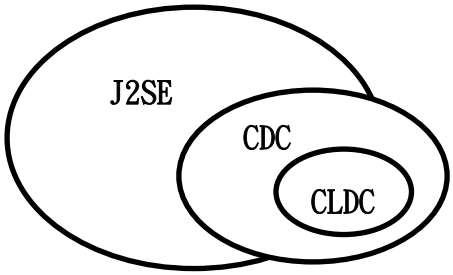


圖 2.21 J2SE、CDC 及 CLDC 關係圖

2.5.3 J2ME 範本

雖然透過組態的規格定義可以已經將設備分為連結設備組態及有限連結設備組態兩類，但是各種設備的特性仍有許多不同點，例如手機和個人數位助理雖然同屬於有限連結設備組態，然而其顯示螢幕大小及解析度等則大相逕庭，因此，J2ME 進一步利用範本針對各個不同的設備做更細部的分類，也就是在組態之上另外定義一組更細節的額外類別函式庫。對於手機而言，Mobile Information Device Profile(MIDP)即為 J2ME 對手機特別定義的函式庫，MIDP 針對手機定義了適用於手機的類別函式庫，如簡訊傳送及聲音的處理等。

由於利用 J2ME 開發的設備在硬體上往往不及個人電腦，因此為了連結設備組態及有限連結設備組態另外最佳化了專屬的爪哇虛擬機器，分別稱之為 CVM 及 KVM。對於連結設備組態所使用的 CVM 主要運作於 32 位元處理器，且其記憶體容量大於 2MB 以上的設備；而 KVM 則為有限連結設備組態所專屬的爪哇虛擬機器，KVM 為 J2ME 中最小型且最精簡的虛擬機器，在功能支援上比起 CVM 少得多，主要運作於記憶體容量僅在 160KB 至 512KB 的設備上，並且具備了低耗電的特性。雖然 J2ME 為不同規格的設備提供了專屬的虛擬機器，但程式本身仍然可以運作於一般的爪哇虛擬機器上，使程式開發更為便利。圖 2.22 為整個爪哇解決方案的示意圖，由圖中便可得知 J2ME 在爪哇平台上所扮演的角色。

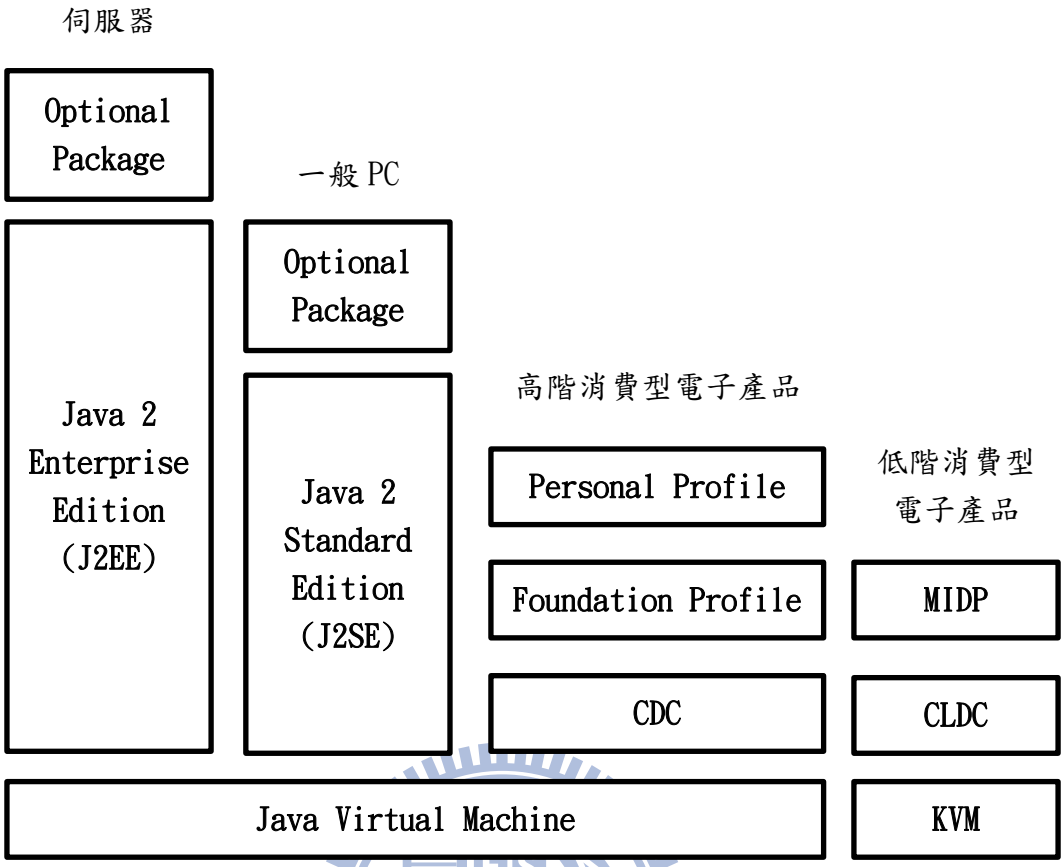


圖 2.22 爪哇解決方案示意圖[12]

第三章 無線競價系統的設計

本論文所建立的公正拍賣系統網絡，旨在透過無線競價(Wireless Bidding)進行公正拍賣(Justice-based Auction)[13]。此公正拍賣系統可分為兩大部份，分別為無線競價系統及敘核計算系統。本章將介紹公正拍賣中的無線競價系統設計。在第 3.1 節中，首先探討拍賣的制度，分析本論文所著重的拍賣制度。在第 3.2 節中，介紹利用秘密分享技術來建立無線競價系統的架構。第 3.3 節則說明如何取得各競價者利用無線競價系統出價後的標單，並整理為標單矩陣。

3.1 拍賣制度

在拍賣的型式中，同時拍賣常依待拍賣物數目、標單內容及成交價格三項特性來分類。就拍賣物數目而言，可以分為單件拍賣與多件拍賣兩大類。就標單內容而言，可根據標單內容是否公開而分為開式拍賣及閉式拍賣兩大類。開式拍賣指的是在此場拍賣中的所有競價者在出價過程中所出的價錢是公開的；相反地，閉式拍賣則是指在出價過程中，競價者所出的價錢是非公開的。就成交價格而言，依據標單價格的最高價、次價及公正價而分為首價拍賣、次價拍賣及公正價拍賣三大類。首價拍賣表示在一場拍賣中，得標者所需付的得標金額即其本身出價的金額；而次價拍賣則是指得標者僅需支付在此場拍賣中，出價金額第二高的金額；公正價格則為本論文結合勞爾斯教授的正義原則及賽局理論的敘核所求出的得標者公正價格。傳統上首價開式拍賣又稱為荷蘭式拍賣(Dutch Auction)，首價閉式拍賣則稱為投標法，次價開式拍賣為英國式拍賣(English Auction)，而次價閉式拍賣則為維氏拍賣(Vickrey Auction)。由拍賣物數目、標單內容及成交價格三項特性，表 3.1 顯示共十二種的拍賣型式，且用其特性來命名。

表 3.1 拍賣型式[13]

	拍賣型式	開式	閉式
單件拍賣	首價	荷蘭式拍賣	投標法
	次價	英國式拍賣	維氏拍賣
	公正價	開式公正拍賣	閉式公正拍賣
多件拍賣(指派賽局)	首價	同時荷蘭式拍賣	同時投標法
	次價	同時英國式拍賣	同時維氏拍賣
	公正價	同時開式公正拍賣	同時閉式公正拍賣

第 3.1.1 節中，首先介紹公正拍賣的架構。第 3.1.2 節中，進一步介紹利用網際網路來進行同時閉式公正拍賣的網路型閉式公正拍賣。第 3.1.3 節則說明網路型閉式公正拍賣的安全性。

3.1.1 公正拍賣架構

公正拍賣的內容包括標單、得標者名單與得標價格三項要素，而這三者可用標單矩陣、利潤矩陣與滿意度矩陣來產生。由此角度來看，公正拍賣的作業流程可分為三項步驟，如圖 3.1 所示。第一步驟為用秘密分享技術產生標單矩陣：這是開標方將所有投標者的標單透過秘密分享技術解密並整理於標單矩陣中。第二步驟為用利潤矩陣產生得標者名單：這是將所有競價者的出價減去拍賣物底價後所產生的利潤矩陣。再利用匈牙利人法求出最佳指派的得標者。第三步驟為用滿意度矩陣求出得標價格：這是透過滿意度轉移方式將最小滿意度團體給予極大化，也就是求其斂核的成交價格解。

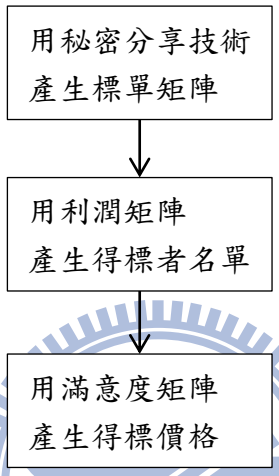


圖 3.1 公正拍賣的作業流程[13]

3.1.2 網路型閉式公正拍賣

在第 3.1 節中，可以得知首價閉式拍賣指的就是得標者需付其本身出價的金額，且出價過程是非公開性的，此類型的拍賣方式也就是投標法，其又可稱為同時出價(Simultaneous-Bidding)制度[7]。投標的進行方式是由競價者將其對於競價物所出的投標金額寫於紙上並密封，當所有競價者皆將其投標金額寫於紙上密封後，由拍賣主辦者收齊標單一次全部打開進行比價，並由出價最高者得標，投標通常以單一賽局的方式出現。

在傳統的首價閉式拍賣中，由於拍賣過程皆以人工方式進行，其中隱含著許多缺失。在出價過程，由於必須將投標金額實際書寫於紙上，在書寫時，其投標金額可能會遭到其他競價者竊看；此外，拍賣進行時，往往是將所有競價者集中在單一場地，一旦有不法利潤團體試圖以威脅利誘的方式來進行不公平的拍賣，例如：圍標及綁標等，將使得此拍賣結果失去公平性；再者，傳統的投標方式會在特定的地點進行拍賣，對該項拍賣有興趣的競價者都必須前往拍賣地點才能參與競價，如此作法往往造成人力、金錢及時間的浪費。

閉式公正拍賣所採用的競價方式與投標法相同，但成交價格則是以公正價格做為最終成交價格。因此，傳統首價閉式拍賣的缺點同樣存在於閉式公正拍賣中。為了能夠改善上述缺點，本論文結合網際網路來進行拍賣，也就是『網路型閉式公正拍賣』。所謂網路型閉式公正拍賣，顧名思義就是利用網際網路的方式來進行投標，在網路發達的今日，透過

網路來進行投標將可充分改善原有傳統首價閉式拍賣的缺點。利用網路的方式，競價者可以利用手機直接進行出價，不需要出席拍賣現場，就可以避免投標金額被竊看等安全性問題。透過電腦處理更能快速地完成拍賣，使得整場拍賣更具效率性。

3.1.3 網路型閉式公正拍賣的安全性

利用網路型閉式公正拍賣是相當高效率的作法。然而，傳統首價閉式拍賣時所需注意的保密性在網路型閉式公正拍賣同樣必須考慮。在標單部份，必須確保所有標單在開標前都是未公開的，投標時段及開標時機的掌握也必須謹慎考慮，避免因為操縱投開標時機而使得某些有意競價的廠商被排除在外。除了傳統首價閉式拍賣的保密性必須考慮外，網路型閉式公正拍賣還須注意在設計上必須能夠確認競價者的身分，防止假冒競價者出價的情況產生；至於標單設計還需考慮其認證方式，目的在於當開標完後，競價者無法否認標單的內容。最重要的一點，網路型閉式公正拍賣對於競價者的身分及資訊傳輸都要能夠做到保密的效果。

3.2 秘密分享的架構

閉式拍賣在現今社會還是相當常見，而在本論文中，更進一步建立網路型閉式公正拍賣的環境，使得投標方式更加多元，開標速度更加快速，拍賣更具公平性，讓閉式公正拍賣能夠更有效率地進行。面對網路型閉式公正拍賣所遇到最根本的問題就在於，當資訊在網路上傳輸時必須保證資訊不會外洩，包含競價者身分、出價金額及所在位置等等。在本章所建立的無線競價系統中，利用在第 2.4 節提到的秘密分享進行資料加密，建立一套秘密分享無線競價系統。無線競價系統必須針對整個公正拍賣系統中，不同角色的單位進行設計。一個典型的公正拍賣系統可分為競價方、公證方及開標方三個單位。在競價方而言，欲競價的人可以利用網路或是手機方式進行出價，出價完成後，其標單資料會以秘密分享方式加密後傳送給各個公證方。對公證方而言，則假設有產、官、學、研四個公證方[5]。

開標方的部份，則由拍賣主辦單位取得各公證方所擁有的部份加密資料後，進行解密並開標。利用秘密分享所建立的無線競價系統，對於競價方、公證方及開標方三個單位的架構如圖 3.2 所示。

