

國立交通大學
運輸與物流管理學系

碩士論文

整合O2O餐點外送之生產與配送問題

Integrating Production and Distribution in
O2O Meal Delivery

研究生：鄭介東

指導教授：張宗勝 教授

中華民國 一零八 年 一 月

整合O2O餐點外送之生產與配送問題

Integrating Production and Distribution in O2O Meal Delivery

研 究 生：鄭介東

Student：Chieh-Tung Cheng

指導教授：張宗勝 教授

Advisor：Dr. Tsung-Sheng Chang

國立交通大學

運輸與物流管理學系

碩 士 論 文

A Thesis

Submitted to Department of Transportation and Logistics Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Logistics Management

中華民國一零八年一月

整合O2O餐點外送之生產與配送問題

研究生：鄭介東

指導教授：張宗勝 教授

國立交通大學
運輸與物流管理學系碩士班

摘要

近年來，O2O (online-to-offline) 餐點外送平台在世界各地蓬勃發展，例如：空腹熊貓 (foodpanda)、優步快送 (UberEats)、誠實蜜蜂 (Honestbee)。顧客可以利用手機 app 訂購向平台訂購與平台合作餐廳的餐點，平台接到訂單後必須與餐廳決策是否接受該訂單。而餐廳與平台是否接受訂單的決策會受到餐廳內部生產排程與平台外送員路線規劃的影響。於是，本研究提出一個新的方式，站在平台的角度，整合餐廳內部的生產排程與平台本身的外送員路線規劃，並決策接受哪些外送訂單。此外，由於未來的訂單情形也同樣影響餐廳與平台決策目前所接到的訂單，所以本研究將未來可能出現的預測訂單也納入決策，這個方式類似滾動時域架構 (rolling horizon approach)，即同時決策目前接到與預測訂單，但最後僅使用目前接到訂單的結果。最後，本研究的目標為最大化平台利潤。

本研究首先會介紹餐廳生產排程問題，利用彈性零工問題 (flexible job shop scheduling problem) 描述實際上的餐廳生產排程問題，並將其轉換成平行機台排程 (parallel machine scheduling problem) 問題，再轉換成與配送問題同樣形式的網路問題，及有時窗限制的團隊尋寶問題 (team orienteering problem with time window)。再介紹平台外送員配送路線規劃問題，即有時窗限制的收送貨問題 (pickup and delivery problem with time window)。最後介紹如何整合生產與配送問題，並針對該問題設計一個混合整數規劃最佳化模型 (mix integer programming model)。

為了使本研究符合實務上緊迫的求解時間，通常為 1 分鐘。本研究設計一個分層最佳化架構 (hierarchical optimization)。利用 IterILS (Iterated Local Search) 求解生產問題再將其結果代入配送問題內，利用 ALNS (Alternative Large Neighborhood Search) 求解。接著，設計小例題題組使用 Gurobi 求出本問題的最佳解與使用 IterILS 和 ALNS 求解本問題的近似解進行績效測試。最後利用實務 O2O 餐點外送平台的資料進行實務例題分析。

關鍵詞：O2O 餐點外送平台、整合生產與配送問題、滾動時域架構、彈性零工排程問題、平行機台排程問題、有時窗限制的團隊尋寶問題、有時窗限制的收送貨問題、ILS 演算法、ALNS 演算法。

Integrating Production and Distribution in O2O Meal Delivery

Student: Chieh-Tung Cheng

Advisor: Dr. Tsung-Sheng Chang

Department of Transportation and Logistics Management
National Chiao Tung University

ABSTRACT

These years, O2O (online-to-offline) meal delivery platforms, such as Foodpanda, Ubereats and Honestbee, are becoming more and more popular. The customers can order the meals from the restaurants which coordinate with platform. After the customers make the orders, the platform and restaurants decide if this order is accepted. However, the acceptance of the orders will be influenced by the production schedule in the restaurants and the delivery plan of the couriers. Therefore, a unique idea is proposed, that is, the platform integrates the production schedule in the restaurants and the distribution plan of the couriers, decides which order should be accepted. Moreover, the predicted orders are included in the planning. By doing so, the characteristic of this process is similar to the rolling horizon approach. The target of this research is maximizing the profit of the O2O meal delivery platform.

