# 以梯度投影法結合內點運算與單體法之研究

莊文華 陸行*
政治大學應用數學系
116 台北市文山區指南路二段 64 號

## 摘要

線性規劃的求解方法依幾何觀點可區分爲單體法(Simplex method)和起源於 Karmarkar 的內點法(Interior-point algorithm)，無論在理論上或應用上這兩者的許多變體仍然不斷地在改進中。儘管內點法求解速度較單體法爲快，但內點法所求得之最佳解，僅是一極爲靠近最佳點的解，並非實際的最佳解。因超平面分割定理(Separating hyperplane theorem)告訴我們，一線性規劃問題的最佳解一定發生在邊界點上，而 Karmarkar 的內點法及其變體的演算法，所得之最佳點卻是可行區域的一個內點，也就是說這些演算法在靠近最佳點時會產生 Z 字形的收斂過程，而無法到達最佳點。因此，在內點法提出之後，即有許多結合點法與單體法的研究，討論如何改進演算效率的問題。本文主要工作在研究結合內點法與單體法之演算法，提出如何利用 QR 分解的特性解決問題，討論放寬一般求解線性規劃問題之演算法中一重要假設：限制式之係數矩陣 A 必須爲滿秩，最後分析此方法的複雜度，確保所需計算量之上限爲 $O(n^3)$。

關鍵詞：線性規劃，梯度投影法，內點法，單體法，QR 分解

## 1. 序論

### 1.1 研究動機與目的

從 Terlaky[13]的文獻當中，得知利用內點法求解線性規劃問題時，欲產生一較容易求最佳解之起始點(Warm start)的研究工作仍持續地在進行，但成果進展地很慢，且對於線性規劃問題之敏感度分析(Postoptimal analysis)，仍無法和單體法競爭。故實際上到目前爲止，當用內點法尋找有界凸多面體(Polytope)之一近似解析中心(作爲起始點)時，僅有一些特定結果。再者，當利用內點法作敏感度分析時，例如確認一線性規劃問題的最佳基底(Optimal basis)，其求解往往遭遇到許多困難，而借用單體法求解，似乎是現今最好的選擇。

這種利用內點法求解線性規劃問題，並確認一線性規劃問題的最佳基底的方法，可在 Megiddocite[11]、Bixby 等人[3]、Vavasis 和 Ye[17]及 Luh 和 Tsaih[10]的文獻中找到演算法的例子。不謀而合的是，這些演算法都利用投影法解決此一問題，只是所建構投影矩陣的方式有所差異。而這種將梯度投影至可行解區域產生可行之搜尋方向，用以計算具有線性限制式之非線性規劃問題的演算法：梯度投影法(Gradient Projection Method; GPM)，最早是由 Rosen[12]所提出。其運用於求解線性規劃問題之技巧則因人而異，一般而言，此技巧必須透過線性轉換以滿足投影後之方向爲可行且得以改進目標函數值的需求，到了最近才由 Luh 和 Tsaih[10]在不透過線性轉換的情況下，利用其所建構之投影矩陣，以達成結合內點法與單體法產生新演算法的目的。此演算法不同於過去利用內點法求解最佳解的原理：使用一個可行解區域內部的點(或稱爲中心)作爲每次疊代的起點；而改以可行解區域的邊界點爲起點，然後建構一內點搜尋方向，以改進目標函數值，最後利用投影法解決內點法 Z 字形的收斂過程，並找出一基本可行解(Basic feasible solution)，搭配單體法求解。其研究結果，使用此種輔助單體法之有效搜尋法求解線性規劃問題，要比直接用單體法求解所需之疊代數，平均值約降低 40%，且整個投影過程所需疊代數不超過變數個數。

將搜尋方向投影在邊界上，然後沿著邊界上移動改進目標函數值，和 Karmarkar[9]的演算法利用

*連絡人: paul@math.nccu.edu.tw

投影最陡峭的下降搜查方向有所不同，最明顯的差別在於有效搜尋法是求得一基本可行解， 而Karmarkar[9]的演算法則是沿內點方向移動且目的在於求得一靠近最佳點之內點，這其中所可能遭遇的問題亦不盡相同。另外雖說整個投影過程所需疊代數不超過變數個數，但計算量若隨著所需疊代數增加而增加，則使得這種新演算法不具實用價值。因此，探討搜尋方向投影在邊界上所可能遭遇的問題，以及尋求一數值計算方法使得整個投影過程之計算量不隨著所需疊代數而改變，是本文主要的研究目的。

### 1.2 研究架構與範圍

我們考慮求解線性規劃問題標準式：

$$\begin{aligned} \min\ & \tilde{c}^T x \\ s.t.\ & \tilde{A}x \geq \tilde{b}, \\ & x \geq \mathbf{0}, \end{aligned} \tag{1}$$

其中，$\tilde{A}$ 是一秩爲 m 的 m×n 矩陣(m ≤ n)，且 $\tilde{c}, x \in \Re^n$ 是二行向量，$\tilde{b} \in \Re^m$ 也是行向量。

將(1)式問題形式單位化後得(2)式，亦即 $c \equiv \frac{\tilde{c}}{\|\tilde{c}\|}$，$A_{i,:} \equiv \frac{\tilde{A}_{i,:}}{\|\tilde{A}_{i,:}\|}$，$b_i \equiv \frac{\tilde{b}_i}{\|\tilde{A}_{i,:}\|}$ 其中≡在本文意指定義之意，爲 $\boldsymbol{A}$ 第 $i$ 列，$\tilde{b}_i$ 爲 $\tilde{b}$ 第 $i$ 個元素，$\|\cdot\|$ 記作歐氏模。檢查(1)和(2)式，很容易發現它們有著相同的可行解及最佳解。

$$\begin{aligned} \min\ & c^T x \\ s.t.\ & Ax \geq b, \\ & x \geq \mathbf{0}, \end{aligned} \tag{2}$$

在不失一般性的條件下，本文將(2)式當成標準式。

**定義 1**：令 $F \equiv \{ x : Ax \geq b, x \geq 0 \}$ 爲(2)式之可行解區域。若 $y$ 是可行解且 $y \in \{ x : Ax > b,\ \ x > 0 \}$，則稱 $y$ 爲 $F$ 之內點；反之，則稱爲 $F$ 之邊界點。

本文所提搜尋方向之「可行性」意指改進方向指向可行解區域，至於研究重點和一般內點法最大的不同是每一疊代所得之可行解，皆爲一邊界點，$\boldsymbol{x}^{(k)}$ 表示第 $k$ 次疊代的可行解。爲簡化問題，假設(1)式有可行解且最佳解有界，並限制在非退化解情況下討論。

針對上述研究重點及範圍，首先整理前人所做過的相關研究，且設計程式，實際觀察搜尋方向在邊界上可能會發生的問題。第 3 節尋找問題發生的原因並提供解決的方法。第 4 節探討如何利用 QR 分解的特性，使得整個投影過程的計算量爲 $O(\mathrm{n}^3)$。

## 2. 文獻回顧

本節首先回顧梯度投影法(GPM) [12]，並探討將其應用在求解線性規劃問題時可能遭遇的問題。其次介紹仿射演算法(Affine scaling algorithm)[1,13,19]及有效搜尋方法(Efficient search direction) [10]兩種全然不同之內點演算法，然後比較兩者的差異性，並觀察投影目的及方法之不同處，藉以點出探討有效搜尋法投影過程計算量之重要性。

### 2.1 梯度投影法

利用梯度作爲搜尋方向是求解無限制式的最佳化問題之一種很自然的方法，但對於具有限制式之問題來說，由於梯度方向可能指向可行解區域外，所以無法執行求解工作。1960 年 Rosen [12]首先提出克服此困難之技巧：將梯度投影至可行解區域內以獲得可行之求解方向。其考慮之標準式同(1)式，並定義：

$$\begin{aligned} &\Omega_1^{(k)} = \left\{ i : A_{i,:} x^{(k)} = b_i \right\} \\ &\text{與 } \Omega_2^{(k)} = \left\{ j : x_j^{(k)} = 0 \right\} \end{aligned} \tag{3}$$

其中 $\boldsymbol{A}_{i,:}$ 表 $\boldsymbol{A}$ 第 $i$ 列，以及

$$M^{(k)} = I - \left(V^{(k)}\right)^T \left[ V^{(k)} \left( V^{(k)} \right)^T \right]^{-1} V^{(k)} \tag{4}$$

爲一投影在 $V^{(k)}$ 核空間之投影矩陣；其中 $\boldsymbol{I}$ 是一 n×n 之單位矩陣，$V^{(k)}$ 是由 $A_{i,:}, \forall i \in \Omega_1^{(k)}$ 及 $e_j^T$, $\forall j \in \Omega_2^{(k)}$ 等列向量組合而成，$\boldsymbol{e}_j$ 表爲 n×n 單位矩陣 $\boldsymbol{I}$ 的第 $j$ 行。

**引理 1**：令 $\eta \in \Re$，$\eta > 0$，$d = -M^{(k)} c$ 且 $\boldsymbol{x}^{(k)}$ 爲一邊界點，若 $x^{(k+1)} = x^{(k)} + \eta d$，則 $x^{(k+1)}$ 亦爲一邊界點。