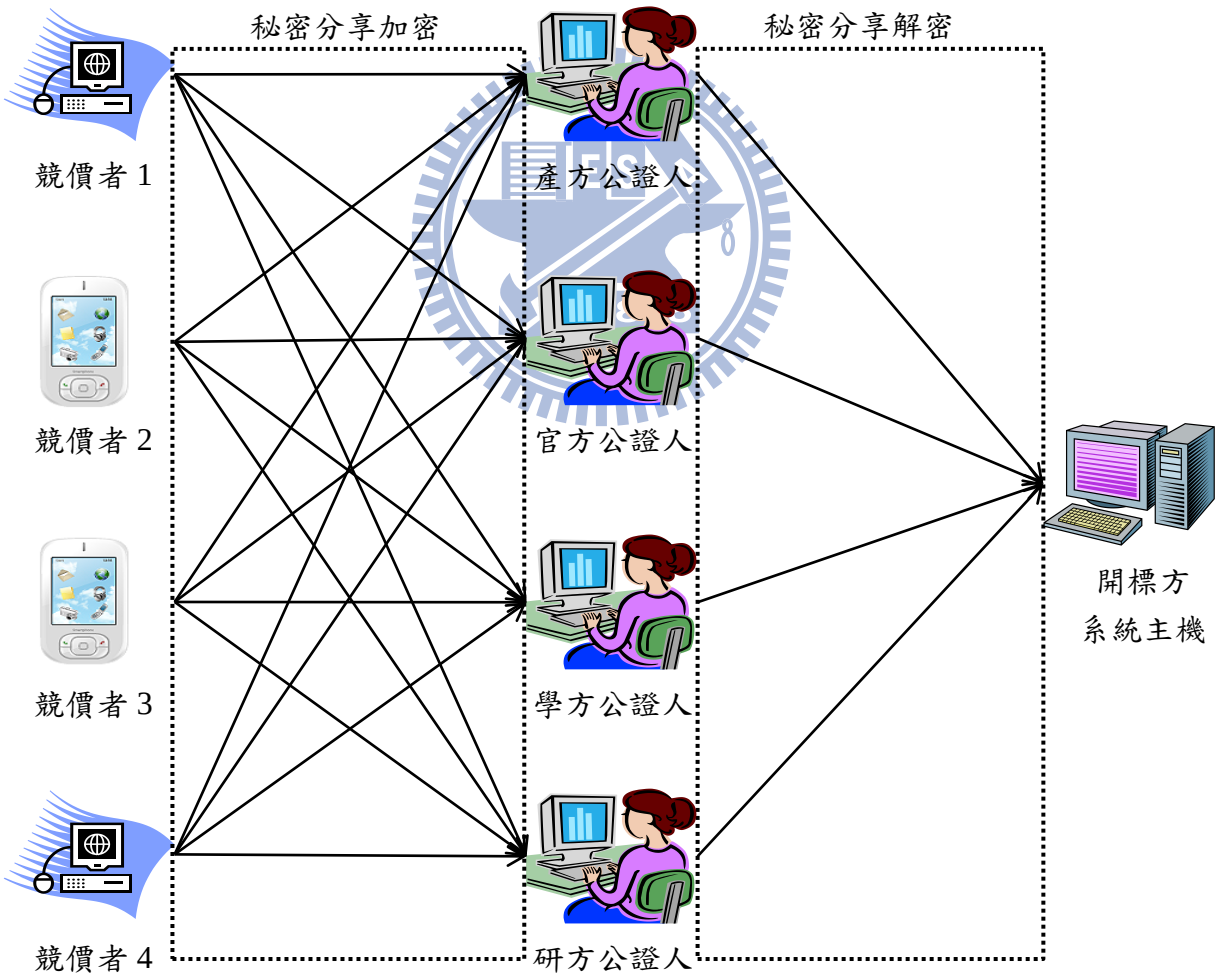


圖 3.2 無線競價系統架構圖

在第 3.2 節說明了秘密分享的架構後，第 3.2.1 節介紹競價方規格設計。第 3.2.2 節介紹公證方規則設計，而第 3.2.3 節則介紹開標方規則設計。

3.2.1 競價方規則設計

在本論文建置的無線競價系統中，所採用的是網路型閉式公正拍賣的方式，競價方可以透過網路或手機的方式進行投標。競價方對於拍賣物的出價屬於單一賽局，也就是所謂的投標。對於競價方而言，首先必須到公正拍賣系統網站上登錄個人資料並註冊專屬帳號及密碼，經系統確認無誤後，競價方始能利用此帳號進行投標行為。競價方的投標必須在投標時間內進行，一旦投標時間結束，便立即停止投標。在進行投標程序時，競價方填妥相關標單後，系統會自動將競價方的標單、帳號及密碼等個人資料以秘密分享的方式加密後，再分別傳送到四個公證方系統的伺服端，並存入資料庫中，競價方投標流程如圖 3.3 所示。

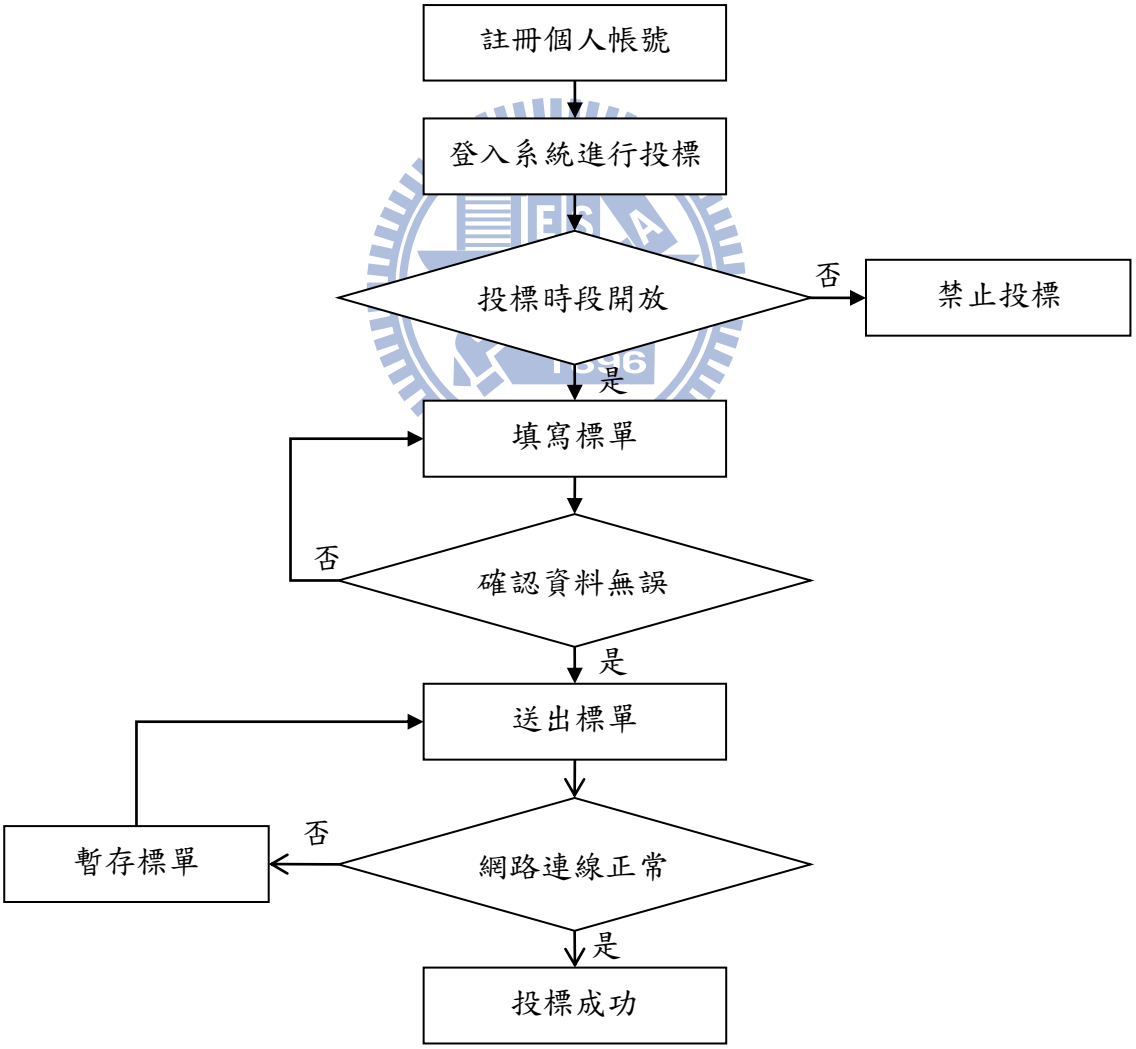


圖 3.3 競價方投標流程

3.2.2 公證方規則設計

在公正拍賣系統中，設計了產、官、學、研四個公證方，負責保管所有競價者的加密標單。為了避免公證方徇私舞弊造成洩標的情形產生，必須將傳送給公證方的標單進行加密的動作，在第 2.4.1 節所提到的沙氏秘密分享法就是此系統用以加密資料的方式。當競價方投標成功後，系統會自動將原始標單分為四個標單拼圖資料，並且傳送給產、官、學、研四個公證方。在公證方的部份，分別利用四個伺服端來接收標單拼圖，並將資料存入資料庫中。等待投標時間結束後，開標方系統主機會發出要求，要求四個公證方伺服端將資料庫中的標單拼圖資料傳送至開標方系統主機，以進行開標動作。公證方設計的架構圖如圖 3.4 所示。

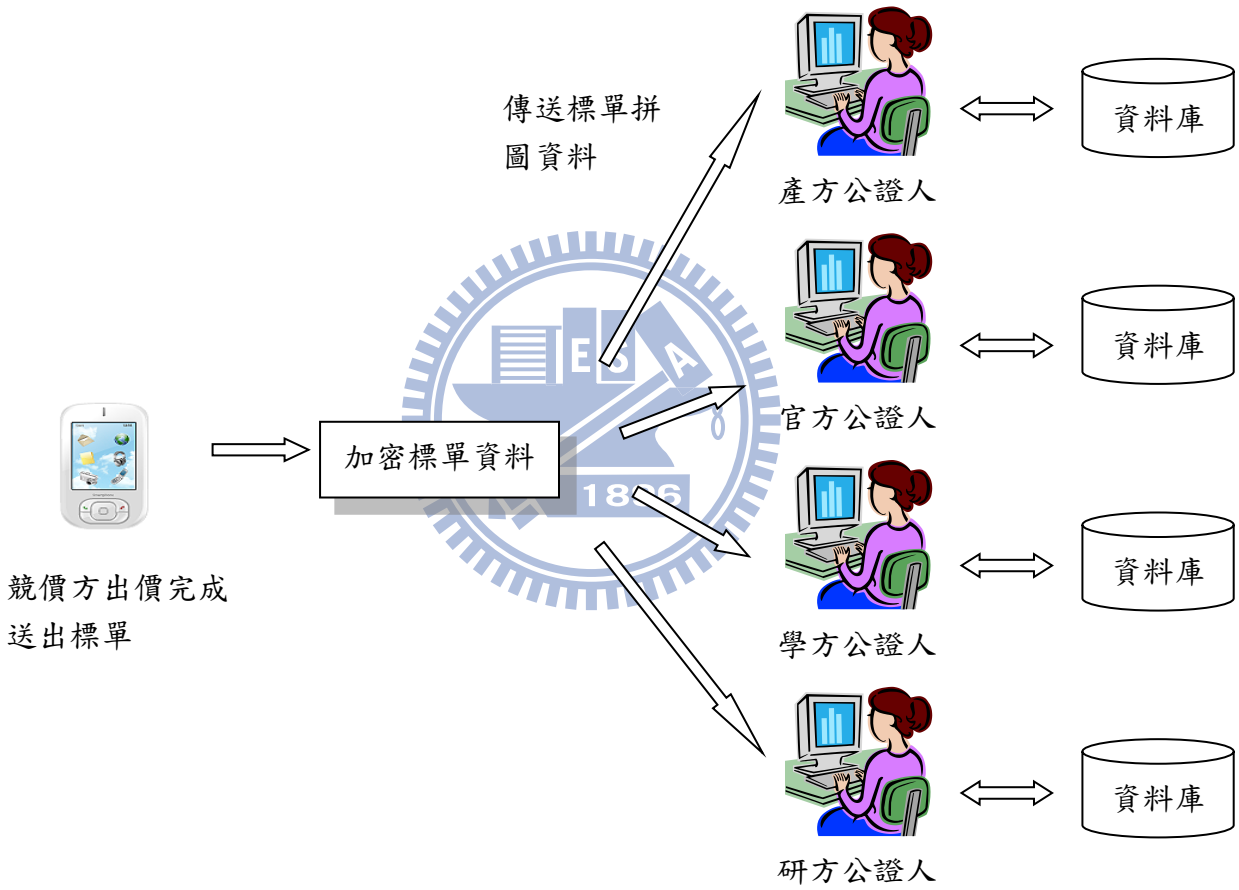


圖 3.4 公證方設計架構圖

3.2.3 開標方規則設計

當投標時間結束後，公正拍賣系統即停止投標作業，準備進入開標作業，開標方角色通常由拍賣主辦單位來擔任。當規定投標時間結束後，拍賣主辦單位宣布進入開標階段，此時開標方系統主機會發出要求給四位公證方，要求公證方將資料庫中各個競價者的標單拼圖資訊傳送至開標方系統主機，而開標方系統主機會根據得到的標單拼圖進行解密的動作。透過解密可以得到各個競價者的原始標單資料，此時系統會自動比對所有標單找出投標金額最高者為得標者，接著求解出得標價格，也就是公正價格。主辦單位宣布得標者並公告得標價格後，此拍賣便告完成。公正拍賣系統的開標流程如圖 3.5 所示。

