隨 機 規 劃 期 末 專 題 報 告

學生：馮成林
學號：311703003
系所：運輸與管理學系物流研究所
年級：一年級

目錄

圖目錄

表目錄

1 緒論

Mousavi 等人在 2022 年的文獻[1]當中，結合移動倉(mobile depots)以及群眾配送(crowd-shipping)這二種新型態的物流作業模式，提出一個兩階段的配送模式，而本次的期末專題是去實作這篇文獻當中的求解方法。以下先說明移動倉以及群眾配送的運作模式，而後介紹此篇文獻提出的兩階段配送模式。

1.1 移動倉(Mobile depots)

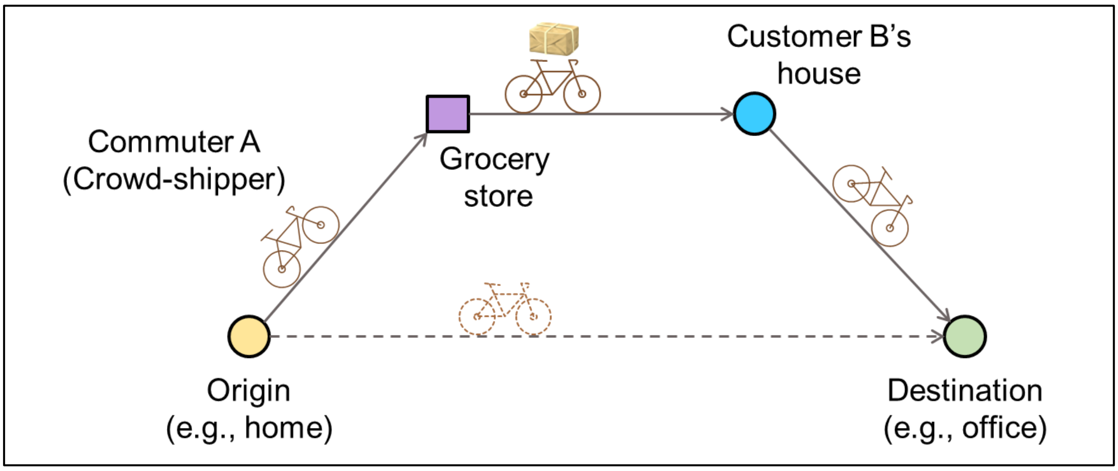
移動倉是具有移動功能的小型倉庫，實務上最常見的形式是拖車。移動倉閒置時，會停在郊區的空曠處，如物流中心的停車場或是郊區的大型停車場，而當有貨物要進行配送時，就會先將貨物存放在移動倉裡，並將移動倉牽引至市區的某處停靠(如賣場的停車場)，以作為臨時的市區倉儲。接著，移動倉的送貨員就會進行移動倉的貨物配送，待所有的貨物配送任務都結束後，移動倉會再回到郊區停靠，等待下次的任務。

移動倉具有市區倉儲的功能，有利於物流公司進行更快速的配送，以提供更優良的服務品質，而同時又能免除在市區建設倉儲的龐大成本。

1.2 群眾配送(Crowd-shipping)

群眾配送的基本概念是委託一般民眾進行貨物的配送，本篇文獻討論的群眾是平日工作的通勤者，舉例而言，通勤者 A 上班的路線如圖一中底下的虛線所示，然而在某天上班以前，通勤者 A 接受了超市的委託，在他上班的路上稍微繞路，先到超市取顧客 B 在線上訂購的商品，接著將商品送到顧客 B 的家以後，他才去到公司上班。完成這項任務以後，超市會給予通勤者 A 商品優惠，以此作為誘因吸引通勤者協助送貨。

對於物流公司來說，部分貨物若能委託群眾配送，公司將不用花費龐大的成本購置及養護貨車，亦能省去僱用大量司機的人力成本。



圖一 群眾配送例子

1.3 結合移動倉和群眾配送的兩階段配送方式

Mousavi 等人(2022)結合了移動倉以及群眾配送的概念，提出兩階段的配送模式，並將此配送模式的派送規劃問題稱為 stochastic mobile depots and crowd-shipping problem，簡稱為 SMDCP。在 SMDCP 的第一階段當中，公司需要決定移動倉停在市區的哪些地方，以及各個移動倉要存放哪些顧客的貨物；在第二階段當中，公司會依照第一階段移動倉的決策，尋找合適的通勤者協助送貨，並指派各個通勤者要配送的貨物。在配送的前一天，公司會先尋找願意協助配送的通勤者，然而，公司並不是跟這些通勤者簽訂契約，通勤者有隨時拒絕委託的權利，因此，於配送當天，各個通勤者有一定的機率會臨時拒絕委託，然而第一階段的決策在配送前一天就必須完成，當時仍無法確定各個通勤者最終是否會臨時拒絕，此即為 Mousavi 等人(2022)在研究當中考慮的不確定性。