The production problem is described as the flexible job shop scheduling problem. Then, the new method to transform the flexible job shop scheduling problem into the parallel machine scheduling problem is proposed. Furthermore, in order to coordinate production problem in the restaurants and the distribution problem of the couriers, we transform the parallel machine scheduling problem into team orienteering problem with time window. The pickup and delivery problem with time window is applied to describe the distribution problem. Finally, the integrated problem is formulated as a mix integer programming model.

In general, the platform should decide which order should be accepted in 1 minute. Due to the urgent time duration, we design a hierarchical optimization scheme to solve it, that is, we use Iterated Local Search algorithm to solve the production problem, and solve the distribution problem by using Alternative Large Neighborhood Search algorithm.

Lastly, the proposed algorithm is applied to the small data set and the empirical data set to test its performance, and the results are analyzed in detail.

Keywords: O2O meal delivery platform, integrated production and distribution problem, rolling horizon approach, flexible job shop scheduling problem, parallel machine scheduling problem, team orienteering problem with time window, pickup and delivery problem with time window, Iterated Local Search algorithm, Alternative Large Neighborhood Search algorithm.

誌謝

謝謝 張宗勝老師 不辭辛勞的指導

謝謝 王逸琳老師 丁慶榮老師 給予的口試審查意見

謝謝 我的父母 沒有給我太多情緒上的壓力

謝謝 方家小館 方老闆 提供餐廳實務流程與資訊

謝謝 饕客送 周經理 提供外送平台實務流程與資訊

謝謝 我的朋友 鼓勵我而不是不看好我

學生必須將完成這篇稍微特別的論文的功勞歸功於您們

鄭介東謹誌

于交大 TS 實驗室 2019.01

目錄

1	緒論	1
1.1	研究背景與目的.....	1
1.2	研究流程.....	23
1.3	章節簡述.....	25
2	文獻回顧	25
3	問題描述	27
3.1	O2O 餐點外送模式之利害關係人.....	27
3.2	規劃長度與決策時段定義.....	33
3.3	餐點完成時間與取餐和送達時間之關係.....	40
3.4	生產排程問題.....	47
3.5	配送路線規劃問題.....	72
3.6	整合生產排程與配送路線規劃問題.....	90
4	數學模型	92
5	演算法	111
5.1	子問題分割與分層最佳化架構.....	111
5.2	求解 TOPTWRC 問題之演算法	116
5.3	求解 MDPDPTWPR 問題之演算法.....	138
6	實例分析	160
6.1	小例題分析.....	160
6.2	實務例題分析.....	166
7	結論與建議	175
8	附錄	177
9	參考資料	177