證明：證明分做 $V^{(k)}$ 爲滿秩和不滿秩二部份討論。

1. 設 $V^{(k)}$ 爲滿秩

   由 $V^{(k)} M^{(k)}$

   $$= V^{(k)} - V^{(k)} \left(V^{(k)}\right)^T \left[ V^{(k)} \left(V^{(k)}\right)^T \right]^{-1} V^{(k)}$$

$= V^{(k)} - V^{(k)}$

$= O$ 零矩陣

得 $V^{(k)}\ x^{(k+1)}$

$= V^{(k)}x^{(k)} + \eta V^{(k)}d$

$= V^{(k)}x^{(k)} + 0$

$= V^{(k)}x^{(k)}$

故得 $\boldsymbol{x^{(k+1)}}$ 爲一邊界點。

2. 當 $V^{(k)}$ 不爲滿秩時，留待第 4 節，利用定理 1 可推得相同之結果。

因此，令 $\eta \in \Re$ 且 $\eta > 0$ ，若 $d \neq 0$ ，由 $x^{(k+1)} = x^{(k)} + \eta d \neq x^{(k)}$ ，我們得到 $cx^{(k+1)} < c^T x^{(k)}$ ，也就是說 $\boldsymbol{d}$ 爲邊界上降低目標函數值的可行方向。由此可觀察出，此演算法和單體法的共同點是沿可行解區域邊界上移動改進目標函數值，所不同的是單體法每一步皆是由角點移動到另一個角點，梯度投影法則不完全如此。下面先簡單地介紹簡述該演算法的求解過程，其證明過程詳見[2]：

步驟一：設 $x^{(k)}$ 爲此演算法第 $k$ 疊代所得之可行解區域邊界上之一點。

步驟二：計算 $\Omega_1^{(k)}$、$\Omega_2^{(k)}$ 及 $M^{(k)}$ （如(3)與(4)式所定義）。

步驟三：令 $u = -\left[ V^{(k)}\left(V^{(k)}\right)^T \right]^{-1} V^{(k)}c$ 若 $-M^{(k)}c = 0$ 且 $u \leq 0$ ，則若且唯若 $x^{(k)}$ 爲 Kuhn-Tucker 點（即最佳解），故停止執行。

步驟四：若 $-M^{(k)}c \neq 0$ ，則沿 $-M^{(k)}c$ 方向移動求得 $x^{(k+1)}$ 令 $k = k+1$ 回步驟二；否則，執行步驟五。

步驟五：令 $u_q = \max\{u_j \mid u_j$ 是的 $\boldsymbol{u}$ 第 $j$ 個元素 $\}$ ，$\overline{V}^{(k)}$ 爲 $V^{(k)}$ 刪除第 $q$ 列所構成之矩陣，

令 $\overline{M}^{(k)} = I - \left(\overline{V}^{(k)}\right)^T \left[ \overline{V}^{(k)}\left(\overline{V}^{(k)}\right)^T \right]^{-1} \overline{V}^{(k)}$

，再沿 $-M^{(k)}c$ 方向移動求得 $\boldsymbol{x^{(k+1)}}$ ，令 $k = k+1$ 回步驟二。

自從 Rosen[12]的演算法發表後，即吸引了很多學者的目光，這些學者的研究重點在於求解具線性限制式之非線性規劃問題時，其總體性收斂(Global convergence)之理論證明。至於利用於求解線性規劃問題，則無論是期刊文獻，或是以線性規劃爲主軸的書籍，都乏人提及，作者認爲其原因有三：

1. 如同單體法一樣，其求解速度與起始點之給定有關；換句話說，求解所需疊代數會隨著起始點與最佳點的距離增加而增加。
2. 投影所需計算量相當可觀。
3. 在步驟五當中，爲保證 $-\overline{M}c \neq 0$ ，必須假設 $V^{(k)}$ 爲滿秩。

這些是利用梯度投影法求解線性規劃問題所可能遭遇之棘手問題，至於解決方法，將分別於第 2.3、4 及 5 節討論。

## 2.2 仿射演算法

此演算法考慮之標準式與(1)式相較，差異在於限制式爲一等式（即 $Ax=b$）。首先，假設給定一個初始可行的內點 $x^{(0)}\left(x^{(0)} > 0\right)$ 性意味著 $Ax^{(0)} = b$ 。

其次，我們想要找到一步伐方向向量 $d$ ，它使得在下降方向得一新的點 $x^{(new)}$ 並保持可行性。如果透 $x^{(new)} = x^{(0)} + d$ 過獲得新點，並維持可行性，則得

$$Ax^{(new)} = A\left(x^{(0)} + d\right) = Ax^{(0)} + Ad = b$$

但是，因爲 $Ax^{(0)} = b$ ，它導出

$$Ad = 0 \tag{5}$$

此外，如果步伐方向向量引導在下降方向，則下面的條件必須滿足 $c^T x^{(new)} \leq c^T x^{(0)}$ ，因此由

$$c^T x^{(new)} = c^T\left(x^{(0)} + d\right) = c^T x^{(0)} + c^T d \leq c^T x^{(0)}$$

導出

$$c^T x^{(0)} \leq 0 \tag{6}$$

因此，所得之步伐向量 $\boldsymbol{d}$，必須同時滿足(5)和(6)兩個限制式。也就是說，步伐向量 $\boldsymbol{d}$ 第一個要求的是屬於矩陣 $\boldsymbol{A}$ 的核空間，而第二則是與目標向量 $\boldsymbol{c}$ 的內積小於等於零。

現在考慮我們先前與梯度的相反的方向走一步的假設。很明顯，僅當移動後所停留點爲可行解以及同時能降低目標函數值時是有效的方向。因此，如果步伐方向滿足(5)和(6)式，這是有效方向；否則必須改進它。我們下一步考慮如何改進梯度方向使得(5)和(6)二式皆滿足。如同由(5)式需要，步伐方向向量必須屬於 m×n 矩陣 $\boldsymbol{A}$ 的核空間。透過

$$N(A) = \left\{x : Ax = 0, x \in \Re^n\right\}$$

定義了此空間上一投影算子 n×n 矩陣 $P$，

$$P = I - A^T\left(AA^T\right)^{-1}A$$

之所以稱作投影算子是因爲它將任意給定 n 維向量轉換成另一個矩陣 $A$ 之核空間中的向量。意即，如果 $y = Px$ ，則對任意向量 $z$ 而言很容易驗證 $Ay = 0$ ，其中 $y$ 屬於矩陣 $A$ 之核空間。這個投影矩陣，除滿足 $AP = O$ （$O$ 爲一 m×n 之零矩陣）外，另有下列兩種性質：

$$P = P^2 \text{ 及 } P = P^T \tag{7}$$

用此投影算子 $P$ 來投影向量 $-c$ ，透過 $d = P(-c)$ 找到了移動方向，因此，將這兩種性質用於(7)我們得

$$c^T d = -c^T Pc = -c^{(T)}P^2c = -(Pc)^T(Pc) = -\|Pc\|^2 \le 0$$

並且它的確是(6)需要的下降方向。

目前爲止我們展示一個給定的可行和內點方向，此方向是將梯度 $\boldsymbol{c}$ 投影在矩陣 $\boldsymbol{A}$ 的核空間上，然後沿投影後的相反方向移動，使得降低目標函數值和保持可行性。

1967 年原始仿射比例(Primal affine scaling)的基本概念首先由 Dikin [4]提出，並且在 1974 年[5]給了演算法收斂性的證明；然而，Dikin 並未提及「中心」的概念。到了 1986 年，Vanderbei 等人[15]重新研究線性規劃，他們利用 Huard[8]在非線性規劃中刻劃中心的方法，定義仿射比例(Affine scaling)用以解決線性規劃問題，此方法視爲在 Karmarkar 革命性的工作之後的一個變體。

下面將介紹此演算法的求解流程：

步驟一：給定一起始點 $x^{(0)}$ 使得， $Ax^{(0)} = b,\ x^{(0)} > 0$ ，及非常接近 0 之正實數 $\boldsymbol{\varepsilon}$ ，令 $k = 0$ 。

步驟二：定義 $D^{(k)} \equiv diag\left[x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, ...., x_n^{(k)}\right]$ 其中 $x_i^{(k)}$ 爲 $x^{(k)}$ 之第 $i$ 個元素。

步驟三：利用 $\left[A\left(D^{(k)}\right)^2A^T\right]y^{(k)} = A\left(D^{(k)}\right)^2c$ ，求出 $y^{(k)}$ ，其中 $y^{(k)} \in \Re^{\mathrm{m}}$

步驟四：計算 $z^{(k)} = c - A^Ty^{(k)}$ 及 $dx^{(k)} = -\left(D^{(k)}\right)^2 z^{(k)}$ 。

步驟五：由 $\alpha = \left\{-x_i^{(k)} / dx_i^{(k)} \mid \forall dx_i^{(k)} \le 0,\ 1 \le i \le \mathrm{n}\right\}$ ，求出 $x^{(k+1)} = x^{(k)} + \rho\alpha dx^{(k)}$ ，其中 $0 < \rho \le 1$ 。

步驟六：當 $\left\|c^Tx^{(k+1)} - b^Ty^{(k)}\right\| / \left(1 + \left\|c^Tx^{(k+1)}\right\|\right) \le \varepsilon$ 時，則停止，否則回步驟二。