此篇文獻提出了兩階段隨機整數規劃模型(two stage stochastic integer programming model)以進行 SMDCP 的研究，並使用 Sample Average Approximation (簡稱 SAA) 進行目標值估計，以及採用 Multicut L-shaped 的求解方法進行 SMDCP 的求解。本次期末專題實作這兩種方法，求解此篇文獻提出之兩階段隨機整數規劃模型。

以下的報告內容，於第 2 章先說明 Mousavi 等人(2022)研究當中的問題定義以及數學模型，接著在第 3 章介紹此篇文獻使用的求解方法，亦為本次期末專題實作的部分；並於第 4 章列出實作內容的相關細節；而第 5 章則呈現實作當中的數項實驗結果；最後第 6 章為結論。

2 問題定義及數學模型

2.1 問題定義

Mousavi 等人(2022)提出的兩階段模型當中，最主要有三個角色：移動倉、顧客、通勤者，分別以符號 i、j、k 表示，而集合 $I=\{1,\dots,i,\dots,|I|\}$, $J=\{1,\dots,j,\dots,|J|\}$, $K=\{1,\dots,k,\dots,|K|\}$則分別代表移動倉、顧客以及通勤者的集合。SMDCP 當中的各個移動倉都有相同的容量，以參數C表示；而每位通勤者 k 都有各自的通勤旅次起訖點，分別為o_k、d_k，且各個通勤者最多只能配送一個顧客的貨物。問題當中，各個移動倉 i 具有各自的服務時段，以參數$t(i)$表示，各個通勤者 k 也有可協助送貨的服務時段，以參數$t(k)$表示，兩者的服務時段皆分成上午及下午兩個時段，因此$t(i)$、$t(k)$皆為二元參數：0 代表服務時段為上午，1 則服務時段為下午。假設通勤者 k 的服務時段為早上，則該通勤者只能前往服務時段亦為早上的移動倉取得顧客的貨物；反之亦然。

為了處理通勤者出現與否的隨機性，問題中以二元隨機變數 ξ_k 代表通勤者 k 出現的情況，$\xi_k=1$ 表示通勤者 k 有出現，反之為 0；ξ_k 服從白努利分佈，$\xi_k=1$的機率為 p_k，$\xi_k=0$的機率則為 $1-p_k$。隨機向量 $\boldsymbol{\xi}=(\xi_1,\dots,\xi_{|K|})$ 則代

表各個通勤者出現情況所對應的情境(scenario)。

SMDCP 考慮了以下四項營運成本：

(i) 移動倉 i 的旅行成本 c_i^z：移動倉 i 從郊區的移動倉基地移動至市區內指定地點的旅行成本。

(ii) 群眾配送的補償成本：由於通勤者要完成配送任務，需要在通勤的路上繞道而行，因此會產生額外的旅行成本，公司必須給予相應的補償作為誘因，通勤者才會願意協助配送。通勤者 k 的補償成本，c_{ijk}^p，其計算方式是先將配送過程的總旅行距離，減去原先通勤者 k 從o_k到d_k的旅行距離後，得到通勤者相較於平常多行走的旅行距離 l_{ijk}，將此多出的旅行距離乘上單位距離的補償費 c_2，再加上基本的補償費用 c_1，即 $c_{ijk}^p = c_1 + c_2 l_{ijk}$；配送過程的總旅行距離則由三段距離所構成，首先通勤者 k 會從旅次起點 o_k 移動到移動倉 i，以取得顧客 j 的貨物；而後從移動倉移動到顧客的所在地；將貨物交給顧客後，最後從顧客的所在地移動到旅次終點 d_k。

(iii) 第一階段的延後配送成本：在第一階段的決策當中，公司考量到第二階段不容易找到合適的通勤者配送某一顧客 j 的貨物，因此可以選擇當天不進行顧客 j 的貨物配送，而移動倉不會將該顧客的貨物載往市區，但這樣的決定會產生第一階段的延後配送成本 c_j^y，此成本為顧客的補償成本。

(iv) 第二階段的延後配送成本：在第一階段當中，若公司決定要進行顧客 j 的配送，並且讓某一移動倉存放該顧客的貨物，但在第二階段當中，由於某些通勤者最終無法協助配送工作，因此配送當天找不到合適的通勤者來送顧客 j 的貨物，導致公司需延後顧客 j 的配送，而產生第二階段的延後配送成本 c_j^v。由於貨物的準備有相對應的成本，因此第二階段的延後配送成本不僅有顧客的補償成本，亦含有準備貨物的相關成本，故 c_j^v 大於 c_j^y。

在 SMDCP 中，第一階段的決策包括各個移動倉是否要出勤，以及出勤的移動倉要存放哪些顧客的貨物；第二階段則是在某一通勤者出現的情境底下，決定顧客貨物以及通勤者的指派。