表目錄

表 1.1 全球餐點外送平台獲得的創投投資金額.....	1
表 1.2 台灣四大餐點外送平台簡介.....	2
表 1.3 平台與餐廳針對即刻送與預訂單分別所需做的決策.....	22
表 2.1 本研究相關文獻整理.....	27
表 3.1 考慮未來可能出現的訂單之優點.....	29
表 3.2 平台在決策時擁有的訂單資訊.....	30
表 3.3 內用與外送訂單詳細資訊.....	30
表 3.4 平台評估餐廳與平台接受或拒絕訂單的可能情境.....	32
表 3.5 已規劃時段、目前規劃時段與預測時段所包含的訂單.....	35
表 3.6 已規劃時段、目前規劃時段與預測時段所包含的訂單狀態與餐點狀態之對應.....	48
表 3.7 餐點狀態之說明.....	48
表 3.8 FJSPSO 問題之範例.....	52
表 3.9 生產排程問題範例一的餐點狀態.....	61
表 3.10 範例一內用餐點完成時間的可行性檢驗.....	62
表 3.11 範例一外送餐點完成時間的可行性檢驗.....	63
表 3.12 範例二的餐點狀態.....	66
表 3.13 範例二正在製作的餐點完成時間的可行性檢驗.....	68
表 3.14 範例二已被餐廳與平台同時接受但尚未開始製作外送餐點完成時間的可行性檢驗	68
表 3.15 轉換成路徑規劃問題之方式.....	70
表 3.16 已規劃時段、目前規劃時段與預測時段所包含的訂單與取餐節點狀態.....	76
表 3.17 虛擬取餐節點狀態之說明.....	77
表 3.18 已規劃時段、目前規劃時段與預測時段所包含的訂單與顧客節點狀態.....	78
表 3.19 顧客節點狀態之說明.....	78
表 3.20 配送路線規劃範例一的餐點狀態.....	81
表 3.21 配送路線規劃範例一的虛擬取餐與顧客節點狀態.....	81
表 3.22 範例一的外送訂單配送可行性檢驗.....	83
表 3.23 配送路線規劃範例二出現的餐點狀態.....	86
表 3.24 配送路線規劃範例二出現的虛擬取餐與顧客節點狀態.....	86
表 3.25 範例二的外送訂單配送可行性檢驗.....	89
表 5.1 為 TOPTW 與 TOPTWRC 的問題環境之整理.....	114
表 5.2 PDPTWSR 與 MDPDPTWSR 的問題環境之整理.....	115
表 6.1 小例題之餐廳與餐點資訊.....	161
表 6.2 外送平台資訊.....	161
表 6.3 本研究設計之小例題資料.....	162

表 6.4 子問題 1 的 Gurobi 求出的解與 IterILS 和 ALNS 演算法求出的解之比較	163
表 6.5 子問題 2 的 Gurobi 求出的解與 IterILS 和 ALNS 演算法求出的解比較	163
表 6.6 整合問題的 Gurobi 求得的解與 IterILS 和 ALNS 演算法求得的解比較	165
表 6.7 T 公司提供之實務資料	166
表 6.8 本研究模擬之空腹熊貓實務資料.....	166
表 6.9 T 公司星期天晚上之訂單點餐時間	168
表 6.10 本研究模擬之空腹熊貓星期天晚上之顧客點餐時間與訂單數量.....	169
表 6.11 本研究預測之空腹熊貓星期天晚上之顧客點餐時間與訂單數量.....	169
表 6.12 本研究之實務例題對應之題組.....	169
表 6.13 實務例題之餐廳與餐點資訊.....	170
表 6.14 T 公司一周晚上之訂單點餐時間	171
表 6.15 實務例題之餐廳數、生產網路節點數與配送網路節點數.....	172
表 6.16 實務例題之每個餐廳的內用與外送餐點段應的生產網路節點數.....	172
表 6.17 實務例題之求解結果.....	172
表 6.18 例題 1 的平台拒絕餐廳接受訂單之訂單資訊.....	174