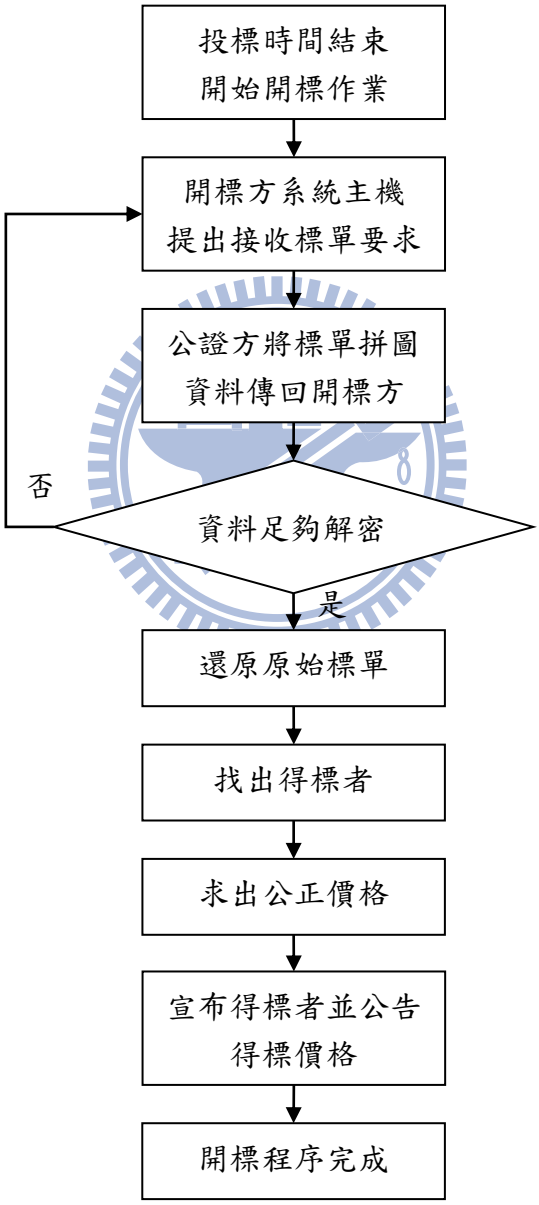


圖 3.5 公正拍賣系統開標流程

開標方完成開標程序後，將開標結果儲存於資料庫中，以供日後比對之用。開標方設計的架構圖如圖 3.6 所示。

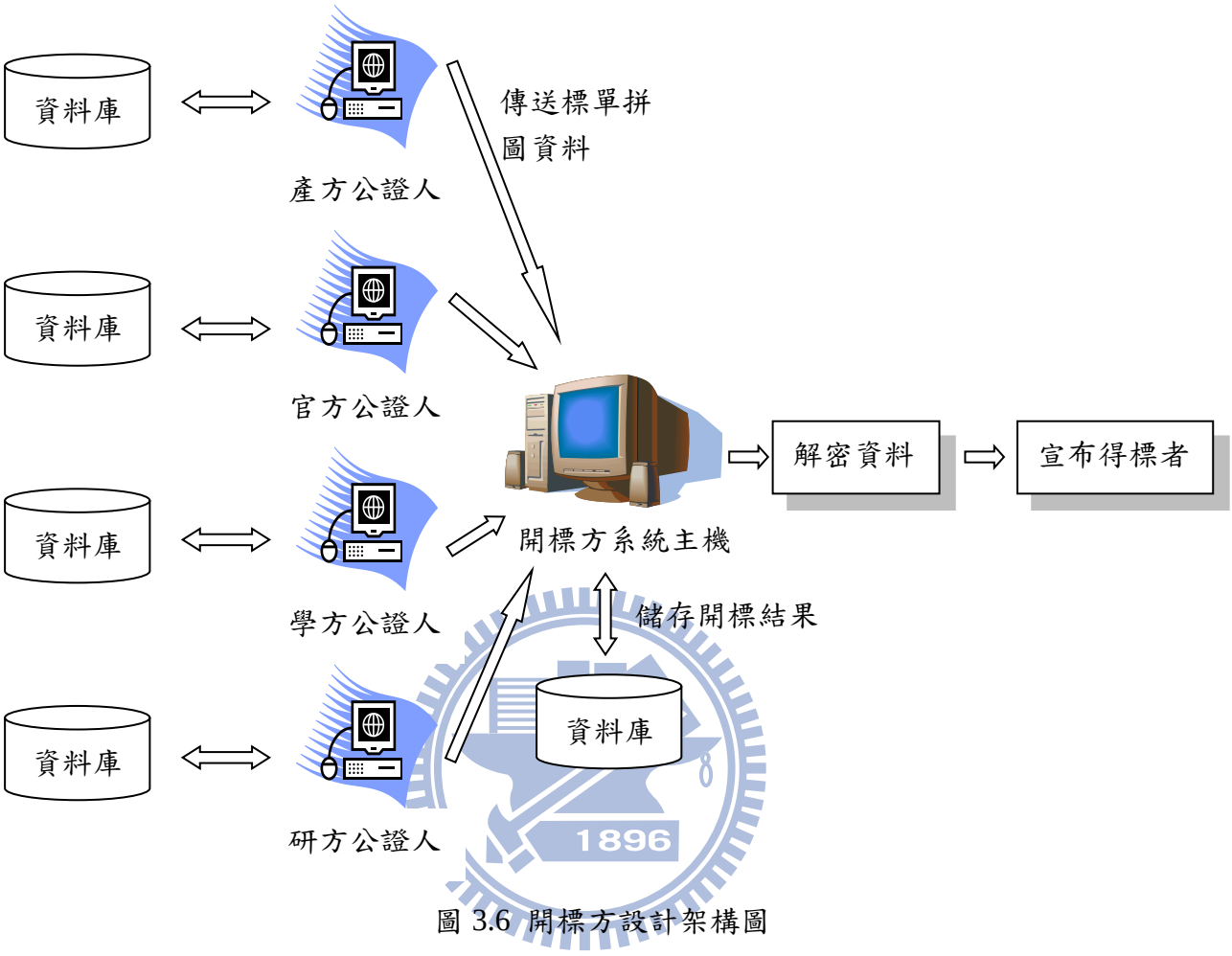


圖 3.6 開標方設計架構圖

3.3 標單矩陣的取得

第 3.2 節秘密分享的架構提到公正拍賣系統可分為競價方、公證方與開標方三個方面。競價方利用本論文所設計的無線競價系統來進行出價的動作，當競價者完成出價後，競價者所出價的金額稱為原始標單。此時，無線競價系統會將原始標單利用秘密分享加密後，傳送至各個公證方伺服端，加密後的標單稱為標單拼圖。當投標時間結束後，開標方系統向各公證方取得標單拼圖後，便可利用秘密分享解密標單拼圖，並且將所有解密後標單整理成標單矩陣，以簡化計算。表 3.2 為公正拍賣系統的各個單位所擁有的標單資訊與使用的系統。

表 3.2 各方標單資訊與系統

單位名稱	標單資訊	使用系統
競價方	原始標單	無線競價系統
公證方	標單拼圖	資料庫系統
開標方	標單矩陣	敘核計算系統

在第 3.3.1 節中，介紹無線競價系統如何自動加密競價者的原始標單。而第 3.3.2 節則說明透過解密標單拼圖來取得標單矩陣。

3.3.1 加密原始標單

在第 2.4 節所提到的秘密分享技術，便是無線競價系統用來加密標單的方法。對秘密分享技術而言，這是將標單以不同的方法加密後，再寄給不同的公證方；但如果開標方接收全到公證方的加密標單後，則可透過解密技術還原出原始標單的內容。底下舉一個簡單的例子來說明加密原始標單的結果。

今假設在一場公正拍賣中，有三樣待拍賣的物品，分別為 X、Y 和 Z。現有四位競價方 A、B、C 和 D 分別對 X、Y 和 Z 三項拍賣物進行投標。已知 A 對 X、Y 和 Z 各別的出價分別為 370 萬元、490 萬元及 500 萬元。此時，A 的原始標單如表 3.3 所示。

表 3.3 原始標單(單位：萬元)

拍賣物	拍賣底價	A 出價
X	100	370
Y	200	490
Z	300	500

利用第 2.4.1 節的沙氏秘密分享法，選定(3, 4)門檻方案後，便可將表 3.3 中 A 對拍賣物 X、Y 和 Z 的出價進行加密。每一個出價金額在加密後，會轉換成四個標單拼圖內容及相對應的標單拼圖鍵值如表 3.4 所示。

表 3.4 標單拼圖(單位：萬元)

拍賣物	競價方	標單拼圖鍵值	標單拼圖內容
X	A	1	472
X	A	2	716
X	A	3	1102
X	A	4	1630
Y	A	1	608
Y	A	2	888
Y	A	3	1330
Y	A	4	1934
Z	A	1	559
Z	A	2	724
Z	A	3	995
Z	A	4	1372

今令競價者 B 對 X、Y 和 Z 的出價分別為 310 萬元、220 萬元及 460 萬元；競價者 C 對 X、Y 和 Z 的出價分別為 140 萬元、260 萬元及 300 萬元；競價者 D 對 X、Y 和 Z 的出價分別為 380 萬元、280 萬元及 320 萬元。則 B、C 和 D 的標單亦同樣能以沙氏秘密分享法加密得到如表 3.4 的標單拼圖，並儲存於四位公證方資料庫中。

3.3.2 解密標單拼圖

在第 3.3.1 節提到，各競價者的標單透過秘密分享轉換為標單拼圖，並且儲存於公證方資料庫中。當投標時間結束後，開標方必須解密由四位公證方收集到的標單拼圖，以還原競價者的原始標單。以表 3.4 的標單拼圖為例，利用第 2.4.1 節沙氏秘密分享法的解密資料步驟，可以解密出競價者 A 對於拍賣物 X、Y 和 Z 的出價如表 3.5 所示。

表 3.5 還原後原始標單(單位：萬元)

拍賣物	拍賣底價	A 出價
X	100	370
Y	200	490
Z	300	500

同樣地，競價者 B、競價者 C 及競價者 D 的標單拼圖亦可還原如表 3.5 的原始標單。為了簡化計算，可以將 A、B、C 和 D 各別還原後的原始標單整理為單一矩陣，稱為標單矩陣。如表 3.6 所示。

表 3.6 標單矩陣(單位：萬元)

拍賣物	拍賣底價	A 出價	B 出價	C 出價	D 出價
X	100	370	310	140	380
Y	200	490	220	260	280
Z	300	500	460	300	320

在表 3.6 中，第三列第四行的數字 220 表示競價者 B 對於拍賣物 Y 的出價為 220 萬元。透過解密標單拼圖得到標單矩陣，可以將四位競價者對於三樣拍賣物的出價整理為一個 3x5 的標單矩陣，提供敘核計算系統更快速地找出得標者並計算公正價格。

第四章 敛核計算系統的實作

在本論文所建立的公正拍賣系統第二部份，為敛核計算系統。在第三章提到，利用秘密分享技術建立無線競價系統，提供競價方於網際網路進行出價；而本章則是利用舒-瑞氏演算法來建立敛核計算系統，提供開標方快速找出得標者，並求解出得標者的得標價格，也就是公正價格。

在第 4.1 節利潤矩陣的使用，說明標單矩陣轉換為利潤矩陣的方式，再用利潤矩陣以匈牙利人法求出得標者。第 4.2 節滿意度矩陣的使用，介紹舒-瑞氏演算法的滿意度矩陣。第 4.3 節有向圖的使用，則解釋有向圖與利潤的關係。第 4.4 節計算的簡化，闡述利用舒-瑞氏演算法來讓公正價格的計算更加快速。

4.1 利潤矩陣的使用

第 3.3.2 節中，表 3.6 所顯示的標單矩陣，包含了拍賣物底價及各競價者的出價。然而在開標過程中，僅需要以拍賣可得利潤做為考量依據，因此，可進一步將標單矩陣轉換為利潤矩陣，以提高計算的效率。

在 4.1.1 節由標單矩陣轉換為利潤矩陣，介紹利潤矩陣的獲得。而第 4.1.2 節由匈牙利人法求出得標者，則說明如何利用第 2.2.2 節的匈牙利人法，由利潤矩陣中找出得標者。

4.1.1 由標單矩陣轉換為利潤矩陣

由標單矩陣轉換為利潤矩陣的方式相當簡單。在表 3.6 中，第二行代表各拍賣物的底價，而第三行至第六行則為 A、B、C 和 D 的出價。將各競價者的出價減去第二行中拍賣物的底價後，便可得到利潤矩陣如表 4.1 所示。

表 4.1 利潤矩陣(單位：萬元)

拍賣物	A 出價	B 出價	C 出價	D 出價
X	270	210	40	280
Y	290	20	60	80
Z	200	160	0	20

在表 4.1 中，第二列第二行的金額為 270 萬元($= 370 - 100 = a_{XA}$)，代表競價者 A 取得拍賣物 X 時得到的利潤。餘此類推，就可由表 3.6 產生表 4.1。由於出價一定要高或等於拍賣底價，所以利潤矩陣的元素a_{XA}沒有負值，即$a_{XA} \geq 0$。a_{XA}又稱為利潤的理論值，它代表利潤團體理論上可以共同得到的利潤。由於利潤矩陣的元素數目比標單矩陣少，又可降低計算得標者名單的計算複雜度，故採用利潤矩陣來計算得標者名單。

4.1.2 由匈牙利人法求出得標者

對利潤矩陣來說，最大的功能是用來產生得標者名單。由於拍賣的目的在產生核裡價格，也就是柏瑞圖最佳化。柏瑞圖最佳化下的得標者名單結果可透過匈牙利人法來計算。例如表 4.2 加註星號(*)的部份就是得標者名單；換言之，拍賣物 X 的得標者為 D，拍賣物 Y 的得標者為 A，拍賣物 Z 的得標者為 B。匈牙利人法的執行細節可參考第 2.2.2 節。

表 4.2 得標者名單(單位：萬元)

拍賣物	A	B	C	D
X	270	210	40	280*
Y	290*	20	60	80
Z	200	160*	0	20

雖然匈牙利人法可以產生得標者名單，它卻無法產生公正的得標價格。例如買方 D 的得標金額為 280，它代表$a_{XD} = 280$，但卻不知道買方 D 應付多少利潤給 X。同理，買方 A 與 B 也有會面對同樣的問題。又公正的得標價格要求最小滿意度團體的利潤最大化。這最小滿意度最大化過程需要有滿意度的初始解，然後再針對初始解改進直到收斂[25]為止。此收斂的解就稱為斂核。為了產生斂核，這必需先將利潤矩陣轉為滿意度矩陣方可進行，第 4.2 節滿意度矩陣的使用將詳細說明之。

4.2 滿意度矩陣的使用

在第 4.1.2 節產生得標者名單後，接著是用滿意度矩陣產生得標價格。得標價格的產生可分為(1)建立滿意度矩陣、(2)滿意度矩陣性質及(3)最小滿意度最大化來說明清楚。對於滿意度矩陣的建立來說，是來自得標者名單。以表 4.2 的得標者名單來說，這是先將競價者重新整理，令未得標者在前而得標者在後，使得得標者呈對角線化排列，如表 4.3 所示。