2.2 數學模型

Mousavi 等人(2022)提出兩階段隨機整數規劃模型，以求解 SMDCP，表一整理模型當中使用到的符號。

以下為第一階段的整數規劃模型：

$$\min \sum_{i\in I} c_i^z z_i + \sum_{j\in J} c_j^y y_j + \mathbb{E}_\xi[Q(\mathbf{y},\mathbf{w},\boldsymbol{\xi})] \quad (1a)$$

$$\text{s.t. } w_{ij} \le z_i \qquad \forall i\in I, j\in J, \quad (1b)$$

$$\sum_{i\in I} w_{ij} + y_j = 1 \qquad \forall j\in J, \quad (1c)$$

$$\sum_{j\in J} w_{ij} \le C \qquad \forall i\in I, \quad (1d)$$

$$\mathbf{z}\in\{0,1\}^{|I|}, \mathbf{y}\in\{0,1\}^{|J|}, \mathbf{w}\in\{0,1\}^{|I||J|}. \quad (1e)$$

目標式(1a)是最小化總成本期望值，其中第一項為移動倉的總旅行成本，第二項則為第一階段的延後配送成本，而第三項為第二階段成本的期望值。限制式(1b)用來約束顧客貨物與移動倉的分配，即顧客貨物只能分配給有出勤的移動倉；限制式(1c)則是有關於第一階段延後配送的決定；限制式(1d)為移動倉的容量限制；限制式(1e)為整數限制。

第二階段的整數規劃模型如下：

$$Q(\mathbf{y},\mathbf{w},\boldsymbol{\xi}) = \min \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk}^p p_{ijk} + \sum_{j\in J} c_j^v v_j \quad (2a)$$

$$\text{s.t. } \sum_{i\in I}\sum_{j\in J} p_{ijk} \le \xi_k \qquad \forall k\in K, \quad (2b)$$

$$\sum_{i\in I}\sum_{k\in K} p_{ijk} + v_j = 1 - y_j \qquad \forall j\in J, \quad (2c)$$

$$\sum_{k\in K} p_{ijk} \le w_{ij} \qquad \forall i\in I, j\in J, \quad (2d)$$

$$p_{ijk} = 0 \qquad \forall j\in J, (i,k)\in I\times K \text{ s.t. } t(i)\ne t(k), \quad (2e)$$

$$\mathbf{p}\in\{0,1\}^{|I||J||K|}, \mathbf{v}\in\{0,1\}^{|J|}. \quad (2f)$$

目標式(2a)是最小化通勤者的補償成本以及第二階段延後配送成本的總和。限制式(2b)用來約束顧客貨物配送與通勤者的指派，即各顧客的貨物配送任務只能指派給配送當天有出現的通勤者；限制式(2c)則是有關於第二階段延後配送的決定；限制式(2d)確保通勤者前往正確的移動倉取得顧客的貨物；限制式(2e)確保通勤者與移動倉的服務時段有正確的配對；限制式(2f)為整數限制。

表一 隨機整數規劃模型使用之符號

Sets	
I	Set of mobile depot stopping locations (indexed by i)
J	Set of customers (indexed by j)
K	Set of crowd-shippers (indexed by k)
Parameters	
c_i^z	Cost of sending a mobile depot to location i
c_{ijk}^p	Cost of serving customer j with crowd-shipper k through location i
c_j^y	Cost of postponing customer j's delivery in the first stage
c_j^v	Cost of postponing customer j's delivery in the second stage
$t(i)$	0 if the operation window of location i is in the morning, 1 if it is in the evening
$t(k)$	0 if crowd-shipper k is available in the morning, 1 if available in the evening
C	Capacity of (homogeneous) mobile depots
Random variable	
ξ_k	1 if crowd-shipper k is available, 0 otherwise
Decision variables	
z_i	1 if a mobile depot is sent to location i, 0 otherwise
y_j	1 if customer j's delivery is postponed in the first stage, 0 otherwise
w_{ij}	1 if customer j's package is sent by mobile depot to location i, 0 otherwise
p_{ijk}	1 if customer j is served through mobile depot at location i by crowd-shipper k, 0 otherwise
v_j	1 if customer j is not served in the second stage, 0 otherwise