圖目錄

圖 1.1 利害關係人間的即刻送訂單資訊流向圖(空腹熊貓).....	3
圖 1.2 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(空腹熊貓).....	4
圖 1.3 若餐廳接受即刻送訂單-平台告知顧客與外送員之資訊(空腹熊貓).....	4
圖 1.4 若餐廳拒絕即刻送訂單-平台告知顧客之資訊(空腹熊貓).....	5
圖 1.5 平台承諾顧客的即刻送訂單送達時窗(空腹熊貓).....	5
圖 1.6 利害關係人間的即刻送訂單資訊流向圖(誠實蜜蜂).....	6
圖 1.7 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(誠實蜜蜂).....	7
圖 1.8 若餐廳接受即刻送訂單-平台告知顧客與外送員之資訊(誠實蜜蜂).....	7
圖 1.9 若餐廳拒絕即刻送訂單-平台告知顧客之資訊(誠實蜜蜂).....	8
圖 1.10 平台承諾顧客的即刻送訂單送達時窗(誠實蜜蜂).....	8
圖 1.11 利害關係人間的即刻送訂單資訊流向圖(優步快送).....	9
圖 1.12 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(優步快送).....	10
圖 1.13 若餐廳接受即刻送訂單-平台告知顧客與外送員之資訊(優步快送).....	11
圖 1.14 若餐廳接受即刻送訂單-平台告知顧客之資訊(優步快送).....	11
圖 1.15 平台承諾顧客的即刻送訂單送達時窗(優步快送).....	11
圖 1.16 利害關係人間的即刻送訂單資訊流向圖(饕客送).....	12
圖 1.17 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(饕客送).....	12
圖 1.18 若餐廳接受即刻送訂單-平台告知顧客與外送員之資訊(饕客送).....	13
圖 1.19 若餐廳拒絕即刻送訂單-平台告知顧客之資訊(饕客送).....	13
圖 1.20 平台承諾顧客的即刻送訂單送達時窗(饕客送).....	14
圖 1.21 利害關係人間的預訂單資訊流向圖(空腹熊貓).....	15
圖 1.22 預訂單訊息和餐廳與平台決策結果傳遞方向(空腹熊貓).....	15
圖 1.23 平台告知顧客與外送員之資訊(空腹熊貓).....	16
圖 1.24 空腹熊貓-平台提供顧客選擇的預訂單送達時間點.....	16
圖 1.25 空腹熊貓-顧客可指定的預訂單送達時間點.....	17
圖 1.26 利害關係人間的預訂單資訊流向圖(誠實蜜蜂).....	17
圖 1.27 誠實蜜蜂-平台供顧客選擇的預訂單送達時窗.....	18
圖 1.28 誠實蜜蜂承諾預訂單的送達時間.....	18
圖 1.29 利害關係人間的預訂單資訊流向圖(優步快送).....	19
圖 1.30 優步快送-平台供顧客選擇的預訂單送達時窗.....	19
圖 1.31 優步快送-顧客可指定的預訂單送達時窗.....	20
圖 1.32 利害關係人間的預訂單資訊流向圖(饕客送).....	20
圖 1.33 饕客送-平台供顧客選擇的預訂單送達時間點.....	20
圖 1.34 饕客送-顧客可指定的預訂單送達時窗.....	21
圖 1.35 研究流程圖.....	24