## 2.3 有效搜尋法

此演算法標準式同(2)式，研究重點在於探討從可行解區域之邊界點出發，以內點法爲搜尋方向，並設法結合單體法的特性，產生有別於其它內點法之新演算法。然而當以內點法爲搜尋方向時，在靠近最佳點時依舊會產生 Z 字形的收斂過程。針對此一情況，兩位學者亦提出其解決方法：即利用 Rosen[12]的梯度投影法。下列是 Luh 和 Tsaih[10] 演算法的求解步驟：

步驟一：設 $\boldsymbol{x^{(k)}}$ 表第次 $k$ 疊代所得之可行解區域 $\boldsymbol{F}$ 一邊界點。

步驟二：同(3)與(4)式，建構 $\Omega_1^{(k)}$、$\Omega_2^{(k)}$ 及 $M^{(k)}$ 。

步驟三：定義：

$$h^{(k)} = \frac{\Sigma_{i\in\Omega_1^{(k)}}A_{i,:}^T + \Sigma_{j\in\Omega_2^{(k)}}e_j}{\left\|\Sigma_{i\in\Omega_1^{(k)}}A_{i,:}^T + \Sigma_{j\in\Omega_2^{(k)}}e_j\right\|} \tag{8}$$

步驟四：若 $h^{(k)} = c$ ，則 $x^{(k)}$ 爲最佳點，故停止執行程式。

步驟五：定義：

$$g^{(k)} = \frac{h^{(k)} - c}{\left\|h^{(k)} - c\right\|} \tag{9}$$

步驟六：若 $g^{(k)}$ 爲可行方向，則令搜尋方向 $d = g^{(k)}$ ，移動步伐 $\eta$ ，令

$$\eta = \min\left\{\frac{-x_j^{(k)}}{d_j}, d_j, < 0, 1 \le j \le \mathrm{n}, j \notin \Omega_2^{(k)};\ \frac{b_i - A_{i,:}x^{(k)}}{A_{i,:}d}, A_{i,:}d < 0, 1 \le i \le \mathrm{m}, i \notin \Omega_1^{(k)}\right\}$$

求得 $x^{(k+1)}$ ，跳至步驟八。

步驟七：若 $g^{(k)}$ 不可行，則令搜尋方向 $d = M^{(k)}g^{(k)}$ ，移動步伐

$$\eta = \min\left\{ \frac{-x_j^{(k)}}{d_j}, d_j, < 0, 1 \le j \le \mathrm{n}, j \notin \Omega_2^{(k)}; \right.$$
$$\left. \frac{b_i - A_{i,:}x^{(k)}}{A_{i,:}d}, A_{i,:}d < 0, 1 \le i \le \mathrm{m}, i \notin \Omega_1^{(k)} \right\}$$

求得 $x^{(k+1)}$ 。

步驟八：若移動步伐 $\eta^{(k)} > \varepsilon$ 則令 $k = k+1$ 回步驟二；否則，執行步驟九。

步驟九：令 $k = k+1$，建構 $\Omega_1{}^{(k)}$ 及 $\Omega_2{}^{(k)}$ 。

步驟十：建構 $M^{(k)}$，令搜尋方向 $d = -M^{(k)}c$，移動步伐，

$$\eta = \min\left\{ \frac{-x_j^{(k)}}{d_j}, d_j, < 0, 1 \le j \le \mathrm{n}, j \notin \Omega_2^{(k)}; \right.$$
$$\left. \frac{b_i - A_{i,:}x^{(k)}}{A_{i,:}d}, A_{i,:}d < 0, 1 \le i \le \mathrm{m}, i \notin \Omega_1^{(k)} \right\}$$

求得 $x^{(k+1)}$ 。

步驟十一：令 $k = k+1$，建構 $\Omega_1{}^{(k)}$ 及 $\Omega_2{}^{(k)}$，若 $\left|\Omega_1{}^{(k)}\right| + \left|\Omega_2{}^{(k)}\right| \ge \mathrm{n}$ 則表示已求得一基本可行解，改以單體法求解。否則回步驟十。

Luh 和 Tsaih[10]利用投影法的目的是爲獲得一基本可行解，再搭配單體法求解，以解決 Z 字形的收斂過程。兩位學者的研究發現，使用此種輔助單體法之演算法求解線性規劃問題，要比直接用單體法求解所需之疊代數，平均值約降低 40%，換句話說，有效搜尋法提供了單體法一接近最佳點之起始點；再加上獲得一基本可行解時，可檢查最多只有個疊代，因此，對於利用梯度投影法求解線性規劃問題而言，有效搜尋法亦提供該演算法接近最佳解之起始點，這表示儘管遭遇最壞情況，有效搜尋法可藉由內點法快速收斂至最佳解的特性，使求解所需疊代數不會隨著起始點與最佳點的距離增加而增加。

## 2.4 仿射演算法與有效搜尋法之差異性

仿射演算法與有效搜尋法都是希望利用內點法快速收斂至最佳解的特性，且同樣會將**-*c*** 投影至可行解區域作爲搜尋方向，但仿射演算法每一疊代都必須透過線性空間轉換來達成，有效搜尋法則不需要，故兩者之間仍有很大的差異性。下列是兩演算法相異之處：

1. 仿射演算法標準式之限制式爲等式，有效搜尋法則爲不等式。
2. 仿射演算法必須先求得一可行解 $x^{(0)}$ 使得 $Ax^{(0)} = b$ 且 $x^{(0)} > 0$，有效搜尋法則可以任意一組數列作爲起始解。
3. 仿射演算法求解過程皆沿可行解區域內部移動求解，有效搜尋法則分爲三階段： 第一階段沿 $g^{(k)}$ 方向移動改進目標函數值，當移動步伐小於一預設值時，進入第二階段改以 $-M^{(k)}c$ 爲搜尋方向求解，順利求得一基本可行解後，再利用單體法求解。
4. 仿射演算法每次疊代所求得之新解爲可行解區域之一內點，有效搜尋法則爲一邊界點。
5. 仿射演算法的目的在於求得一靠近最佳點之內點，有效搜尋法則是在求得一基本可行解。
6. 仿射演算法所使用之投影矩陣 $\boldsymbol{P}$ 會隨著每一疊代 $D^{(k)}$ 而改變，有效搜尋法所建構之投影矩陣 $M^{(k)}$ 則隨著 $V^{(k)}$ 而改變。

# 3. 投影法的目的及可能遭遇的問題

在 Luh 和 Tsaih[10]文章中，其解決內點法 Z 字形收斂過程的方法，即沿 $-M^{(k)}c$ 方向改進，最後停在一角點上再利用單體法求得最佳解，本節要說明的是此方法得以成功地移動到角點的原因，以及可能遭遇的問題。

首先介紹利用梯度投影法或有效搜尋法求解線性規劃(2)式，給定起始點的方式：任意給定一組變數值 $\boldsymbol{x'}$ ，帶入所有限制式，若 $\boldsymbol{A}_{i,:}\boldsymbol{x'} \ne b_i$ ，則在該限制式減去剩餘變數或加上差額變數後，得一新起始解 $x^{(0)}$ 使得 $\left|\Omega_1{}^{(0)}\right| = \mathrm{m}$ ，其中 $x^{(0)} \in \Re^{\mathrm{n+m}}$ 。此起始點給定方式如同單體法一樣，目的在將限制式不等式部分化爲等式，並保證爲一邊界點。

**引理 2：** 當利用有效搜尋法求解(1)式，且進入以投影方向移動改進過程(步驟九到十一)，則存在 $k''$ ,使得 $rank\left(V^{(k'')}\right) = \mathrm{n}$ (意即矩陣 $V^{(k'')}$ 之秩爲 n)

證明：上述給定起始點的方法，說明演算法的第一步從邊界點開始出發，而由有效搜尋法步驟六或步驟七，得知每一疊代均停留在邊界點上；因此，令疊代 $k'$ 表示開始以投影方向移動改進，可推得 $\left|\Omega_1{}^{(k')}\right| + \left|\Omega_2{}^{(k')}\right| \ne 0$ 。此外 $\forall k \ge k'$ ，同步驟十令 $x^{(k+1)} = x^{(k)} - \eta M^{(k)}c$ ，預證

◎1. $\left|\Omega_1{}^{(k)}\right| + \left|\Omega_2{}^{(k)}\right| \le \left|\Omega_1{}^{(k+1)}\right| + \left|\Omega_2{}^{(k+1)}\right|$ 。

◎2. $rank\left(V^{(k)}\right) \le rank\left(V^{(k+1)}\right)$ 。

1. 由引理 1 得，$V^{(k)}x^{(k+1)} = V^{(k)}x^{(k)}$ 因此

$$\left|\Omega_1{}^{(k)}\right| + \left|\Omega_2{}^{(k)}\right| \le \left|\Omega_1{}^{(k+1)}\right| + \left|\Omega_2{}^{(k+1)}\right|$$

2. 根據假設(1)式有可行解且最佳解有界，及由上述結果和步驟十可推得 $\left|\Omega_1{}^{(k)}\right| + \left|\Omega_2{}^{(k)}\right| < \left|\Omega_1{}^{(k+1)}\right| + \left|\Omega_2{}^{(k+1)}\right|$ 因此，僅需證明 $rank\left(V^{(k)}\right) \neq rank\left(V^{(k+1)}\right)$。