3 求解方法

3.1 Sample Average Approximation (SAA)

由於情境的總數量是$2^{|K|}$，此數量以指數的形式成長，當考慮的通勤者數量增加時，就會造成第二階段模型的求解十分困難。Mousavi 等人採用 sample average approximation (簡稱 SAA)的方法克服此運算上的困難。SAA 會從所有可能的情境當中，抽取出獨立且相同分布(independent and identically distributed, i.i.d.)的樣本 S，並針對此樣本去求解出相對應的 extensive form (EF)：

$$\min \sum_{i\in I} c_i^z z_i + \sum_{j\in J} c_j^y y_j + \frac{1}{|S|}\sum_{s\in S}\left(\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk}^p p_{ijks} + \sum_{j\in J} c_j^v v_{js}\right) \quad (7a)$$

$$\text{s.t.} \quad w_{ij} \le z_i \qquad \forall i\in I, j\in J, \quad (7b)$$

$$\sum_{i\in I} w_{ij} + y_j = 1 \qquad \forall j\in J, \quad (7c)$$

$$\sum_{j\in J} w_{ij} \le C \qquad \forall i\in I, \quad (7d)$$

$$\sum_{i\in I}\sum_{j\in J} p_{ijks} \le \xi_k^s \qquad \forall k\in K, s\in S, \quad (7e)$$

$$\sum_{i\in I}\sum_{k\in K} p_{ijks} + v_{js} = 1 - y_j \qquad \forall j\in J, s\in S, \quad (7f)$$

$$\sum_{k\in K} p_{ijks} \le w_{ij} \qquad \forall i\in I, j\in J, s\in S, \quad (7g)$$

$$p_{ijks} = 0 \qquad \forall j\in J, s\in S, (i,k)\in I\times K \text{ s.t. } t(i)\ne t(k), \quad (7h)$$

$$\mathbf{z}\in\{0,1\}^{|I|}, \mathbf{y}\in\{0,1\}^{|J|}, \mathbf{w}\in\{0,1\}^{|I||J|}, \quad (7i)$$

$$\mathbf{p}\in\{0,1\}^{|I||J||K||S|}, \mathbf{v}\in\{0,1\}^{|I||S|}. \quad (7j)$$

，以此 EF 的目標值來進行統計估計。

研究中以 SAA 估計 SMDCP 目標值的上界(upper bound)及下界(lower bound)的信賴區間，兩者分別的作法如下：

(i) 上界之信賴區間：

首先抽取出一個 i.i.d.的樣本 S，並求解 S 對應的 EF 後，取得一組解$(\hat{z}(S), \hat{y}(S), \hat{w}(S))$，以及這組解在 S 對應的 EF 當中的目標值 $\hat{v}(S)$，但 $\hat{v}(S)$ 並非$(\hat{z}(S), \hat{y}(S), \hat{w}(S))$真正的目標值，其真正的目標值為

$$U(S) := F(\hat{z}(S), \hat{y}(S)) + \mathbb{E}_{\xi}[Q(\hat{y}(S), \hat{w}(S), \xi)] \text{。}$$

由於$(\hat{z}(S), \hat{y}(S), \hat{w}(S))$是一組可行解，因此$U(S)$ 會是原問題的一個上界。研究當中再額外抽取出另一個更大的樣本，稱為 evaluation sample，以符號S^{eval}表示，以此樣本估計$(\hat{z}(S), \hat{y}(S), \hat{w}(S))$真正的目標值$U(S)$，其估計之信賴區間即為

上界之信賴區間：

$$U^{mean}(S) \pm (z-score)_{\alpha/2} U^{std}(S)\ ,$$

其中 $U^{mean}(S)$ 及 $U^{std}(S)$ 分別為 $\{F(\hat{z}(S),\hat{y}(S))+Q(\hat{y}(S),\hat{w}(S),\xi^S)\}_{s\in S^{eval}}$ 這組數據之平均值與標準差。

(ⅱ) 下界之信賴區間：

下界之估計來自於以下的關係式：

$$\mathbb{E}_{S\in\Omega_N}[\hat{v}(S)] \le v^*,\ \ \forall N \in \mathbb{N} \quad (3),$$

其中 $\hat{v}(S)$ 為求解樣本 S 對應的 EF 後得到的目標值，Ω_N 為所有大小為 N 的樣本形成的集合，v^* 則為 SMDCP 真正的目標值。由關係式(3)可知，在給定任一樣本大小 N 時，$\mathbb{E}_{S\in\Omega_N}[\hat{v}(S)]$ 為此問題的一個下界。研究中抽取出 M 組大小為 N 的 i.i.d. 樣本 $\{S_1,S_2,...,S_M\}$，且各組樣本之間也滿足 i.i.d.的特性，分別求解各組樣本對應的 EF 後，會得到 M 項目標值 $\{\hat{v}(S_1),\hat{v}(S_2),...,\hat{v}(S_M)\}$，以此組數據估計下界 $\mathbb{E}_{S\in\Omega_N}[\hat{v}(S)]$ 之信賴區間：

$$L^{mean}(S) \pm (t-score)_{\alpha/2,M-1} L^{std}(S)\ ,$$

其中 $L^{mean}(S)=\dfrac{1}{M}\sum_{i=1}^{M}\hat{v}(S_i)$，$L^{std}(S)=\dfrac{1}{M-1}\sum_{i=1}^{M}[\hat{v}(S_i)-L^{mean}(S)]^2$。