表 4.3 對角線化得標者名單(單位：萬元)

拍賣物	C	D	A	B
X	40	280*	270	210
Y	60	80	290*	20
Z	0	20	200	160*

而所謂滿意度，指的就是在一場公正拍賣中，參與拍賣的利潤團體對於交易結果的滿意程度。在第 2.1.2 節中，對於抱怨函數及滿意度函數的介紹，說明當滿意程度高時，也就代表著其抱怨程度低，抱怨與滿意度是一體兩面的。對於一個由賣方 i 及買方 j 所組成的利潤團體{i, j}而言，其滿意度計算可以寫成

$$f_{ij} = u_i + v_j - a_{ij}$$

其中f_{ij}表示{i, j}這個利潤團體的滿意度，u_i為賣方 i 實際分得的利潤，v_j為買方實際分得的利潤，a_{ij}為第 4.1.1 節所述的理論利潤值。在表 4.3 中，增加代表賣方實際利潤的 U 欄與買方實際利潤的 V 列，如表 4.4 所示。滿意度矩陣可以用對角線化值來設定，例如在建立初始滿意度矩陣時，將得標者的得標金額放入 U 欄，而 V 列的初始值則設定為 0。因此，當實際利潤$(u_i + v_j)$大於理論利潤(a_{ij})時，滿意度為正值；反之，滿意度為負值。在表 4.4 第三列第三行中，利潤團體{X, C}的滿意度為$f_{XC} = 240 = u_X + v_C - a_{XC} = 280 + 0 - 40$，而利潤團體{X, D}的滿意度為$f_{XD} = 0 = u_X + v_D - a_{XD} = 280 + 0 - 280$，餘則類推。

表 4.4 滿意度矩陣(單位：萬元)

	賣方利潤 U (買方 0)	C	D	A	B
買方利潤 V (賣方 0)	0	0	0	0	0
X	280	240	0	10	70
Y	290	230	210	0	270
Z	160	160	140	-40	0

對滿意度矩陣性質而言，可以分為(1)得標者團體與(2)非得標者團體兩部份來說明。表 4.4 的滿意度矩陣中，新增了虛擬的買方 0 及賣方 0，並且將所有未成功交易到的賣方或買方指派給買方 0 及賣方 0，形成虛擬的得標者團體，如{V, C}。對得標者團體來說，這是指表 4.4 數字被框起的部份。在滿意度矩陣中，得標者團體的滿意度一定為 0，故又稱為固定滿意度(Settled Satisfaction)。就非得標者團體來說，這是指表 4.4 數字未被框起的部份。其滿意度可透過買賣雙方利潤的流動來調整，故又稱為變動滿意度(Unsettled Satisfaction)。觀察表 4.4 滿意度矩陣可以發現初始的利潤全數歸於賣方，因此賣方 1 的利潤為 280 萬元，賣方 2 的利潤為 290 萬元，賣方 3 的利潤則為 160 萬元，而賣方 0 由於是虛擬的賣方，因此其利潤為 0。買方初始的利潤則皆設為 0。則初始的利潤向量(Payoff Vector)為

$$(u, v) = \left(u_{賣方0}, u_X, u_Y, u_Z, v_{買方0}, v_C, v_D, v_A, v_B\right) = (0, 280, 290, 160, 0, 0, 0, 0)$$

在表 4.5 中，若 Y 的利潤減少 1 萬元，則$f_{YC} = 229$，$f_{YD} = 209$，$f_{YB} = 269$，但為了保持得標者團體$f_{YA} = 0$的性質，因此會產生$v_A = 1$，$f_{XA} = 11$，$f_{ZA} = -39$的結果。由此可看出利潤的流動會使得滿意度獲得改善，例如表 4.4 的最小滿意度為$f_{ZA} = -40$，而改善後的$f_{ZA} = -39$。持續最小滿意度的最大化過程即可達到計算斂核的目的。

表 4.5 滿意度矩陣的利潤流動(單位：萬元)

	賣方利潤 U (買方 0)	C	D	A	B
買方利潤 V (賣方 0)	0	0	0	1	0
X	280	240	0	11	70
Y	289	229	209	0	269
Z	160	160	140	-39	0

4.2.1 最小滿意度值 α 的改善

對最小滿意度最大化而言，這是透過利潤的流動來提升最小滿意度。為了表達利潤的流動，將表 4.5 加入座標系$\{x, y\}$可得表 4.6。由於買方 C 是與虛擬賣方 0 交易，因此其座標值為 0。令s_x表示賣方 x 的利潤單位增量，t_y表示買方 y 的利潤單位增量。由於利潤的流動必須由賣方流向買方，因此$s_x \leq 0$，$t_y \geq 0$。例如表 4.6 利潤的流動為$s_2 = -1$，$t_2 = +1$。表 4.6 的最後一欄顯示s_x的值；最後一列則顯示t_y的值。觀察座標化滿意度矩陣可以發現，當$x = y$時，$s_x + t_y = 0$，即表示得標者團體的利潤流動恆等於 0。

表 4.6 座標化滿意度矩陣(單位：萬元)

座標		0	0	1	2	3	
		賣方利潤 U (買方 0)	C	D	A	B	s_i
0	買方利潤 V (賣方 0)	0	0	0	1	0	0
1	X	280	240	0	11	70	0
2	Y	289	229	209	0	269	-1
3	Z	160	160	140	-39	0	0
	t_j		0	0	+1	0	

現以表 4.4 為例說明最小滿意度值 α 的改善。表 4.4 加入座標系後可轉換為表 4.7。在表 4.7 中，可以明顯看出最小滿意度是$\alpha = f_{ZA} = -40$，為了使得最小滿意度值獲得改善，因此可以得到$s_3 + t_2 \geq 1$的不等式[19]。又因為$s_3 + t_3 = 0$，則不等式可改寫為$t_2 - t_3 \geq 1$。當t_3增加時，並不會對最小滿意度值f_{ZA}有任何改善，因此令$t_3 = 0$，則$t_2 \geq 1$。由於t_2僅代表買方 2 的單位增量，因此取最小增量$t_2 = 1$。故$s_2 = -1$。利潤流動的結果如表 4.7 所示。

表 4.7 滿意度矩陣的利潤流動(單位：萬元)

座標		0	0	1	2	3	
		賣方利潤 U (買方 0)	C	D	A	B	s_i
0	買方利潤 V (賣方 0)	0	0	0	0	0	0
1	X	280	240	0	10	70	0
2	Y	290	230	210	0	270	-1
3	Z	160	160	140	-40*	0	0
	t_j		0	0	+1	0	

4.2.2 最大調整值 β 的計算

第 4.2.1 節決定利潤流向後，接著必須決定利潤流動的上限值，也就是最大調整值 β。由於最小滿意度值的可調整上限為α + β，而利潤團體{x, y}改善後的滿意度$f'_{xy} = f_{xy} + \beta(s_x + t_y)$，為了保持$f_{32} = \alpha + \beta$是最小滿意度，則最大調整值 β 可寫成表 4.8 上方的通式。表 4.8 上方的通式可再簡化而成為表 4.8 下方的公式。

表 4.8 最大調整值 β 的計算[13]

Max β $\alpha + \beta \le f_{xy} + \beta(s_x + t_y)$
Max β $\beta \le \dfrac{f_{xy} - \alpha}{1 - (s_x + t_y)}$

例如表 4.7 的最大調整值

$$\beta = \frac{f_{xy} - \alpha}{1 - (s_x + t_y)} = \frac{f_{01} - (-40)}{1 - (s_0 + t_1)} = \frac{0 - (-40)}{1 - (0 + 0)} = 40$$

改善後的滿意度矩陣如表 4.9 所示。

表 4.9 改善後滿意度矩陣(單位：萬元)

座標		0	0	1	2	3	
		賣方利潤 U (買方 0)	C	D	A	B	s_i
0	買方利潤 V (賣方 0)	0	0	0*	40	0*	
1	X	280	240	0	50	70	
2	Y	250	190	170	0	230	
3	Z	160	160	140	0*	0	
	t_j						

4.3 有向圖的使用

在第 4.2 節中，利用最小滿意度值 α 的改善及最大調整值 β 的計算，便可持續更新滿意度矩陣以計算出斂核。然而改善的過程較為繁瑣。在舒-瑞氏演算法中，其最大特色就是利用有向圖的方式去求解滿意度矩陣的利潤流向並重新分配利潤，使得利潤往斂核前進。

在第 4.3.1 節首先介紹有向圖與利潤流動的關係。第 4.3.2 則說明當有向圖出現迴徑時，透過迴徑的融合來形成新的有向圖。

4.3.1 有向圖與利潤流動的關係

對於舒-瑞氏演算法中的有向圖來說，可以分為(1)有向圖座標及(2)有向圖性質來說明。就有向圖座標而言，第 4.2.1 節提到將滿意度矩陣加入座標系$\{x, y\}$，如表 4.7 的滿意度矩陣中，所有未指派到的買方以及虛擬的買賣方，其座標皆為 0，其餘買賣方則依序編號為 1、2 及 3。此座標系可以用有向圖來重新詮釋。圖 4.1 為根據表 4.7 的滿意度矩陣所繪製成的典型有向圖。此有向圖共有兩列，第一列節點 p 用來表達得標者團體。節點 0 表示得標者團體$\{0, 0\}$，即{賣方 0, 買方 0}與{賣方 0, 買方 C}這兩組得標者團體；節點 1 則表示得標者團體$\{1, 1\}$，即{賣方 X, 買方 D}此組得標者團體，餘則類推。在有向圖中，箭號的指向用來表示對應的滿意度矩陣中，非得標者團體最小滿意度值 α 的座標。表 4.7 中，最小滿意度發生在座標$\{3, 2\}$，因此圖 4.1 會產生一個由 3 指向 2 的箭號，用來代表座標$(3, 2)$。

對有向圖性質而言，這是說明有向圖如何找出利潤的流向。圖 4.1 的第二列 l(p)表示某節點到達節點 p 的最長路徑值。例如節點 2 的最長路徑為 3→2，則節點 2 的最長路徑值 $l(2) = 1$。有向圖的最長路徑值可透過第 2.3.2 節「深度優先搜尋」所介紹的方法求得。今令買方 y 的利潤單位增量$t_y = l(y)$，而賣方 x 的利潤單位增量$s_x = -l(x)$。則由圖 4.1 可以觀察出，$t_2 = l(2) = +1$，$s_2 = -l(2) = -1$，此結果與第 4.2.1 節利用不等式所求出的結果相同。

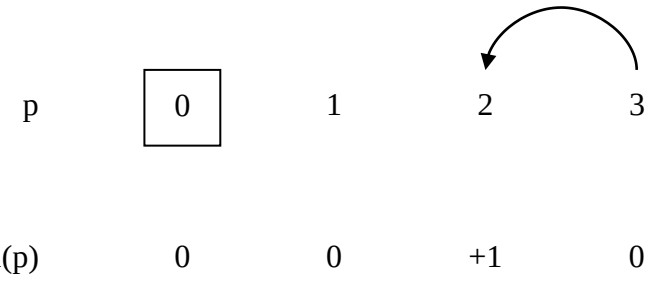


圖 4.1 有向圖

考慮第 4.2.2 節表 4.9 的滿意度矩陣，利用第 4.2.1 節的不等式方法可以求出其利潤流動為$t_1 = 1$，$t_2 = 2$，$t_3 = 1$，故$s_1 = -1$，$s_2 = -2$，$s_3 = -1$。如表 4.10 所示。

表 4.10 第一階段滿意度矩陣(單位：萬元)

座標		0	0	1	2	3	
		賣方利潤 U (買方 0)	C	D	A	B	s_i
0	買方利潤 V (賣方 0)	[0]	[0]	0*	40	0*	0
1	X	280	240	[0]	50	70	-1
2	Y	250	190	170	[0]	230	-2
3	Z	160	160	140	0*	[0]	-1
	t_j	0	0	+1	+2	+1	

今利用有向圖來重新求出利潤的流動，根據表 4.9 的滿意度矩陣可以繪製出有向圖如圖 4.2 所示。由圖形可以快速求出$t_1 = l(1) = 1$，$t_2 = l(2) = 2$，$t_3 = l(3) = 1$，且$s_1 = -l(1) = -1$，$s_2 = -l(2) = -2$，$s_3 = -l(3) = -1$。此結果與上述不等式法求解出相同的結果。因此，透過有向圖可以更快速地找出滿意度矩陣中利潤的流動關係。

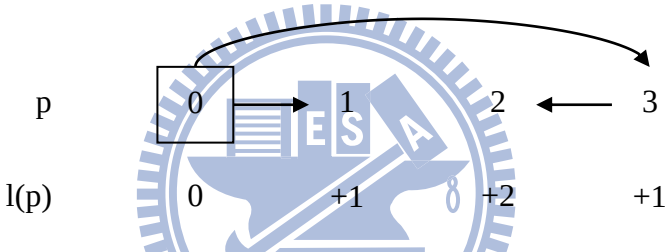


圖 4.2 第一階段有向圖

4.3.2 迴徑的融合

在舒-瑞氏演算法的有向圖中，必須滿足(1)所有節點不能指向節點 0 及(2)不可產生任何迴徑兩個條件限制。條件(1)的限制相當淺而易見，由於節點 0 所表示的買家其利潤已固定，即$t_0 = 0$。當有節點指向 0 時，會使得$t_0 > 0$而破壞利潤結構。條件(2)的限制則可用圖 4.3 說明。圖 4.3 中包含箭號 2→3 及箭號 3→2，此時節點 2 與節點 3 的最長路徑值由於迴徑的影響無法求得。就滿意度矩陣而言，表示無法同時提升{2, 3}及{3, 2}這兩個包含最小滿意度值的利潤團體。當有向圖滿足上述兩個條件限制時，稱此有向圖為正確圖形(Proper Graph)；反之，則將此圖形稱為不正確圖形(Improper Graph)。