圖 3.1 利害關係人間的訂單資訊與流向圖 (本研究提出).....	28
圖 3.2 餐廳內部資訊與訂單資訊流向圖 (本研究提出).....	28
圖 3.3 平台決策結果傳遞方向-平台建議餐廳接受訂單 (本研究提出).....	31
圖 3.4 平台決策結果傳遞方向-平台建議餐廳拒絕訂單 (本研究提出).....	32
圖 3.5 餐廳營業時間.....	33
圖 3.6 規劃長度與決策時段之關係圖.....	34
圖 3.7 目前外送平台的決策時間點.....	35
圖 3.8 規劃時段時間長度與模型計算時間長度.....	36
圖 3.9 規劃長度中的規劃時段劃分方式.....	37
圖 3.10 已規劃時段與規劃時段劃分方式.....	37
圖 3.11 預測時段時間區間.....	39
圖 3.12 規劃長度中的已規劃時段、規劃時段與預測時段劃分方式.....	39
圖 3.13 內用訂單餐點最早可完成時間.....	40
圖 3.14 內用訂單的服務時窗.....	41
圖 3.15 待平台評估餐廳與平台是否接受的外送訂單餐點可完成時間.....	41
圖 3.16 待平台評估餐廳與平台是否接受的外送訂單取餐時間與餐點完成時間之關係	42
圖 3.17 待平台評估餐廳與平台是否接受的外送訂單的服務時窗.....	42
圖 3.18 目前規劃時段的外送訂單外送員的必須送達時段.....	43
圖 3.19 餐廳可完成已被餐廳與平台同時接受即刻送訂單餐點的時間段.....	44
圖 3.20 已被餐廳與平台同時接受的即刻送訂單之必須取餐時間段.....	44
圖 3.21 已被餐廳與平台同時接受的即刻送訂單的必須送達時間段.....	45
圖 3.22 已被餐廳與平台同時接受的預訂單的餐點須完成時間段.....	46
圖 3.23 已被餐廳與平台同時接受的預訂單的送達時間段.....	46
圖 3.24 餐廳的餐點可能的製作步驟(e.g. 漢堡與薯條).....	50
圖 3.25 餐廳廚房的工作站組成範例.....	51
圖 3.26 內用餐點與外送餐點的可能的完成時間.....	51
圖 3.27 FJSPSO 問題的分離節線圖.....	53
圖 3.28 餐點製作步驟與該步驟對應的工作站.....	55
圖 3.29 某餐廳在樂觀且長期穩定狀態下製作漢堡的情形.....	56
圖 3.30 漢堡製作時間長度計算方式.....	57
圖 3.31 漢堡每分鐘只能同時在 2 台平行機台上生產的情形.....	58
圖 3.32 PMSPRS 問題之示意圖.....	59
圖 3.33 範例一、二和三對應的決策時段圖.....	59
圖 3.34 範例一的可行生產排程甘特圖.....	61
圖 3.35 範例一餐點種類與生產排程對照圖.....	64
圖 3.36 規劃時段 2 決策開始時正在製作的餐點.....	66
圖 3.37 規劃時段 2 決策開始時尚未製作的外送餐點.....	67