取得上下界之信賴區間後，Mousavi 等人取出目標值上界之信賴區間上界，以及目標值下界之信賴區間下界，以衡量 optimality gap，即

$$\text{Optimality gap} = \frac{\left\{\begin{aligned}&[U^{mean}(S)+(z-score)_{\alpha/2}U^{std}(S)]\\&\qquad\qquad -[L^{mean}(S)-(t-score)_{\alpha/2,M-1}L^{std}(S)]\end{aligned}\right\}}{U^{mean}(S)+(z-score)_{\alpha/2}U^{std}(S)}\ 。$$

衡量上界的信賴區間需要求解一個樣本對應的 EF，得到一個問題的可行解，而這個解真正的目標值需盡可能靠近 SMDCP 真正的目標值，才能估計出比較窄(tight)的 optimality gap。為了找到這樣的解，研究中先抽取用來尋找下界的 M 組

樣本$\{S_1, S_2, ..., S_M\}$，並求解各組樣本的 EF 後，得到 M 組可行解，接著再另外抽取出一個中型的樣本(midsize evaluation sample)，以符號S^{pick}表示，並將 M 組可行解分別代入到S^{pick}對應的 EF 當中，以求得$\{U^{mean}(S_1), U^{mean}(S_2), ..., U^{mean}(S_M)\}$，從中找到樣本$\tilde{S}=S_\ell$，其中$\ell = \arg\min_{i=1,...,M}\{U^{mean}(S_i)\}$，並以$\tilde{S}$對應的可行解來估計 SMDCP 的上界信賴區間。

3.2 Multicut L-shaped 方法

SMDCP 的第二階段模型含有整數限制，看似無法採用 multicut L-shaped 方法，而應採用 integer L-shaped 方法進行求解，但 Mousavi 等人證明 SMDCP 的第二階段模型具有 total unimodularity 的性質，將整數限制放鬆進行求解，仍可以得到整數解，因此第二階段模型可以使用放鬆整數限制的模型來求解，且不需要進行 branch-and-bound 的過程就可以得到最佳解，而以此放鬆過後的模型求解，即可取得各個限制式的對偶值，如此 multicut L-shaped 方法便得以用於 SMDCP 的求解。

研究當中，multicut L-shaped 方法用於求解樣本 S 對應的 EF，以下為詳細步驟：

步驟 1. 抽取 i.i.d.樣本 S。

步驟 2. 設定 $r = t = 0$ 和 $q_s = 0, \ \forall s \in S$。

步驟 3. 設定 $t = t + 1$，使用 branch-and-bound 求解以下模型：

$$\min \ \sum_{i\in I} c_i^z z_i + \sum_{j\in J} c_j^y y_j + \frac{1}{|S|}\sum_{s\in S}\theta_s \tag{8a}$$

$$\text{s.t.} \quad w_{ij} \le z_i \qquad \forall i \in I, j \in J, \tag{8b}$$

$$\sum_{i\in I} w_{ij} + y_j = 1 \qquad \forall j \in J, \tag{8c}$$

$$\sum_{j\in J} w_{ij} \le C \qquad \forall i \in I, \tag{8d}$$

$$\theta_x \ge \sum_{k\in K}\hat{\gamma}_{k,\ell(s)}\xi_k^s + \sum_{j\in J}\hat{\pi}_{j,\ell(s)}(1-y_j) + \sum_{i\in I}\sum_{j\in J}\hat{\beta}_{ij,\ell(s)}w_{ij} \qquad \forall s \in S, \ell(s) = 1, ..., q_s \tag{8e}$$

$$\mathbf{z} \in \{0,1\}^{|I|}, \mathbf{y} \in \{0,1\}^{|J|}, \mathbf{w} \in \{0,1\}^{|I||J|}. \tag{8f}$$

其中(8e)為 optimality cut。若限制式(8e)尚未出現在模型中，則設定$\theta_s^t = -\infty$，並於求解之前暫時將 θ_s 從目標式(8a)當中除去。

步驟 4. 令 $[(\hat{z}_i^t)_{i\in I}, (\hat{y}_j^t)_{j\in J}, (\widehat{w}_{ij}^t)_{i\in I, j\in J}, (\theta_s^t)_{s\in S}]$ 為模型(8a)-(8f)的最佳解。對於所有$s \in S$，求解以下模型：

$$\min \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk}^{p} p_{ijk} + \sum_{j\in J} c_j^{v} v_j \tag{9a}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{j\in J} p_{ijk} \le \xi_k^s \qquad \forall k \in K, \tag{9b}$$

$$\sum_{i\in I}\sum_{k\in K} p_{ijk} + v_{js} = 1 - y_j \qquad \forall j \in J, \tag{9c}$$