令 $\tilde{s} = \left|\Omega_1{}^{(k+1)}\right| + \left|\Omega_2{}^{(k+1)}\right|$ ， $s = rank\left(V^{(k)}\right)$ ，$V_{\tilde{s},:}{}^{(k+1)}$ 表示 $V^{(k+1)}$ 之第 $\tilde{s}$ 列，$row\left(V^{(k)}\right)$ 表示 $V^{(k)}$ 之列所組成的集合，利用矛盾證法先假設 $rank\left(V^{(k)}\right) = rank\left(V^{(k+1)}\right)$ ，由 $\left|\Omega_1{}^{(k)}\right| + \left|\Omega_2{}^{(k)}\right| < \left|\Omega_1{}^{(k+1)}\right| + \left|\Omega_2{}^{(k+1)}\right|$ ，可推得存在 $\tilde{s} > s$ 使得 $V_{\tilde{s},:}{}^{(k+1)} \notin row\left(V^{(k)}\right)$，故 $V_{\tilde{s},:}{}^{(k)}x^{(k)} > b_{\tilde{s}}$，根據假設存在 $u \neq 0$ 使得 $V_{\tilde{s},:}{}^{(k)}x^{(k)} = u^T V^{(k)},\ u \in \Re^s$ ，因此

$$\begin{aligned}
&V_{\tilde{s},:}{}^{(k+1)}x^{(k+1)} \\
&= V_{\tilde{s},:}{}^{(k+1)}x^{(k)} - \eta V_{\tilde{s},:}{}^{(k+1)}M^{(k)}c \\
&= V_{\tilde{s},:}{}^{(k+1)}x^{(k)} - \eta u^T V^{(k)}M^{(k)}c \\
&= V_{\tilde{s},:}{}^{(k+1)}x^{(k)} - \eta u^T \left\{V^{(k)} - V^{(k)}\right\}c \\
&= V_{\tilde{s},:}{}^{(k+1)}x^{(k)} > b_{\tilde{s}}
\end{aligned}$$

故 $V_{\tilde{s},:}{}^{(k+1)} \notin row\left(V^{(k)}\right)$，矛盾；因此 $rank\left(V^{(k)}\right) \neq rank\left(V^{(k+1)}\right)$ 最後利用數學歸納法，由 $\left|\Omega_1^{(k')}\right| + \left|\Omega_2^{(k')}\right| \neq 0$ 及(b)，可證得存在 $k'', rank\left(V^{(k'')}\right)$。

舉例說明，對三維空間任一凸多面體，任意給定此多面體一邊界點，要如何移動到一未知的角點上？首先沿著該點所在平面任意方向移動，不難想像必定會經過此多面體的一個邊，此時若停留在此邊上，然後再沿邊上移動，就能找到此多面體的一個角點。此過程推廣至 n 維空間亦是如此，因投影矩陣 $M^{(k)}$ 就是要將 $-c$ 投影在 $V^{(k)}$ 的核空間上，沿投影後方向移動，而每次投影都是在增加 $V^{(k)}$ 線性獨立列的個數。最後當 $V^{(k)}$ 線性獨立列的個數等於 n 時，即表示 $x^{(k)}$ 為可行解區域上的一個角點。因為在可行解的 n 維空間中，任意 n 個線性獨立的方程式有唯一解。

無疑的，$-M^{(k)}c$ 是一可行的邊界方向，不僅如此，$-M^{(k)}c$ 亦可改進目標函數值，這可由一簡單式子觀察出：

$$\begin{aligned}
c^T\left(-M^{(k)}c\right) &= c^T\left[-\left(M^{(k)}\right)^2 c\right] \\
&= -c^T\left(M^{(k)T}\,M^{(k)}c\right) \\
&= -\left(M^{(k)}c\right)^T\left(M^{(k)}c\right) \\
&= -\left\|M^{(k)}c\right\|^2 \le 0
\end{aligned}$$

儘管 $-M^{(k)}c$ 是一可行的改進方向，但由於 $-M^{(k)}c$ 為邊界上的方向，故有兩種情況必須注意：

1. $\left[V^{(k)}\left(V^{(k)}\right)^T\right]$ 不一定有反矩陣。
2. 不保證 $-M^{(k)}c = 0$ 時即求得最佳解。

第一種情況發生的原因與 $V^{(k)}$ 的建構方式有關。雖說 $A$ 為滿秩，但所建構之 $V^{(k)}$ 卻不一定為滿秩，此即 $\left[V^{(k)}\left(V^{(k)}\right)^T\right]$ 反矩陣不存在的充要條件。而此一情況之發生恰好說明原始線性規劃(1)式為退化解問題。有關退化解的原始定義請參閱附錄一。

**引理 3**：若存在 $k$ 使得 $V^{(k)}$ 不為滿秩，則原始線性規劃(1)式為退化解問題。

證明：假設疊代第 $k$ 次時，即發生在角點上。因為 $V^{(k)}$ 不滿秩，由(3)得此投影後之角點所對應之 $\Omega_1^{(k)}$ 及 $\Omega_2^{(k)}$ 有著下列不等式關係：

$$\left|\Omega_1^{(k)}\right| + \left|\Omega_2^{(k)}\right| > \mathrm{n}$$

因 $\Omega_1^{(k)} \le \mathrm{m}$ ，因此得

$$\left|\Omega_2^{(k)}\right| > \mathrm{n} - \left|\Omega_1^{(k)}\right| \Rightarrow \left|\Omega_2^{(k)}\right| > \mathrm{n} - \mathrm{m}$$

這表示此角點變數為零的個數大於 n-m，此即所對應之基本可行解至少存有一變數為零，故此線性規劃為一退化解問題。

Rosen[12]的梯度投影法為保證求得最佳解，必須假設對任意疊代 $k$，$V^{(k)}$ 皆為滿秩。時至今日，仍無學者提出放寬此一假設的方法。由引理 3 可知，若原始線性規劃(1)式為非退化解問題，則所有的 $k$ 皆使得 $V^{(k)}$ 為滿秩。 因此，我們想藉由一數據實驗，來觀察利用梯度投影法求解線性規劃問題時， $V^{(k)}$ 發生不為滿秩情況的機率。首先從網站(/www.netlib.org/lp/data/)下載六個線性規劃問題(afiro、adlittle、sc105、sc205、sc50a、sc50b)。因

爲基底的不同選擇會影響基底變數以及求解時退化解的產生，所以分別對這六個題目，給定 1000 個不同之起始點，再利用梯度投影法求解，觀察在尚未求得最佳解前，$V^{(k)}$存有線性相依列情況的次數。結果如表 1：

表 1. $V^{(k)}$存有線性相依列情況的次數

| 問題名稱 | m | n | 產生退化解次數 |
|---|---|---|---|
| adlittle | 56 | 97 | 1000 |
| afiro | 27 | 32 | 1000 |
| Sc105 | 104 | 103 | 911 |
| Sc205 | 204 | 203 | 1000 |
| Sc50a | 49 | 48 | 764 |
| Sc50b | 48 | 48 | 390 |

表 1 告訴我們在求解特定六個問題時，$V^{(k)}$存有線性相依列情況的機率高達 84.42%，因此作者認爲，梯度投影法必須放寬對任意疊代 $k$ 而言，$V^{(k)}$皆爲滿秩的假設，才具有實用價值。

第二種情況則是因$V^{(k)}$的核空間在 n 維空間中並非一超平面(Hyperplane)，意指可經由適當地平移而成爲 $n$ 維向量空間中 n-1 維之子空間，使得與其正交的方向有無窮多個，下面借由引理 4 說明此一情況發生的原因。

**引理 4**：令 $N\left(V^{(k)}\right)$表 $V^{(k)}$ 的核空間，$\Psi \equiv \{x:\ x^T y = 0, \forall y \in N\left(V^{(k)}\right), x \in \Re^n\}$，且 $k, \dim(\Psi)$ 分別表示向量空間 $N\left(V^k\right), \Psi$ 的維度。若 $\dim\left(N\left(V^{(k)}\right)\right) < \mathrm{n}-1$，則 $\dim(\Psi) > 1$。

證明：由 $\dim\left(\Re^n\right) = \mathrm{n}$

且 $\forall x \in \Psi, y \in N\left(V^{(k)}\right), x^T y = 0$

$\Rightarrow \dim\left(N\left(V^{(k)}\right)\right) + \dim(\Psi) = \dim\left(\Re^n\right) = \mathrm{n}$

因 $\dim\left(N\left(V^{(k)}\right)\right) < \mathrm{n}-1$

$\Rightarrow \dim(\Psi) > 1$

由引理 4 得知，在 n 維空間中，當$V^{(k)}$核空間的維度小於 n-1 時，則$V^{(k)}$核空間的正交補空間的維度大於 1。換句話說，當$V^{(k)}$核空間非一超平面時，則與其核空間正交的方向有無窮多個。故將***-c*** 投影在$V^{(k)}$之核空間上的方法，無法使得演算法順利進行，若選擇$V^{(k)}$的其中一列建構投影矩陣作投影，雖可避免此一問題的發生，但要選擇哪一平面投影，仍是令人頭疼的問題，除此之外，投影後的方向還必須是有效搜尋方向。更糟的是，選擇$V^{(k)}$其中一列建構投影矩陣作投影，即破壞了演算法欲找到角點並藉此解決 Z 字型收斂過程的目的。下一節將針對投影矩陣 $M^{(k)}$ 做進一步的探討，並提供解決上述兩個問題的計算方法。