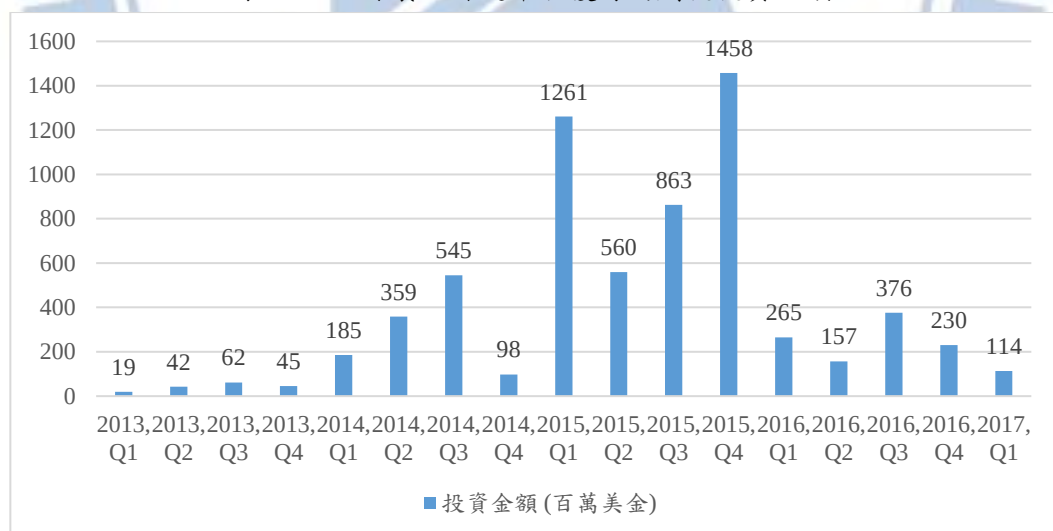
圖 3.38 範例二的可行生產排程甘特圖	67
圖 3.39 餐廳可完成範例二餐點 5 與 6 的時間段	69
圖 3.40 範例二餐點種類與生產排程對照圖	69
圖 3.41 範例二對應的 TOPTWRC 問題之網路圖	72
圖 3.42 範例二的可行生產排程甘特圖與可行路線規劃圖的對照圖	72
圖 3.43 平台在規劃時段 1 決策的外送員 1 與 2 的路線範例	73
圖 3.44 外送員 1 與 2 在規劃時段 2 決策開始時的出發位置	74
圖 3.45 平台在規劃時段 1 所規劃的外送員 1 與 2 的路線範例	74
圖 3.46 外送員 1 與 2 在決策時段 2 決策時的車廂剩餘容量	75
圖 3.47 在某決策時段規劃的有預測訂單之外送員路線圖	75
圖 3.48 MDPDPTWSR 問題之示意圖	79
圖 3.49 配送路線規劃問題範例一的配送網路圖	82
圖 3.50 範例一的可行配送路線圖	83
圖 3.51 範例一路線圖上虛擬取餐與顧客節點的訂單餐點體積	84
圖 3.52 配送路線規劃問題範例二的配送網路圖	88
圖 3.53 範例二的可行配送路線圖	88
圖 3.54 已被餐廳與平台同時接受的外送訂單(即刻送訂單)的外送員必須取餐時段	89
圖 3.55 已被餐廳與平台同時接受的外送訂單的外送員必須送達時段	90
圖 3.56 範例二路線圖上虛擬取餐與顧客節點的訂單餐點體積	90
圖 3.57 連接外送餐點對應的節點、虛擬取餐節點與顧客位置節點的節線	91
圖 3.58 整合生產排程問題與配送路線規劃問題之網路圖	92
圖 5.1 數學模型目標式與限制式的決策變數	112
圖 5.2 子問題 1、2 與複雜限制式	113
圖 5.3 分層最佳化求解架構圖	116
圖 5.4 本研究提出之 IterILS 演算法流程圖	117
圖 5.5 車輛 k 抵達節點 $i-1$ 的時間小於車輛 k 最早能抵達節點 i 的時間	123
圖 5.6 車輛 k' 抵達節點 $i-1$ 的時間大於或等於車輛 k 最早能抵達節點 i 的時間	124
圖 5.7 若插入節點 i 後其 s'_{ik} 較節點 i 服務時窗上界 a_i 早所需的調整	125
圖 5.8 車輛 k' 抵達節點 $i-1$ 的時間大於或等於車輛 k 最早能抵達節點 i 所需的調整	125
圖 5.9 因車輛 k 插入節點 i 造成節點 i 之後的節點 j 的抵達時間向後推遲之示意圖	126
圖 5.10 $wait'_{jk}$ 與 $shift'_{ik}$ 的更新方式示意圖	128
圖 5.11 拔出某些節點後剩餘節點之 $wait$ 與 m_shift 之計算方式	133
圖 5.12 本研究提出之 ALNS 演算法流程圖	139
圖 6.1 空腹熊貓於台北市的合作餐廳分布圖	167
圖 6.2 優步快送於台北市的外送員送餐熱區分布圖	167

1 緒論

1.1 研究背景與目的

直至 2016 年，全球約有 2.1 億的智慧型手機用戶 (Statista, 2016)。由於智慧型手機的普及化，消費者能便利的在網路上訂購商品，因此，近年來全球興起一種新的 O2O (online-to-offline) 商業模式。O2O 商業模式是指業者的線下實體商店可以與線上平台合作，消費者能夠透過手機 app 在平台上訂購線下實體商店的商品，線下業者能藉由與外送平台合作提升本身的營業額 (The Online Economy: Strategy and Entrepreneurship, 2016)。在眾多 O2O 商業模式中，餐點外送平台 (meal delivery platform) 便是一個近年來蓬勃發展的例子，顧客可以利用手機 app 訂購平台上的餐點，支付外送費用與餐點費用給平台，由平台自建的車隊、以群眾外包 (crowd-sourcing) 的方式或餐廳本身的外送員將餐點送達顧客手上。表 1.1 表示 2013 年第一季至 2017 年第一季，全球創投公司對新創 O2O 餐點外送平台的投資金額。