$$\sum_{k\in K} p_{ijk} \le w_{ij} \qquad \forall i \in I, j \in J, \tag{9d}$$

$$p_{ijk} = 0 \qquad \forall j \in J, (i,k) \in I \times K \text{ s.t. } t(i) \ne t(k), \tag{9e}$$

$$\mathbf{p} \in \{0,1\}^{|I||J||K|}, \mathbf{v} \in \{0,1\}^{|I|}. \tag{9f}$$

令 $\hat{\gamma}_{k,s}^{t}$，$\hat{\pi}_{j,s}^{v}$，$\hat{\beta}_{ij,s}^{v}$ 分別為限制式(9b)、(9c)、(9d)的對偶值，若關係式

$$\theta_s^t < \sum_{k\in K} \hat{\gamma}_{k,s}^{t} \xi_k^s + \sum_{j\in J} \hat{\pi}_{j,s}^{t} (1 - \hat{y}_j^t) + \sum_{i\in I}\sum_{j\in J} \hat{\beta}_{ij,s}^{t} \hat{w}_{ij}^t \tag{10}$$

成立，則將以下 optimality cut 加入到模型(8a)-(8f)當中：

$$\theta_s \ge \sum_{k\in K} \hat{\gamma}_{k,s}^{t} \xi_k^s + \sum_{j\in J} \hat{\pi}_{j,s}^{t} (1 - y_j) + \sum_{i\in I}\sum_{j\in J} \hat{\beta}_{ij,s}^{t} w_{ij} \quad ,$$

並設定$q_s = q_s + 1$。

若關係式(10)在所有$s \in S$的求解結果當中都不成立，則

$[(\hat{z}_i^t)_{i\in I}, (\hat{y}_j^t)_{j\in J}, (\hat{w}_{ij}^t)_{i\in I, j\in J}]$為最佳解，前往步驟 5；否則，返回步驟 3。

步驟 5. 給定最佳解$[(\hat{z}_i^t)_{i\in I}, (\hat{y}_j^t)_{j\in J}, (\hat{w}_{ij}^t)_{i\in I, j\in J}]$，求解模型(7a)-(7j)以取得樣本 S 的目標值$\hat{v}(S)$。

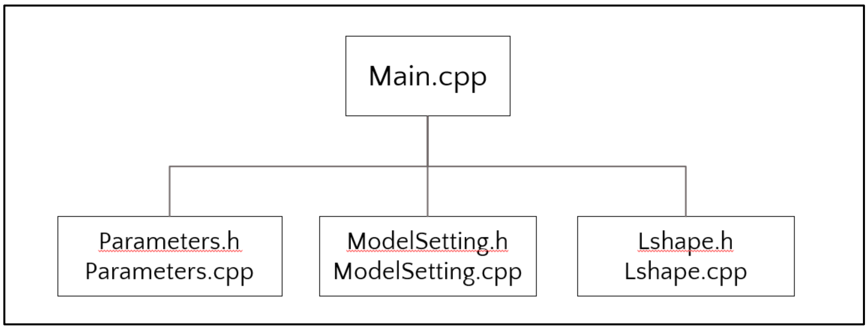
由於第二階段模型允許延後配送，因此第一階段的任意可行解，在第二階段當中也會可行，故 SMDCP 的數學模型具有 relatively complete recourse，在這樣的特性下，multicut L-shaped 方法可以跳過產生 feasibility cut 的步驟。

4 實作求解方法之細節

4.1 程式碼架構及細節

程式碼架構如圖二所示，其中 main.cpp 的流程以圖三的虛擬碼呈現。在 main.cpp 底下可分成三個部分：Parameters、modelSetting、Lshaped。Parameters 的主要功能包括參數的輸入、抽樣以及計算數據的平均值和標準差；modelSetting 的工作為設定模型的目標式和限制式；Lshped 的功能為求解各個樣本的 EF，其中包括 multicut L-shaped 方法、第一階段模型的 branch-and-bound(檢查整數限制是否滿足、分支變數的選擇、產生分支(branching)、結束分支(fathom)等工作)、