## 4. 投影矩陣的數值計算方法

令 $\left|\Omega_1^{(k)}\right| + \left|\Omega_2^{(k)}\right| = \tilde{s}^{(k)}$，而其中線性獨立列的個數爲 $\boldsymbol{s^{(k)}}$。由

$$-M^{(k)}c = -\left\{I - \left[V^{(k)}\left(V^{(k)}\right)^T\right]^{-1} V^{(k)}\right\}c$$

$$= -c + \left(V^{(k)}\right)^T\left[V^{(k)}\left(V^{(k)}\right)^T\right]^{-1} V^{(k)}c$$

不難觀察其中計算量最大的是在求$\left[V^{(k)}\left(V^{(k)}\right)^T\right]^{-1}$，本文考慮將$V^{(k)}$作 QR 分解，令 $V^{(k)} = L^{(k)}Q^{(k)}$，其中 $L^{(k)} \in \Re^{\mathrm{m}\times\mathrm{n}}$，($l_{ij}$ 表矩陣 $L^{(k)}$ 第 $i$ 列第 $j$ 行之元素，根據 QR 分解的性質，$l_{ij} = 0, \forall j > i$)，$Q^{(k)} \in \Re^{\mathrm{n}\times\mathrm{n}}$，爲一正交矩陣，得

$$\left[V^{(k)}\left(V^{(k)}\right)^T\right]^{-1} = \left[L^{(k)}Q^{(k)}\left(L^{(k)}Q^{(k)}\right)^T\right]^{-1}$$

$$= \left[L^{(k)}Q^{(k)}\left(Q^{(k)}\right)^T\left(L^{(k)}\right)^T\right]^{-1}$$

$$= \left[L^{(k)}\left(L^{(k)}\right)^T\right]^{-1}$$

$$= \left[\left(L^{(k)}\right)^T\right]^{+}\left(L^{(k)}\right)^{+}$$

其中 $\left(L^{(k)}\right)^{+} \equiv \left[\left(\bar{L}^{(k)}\right)^{-1}, O\right]$，$\left[\left(L^{(k)}\right)^T\right]^{+} \equiv \left[\left(\bar{L}^{(k)}\right)^{-1}, O\right]^T$ $\bar{L}^{(k)} \equiv L^{(k)}$ 的首 $s^{(k)}$ 行，$O$ 爲一 $s^{(k)} \times \left(\mathrm{n} - s^{(k)}\right)$ 之零矩陣，此時無須求解 $\left(L^{(k)}\right)^{+}$ 或 $\left[\left(L^{(k)}\right)^T\right]^{+}$，而是利用 $L^{(k)}$ 下三角非零元素的特性，以後向前(Backward)和前向後(Forward)的運算法以遞迴關係式逐一帶入求解得之。做法見 5.3 節。

當 $V^{(k)}$ 非滿秩時可利用 QR 分解求 $\left[V^{(k)}\left(V^{(k)}\right)^T\right]^{-1}$ 的原因是：當$V^{(k)}$在執行 QR 分解時，若有線性相依列，則該列在轉換過程中，由 $L^{(k)}$ 下三角非零元素的性質，可立即被判斷出，此時僅需逐一將相依列淘汰。只要最後保留一組線性獨立列，仍不失其一般性。

**定理 1**： 對任一 $k$ 而言，若 $V^{(k)}$ 存有線性相依列，令 $V_i^{(k)}$ 、$V_j^{(k)}$ 分別表示 $V^{(k)}$ 相異之兩組線性獨立列所構成之矩陣，

$$M_i^{(k)} = I - \left(V^{(k)}\right)^T \left[ V^{(k)} \left(V^{(k)}\right)^T \right]^{-1} V_i^{(k)}$$

$$M_j^{(k)} = I - \left(V^{(k)}\right)^T \left[ V^{(k)} \left(V^{(k)}\right)^T \right]^{-1} V_j^{(k)}$$

則 $-M_i^{(k)} c = -M_j^{(k)} c$ 。

證明：因 $V^{(k)}$ 之線性相依列皆可表示成 $V_i^{(k)}$ 或 $V_j^{(k)}$ 所有列之線性組合，所以 $V^{(k)}$ 之線性相依列皆屬於 $V_i^{(k)}$ 或 $V_j^{(k)}$ 的核空間，且 $V^{(k)}$ 與 $V_i^{(k)}$ 或 $V_j^{(k)}$ 之線性獨立列數相等，故 $V^{(k)}$ 之核空間與 $V_i^{(k)}$ 或 $V_j^{(k)}$ 之核空間相同，因此， $V_i^{(k)}$ 與 $V_j^{(k)}$ 之核空間亦相同。由此得知， $-M_i^{(k)} c = -M_j^{(k)} c$ 。

由定理 1 得知，$V^{(k)}$ 無論保留哪一組線性獨立列，都會求出相同之 $-M^{(k)} c$ ，舉例說明：

例題一：

$$\begin{array}{llllllll} \min & x_1 & +\frac{1}{4}x_2 & +x_3 & +\frac{1}{4}x_4 & +x_5 & & \\ s.t. & x_1 & & x_3 & & & \geq & 1 \\ & & x_2 & & +x_4 & & \geq & 1 \\ & x_1 & & & & +x_5 & \geq & 1 \\ & x_1 & ,x_2 & ,x_3 & ,x_4 & ,x_5 & \geq & 0 \end{array}$$

疊代一：

給定起始解 $x^{(0)} = (1,1,1,1,0)^T, \Omega_1^{(0)} = \{3\}$ 且 $\Omega_2^{(0)} = \{5\}$ ， $V^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

$-M^{(0)} c = (0, \frac{-1}{4}, -1, \frac{-1}{4}, 0)^T$ ，得 $x^{(1)} = \left(1, \frac{3}{4}, 0, \frac{3}{4}, 0\right)^T$ 。

疊代二：

$\Omega_1^{(1)} = \{1,3\}$ 且 $\Omega_2^{(1)} = \{3,5\}$ ， $V^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}$

此時 $V_1^{(1)}$ 存有線性相依列，因此令

$$V_1^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$V_2^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

得

$$\left(V_1^{(1)}\right)^T \left[ V_1^{(1)} \left(V_1^{(1)}\right)^T \right]^{-1} V_1^{(1)}$$

$$= \left(V_2^{(1)}\right)^T \left[ V_2^{(1)} \left(V_2^{(1)}\right)^T \right]^{-1} V_2^{(1)}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

故令 $V^{(1)} = V_1^{(1)}$ 得 $-M^{(1)} c = \left(0, \frac{-1}{4}, 0, \frac{-1}{4}, 0\right)^T$ ，以及 $x^{(2)} = \left(1, \frac{1}{2}, 0, \frac{1}{2}, 0\right)$ 。

正因爲如此，本文並沒有嚴格區分當 $\left[ V^{(k)} \left(V^{(k)}\right)^T \right]$ 反矩陣存在時才使用 $\left[ V^{(k)} \left(V^{(k)}\right)^T \right]^{-1}$ 這個符號，且爲便於討論，在此令 $V^{(k)}$ 爲 $\left\{ A_{i,:}, e_j^T : \forall i \in \Omega_1, \forall j \in \Omega_2 \right\}$ 其中一組線性獨立列向量所構成之矩陣。

當 $A$ 爲滿秩時，亦不能保證 $V^{(k)}$ 爲滿秩，但由定理一，無論 $V^{(k)}$ 是否爲滿秩，都可以 $-M^{(k)} c$ 爲搜尋方向，然後移動至可行解區域一角點上，並得以保留 $V^{(k)}$ 之一組線性獨立列，再改以單體法求解；故對於有效搜尋法而言，同樣可藉由定理 1，放寬限制式之係數矩陣 $A$ 必須爲滿秩的假設。

其次，探討當發生 $-M^{(k)} c = 0$ 情況時，如何利用 QR 分解來求得一向量 $d$ 使得 $-M^{(k)} d \neq 0$ ，且保證 $c^T d \leq 0$ 。由完全 QR 分解(Full QR factorization)及簡化 QR 分解(Reduced QR factorization) [14]，得知 $V^{(k)} = L^{(k)} Q^{(k)} = \overline{L}^{(k)} \hat{Q}^{(k)}$ ，其中，存在 $s^{(k)} > 0$ 使得 $\overline{L}^{(k)} \equiv L^{(k)}$ 的首 $s^{(k)}$ 行，$\hat{Q}^{(k)}$ 表矩陣 $Q^{(k)}$ 之首 $s^{(k)}$ 列所成之矩陣。由於 $Q_{s^{(k)}+1,:}^{(k)}, Q_{s^{(k)}+2,:}^{(k)}, \cdots, \cdots Q_{\mathrm{n}+1,:}^{(k)}$ 與 $Q_{1,:}^{(k)}, Q_{2,:}^{(k)}, \cdots, Q_{s^{(k)},:}^{(k)}$ 相互正交，得 $Q_{s^{(k)}+1,:}^{(k)}, Q_{s^{(k)}+2,}$ $, \cdots, Q_{\mathrm{n},:}^{(k)}$ 與 $V^{(k)}$ 所有列皆線性獨立，故當 $-M^{(k)} c = 0$ 時， $M^{(k)} (Q_{i,:}^{(k)})^T \neq 0, s^{(k)} + 1 \leq i \leq \mathrm{n}$ 但 $c^T M^{(k)} \left(Q_{i,:}^{(k)}\right)^T = \left(M^{(k)} c\right)^T \left(Q_{i,:}^{(k)}\right)^T = 0^T \left(Q_{i,:}^{(k)}\right)^T = 0$ 這