表 1.1 全球餐點外送平台獲得的創投投資金額



(資料來源：CBinsights, 2017)

在台灣，民眾大約將三分之一的年收入花費在外食上面 (王柔雅，2016)，這個市場特性吸引了國際或本土的餐點外送平台開始經營台灣市場。在台灣依合作餐廳的數量排名，前四名的餐點外送平台分別為：空腹熊貓 (foodpanda)、優步快送 (UberEats)、

誠實蜜蜂 (Honestbee) 與饕客送 (Taokesong)。本研究整理上述四家餐點外送平台的開始營運時間、服務範圍、合作餐廳數量、平均每日接單量、外送員來源與配送工具。

表 1.2 表示這四家餐點外送平台目前在台灣的營運範圍，合作餐廳數與平均每日接單數量。

表 1.2 台灣四大餐點外送平台簡介

	空腹熊貓	誠實蜜蜂	優步快送	饕客送
進駐台灣時間	2012 年	2017 年	2016 年	2014 年
服務範圍	台北市、新北市、 台中市、高雄市	台北市、新北市、 新竹縣、新竹市、 台中市、高雄市	台北市、新北市、 台中市、高雄市	台北市、新北市
合作餐廳數	約 1300 家	約 1200 家	約 1000 家	約 90 家
全台每日單量	約 3000 筆*	約 3000 筆*	約 2500 筆**	約 60 筆
外送員來源	自有車隊	自有車隊	群眾外包 (crowd-sourcing)	自有車隊
配送工具	機車	機車	步行、單車、機車	機車

(*以平台外送員人數約 1000 人/每人每日約 3 張單估算)

(**表示營運首 4 日的每日平均接單量)

(資料來源：商業週刊、自由時報、中央社與各平台網站)

在服務範圍方面，即使空腹熊貓、誠實蜜蜂與優步快送號稱有超過 1000 家的合作餐廳。但其主要服務範圍仍集中在台北市與新北市，例如：空腹熊貓、誠實蜜蜂與優步快送在台北市的合作餐廳數量分別約為 600 家、1000 家與 500 家，大約占了一半左右的合作餐廳數量。在外送員來源方面，優步快送 (Ubereats) 不同於另外三家建置自有車隊的外送平台，採用的是群眾外包 (crowd-sourcing) 的方式。外送員與平台並非正式的勞雇關係，外送平台不用負擔聘僱成本但也無法規定外送員的上班時間，外送員可以自由選擇空閒的時間登入平台的 app，上線接單。關於詳細的群眾外包 (crowd-sourcing) 說明可以參照文獻 Dai et al. (2017)。在配送工具方面，雖然優步快送 (Ubereats) 的外送員可以選擇機車、腳踏車或是步行進行外送服務，但是在台北都會區內，考量到塞車情形、載具的速度與可配送距離，大多數的外送訂單仍是由機車外送的方式來配送。

顧客可以透過四大餐點外送平台的手機 app 訂購餐點，並要求餐點外送平台即刻送達或是指定送達的時間。本研究依顧客要求即刻送達的訂單稱之為即刻送訂單，顧客指定送達時間的訂單稱之為預訂單。