以及給定第一階段可行解後計算第二階段模型目標值的功能。



圖二 程式碼架構

Input: One SMDCP(P), $sampleSize$, $numSamples$, $pickSize$, $evalSize$;
$Lowerbounds := \varnothing$;
best solution $(\tilde{\mathbf{z}}, \tilde{\mathbf{y}}, \tilde{\mathbf{w}}) := \varnothing$;
$minUpperMean := \infty$;
$i := 0$;
midesize evaluation sample S^{pick} := Sampling(pickSize);
while $i < numSamples$ **do**
  sample S_i := Sampling(sampleSize);
  solution $(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \hat{\mathbf{w}})$:= Lshaped(S_i);
  Evaluate the objective value of $(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \hat{\mathbf{w}})$ in S_i's EF, $\hat{v}(S_i)$;
  Add $\hat{v}(S_i)$ to $LowerBounds$;
  Evaluate the objective value of $(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \hat{\mathbf{w}})$ in S^{pick}'s EF, $U^{mean}(S_i)$;
  if $U^{mean}(S_i) < minUpperMean$ **then**
    $(\tilde{\mathbf{z}}, \tilde{\mathbf{y}}, \tilde{\mathbf{w}}) := (\hat{\mathbf{z}}, \hat{\mathbf{y}}, \hat{\mathbf{w}})$;
    $minUpperMean := U^{mean}(S_i)$;
  end
  $i := i + 1$;
end
S^{eval} := Sampling($evalSize$);
Evaluate U^{mean} and U^{std} by $(\tilde{\mathbf{z}}, \tilde{\mathbf{y}}, \tilde{\mathbf{w}})$ and S^{eval};
Calculate L^{mean} and L^{std} with data in $LowerBounds$;
Output U^{mean}, U^{std}, L^{mean}, L^{std};

圖三 main.cpp 之流程

4.2 Branch-and-Bound 策略

本實作當中，搜尋分支樹的策略是採用 depth-first-search (DFS)，而分支變數的選擇順序為$\boldsymbol{z}$、$\boldsymbol{y}$、$\boldsymbol{w}$，其次再以下標數字的大小作為分支變數選擇的先後順序，下標數字較小者優先作為分支變數。之所以讓 $\boldsymbol{z}$ 和 $\boldsymbol{y}$ 優先作為分支變數，乃因這兩者的變數數量較少，在分支早期可以減少分支的數量，此外，經由分析限制式的結構，可發現 $\boldsymbol{z}$ 和 $\boldsymbol{y}$ 的變數值給定後，有一定比率的 $\boldsymbol{w}$ 變數值也會跟著確定，因此可以減少分支晚期 $\boldsymbol{w}$ 的分支數。

為了加快尋得整數解的速度，實作當中將限制式(8b)和(8d)整併為以下限制式：

$$\sum_{j \in J} w_{ij} \le C\, z_i, \quad \forall i \in I \qquad (11),$$

如此可減少第一階段模型的限制式數量。此外，經過對問題的分析，額外再第一階段的模型當中加入一條 valid inequality：

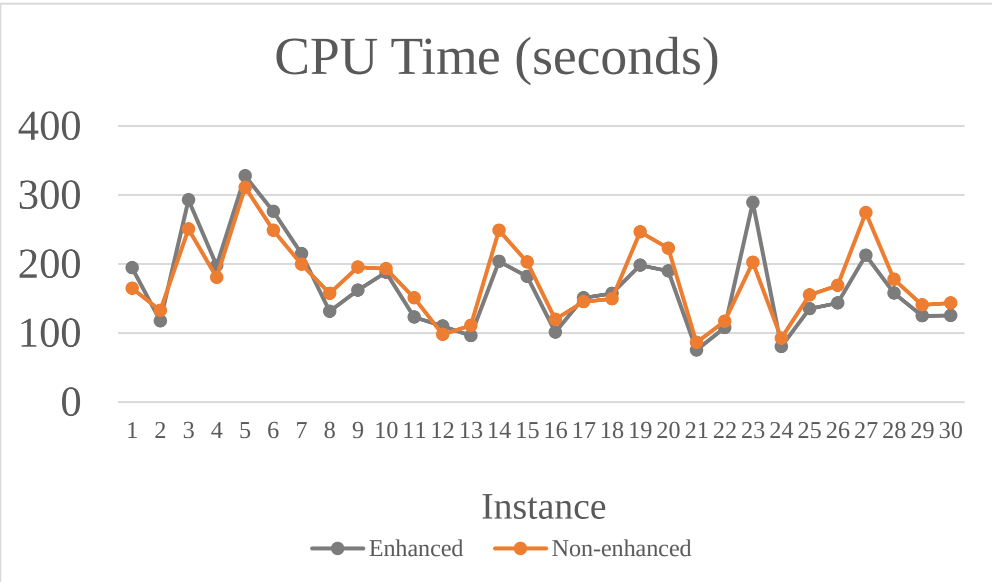
$$C\sum_{i\in I} z_i \geq |J| - \sum_{j\in J} y_j \qquad (12)，$$

限制式(12)的右式代表公司決定當天要配送的貨物總量，而左式則為所有出勤的移動倉之容量總量，此容量總量須大於當天要配送的貨物總量。在限制式(12)的作用下，當 $\boldsymbol{z}$ 的變數值確定後，$\boldsymbol{y}$ 的變數值會更容易變為整數，如此可以減少分支的數量。