解決了當 $-M^{(k)}c=0$ 但未到達最佳點情況發生時，可分別利用 $\left[Q_{s^{(k)}+1,:}^{(k)}\right]^T,\left[Q_{s^{(k)}+2,:}^{(k)}\right]^T,\cdots,\left[Q_{\mathrm{n},:}^{(k)}\right]^T$ 等 $\mathrm{n}-s^{(k)}$ 個向量於第 $k$ 至 n-1 個疊代中投影，使能順利移動到角點，然後改以單體法求解，解決內點法 Z 字型收斂過程。舉例說明：

例題二：

$$\begin{aligned}\min \quad & x_2-x_3\\ s.t. \quad & -2x_1-2x_2-x_3\ge -4\\ & x_1,\ x_2,\ x_3\ge 0\end{aligned}$$

疊代一：

給定起始解 $x^{(0)}=(1,0,0)^T$，$\Omega_1^{(0)}$ 爲空集合而 $\Omega_2^{(0)}=\{2,3\}$，$V^{(0)}=\begin{bmatrix}0&1&0\\0&0&1\end{bmatrix}$ $-M^{(0)}c=0^T$，但 $\boldsymbol{x}^{(0)}$ 並非最佳解，因爲最佳解爲 $(0,0,4)^T$。此時將 $\boldsymbol{V}^{(0)}$ 作 QR 分解，得 $\boldsymbol{Q}_{3,:}^{(0)}=(1,0,0)$，$-M^{(0)}\left(Q_{3,:}^{(0)}\right)^T=(-1,0,0)^T$，以及 $\boldsymbol{x}^{(1)}=(0,0,0)^T$ 爲可行解區域內一角點。

## 5. 投影過程的計算量

本節將探討如何利用 Householder 正交化[14]將 $\boldsymbol{V}^{(k)}$ 做 QR 分解後求 $-\boldsymbol{M}^{(k)}\boldsymbol{c}$。同樣令 $\boldsymbol{V}^{(k)}=\boldsymbol{L}^{(k)}\boldsymbol{Q}^{(k)}$，其中 $\boldsymbol{L}^{(k)}\in\Re^{\mathrm{m}\times\mathrm{n}},l_{ij}=0,\forall j>i,$ $\boldsymbol{Q}^{(k)}\in\Re^{\mathrm{m}\times\mathrm{n}}$，爲一正交矩陣，得

$$\begin{aligned}&\left(V^{(k)}\right)^T\left[V^{(k)}\left(V^{(k)}\right)^T\right]^{-1}V^{(k)}\\ &=\left(V^{(k)}\right)^T\left[L^{(k)}Q^{(k)}\left(L^{(k)}Q^{(k)}\right)^T\right]^{-1}L^{(k)}Q^{(k)}\\ &=\left(V^{(k)}\right)^T\left[L^{(k)}Q^{(k)}\left(Q^{(k)}\right)^T\left(L^{(k)}\right)^T\right]^{-1}L^{(k)}Q^{(k)}\\ &=\left(V^{(k)}\right)^T\left[L^{(k)}\left(L^{(k)}\right)^T\right]^{-1}L^{(k)}Q^{(k)}\\ &=\left(V^{(k)}\right)^T\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}\left(\bar{L}^{(k)}\right)^{-1}L^{(k)}Q^{(k)}\\ &=\left(V^{(k)}\right)^T\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}\left[\bar{I},O\right]Q^{(k)}\\ &=\left(V^{(k)}\right)^T\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}\hat{Q}^{(k)}\end{aligned}$$

其中 $\bar{I}$ 是一 $s^{(k)}\times s^{(k)}$ 之單位矩陣，$O$ 爲一 $s^{(k)}\times\left(\mathrm{n}-s^{(k)}\right)$ 之零矩陣，$\hat{Q}^{(k)}$ 表矩陣 $Q^{(k)}$ 之首 $s^{(k)}$ 列所成之矩陣。

利用 Householder 正交化將 $\boldsymbol{V}^{(k)}$ 做 QR 分解時，$Q^{(k)}$ 可寫成 $H_{s^{(k)}}H_{s^{(k)}-1}\cdots H_1$，其中 $H_i,i=1,2,\cdots,s^{(k)}$ 都是一 Householder 矩陣[14]，因此得

$$\begin{aligned}&-M^{(k)}c\\ &=-c+\left(V^{(k)}\right)^T\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}\hat{Q}^{(k)}c\\ &=-c+\left(V^{(k)}\right)^T\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}\hat{Q}^{(k)}E^{(k)}\hat{Q}^{(k)}c\\ &=-c+\left(V^{(k)}\right)^T\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}\left[E^{(k)}H_{s^{(k)}}\cdots H_1c\right]\\ &=-c+\left(V^{(k)}\right)^T\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}\left\{E^{(k)}\left[H_{s^{(k)}}\left[\cdots\left[H_1c\right]\cdots\right]\right]\right\}\end{aligned}$$

其中 $E^{(k)}$ 是一 n×n 之矩陣，除左上角爲一 $s^{(k)}\times s^{(k)}$ 單位矩陣外，其餘元素皆爲零。

不難觀察 $-\boldsymbol{M}^{(k)}\boldsymbol{c}$ 的計算工作主要可分成下列四個步驟：

1. 利用 Householder 正交化將 $\boldsymbol{V}^{(k)}$ 作 QR 分解
2. 計算 $u=\left\{E^{(k)}\left[H_{s^{(k)}}\left[\cdots\left[H_1c\right]\cdots\right]\right]\right\}$
3. 計算 $\hat{u}=\left[\left(\bar{L}^{(k)}\right)^T\right]^{-1}u$
4. 計算 $\left(V^{(k)}\right)^T\hat{u}$

透過這四個步驟求得每一疊代之 $-M^{(k)}c$，將使得在求得基本可行解前，整個投影過程所需計算量看似隨疊代 $k$ 增加而增加，但實際上第一、二和四步驟僅與變數個數有關，而第三個步驟無論投影過程疊代數爲多少，計算量總和之上界爲：$O\left(\mathrm{n}^3\right)$，下面分四小節探究其中的原因。

### 5.1 執行 QR 分解所需計算量

由引理 2 可推得存在 $k''$ 使得 $V^{(k'')}$ 線性獨立列的個數等於變數個數 n，因此就投影過程而言，需先透過 $V^{(k''-1)}$，由其所建構之投影矩陣 $M^{(k''-1)}$，求得搜尋方向 $d^{(k''-1)}$，然後移動至可行解區域一角點上。設 $V^{(k''-1)}$ 線性獨立列的個數爲 n-t，其中，t 是指從疊代 $k''-1$ 到疊代 $k''$ 所增加線性獨立列的個數，$1\le \mathrm{t}\le \mathrm{n}-1$，即 $V^{(k''-1)}\in\ \Re^{(\mathrm{n}-\mathrm{t})\times\mathrm{n}}$，故當 t=1 時是整個投影過程執行 QR 分解最大計算量的情況。下列將介紹 Householder 正交算子 $H_i$，並藉此說明 $V^{(k''-1)}$ 執行 QR 分解之步驟：

$$H_i = I - 2\frac{\tau_i^T \tau_i}{\tau_i \tau_i^T}$$

其中，$\tau_i = V_{i,i:n}^{(k''-1)} - \left\| V_{i,i:n}^{(k''-1)} \right\| \sigma_i$ 表示矩陣 $V^{(k''-1)}$ 第 $i$ 列中第 $i$ 至第 n 行元素，$V_{i,j}^{(k''-1)}$ 表示 $V^{(k''-1)}$ 第 $i$ 列第 $j$ 行元素，$V_{i,:}^{(k''-1)}$ 表示矩陣 $V^{(k''-1)}$ 的第 $i$ 列，$\sigma_i = (1,0,\cdots,0)^T$，$\sigma_i \in \Re^{(n-i+1)\times 1}$。計算

$$\tau_i \tau_i^T = 2\left\| V_{i,i:n}^{(k''-1)} \right\| \left( \left\| V_{i,i:n}^{(k''-1)} \right\| - V_{i,i}^{(k''-1)} \right)$$

$$V_{j,i:n}^{(k''-1)} H_i = V_{j,i:n}^{(k''-1)} + \frac{V_{j,i:n}^{(k''-1)} \tau_i^T}{\left\| V_{i,i:n}^{(k''-1)} \right\| \left( \left\| V_{i,i:n}^{(k''-1)} \right\| - V_{i,i}^{(k''-1)} \right)} \tau_i$$

其中，$i \le j \le n-1$，接著在執行分解步驟之遞迴式中，先令，$\omega_i = \left\| V_{i,i:n}^{(k''-1)} \right\|$ 再令 $V_{i,i}^{(k''-1)} = V_{i,i}^{(k''-1)} - \omega_i$，則可得

$$V_{j,i:n}^{(k''-1)} H_i = V_{j,i:n}^{(k''-1)} + \frac{V_{j,i:n}^{(k''-1)} \left( V_{i,i:n}^{(k''-1)} \right)^T}{\omega_i V_{i,i}^{(k''-1)}} V_{i,i:n}^{(k''-1)} \qquad (10)$$