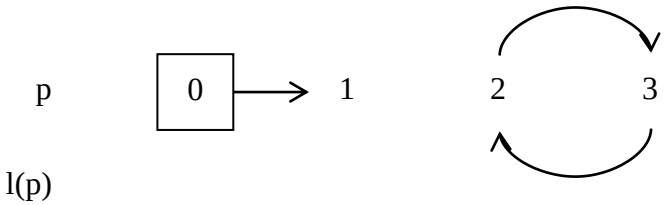


圖 4.3 包含迴徑的有向圖

對於有向圖違反限制條件時，可以利用融合節點的方式來達到正確圖形的形式。例如圖 4.3 的圖形中，將形成迴徑的節點 2 及節點 3 融合為單一節點 2 後可轉換為圖 4.4。此時圖形便達到正確圖形的形式。同理，當節點 p 指向節點 0 時，只需將節點 p 融入節點 0 成為單一節點 0 即可修正為正確圖形。

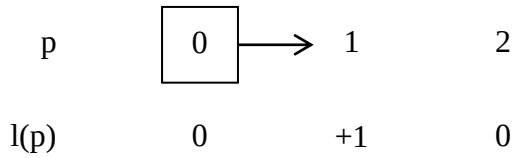


圖 4.4 融合迴徑後的有向圖

有向圖的修正，必須持續反覆直到有向圖滿足上述兩個限制條件。圖 4.5 為有向圖更新的流程圖，當有向圖建立時，必須判斷其正確性，一旦有向圖條件不滿足時，則須融合節點來修正有向圖，並再次檢查有向圖的正確性。當有向圖滿足限制條件後，即可判斷其對應滿意度矩陣的利潤流向。

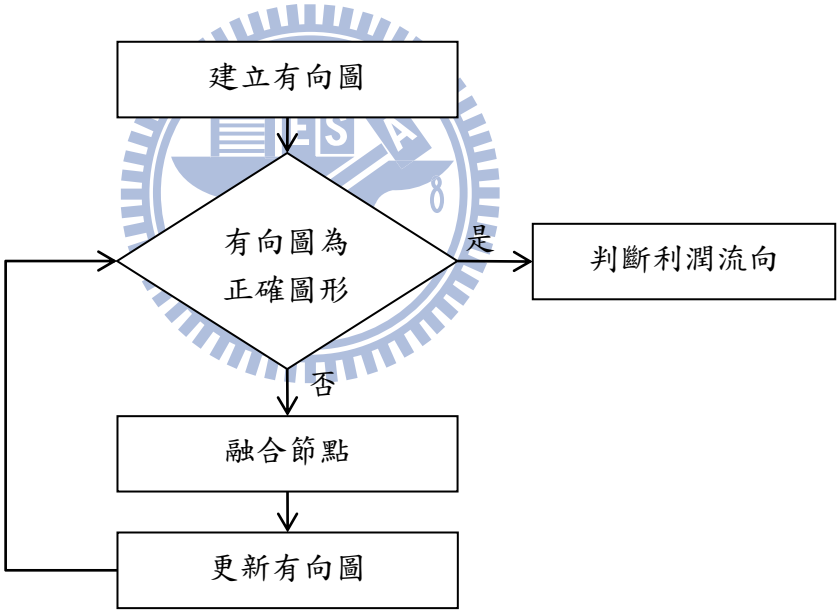


圖 4.5 有向圖更新流程

4.4 計算的簡化

對計算的簡化而言，這是將最小滿意度的利潤團體透過舒-瑞氏演算法給予極大化以達到求解斂核的目的。以表 4.1 為例，在這場公正拍賣中，共有{X、Y、Z、A、B、C、D}等 7 位成員，所以共有 $127(=2^7-1)$個部分集合及利潤團體；但由於此公正拍賣限制每一位買方最多只能得標一項拍賣物，事實上只會形成 $19(=(3+1)\times(4+1)-1)$個實際上的利潤團體，如表 4.11 的陰影部份所示。

表 4.11 實質上的利潤團體(單位：萬元)

	賣方利潤 U (買方 0)	C	D	A	B
買方利潤 V (賣方 0)	0	0	0	0	0
X	280	240	0	10	70
Y	290	230	210	0	270
Z	160	160	140	-40	0

從有向圖的角度來看，其涉及了非得標者團體間的利潤流動，即有 9(= 3 × 4 − 3)種組合的計算，如表 4.12 的陰影部份所示。

表 4.12 非得標者團體的利潤流動(單位：萬元)

	賣方利潤 U (買方 0)	C	D	A	B
買方利潤 V (賣方 0)	0	0	0	0	0
X	280	240	0	10	70
Y	290	230	210	0	270
Z	160	160	140	-40	0

因此，只要針對表 4.12 這 9 種組合來進行最小滿意度的最大化就可算出公正的得標價格。對於舒-瑞氏演算法計算斂核以達到公正得標價格的流程可分成最小滿意度的極大化與迴徑圖的處理兩個步驟來進行。最小滿意度的極大化是指調整實際利潤值 u 及 v 來改善滿意度。迴徑圖的處理是指將迴徑的節點融合來產生沒有迴徑的正確圖形。這兩個步驟的綜合使用如圖 4.6 所示。

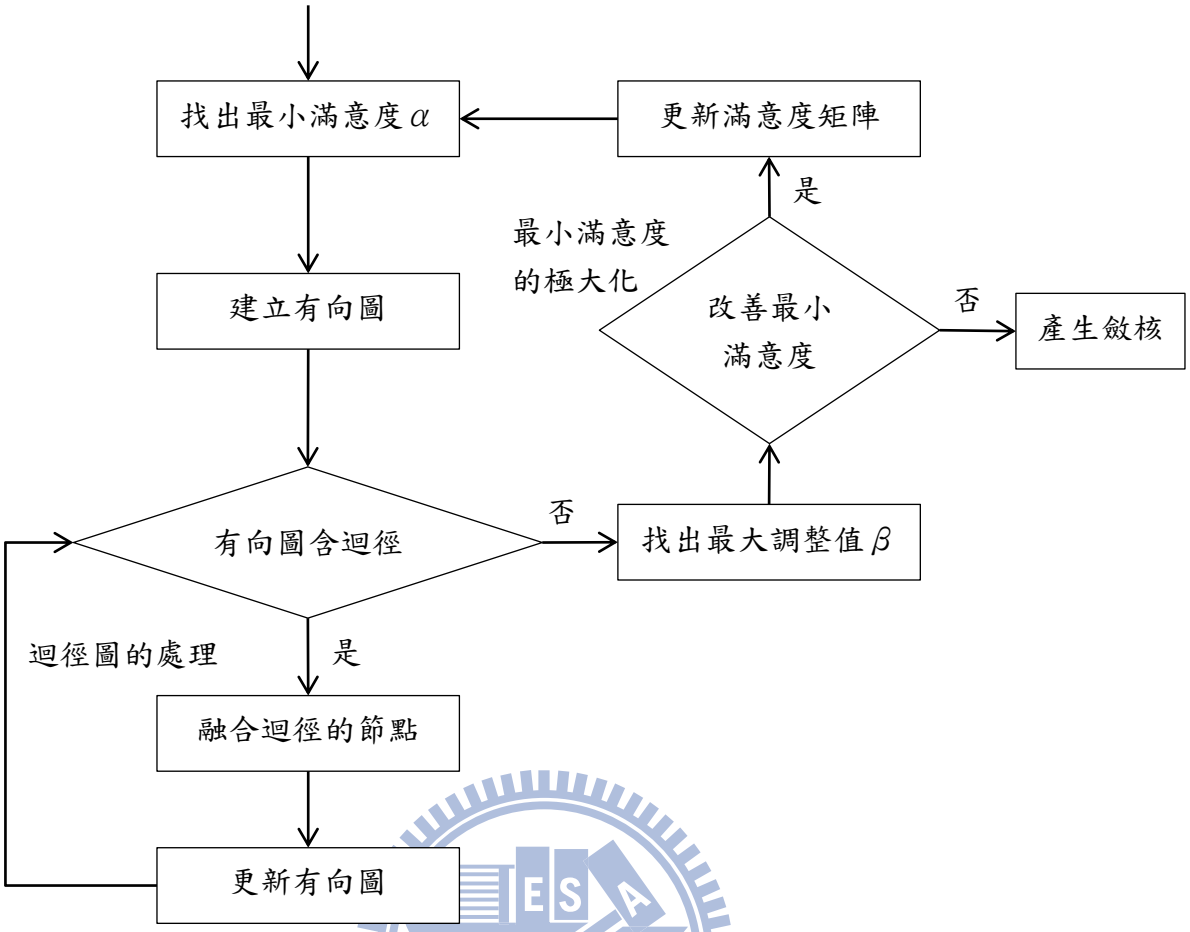


圖 4.6 計算斂核流程[13]

第五章 公正拍賣系統操作與績效分析

對指派賽局的公正拍賣而言，本實作公正拍賣系統由無線競價與公正價格開標兩套系統組成。前者採用 J2ME 技術來進行開發，而後者採用 J2SE 技術來進行開發。由於爪哇語言跨平台的特性，本公正拍賣系統可以在不同作業平台上執行，避免不同作業平台導致無法順利進行拍賣的情況產生。

在第 5.1 節首先介紹無線競價系統的操作實例。第 5.2 節說明敘核計算系統的實例操作。第 5.3 節分析利用本公正拍賣系統的效率。第 5.4 節則統整本公正拍賣系統所需的軟硬體配備。

5.1 無線競價系統操作實例

對無線競價系統而言，這包含鍵入帳號密碼、選擇拍賣場次、選擇拍賣物品、鍵入投標金額及確認投標資訊等動作。這些動作的操作介面可分成登入、投標及確認三頁面來分工，而操作時可依登入頁面、投標頁面及確認頁面順序來執行。競價者利用無線競價系統進行加密投標的作業流程如圖 5.1 所示。

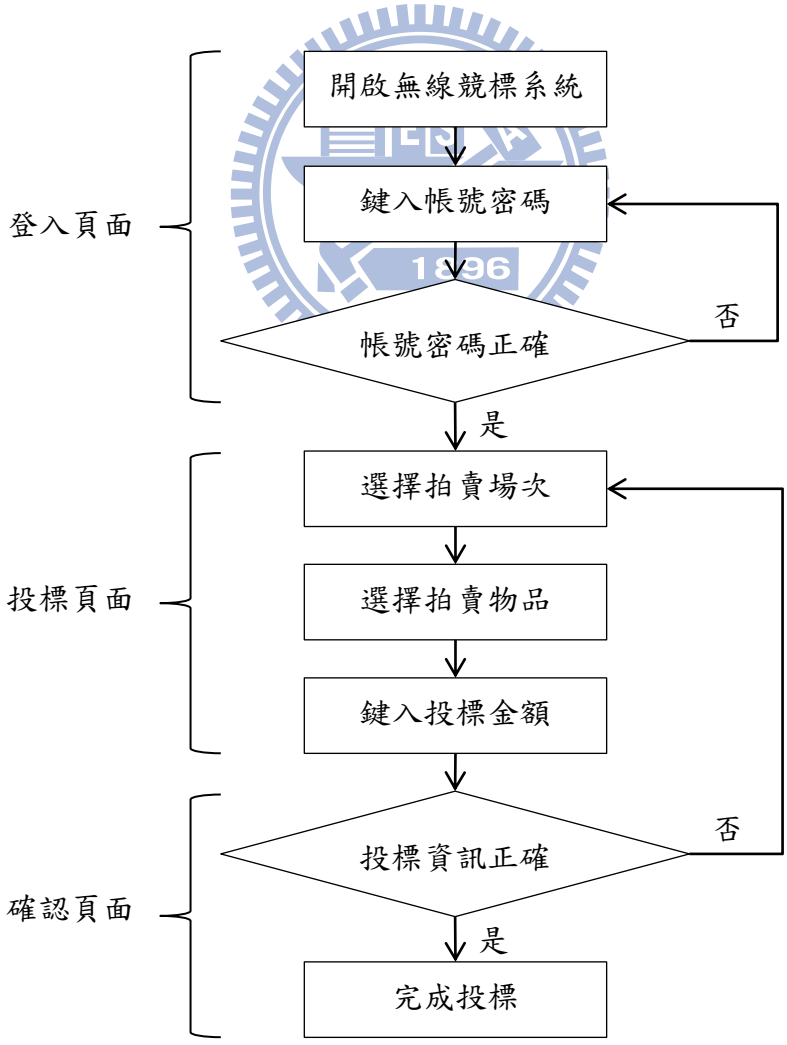


圖 5.1 加密投標的作業流程[13]

對登入頁面來說，這是由圖 5.2 的頁面進行身份辨識，也就是輸入競價者專用的帳號及密碼。更詳細來說，這包含「帳號密碼」及「空值」兩項辨識功能。對前者來說，資料庫將自動比對是否擁有此組帳號密碼。對後者來說，如果帳號密碼處未鍵入任何值，此時畫面會出現警告訊息。若無法通過身份辨識時則會出現圖 5.3 的錯誤訊息。待身份辨識通過後，競價者始能進行投標。

圖 5.2 登入頁面[13]

圖 5.3 錯誤訊息[13]

圖 5.4 投標頁面[13]

圖 5.5 確認頁面[13]

對投標頁面來說，圖 5.4 的畫面包含「拍賣場次」、「拍賣物品」及「出價金額」三個輸入選項與「物品圖片」一個輸出項。競價者先利用拍賣「拍賣場次」的下拉式選單選擇要參與競價的拍賣場次。接著由「拍賣物品」的下拉式選單選取待拍賣物品。這時「物品圖片」會出現該拍賣物品的圖片供競價者進行確認。最後競價者可於「物品圖片」下方輸入出價金額，而此輸入選項亦包含除錯機制。例如出價金額僅能輸入數字，且當無鍵入值時，畫面將出現錯誤訊息。出價金額輸入完後，再按下左下方的「送出」按鈕。