5 實驗結果

5.1 加快 Branch-and-Bound 策略之效果

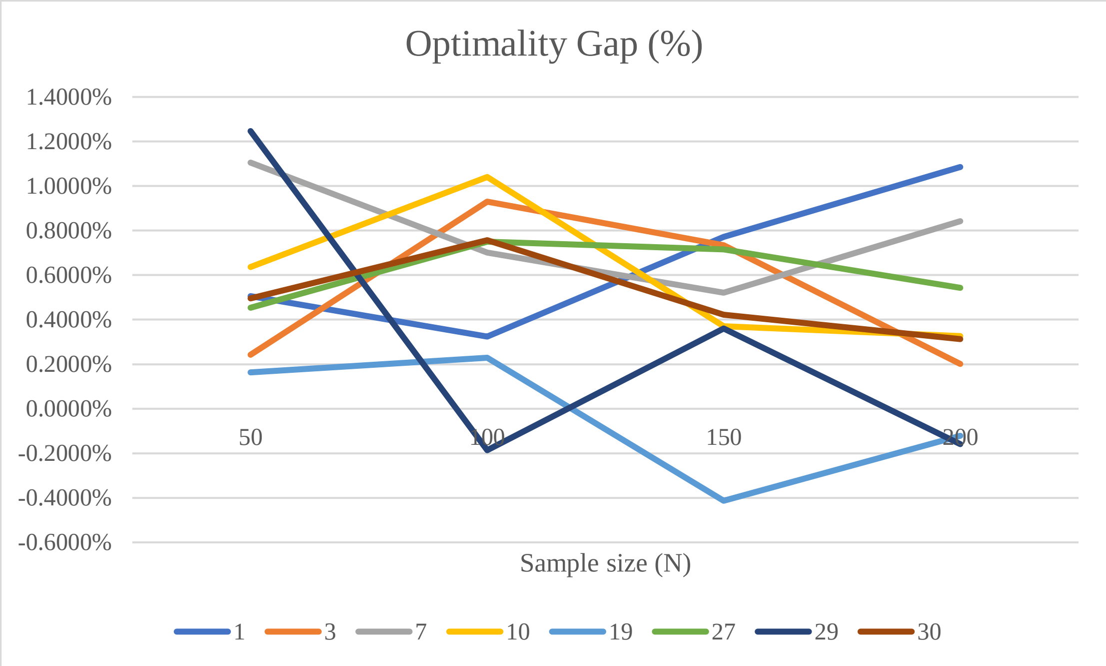
在 4.2 小節當中提到，實作中有將限制式(8b)和(8d)整併為限制式(11)，並加入了限制式(12)作為縮小解空間的 valid inequality。在實驗當中以 30 組例子來檢驗這些加速 branch-and-bound 策略之效果，此 30 組例子之參數如下：$|I|$=10、$|J|$=10、$|K|$=30、N= 30、M=25、$|S^{eval}|$ = 1000。圖四為求解時間的實驗結果，橫軸為例子編號，縱軸為求解時間，其中 enhanced 代表有加入加強策略的求解時間，non-enhanced 則無加入加強策略，從圖上可以看到 enhanced 並不是在所有例子當中都有較少的求解時間，且兩結果的求解時間差異並不大。進一步推測原因後，我認為影響複雜度最大的部分並不是在第一階段的模型大小，而是在樣本大小(N)，因此在 branch-and-bound 部分稍為加速無法讓求解效率顯著增加。



圖四 加快 Branch-and-Bound 策略之測試結果

5.2 Optimality gap 的有效性

一個有效的 optimality gap，其值應隨著樣本大小增加而減少，且最終應收斂至問題真正的目標值。此實驗以 8 組例子來驗證 optimality gap 的有效性，這 8 組例子之參數如下：$|I|$=10、$|J|$=10、$|K|$=30、M=25、$|S^{eval}|$ = 10000。圖五為實驗結果，橫軸為樣本大小(N)，縱軸為 optimality gap，而各曲線之號碼為例子編號，由此圖可知，當樣本大小增加，optimality gap 並沒有減少的趨勢，證實老師在我上台報告時提出的懷疑：3.1 小節提出之下界並不是很有效的下界。



圖五 Optimality gap 有效性實驗結果

6 結論

本次期末專題實作 SMDCP 的求解，採用 Mousavi 等人(2022)在研究中使用的 sample average approximation 以及 multicut L-shapled 方法，並於實作當中設計程式碼架構及內容，以及嘗試加入額外的加速策略以加快 SMDCP 第一階段的求解速度，經過實驗後，發現這些加速策略並沒有顯著的效果。在有關於 optimality gap 有效性的實驗當中，也得知 Mousavi 等人提出的下界確實並不是很有效，而如何找到 SMDCP 更有效的下界，是未來很值得研究的題目。

參考文獻

[1] Mousavi, K., Bodur, M., & Roorda, M. J. (2022). Stochastic last-mile delivery with crowd-shipping and mobile depots. *Transportation Science*, *56*(3), 612-630.