上述說明了演算法各步驟式子的推導過程。其中，對任意疊代 $k$，$1 \le i \le s^{(k)}$，從(10) 式中可看出，除矩陣 $V^{(k''-1)}$ 外，僅需增加變數 $\omega_i$ 以記錄 $\left\| \boldsymbol{V}_{\boldsymbol{i,i:n}}^{(\boldsymbol{k''-1})} \right\|$，再修改 $\boldsymbol{V}_{\boldsymbol{i,i}}^{(\boldsymbol{k''-1})}$，即能計算出 $V_{j,i:n}^{(k''-1)} H_i$。此即所謂「$yQ$ 乘積之隱藏式計算」(Implicit calculation of a product $yQ$ ) [14]。$V^{(k''-1)}$ 執行 QR 分解之演算法如下列所示：

```
for  i = 1  to  n − 1 {
   ω_i = ‖ V_{i,i:n}^{(k''-1)} ‖
   V_{i,i}^{(k''-1)} = V_{i,i}^{(k''-1)} − ω_i
   for  j = i + 1  to  n − 1 {
        v_j = ( V_{j,i:n}^{(k''-1)} )
        tmp = v_j ( V_{i,i:n}^{(k''-1)} )^T
        tmp = tmp / V_{i,i}^{(k''-1)}
        tmp = tmp / ω_i
        V_{j,i:n}^{(k''-1)} = V_{j,i:n}^{(k''-1)} + tmp × V_{i,i:n}^{(k''-1)}
   }
}
```

此演算法的計算量爲 $\frac{4}{3}n^3 - \frac{1}{2}n^2 - \frac{17}{6}n + 2$，其證明請參閱附錄二。

投影改進過程當中，若考慮每一疊代 $k$ 新增加之 $A_{i,:}$ 或 $e_j^T$ 置於 $V^{(k-1)}$ 之下，而構成之 $V^{(k)}$。此時，對於每一疊代 $k$ 在執行 QR 分解時，可藉由(10) 式寫成下列演算法：

```
for  i = 1  to  s^(k−1)
   for  j = s^(k−1) + 1  to  s^(k) {
        v_j = V_{j,i:n}^(k)
        tmp = v_j ( V_{i,i:n}^(k) )^T
        tmp = tmp / V_{i,i}^(k)
        tmp = tmp / ω_i
        V_{j,i:n}^(k) = V_{j,i:n}^(k) + tmp × V_{i,i:n}^(k)
   }

for  i = s^(k−1) + 1  to  s^(k) {
   ω_i = ‖ V_{i,i:n}^(k) ‖
   V_{i,i}^(k) = V_{i,i}^(k) − ω_i
   for  j = i + 1  to  n − 1 {
        v_j = V_{j,i:n}^(k)
        tmp = v_j ( V_{i,i:n}^(k) )^T
        tmp = tmp / V_{i,i}^(k)
        tmp = tmp / ω_i
        V_{j,i:n}^(k) = V_{j,i:n}^(k) + tmp × V_{i,i:n}^(k)
   }
}
```

最後將整個投影過程所執行之 QR 分解的全部計算量加起來，我們發現當利用上述演算法時，計算量之和最多不會超過 $\frac{4}{3}n^3 - \frac{1}{2}n^2 - \frac{17}{6}n + 2$，這顯示整個投影過程所需執行 QR 分解的計算量不隨著疊代數改變而增加，僅僅與變數個數有關。

## 5.2 $E^{(k)}Q^{(k)}c$ 的計算量

由 $E^{(k)}Q^{(k)}c = E^{(k)}\left[ H_{s^{(k)}} \left[ H_{s^{(k)}-1} [\cdots H_1 c \cdots] \right] \right]$ 得此計算過程相當於接連重複 $H_i$ 乘上一行向量，$i = 1,2,\cdots,s^{(k)}$。此時若利用 $Qx$ 乘積之隱藏式計算，技巧同(10) 式，做局部修正便可得到每一階段所需之矩陣或向量，茲將演算法整理如下：

```
u = c
for  i = 1  to  n − 1 {
```

$tmp = V_{i,i:n}^{(k''-1)} u_i$

$tmp = tmp / V_{i,i:n}^{(k''-1)}$

$tmp = tmp / w_i$

$u_i = u_i + tmp \times V_{i,i:n}^{(k''-1)}$

}

其中，$u = (u_1, u_2, \cdots, u_n)^T$，$u_i = (u_i, u_{i+1}, \cdots, u_n)^T$。在整個投影過程中，對於求所有 $k$, $E^{(k)}Q^{(k)}c$ 計算量之總和，等於求 $E^{(k''-1)}Q^{(k''-1)}c$ 的計算量，也就是 $2n^2 + 3n - 5$，其證明請參閱附錄三。故其計算量僅僅與變數個數有關。

## 5.3 $\left[\left(\overline{L}^{(k)}\right)^T\right]^{-1} u$ 的計算量

此時仍無須求解 $\left[\left(\overline{L}^{(k)}\right)^T\right]^{-1}$，而是利用

$$y = \left[\left(\overline{L}^{(k)}\right)^T\right]^{-1} u \text{ 即 } \left(\overline{L}^{(k)}\right)^T y = u \ ,$$

以遞迴關係式逐一帶入求解得 $y$。

不同於前述演算法，此演算法每一疊代 $k$ 求解過程皆相互獨立，亦即不僅與變數個數有關，還會隨著疊代數增加而增加，但無論投影過程疊代數有多少，都小於變數個數，而且計算量總和仍不超過 $\frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n$，其證明請參閱附錄四。

## 5.4 $\left(V^{(k)}\right)^T \hat{u}$ 的計算量

令第 $k$ 次疊代的 $\hat{u}$ 爲 $\hat{u}^{(k)}$，其中的第 $s^{(k)}$ 個元素爲 $\hat{u}_{s^{(k)}}^{(k)}$。觀察 $\left(V^{(k+1)}\right)^T$ 及，$\hat{u}^{(k+1)}$ 與 $\left(V^{(k)}\right)^T$ 及 $\hat{u}^{(k)}$，比較後不難發現分別多了 $\left(V_{s^{(k+1)},:}^{(k+1)}\right)^T$ 及 $\hat{u}_{s^{(k+1)}}^{(k+1)}$。令 $\hat{y}^{(k)} = \left(V^{(k)}\right)^T \hat{u}^{(k)}$，則

$$\hat{y}^{(k+1)} = \hat{y}^{(k)} + \hat{u}_{s^{(k+1)}}^{(k+1)} \left(V_{s^{(k+1)},:}^{(k+1)}\right)^T \qquad (11)$$

因此就整個投影過程而言，$1 < s^{(k)} \le n$，利用(11)式，其所需計算量爲 $(n-1)(2n-1) = 2n^2 - 3n + 1$。

# 6. 結論

本文於第 3 節說明 Luh 和 Tsaih[10]演算法中其投影法可能遭遇到 $V^{(k)}$ 存有線性相依列及不保證當 $-M^{(k)}c = 0$ 時即求得最佳解等兩個問題，而於第 4 節提出利用 QR 分解的正交性解決這兩個問題。值得一提的是本文雖於(1)式中同 Luh 和 Tsaih[10]演算法假設矩陣 $A$ 爲滿秩，但此假設仍無法確保對任一 $k$ 而言，$V^{(k)}$ 爲滿秩。本文因此提出解決方法，此時逐一將相依列淘汰後，最後無論保留哪一組線性獨立列，都會求出相同之 $-M^{(k)}c$，也就是說，Luh 和 Tsaih [10]演算法可以放寬矩陣 $A$ 爲滿秩的假設。

此外，利用第 5 節計算方法求 $-M^{(k)}c$ 最大的優點是，在移動到角點這段過程當中，疊代 $k$ 所求得之 Householder 矩陣仍可在疊代 $k$+1 繼續延用，不需重新計算，但需修正疊代 $k$+1 之新加入的向量 $V_{s^{(k+1)},:}^{(k+1)}$，並且求得新 Householder 矩陣 $H_{s^{(k+1)}}$ 即可。直到求得第 n 個 Householder 矩陣後，便可順利的移動至可行解區域內一角點上，並使得整個投影過程所需計算量不超過 $\frac{5}{3}n^3 + 3n^2 - \frac{8}{3}n - 2$。

若利用 Cholesky 分解法計算 $-M^{(k)}c$ [7]，雖說 Cholesky 分解計算量較 QR 分解約少 $n^3$，但因必須完整求得 $V^{(k'')}(V^{(k'')})^T$，而使得就整個投影過程而言，儘管利用每一疊代 $k$ 之矩陣 $V^{(k)}$ 的特性，其計算量上限爲 $\frac{5}{3}n^3 + \frac{5}{2}n^2 + \frac{43}{6}n + 3$，僅比 QR 分解少 $\frac{1}{2}n^2 + \frac{9}{2}n - 5$，但當遭遇 $-M^{(k)}c = 0$ 且未到達最佳點情況發生時，仍須利用 QR 分解使其能順利移動到角點，此時計算量上限增加成爲 $3n^3 + 2n^2 - 11n + 6$，因此作者認爲在 Luh 和 Tsaih[10]演算法中，應利用 QR 分解計算 $-M^{(k)}c$，此與傳統內點法利用 Cholesky 分解法有所不同。