對確認頁面來說，畫面將再度呈現競價者於投標頁面所鍵入的「拍賣場次」、「拍賣物品」、「物品圖片」及「出價金額」等四項投標資訊供競價者確認。當競價者欲修改投標資訊時，可按下右下方「上一頁」按鈕來修正投標資訊；若競價者確認資料無誤後，這時可按下左下方「確認送出」按鈕即可完成投標，如圖 5.5 所示。換言之，這些投標資訊將透過秘密分享技術加密後傳送至四位公證方，無線競價系統也隨之結束。圖 5.6 顯示競價者進行無線競價的情形。

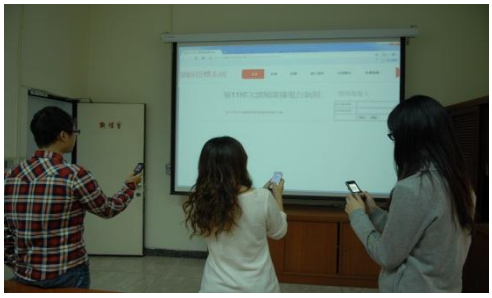


圖 5.6 無線競價的使用情形[13]

在無線競價系統的程式設計上，由於 J2ME 程式力求簡單，因此在設計上僅使用一個類別 BidderProgram 進行設計。BidderProgram 共使用 35 個變數及 9 個方法如圖 5.7 所示。

圖 5.7 BidderProgram 類別圖

對公證方資料庫系統而言，僅使用公證方加密標單一張表格來儲存標單拼圖資訊。公證方加密標單資料表設計如圖 5.8 所示。

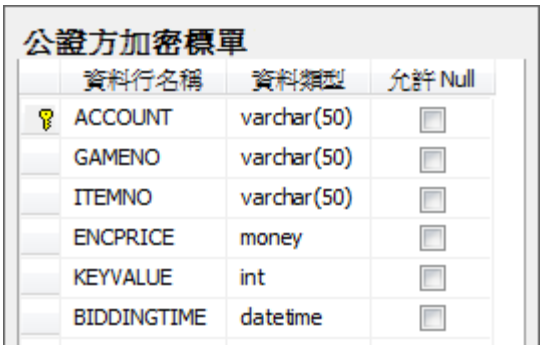


圖 5.8 公證方加密標單設計

5.2 敘核計算系統操作實例

對敘核計算系統而言，其操作主要可分為表 5.1 的三大步驟，且使用時必須依序進行。第一步驟的主要功能在確認開標方的身份，而其畫面為「安全認證」頁面。由於標單的保密性，因此必須確保開標方的帳號與密碼正確後，開標方始能進入本系統操作。第二步驟的主要功能在取得標單矩陣，而其畫面為「標單矩陣」頁面。這是透過秘密分享解密方式來解密出所有競價者的標單拼圖。第三步驟的主要功能在產生利潤矩陣與滿意度矩陣，而其畫面為「公正價格」頁面。利潤矩陣可用來產生得標者名單，而滿意度矩陣則用來產生公正價格。這裡的三大頁面是採用標籤頁面(Tabbed Panel)[16]的使用者圖形介面呈現。底下分別說明「安全認證」頁面、「標單矩陣」頁面及「公正價格」頁面的使用。

表 5.1 敘核計算系統步驟[13]

操作步驟	1	2	3
功能說明	進行身份確認	獲得標單矩陣	以滿意度矩陣獲得公正價格
頁面名稱	安全認證	標單矩陣	公正價格

就「安全認證」頁面來說，當開標系統啟動後，圖 5.9 的畫面就會自動出現。這時開標方可以輸入帳號與密碼來確認身分。當輸入完成後，這時可按下送出鍵以供認證資料庫進行驗證。當驗證成功後，系統將自動切換至「標單矩陣」頁面。

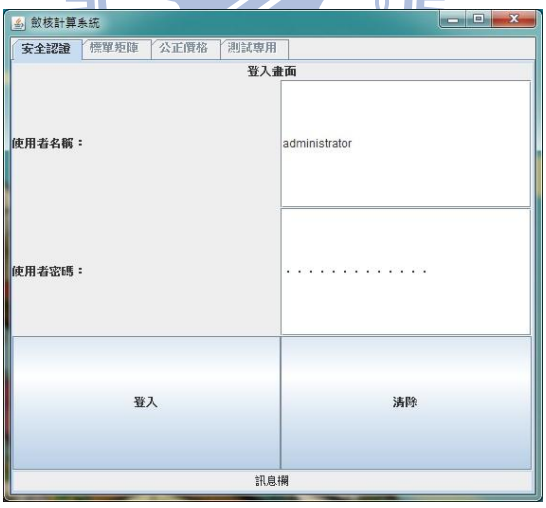


圖 5.9 「安全認證」頁面[13]

就「標單矩陣」頁面來說，該畫面包含(1)下拉式選單、(2)按鈕及(3)表格等三大物件。下拉式選單物件指的是「選擇拍賣場次」。按鈕物件則分別為「清空資料庫」、「傳回加密標單」、「開始解密標單」與「進行開標」等四個按鈕。而表格物件則為「加密標單」與「解密標單」兩個表格。如圖 5.10 所示。

圖 5.10 「標單矩陣」頁面[13]

今以表 3.6 的標單矩陣為例說明敘核計算系統所顯示的內容。在圖 5.10 中，開標方首先利用下拉式選單「選擇拍賣場次」來決定欲開標的場次，假定本次公正拍賣的場次為第 3 場，因此選定拍賣場次為 Game3。選定拍賣場次後，接著按下「清空資料庫」按鈕將所有的資料表清空以避免留下上次開標的殘留資料。再按下「傳回加密標單」按鈕，這時系統會自動向公證方發出收集加密標單的要求，而收集到的加密標單會顯示於「加密標單」表格中。

加密標單表格共有六個欄位，第一個欄位『ACCOUNT』表示競價者名稱；第二個欄位『GAMENO』表示拍賣場次編號；第三個欄位『ITEMNO』表示拍賣物品編號；第四個欄位『ENCPRICE』表示加密後的標單拼圖內容；第五個欄位『KEYVALUE』表示標單拼圖鍵值；第六也是最後一個欄位『BIDDINGTIME』表示競價者投標的時間。確定加密標單接收完成後，再按下「開始解密標單」按鈕，這時系統會將加密標單自動解密，而解密後的標單顯示於「解密標單」表格中。

在解密標單表格中，第四欄位『BIDDINGPRICE』表示競價者的投標金額。投標金額為利用秘密分享將加密標單中的標單拼圖內容及標單拼圖鍵值解密後得到。解密標單的第一行顯示出競價者 A 於第 3 場次的拍賣中，對拍賣物編號 1(拍賣物 X)的出價為 370 萬元；第二行表示競價者 A 於第 3 場次的拍賣中，對拍賣物編號 2(拍賣物 Y)的出價為 490 萬元，餘則類推。觀察後可以發現此解密標單的結果與表 3.6 相同。

在得到各個競價者的原始標單後，接著按下「進行開標」按鈕，這時畫面會自動切換至「公正價格」頁面。

圖 5.11 「公正價格」頁面[13]

就「公正價格」頁面來說，該畫面包含「功能鍵區」區塊的「列印利潤矩陣」、「求出得標者名單」、「計算斂核」及「求出公正價格」四個按鈕，及「利潤矩陣」、「滿意度矩陣」與「斂核變數區」三個反應區塊，如圖 5.11 所示。

操作時，先按下「列印利潤矩陣」的按鈕，這時系統將會自動將標單矩陣轉成利潤矩陣並顯示於「利潤矩陣」區塊中。得到利潤矩陣後，接著按下「求出得標者名單」按鈕，系統會自動求出得標者名單，並以紅色字體標示於「利潤矩陣」區塊裡。找出得標者後就可以按下「計算斂核」按鈕，此時圖 5.11 中的「滿意度矩陣」展示區與「斂核變數區」皆會出現計算結果。前者顯示出當前的滿意度矩陣，而後者則提供開標方進行各階段滿意度矩陣的比對。

在圖 5.11 下方的「斂核變數區」包含「第 i 次滿意度」、「當前最小滿意度」及「當前最大調整值」三個標題。對「第 i 次滿意度」來說，這是以下拉式選單方式提供開標方可選擇不同階段 i 的滿意度矩陣計算過程。當開標方選定欲觀察的階段後，「滿意度矩陣」展示區會顯示對應的滿意度矩陣；同樣地，在「斂核變數區」內亦會顯示對應的最小滿意度值 α 及最大調整值 β 以方便開標方進行資料查核動作。由「利潤矩陣」區塊可以觀察出，利用開標系統所得到的得標者名單與表 4.2 相同。而「滿意度矩陣」展示區所顯示的初始滿意度矩陣也與表 4.4 的滿意度矩陣相同。

當確認資料無誤後，便可按下「求出公正價格」按鈕，此時系統會跳出一個「得標資訊」的新視窗來顯示出本次的拍賣結果。在「得標資訊」視窗各有四行，第一行是拍賣物品名稱，第二行是得標者名稱，第三行是競價金額，而第四行則是最終的公正價格。例如圖 5.12 的視窗中，第二列表示拍賣物編號 1(拍賣物 X)的得標者為買方 D，其競價金額為 380 萬元，而最終斂核所求出來的公正價格約為 313.33 萬元。

圖 5.12 「得標資訊」視窗

敘核計算系統程式設計部份，以 OpenBidServer 做為主套件，其中包含三個子套件「assignment」、「openBid」與「server」。OpenBidServer 套件的內部結構如圖 5.13 所示。

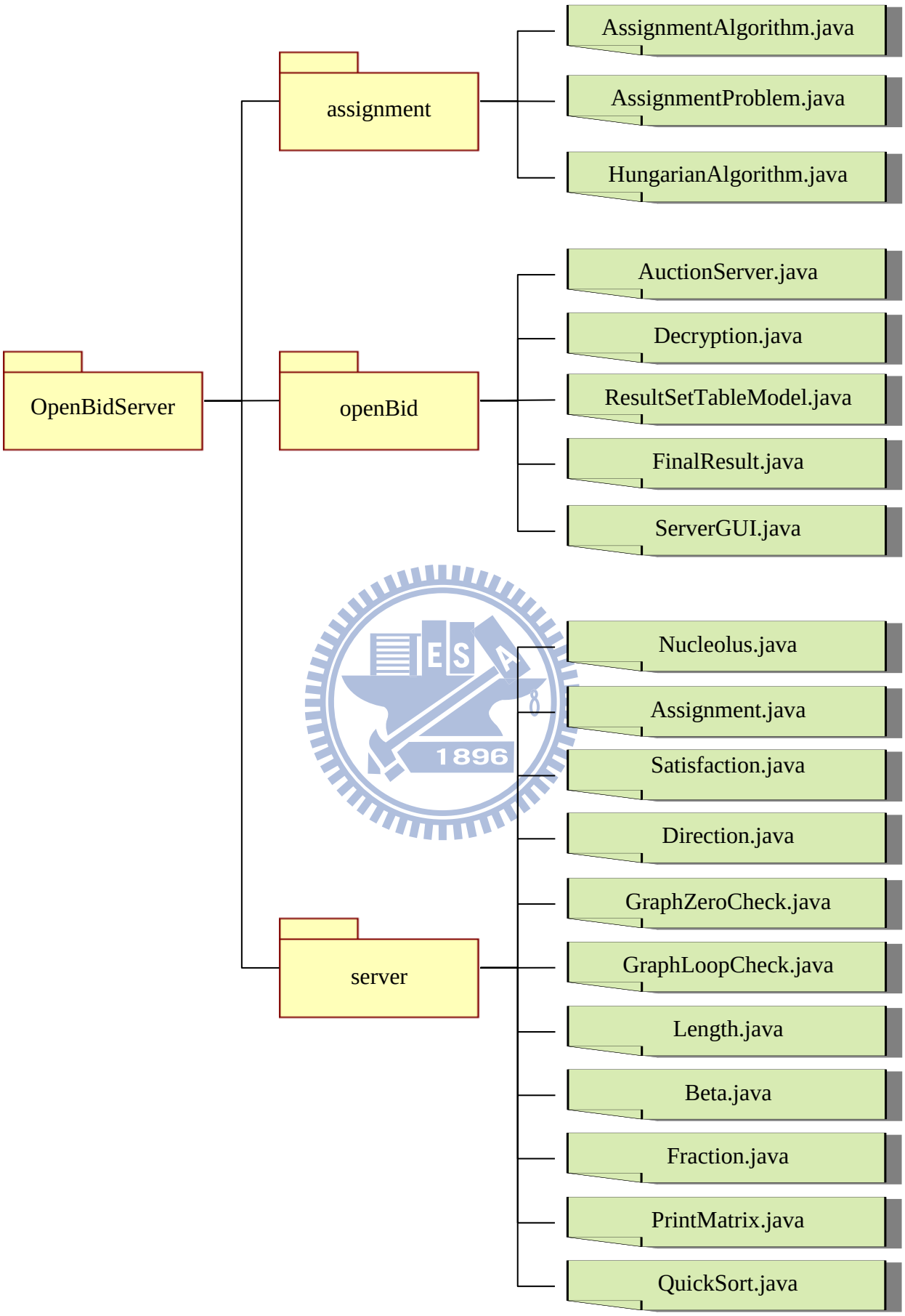


圖 5.13 OpenBidServer 套件架構

表 5.2 為 assignment 套件內容，包含類別檔案及功能簡述。表 5.3 為 openBid 套件內容。而表 5.4 則為 server 套件內容。

表 5.2 assignment 套件內容

類別名稱	功能簡述
AssignmentProblem	將標單矩陣轉為方陣
AssignmentAlgorithm	啟動 HungarianAlgorithm
HungarianAlgorithm	匈牙利人法核心