# 附錄一

在定義退化解前必須先定義基本可行解：一線性系統 $A'x = b$ 的基底解(Basic solution)可由令 $n' - m$ 個變數爲 0 且僅求解剩餘 m 個變數值當中獲得。其中，$A'$ 爲一 $m \times n'$ 之矩陣，$n' = n + t$，$1 \le t \le m$，這假設是讓 $n' - m$ 個變數爲 0 而使得剩餘 m 個變數解唯一，這相當於說剩餘 m 個變數所對應矩陣 $A'$ 的行爲線性獨立。任一基底解當所有變數皆大於等於 0 時，則稱此基底解爲基本可行解。

退化解之定義：在線性規劃問題的一基本可行解中，若存在至少一個爲 0 的基底變數(Basic variable)，則稱此線性規劃爲退化。詳細內容請參

考文獻[18]。

## 附錄二

任意兩 n 維向量之內積共需 n 個乘法及 n-1 個加法，下標 $k$ 由 1 累加至 n-1 表開始投影時 $|\Omega_1|+|\Omega_2|=1$，且整個投影過程必須求得 n-1 個線性獨立向量，以獲得一基本可行解，故每一疊代 $k$ 執行 QR 分解所需計算量包含求長度爲 n-$k$+1 之歐氏模及一個減法，再加上修改剩餘 n-$k$-1 個長度爲 n-$k$+1 之向量。因此，其總和除 n-1 個開根號外，餘計算量如下所示：

$$\begin{aligned}
&\sum_{k=1}^{n-1}\Big\{(2n-2k+1)+1 \\
&\quad+\sum_{j=k}^{n-1}\big[(2n-2k+1)+2+2(n-k+1)\big]\Big\} \\
&=(n-1)(2n+2)+\sum_{k=1}^{n-1}\big[-2k+(n-k-1)(4n-4k+5)\big] \\
&=2n^2-2+\sum_{k=1}^{n-1}\Big[\big(4n^2+n-5\big)-(8n+3)k+4k^2\Big] \\
&=2n^2-2+(n-1)\big(4n^2+n-5\big) \\
&\qquad-(8n+3)\frac{n(n-1)}{2}+4\frac{n(n-1)(2n-1)}{6} \\
&=\frac{4}{3}n^3-\frac{1}{2}n^2-\frac{23}{6}n+3
\end{aligned}$$

## 附錄三

當 $V^{(k)}$ 列數爲 $n-1$ 時，沿 $-M^{(k)}c$ 方向即可順利移動至可行解區域內之角點上，且整個投影過程 $E^{(k)}Q^{(k)}c$ 所需計算量之總和等於求 $H_{n-1}\big[H_{n-2}[\cdots[H_1c]\cdots]\big]$，故每一疊代 $k$ 之計算皆包含長度爲 n-$k$+1 之向量內積，再加上 2 個除法及長度爲 n-$k$+1 之向量乘上一實數後加至另一向量，如下所示：

$$\begin{aligned}
&\sum_{k=1}^{n-1}\big[(2n-2k+1)+2+2(n-k+1)\big] \\
&=(n-1)(4n+5)-4\sum_{k=1}^{n-1}k \\
&=4n^2+n-5-2n(n-1) \\
&=2n^2+3n-5
\end{aligned}$$

## 附錄四

因 $(\overline{L}^{(k)})^T$ 爲一上三角矩陣，使得對每一疊代 $k$ 而言，都必須計算 $\sum_{j=1}^{k-1}j=(j+1)$ 個乘法與加法，以及 $k$-1 個減法和 $k$ 個除法，故所需之計算量可由下列表示：

$$\begin{aligned}
&\sum_{k=1}^{n-1}\left\{\left[\sum_{j=1}^{k-1}j+(j-1)\right]+(k-1)+k\right\} \\
&=\sum_{k=1}^{n-1}\left[\left(2\sum_{j=1}^{k-1}j\right)+k\right] \\
&=\sum_{k=1}^{n-1}\big[k(k-1)+k\big] \\
&=\sum_{k=1}^{n-1}k^2 \\
&=\frac{n(n-1)(2n-1)}{6} \\
&=\frac{1}{3}n^3-\frac{1}{2}n^2+\frac{1}{6}n
\end{aligned}$$

## 誌謝

本研究承蒙行政院國科會計畫(編號：NSC-88-2213-E-004-004) 補助研究經費，謹此致謝。作者十分感謝審查委員所提供之寶貴建議，使本文更趨完整。

## 參考文獻

1. Arbel, A., *Exploring Interior-Point Linear Programming*, The MIT Press, Cambridge, Massachusetts (1993).
2. Bazaraa, M. S. and C. M. Shetty, *Nonlinear Programming Theory and Algorithms*, John Wiley and Sons, New York (1979).
3. Bixby, R. E., J. W. Gregory, I. J. Lustig, R. E. Marsten and D. F. Shanno, "Very large-scale linear programming: a case study in combining interior point and simplex methods," *Operations Research*, **40**, 885-897 (1992).
4. Dikin, I. I., "Iterative solution of problems of linear and quadratic programming," *Soviet Mathematics Doklady*, **8**, 674-675 (1967).
5. Dikin, I. I., "O skhodimosti odnogo iteratsionnogo protsessa(in Russian)," *Upravlyaemye Sistemy*, 12, 54-60 (1974).
6. Fang, S. C. and S. Puthenpura, *Linear Optimization*

*and Extensions*, Prentice-Hall, Englewood Cliffs, New Jersey (1993).

7. Gondzio, J., "Implementing Cholesky factorization for interior point methods of linear programming," *Optimization*, 27, 121-140 (1993)
8. Huard, P. and A. Auslender, *Point-to-Set Maps and Mathematical Programming*, North-Holland Pub. Co., Amsterdam, New York (1979).
9. Karmarkar, N. K., "A new polynomial time algorithm for linear programming," *Combinatorica*, **4**, 373-395 (1984).
10. Luh, H. and R. Tsaih, "An efficient search direction for linear programming problems," *Computers and Operations Research*, **29**, 195-203 (2002).
11. Megiddo, N., "On finding primal and dual optimal bases," *ORSA J. on Comput.*, **3**, 63-65 (1991).
12. Rosen, J. B., "The gradient projection method for nonlinear programming. Part 1: linear constraints," *SIAM Jounal on Application Mathematics*, **9**, 514-553 (1960).
13. Terlaky, T., *Interior Point Methods of Mathematical Programming*, Kluwer Academic publishers, Dordrecht, Netherlands (1996).
14. Trefethen, L. N. and D. Bau, "Numerical linear algebra," *The Society for Industrial and Applied Mathematics*, **7**, 276-319(1997).
15. Vanderbei, R. J., M. S. Meketon and B. A. Freedman., "A modiffication of Karmarkar's linear programming algorithm," *Algorithmica*, **1**, 395-407 (1986).
16. Vanderbei, R. J., "Affine-scaling for linear programs with free variables," *Mathematical Programming*, **43**, 31-44 (1989).
17. Vavasis, S. A. and Y. Ye, "Identifying an optimal basis in linear programming," *Annals of Operations Research*, **62**, 565-572 (1996).
18. Winston, W. L., *Operations Research Applitions and Algorithms*, Duxbury Press, New York (1994).
19. Ye, Y., *Interior Point Algorithms Theory and Analysis*, John Wiley and Sons, New York (1997).

## 作者簡介

**莊文華** 政治大學應用數學系碩士。目前服務於中華民國空軍。研究興趣包括作業研究、數值計算。

**陸 行** 美國北卡州立大學作業研究博士。主要研究領域作業研究、等候理論。

(Received August 2001; revised November 2001; accepted February 2002)

# GPM ANALYSIS IN COMBINING INTERIOR POINTS AND THE SIMPLEX METHOD

Wen-Hua Chuang and Hsing Luh*
*Department of Mathematical Sciences, National Chengchi University, 64, Section 2,Chih-nan Road, Wenshan,Taipei, Taiwan, 116, R.O.C.*

## ABSTRACT

By geometric viewpoint in search directions of linear programming problems (LP), there are two major approaches: the simplex method and the interior-point algorithm originated from Karmarkar's approach. Many of their variants developed both in theory and applications are still in progress. Roughly speaking, the main difference among them is that the simplex method is devoted for each the exact optimal solution while Karmarkar-based method is computationally fast in approaching to the neighborhood of the optimal solution, but it becomes slow near the optimal point. By the separating hyperplane theorem, we know that the optimal solution of a linear programming problem would locate at the boundary point. However, Karmarkar's algorithm is claimed as an interior-point approach which takes a solution trajectory path through the interior of the feasible region. On the other hand, these algorithms will coincide the zig-zag situation following the trajectory path before attaining the optimal point meanwhile it cannot achieve the optimal point. Therefore, the purpose of this paper is to study a possible approach by combining the interior point and the simplex method. By means of QR factorization, we give a general analysis on Gradient Projection Method(GPM) in solving linear programming problems. Moreover, we prove that a general assumption-technique constraint matrix A of full rank can be relaxed by using our approach. Finally, we analyze the complexity of this approach to ensure that its upper bound is $O(n^3)$.

***Keywords***: linear programming, gradient projection method, QR factorization, the simplex method, interior-point algorithms.

(*Corresponding author: paul@math.nccu.edu.tw)