表 5.3 openBid 套件內容

類別名稱	功能簡述
AuctionServer	敘核計算系統人機介面
Decryption	解密標單拼圖
ResultTableModel	列印標單拼圖及解密標單
FinalResult	得標資訊人機介面
ServerGUI	敘核計算系統核心

表 5.4 server 套件內容

類別名稱	功能簡述
Nucleolus	舒-瑞氏演算法核心
Assignment	求出最佳指派
Satisfaction	計算滿意度矩陣
Direction	求出最小滿意度值 α 及座標
GraphZeroCheck	判斷有向圖是否有指向節點 0
GraphLoopCheck	判斷有向圖是否包含迴徑
Length	求出有向圖各節點最長路徑
Beta	求出最大調整值 β
Fraction	以分數表示計算數字
PrintMatrix	列印矩陣
QuickSort	快速排序法

對開標方資料庫而言，總共使用七張資料表來儲存拍賣資訊。開標方資料庫的雪花網要圖如圖 5.14 所示。

圖 5.14 開標方資料庫雪花網要圖

開標方資料庫的七張資料表所用功能簡述如表 5.5 所示。

表 5.5 開標方資料庫資料表

資料表名稱	功能簡述
使用者資料	儲存競價者帳號及密碼等資訊
拍賣物品資料	儲存拍賣場次及拍賣物品等資訊
加密標單	儲存由各公證方資料庫收集的標單拼圖
解密標單	儲存以秘密分享解密後的解密標單
滿意度水準	儲存各階段的最小滿意度值 α 及最大調整值 β
滿意度資料	儲存各階段所有利潤團體的滿意度
得標資料	儲存得標者、得標金額及最終公正價格

5.3 績效分析

對於公正拍賣系統的績效分析，這裡利用程式執行的時間來做為績效測量的依據。就無線競價系統來說，程式執行的時間以競價者按下「確認送出按鈕」後，系統完成加密標單並傳送給四位公證方的時間為測量時間長度。表 5.6 為無線競價系統進行 10 次時間測量的結果。

表 5.6 無線競價系統執行時間(單位：毫秒)

第 i 次測量	1	2	3	4	5	6	7	8	9	10
所需時間	92	82	71	81	81	71	71	71	82	81

由表 5.6 可以求得系統平均執行時間為 78.3 毫秒。表示競價者完成出價後，僅需約 0.08 秒的時間便可將競價者的原始標單加密為標單拼圖並傳送至四位公證方。

就敘核計算系統來說，其程式執行時間可分為(1)傳回加密標單、(2)解密標單及(3)計算敘核三個部份來測量。「傳回加密標單」部份指的是當開標方按下「傳回加密標單」按鈕至畫面出現加密標單所需的時間；「解密標單」部份指的是開標方按下「開始解密標單」按鈕至畫面出現解密標單所需的時間；而「計算敘核」部份則指的是開標系統求出敘核結果所需的時間。

案例：5 物 7 人的公正拍賣，競價金額介於 100 萬元至 1000 萬元間。如表 5.7 所示。

表 5.7 包含 5 物 7 人的標單矩陣(單位：萬元)

拍賣物	拍賣底價	A 出價	B 出價	C 出價	D 出價	E 出價	F 出價	G 出價
X	100	370	310	140	380	190	130	200
Y	200	490	220	260	280	300	400	500
Z	300	500	460	300	320	500	600	700
Z_1	400	720	500	410	900	700	800	650
Z_2	500	700	800	520	900	900	650	590

根據表 5.7 的標單矩陣可得其進行 10 次時間測量結果如表 5.8 所示。

表 5.8 程式執行時間(單位：毫秒)

第 i 次測量	傳回加密標單耗時	解密標單耗時	計算斂核耗時	總耗時
1	234	94	375	703
2	214	86	339	639
3	214	91	422	727
4	203	93	328	624
5	227	91	420	738
6	221	92	355	668
7	230	92	382	704
8	212	87	404	703
9	219	95	388	702
10	209	100	379	688

在表 5.8 中，求出平均執行時間為 689.6 毫秒。綜合上述案例的分析結果，可以了解對於斂核計算系統而言，其計算斂核價格的速度相當快，此成果大大提升利用本系統來進行公正拍賣的價值。

5.4 公正拍賣系統軟硬體配備

本公正拍賣系統的開發使用的軟體可分為四個部份。在公正拍賣系統開發上，是利用爪哇程式語言開發環境，包含爪哇環境 Java Runtime Environment(JRE)6 update 21 及整合開發環境(Integrated Development Environment，IDE)軟體 Eclipse IDE。在手機程式部份使用 J2ME 技術利用 Java Platform Micro Edition Software Development Kit 3.0 軟體進行開發。而硬體部份，在伺服端使用了 IBM 公司出產的伺服器 x3650 系列，手機部份使用了 NOKIA 系列支援爪哇環境手機。本系統所用到的軟硬體配備整理如表 5.9 及表 5.10 所示。

表 5.9 公正拍賣系統軟體需求

應用方面	使用軟體
爪哇環境	JRE 6 update 21
電腦程式整合開發環境	Eclipse IDE
手機程式整合開發環境	Java Platform Micro Edition Software Development Kit 3.0
網頁語言	JavaServer Pages(JSP)
網頁編輯	Dreamweaver CS5
資料庫	Microsoft SQL Server 2000
資料庫	Microsoft SQL Server 2008 Express

表 5.10 公正拍賣系統硬體需求

項目		規格
伺服器主機		IBM x3650
手機		NOKIA Series
桌上型電腦主機	中央處理器	Intel Pentium Dual-Core CPU E6300 2.80GHz
	主機板	ASUS P5P41T LE
	記憶體	DDR3 1333 4.00GB
	硬碟	WD SATAII 1TB
	顯示卡	NVIDIA GeForce GT 220
	螢幕	ASUS VH222D

第六章 結論

本章主要目的為說明研究結果與未來研究方向。本章共分為兩小節。第 6.1 節「結論」說明研究結果。第 6.2 節「後續研究」則說明本論文未來可繼續研究方向。

6.1 結論

本研究的目的在於找出拍賣中的公正價格，達成以無線競價方式來進行公正的拍賣，使得拍賣能夠真正滿足公平、公開及公正的環境。在現實生活中，公正的觀念很難以定量分析，而賽局理論的斂核解卻使得這樣的觀念得到解決的方法。本論文以賽局理論中核及斂核的觀念，結合勞爾斯教授所提出的公正理論，使得公正拍賣中，最小滿意度的利潤團體其滿意度最大化，進而求出公正價格。

本研究接著提出達到該目標的兩項關鍵技術，即秘密分享技術與舒-瑞氏演算法技術。前者是為了使得利用無線競價方式進行投標時，標單的內容不會外洩；後者則是為了能夠更快速地找出公正拍賣中的斂核結果，也就是公正價格。在分析完這兩項關鍵技術的各項執行細節後，本論文再提出各項執行細節的組合與操作方式。在本論文的第三章「無線競價系統的設計」與第四章「斂核計算系統的實作」便是以這兩項關鍵技術為基礎，而分別開發出無線競價與斂核計算兩套系統。

對於這兩套系統來說，第五章的「公正拍賣系統操作與績效分析」中，分別展示了無線競價系統與斂核計算系統的實例操作，而第 5.3 節的績效分析更是具體計算出此二系統的快速計算能力。無線競價系統與斂核計算系統為公正拍賣帶來相當大的便利性以及效率性。目前以這兩套系統的操作經驗來說，這已實現以無線競價方式來進行公正拍賣的原始構想。對於本論文利用舒-瑞氏與秘密分享演算法進行公正的資源分配的實現如**錯誤! 找不到參照來源。**所示。

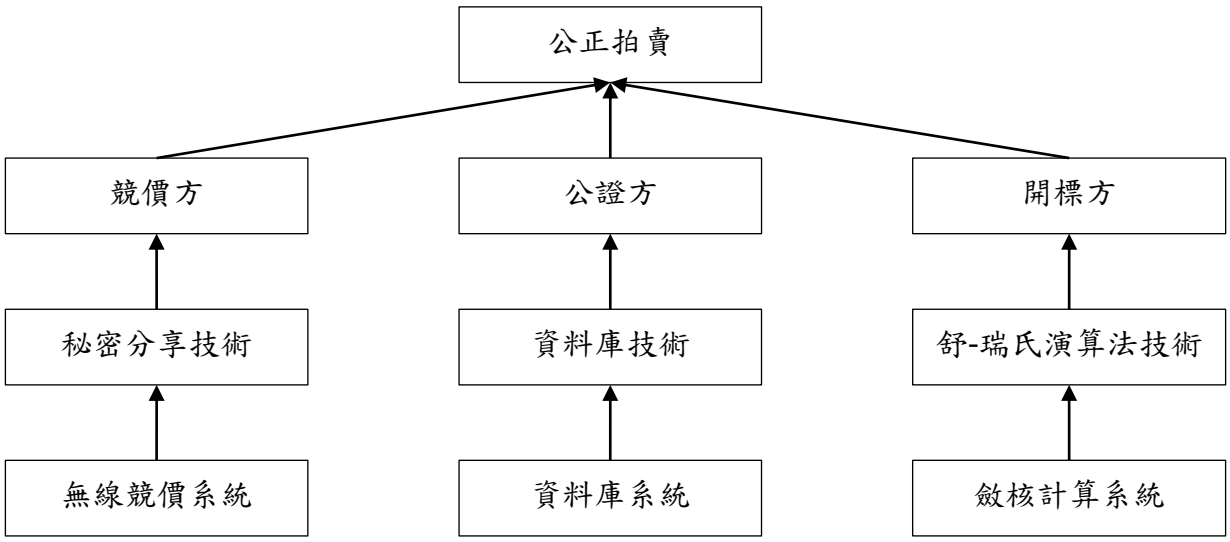


圖 6.1 公正拍賣的實現

6.2 後續研究

本研究在拍賣型式上,主要針對找出同時閉式公正拍賣的公正價格為研究方向。因此,後續研究可以針對更多不同的拍賣型式來找出拍賣結果的公正價格。

此外,在本研究的公正拍賣系統實作中,並沒有探討當有兩位以上競價者的標單完全相同時的處理方式。一般而言,在拍賣開始進行前,必須先制定完整的競價規則,這當中當然包含有發生兩位以上競價者標單相同時的處理方式。在本公正拍賣系統中,對於標單相同的情況會由系統隨機選出其中一位做為得標者,這樣的方式可能導致拍賣公平性受到質疑。因此,在公正拍賣系統實作中,若能加入所有競價者皆同意的處理方式,則拍賣的公平性便無庸置疑。

在無線競價系統的設計上,對於競價者的標單內容能夠以秘密分享達到加密的效果。然而秘密分享的缺點在於若所有標單拼圖都遭到竊取時,則標單內容將被解密開來。這樣的情況會發生在採用手機上網投標時,由於所有標單拼圖的傳送都須透過電信公司提供的手機上網服務,則電信公司就成為半路攔標的最大危險因子。因此,在無線競價系統的設計上,若能夠加入經過 RSA 法則裡的公匙(Public Key)加密後再傳送會較為安全,也令整個公正拍賣系統更加無懈可擊。

參考文獻

[1] 交郵發九十字第六三號令,「第三代行動通信業務管理規則」,交通部,2001 年 10 月。

[2] 李唯爾,「網路型投標系統的設計與實作」,國立交通大學工業工程與管理學系碩士論文,1999 年。

[3] 梁高榮,「作業研究技術幫國庫進帳 488.99 億元台幣」,工業工程雜誌,Vol. 2, No. 1, pp. 31-40, 2002。

[4] 梁高榮,「無線執照的拍賣與挑戰(上)」,通訊雜誌,93 期,82-89 頁,2001 年 10 月。

[5] 梁高榮,「無線執照的拍賣與挑戰(下)」,通訊雜誌,94 期,64-69 頁,2001 年 11 月。

[6] 梁高榮,農產品交易工程學,交大出版社,1999 年。

[7] 梁高榮,農產品批發市場的管理與自動化,財團法人農產機械化研究發展中心,1997 年。

[8] 許春田,「網路型開標系統的設計與實作」,國立交通大學工業工程與管理學系碩士論文,1999 年。

[9] 許鈞豪,「舒-瑞氏演算法在產權指派賽局的應用」,國立交通大學工業工程與管理學系碩士論文,2003 年。

[10] 許鈞豪,梁高榮,陳宗沂,「產權移轉時指派賽局的斂核計算軟體開發」,機械工業,236 期,212-224 頁,2002 年 11 月。

[11] 陳立傳,張克飛,黎秀江,精通 Java 手機遊戲與應用程式設計,文魁資訊,2006 年。

[12] 黃志泰,行動網路應用—Java 手機程式設計,文魁資訊,2009 年。

[13] 劉思宇、梁高榮,「透過無線競價進行公正拍賣」,機械工業,四月,141-151 頁,2011。

[14] 顧佳樺,「調頻廣播執照釋出問題的組合拍賣解法」,國立交通大學工業工程與管理學系碩士論文,2010 年。

[15] Barron, E. N., Game Theory: An Introduction, Wiley-Interscience, 2008.

[16] Deitel, H. M. and Deitel, P. J., Java How to Program, 7th ed., Prentice Hall, 2007.

[17] Floyd, R. W., "Nondeterministic Algorithms," Journal of the ACM, Vol. 14, Issue. 4, pp. 636-644, 1967.

[18] Gillies, D. B., "Solutions to General Non-zero-sum Games," Annuals of Mathematical Studies, Vol. 40, pp. 47-85, 1959.

[19] Haurie, A., Muto S., Petrosjan L. A., and Raghavan, T. E. S., Advances in Dynamic Games: Applications to Economics, Management Science, Engineering, and Environmental Management, Baker & Taylor Books, 2006.

[20] http://en.wikipedia.org/wiki/Hungarian_algorithm, Hungarian algorithm - Wikipedia, the free encyclopedia, 2010/10/22.

[21] http://www.java.com/zh_TW/download/faq/whatis_j2me.xml，何為 J2ME？，2010/8/3.

[22] http://www.ncc.gov.tw/chinese/, 國家通訊傳播委員會 全球資訊網, 2010/8/3.

[23] http://www.oracle.com/us/technologies/java/index.htm, Oracle and Java | Technologies, 2010/10/22.

[24] Jeffreys, H. and Jeffreys, B. S., Methods of Mathematical Physics, 3rd ed., Cambridge

University Press, 1988.
[25] Kohlberg, E., “The Nucleolus as a Solution of a Minimization Problem,” SIAM Journal on Applied Mathematics, Vol. 23, pp. 34-39, 1972.
[26] Kuhn, H.W., “The Hungarian Method for the assignment problem,” Naval Research Logistics Quarterly, Vol. 2, pp. 83-97, 1955.
[27] Owen, G., “A Note on the Nucleolus,” International Journal of Game Theory, Vol. 3, pp. 101-103, 1974.
[28] Rassenti, S. J., Smith, V. L., and Bulfin, R. L., “A Combinatorial Auction Mechanism for Airport Time Slot Allocation,” The Bell Journal of Economics, Vol. 13, No. 2, pp. 402-417, 1982.
[29] Rawls, J., “Justice as Fairness,” The Philosophical Review, Vol. 67, No. 2, pp. 164-194, 1958.
[30] Rawls, J., A Theory of Justice, The Belknap Press of Harvard University Press, 1971.
[31] Sankaran, J. K., “On finding the nucleolus of an n-person cooperative game,” International Journal of Game Theory, Vol. 19, pp. 329-338, 1991.
[32] Schemeidler, D., “The Nucleolus of a Characteristic Function Game,” SIAM Journal on Applied Mathematics, Vol. 17, No. 6, pp. 1163-1170, 1969.
[33] Shamir, A., “How to Share a Secret,” Communication of the ACM 22, pp. 612-613, 1979.
[34] Shapley, L. S. and Shubik, M., “The Assignment Game I: The Core,” International Journal of Game Theory, Vol. 1, pp. 111-130, 1972.
[35] Solymosi, T. and Raghavan, T. E. S., “An Algorithm for Finding the Nucleolus of Assignment Games,” International Journal of Game Theory, pp. 119-143, 1994.
[36] Von Neumann, J. and Morgenstern, O., Theory of Games and Economic Behavior, Princeton University Press, 1944.

附錄一 深度優先搜尋法程式碼

1. 程式執行流程

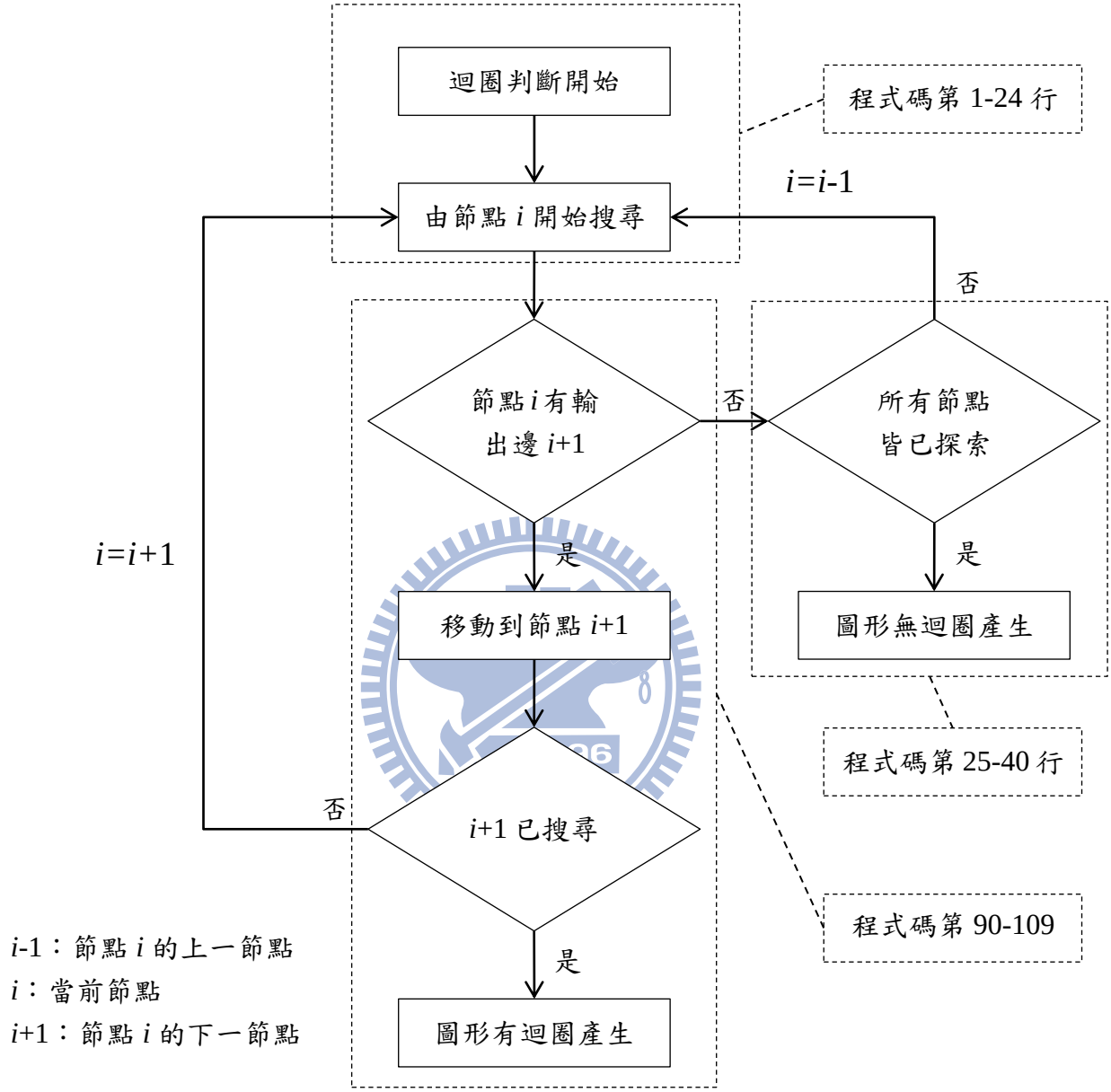


圖 A.1 深度優先搜尋流程圖

2. 原始程式碼

```
1  package server;
2
3  public class Cycle1 {
4
5      int[] visit; // 造訪矩陣
6      int endNode;
7      int counter = 0;
8      int having = 0;
9
10     int alpha[][];
11     float s_Matrix[][];
12
13     public Cycle1(int a1[][], float s1[][]) {
14
15         int exMatrix[][]; // 儲存擴增節點矩陣
16
17         while (having != 2) {
18             printData("初始節點座標", a1);
19             exMatrix = extendedMatrix(a1); // 將初始節點座標陣列轉換成擴增節點
20
21             for (int i = 0; i < exMatrix.length; i++) {
22                 System.out.println("由" + i + "點出發找尋迴徑");
23                 visit_init(visit); // 初始化訪問節點矩陣
24                 findCycle(i, exMatrix);
25                 if (having == 1) {
26                     int cycleNode[] = new int[counter];
27                     for (int j = 0; j < visit.length; j++) {
28                         if (visit[j] >= visit[endNode]) {
29                             cycleNode[visit[j]] = j;
30                         }
31                     }
32                     CycleTest.changeToZero(s1, a1, cycleNode, endNode);
33                     a1 = CycleTest.noZero(a1, endNode);
34                     a1 = CycleTest.compare1(a1);
35                     CycleTest.changeSNode(s1, cycleNode, endNode);
36                     having = 0;
37                     break;
38                 } else {
39                     having = 2;
40                 }
41                 counter = 0; // 重新初始化計數器
42             }
43
44         }
45         setAlpha(a1);
46         setSatisfaction(s1);
47         System.out.println("已無迴徑產生");
48
49     }
50
51     public int findMax(int maxData[][]) { // 找出陣列中最大值
52         int number = 0;
53         for (int i = 0; i < maxData.length; i++) {
54             for (int j = 0; j < maxData[0].length; j++) {
55                 if (number < maxData[i][j]) {
56                     number = maxData[i][j];
57                 }
```

```
58                      }
59              }
60              return number;
61      }
62
63      public int findMax2(int maxData[]) {
64              int number = 0;
65              for (int i = 0; i < maxData.length; i++) {
66                      if (number < maxData[i]) {
67                              number = maxData[i];
68                      }
69              }
70              return number;
71      }
72
73      public int[][] extendedMatrix(int initial[][]) { // 建立擴增節點矩陣
74              int maxLength; // 儲存矩陣長度
75              maxLength = findMax(initial); // 找出矩陣長度
76              visit = new int[maxLength + 1]; // 宣告造訪矩陣長度
77
78              int exMatrix[][] = new int[maxLength + 1][maxLength + 1]; // 宣告擴增節點矩
79
80              for (int i = 0; i < initial.length; i++) {
81                      exMatrix[(int) initial[i][0]][(int) initial[i][1]] = 1; // 將座標矩陣轉換成有向圖矩陣，例如：
    (3, 2)即在有向圖陣列(3, 2)處標註 1
82              }
83
84              printData("擴增節點矩陣", exMatrix);
85              return exMatrix;
86      }
87
88      public void findCycle(int v, int exMatrix[][]) { // 找出迴徑
89
90              visit[v] = counter; // 由 v 點出發
91              for (int i = 0; i < exMatrix[0].length; i++) {
92                      if (exMatrix[v][i] == 1) {
93                              if (visit[i] == -9) {
94                                      counter++;
95                                      findCycle(i, exMatrix);
96                                      break;
97                              } else { // 已找到一個 cycle,for 迴徑停止
98                                      System.out.print("有迴徑產生," + "結束節點為：" + i + ",");
99                                      endNode = i;
100                                     counter++;
101                                     printData2("路徑如下：", visit);
102                                     having = 1; // 當有迴徑產生時，having 布林值為 true
103                                     break;
104                             }
105                     }
106             }
107     }
108
109     public void combineNode(int endNode) {
110             float cbNode[];
111             int max = findMax2(visit);
112             cbNode = new float[max];
113
114             for (int i = 0; i < max; i++) {
115                     // if ( )
116             }
```

```
117         }
118
119         public void visit_init(int v[]) { // 初始化訪問結點
120             for (int i = 0; i < v.length; i++) {
121                 v[i] = -9;
122             }
123         }
124
125         public void setAlpha(int a2[][]) {
126             alpha = a2;
127         }
128
129         public int[][] getAlpha() {
130             return alpha;
131         }
132
133         public void setSatisfaction(float s2[][]) {
134             s_Matrix = s2;
135         }
136
137         public float[][] getSatisfaction() {
138             return s_Matrix;
139         }
140
141         public void printData(String title, int data[][]) { // 列印二維陣列
142             System.out.print(title);
143             for (int i = 0; i < data.length; i++) {
144                 System.out.println();
145                 for (int j = 0; j < data[0].length; j++) {
146                     System.out.print(data[i][j] + ",\t");
147                 }
148             }
149             System.out.println();
150         }
151
152         public void printData2(String title, int data[]) { // 列印一維陣列
153             System.out.println(title);
154             for (int i = 0; i < data.length; i++) {
155                 System.out.print(data[i] + ",\t");
156             }
157             System.out.println();
158         }
159 }
```

附錄二 公正拍賣系統網站簡介

在公正拍賣系統中，競價者必須透過手機進行投標的動作。而公正拍賣系統網站則提供競價者於線上註冊個人帳號，以取得參與競價的資格及無線競價系統程式。

公正拍賣系統網站的網址為 http://auction.nctu.edu.tw:8080/Justice/index.jsp。當使用者於瀏覽器鍵入上述網址後，瀏覽器畫面會進入公正拍賣系統網站的首頁，如圖 B.1 所示。

圖 B.1 公正拍賣首頁

在公正拍賣首頁中，已註冊過的使用者可以直接鍵入帳號密碼登入，以下載無線競價系統程式。而第一次進入本站的使用者，可點選連結『註冊』進入『使用者註冊』畫面，如圖 B.2 所示。

圖 B.2 使用者註冊畫面

在使用者註冊畫面，使用者必須輸入包含『使用者名稱』、『使用者 e-mail』、『使用者電話』、『使用者帳號』和『使用者密碼』共五個欄位。使用者名稱、使用者 e-mail 和使用者電話為使用者的基本資料，而使用者帳號及使用者密碼則做為日後登入系統及競價者身分辨別之用。

圖 B.3 註冊成功畫面

圖 B.3 為註冊成功畫面。當使用者所註冊的帳號密碼已存在時，網頁會出現註冊失敗的訊息，如圖 B.4 所示。

圖 B.4 註冊失敗畫面

使用者成功註冊後，可重新於公正拍賣首頁鍵入註冊時所輸入的帳號密碼，網站將自動檢查帳號密碼是否正確，確認帳號密碼無誤後，網站將顯示登入成功畫面如圖 B.5 所示。

圖 B.5 登入成功畫面

使用者成功登入後，可點選『前往下載頁面』連結，網站將導入無線競價系統下載的網頁如圖 B.6 所示。

圖 B.6 無線競價系統下載畫面

在無線競價系統下載的網頁中，分為左右兩個部份。圖 B.6 左方為 J2ME 版本的無線競價系統，任何支援 JAVA 語言的手機都可以下載此版本進行安裝。而圖 B.6 右方則為 Android 版本，是為 Android 系統手機所量身訂做的無線競價系統。

在這裡特別需要注意的是，本無線競價系統需要 Android 1.6 以上版本才可支援。檔案下載完成後，使用者可按下畫面下方『登出』連結，以登出此網站。登出成功訊息如圖 B.7 所示。

圖 B.7 登出成功畫面

使用者將檔案下載至電腦後，可利用手機傳輸線將檔案傳入手機並安裝。此外，也可利用手機無線上網的功能，利用手機連結本網站，將檔案直接下載至手機內並安裝。安裝完成後，便可執行無線競價系統程式。