下面將會介紹顧客訂購這兩類訂單的方式，以及在顧客訂餐後，各個利害關係人 (stakeholders) 之間的訊息傳遞。

➤ 即刻送訂單

■ 空腹熊貓

◆ 利害關係人

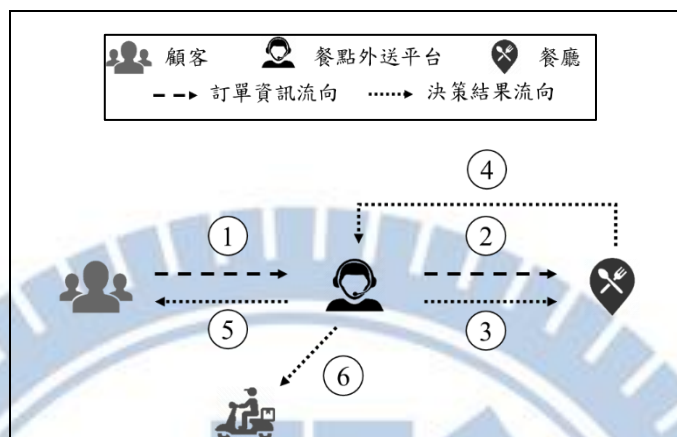


圖 1.1 利害關係人間的即刻送訂單資訊流向圖(空腹熊貓)



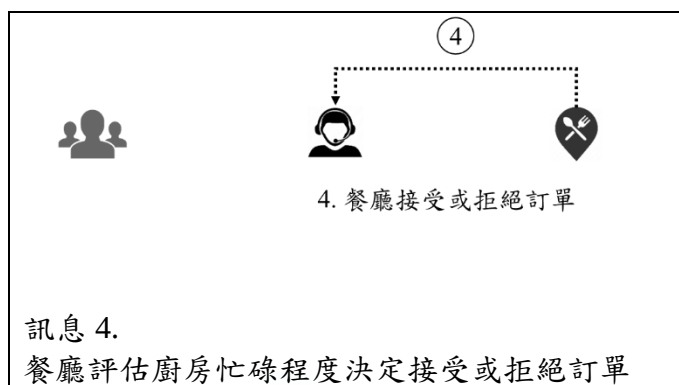


圖 1.2 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(空腹熊貓)

當顧客向空腹熊貓點餐後，平台便會將訂單資訊告知餐廳，同時告知外送員到餐廳取餐的時間。一般而言，空腹熊貓主導外送員取餐時間的決策。平台在決策取餐時間時，不會將餐廳內部廚房情況納入考慮。而是假設訂單可以馬上開始製作，取訂單內製作時間最長的餐點作為該訂單的預估完成時間。並依據各個外送員的位置與外送員的車廂剩餘容量指派離該餐廳最近的外送員取餐。取外送員預估抵達時間與訂單餐點的預估完成時間的最大值作為取餐時間。

若餐廳評估無法在平台決策的取餐時間之前完成訂單裡的餐點，餐廳只能拒絕訂單，不能要求平台更改取餐時間。



圖 1.3 若餐廳接受即刻送訂單-平台告知顧客與外送員之資訊(空腹熊貓)

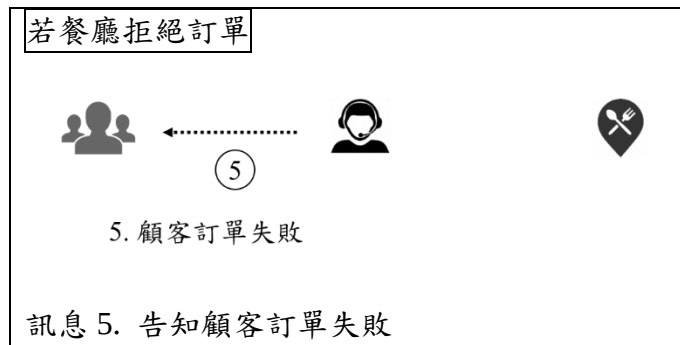


圖 1.4 若餐廳拒絕即刻送訂單-平台告知顧客之資訊(空腹熊貓)

若餐廳接受訂單，平台將會告知顧客訂單成功，並告知預估送達時間。同時，調度外送員進行配送。反之，若餐廳拒絕訂單，則告知顧客訂購失敗。一般而言，空腹熊貓在顧客訂餐五分鐘內，會以簡訊通知訂購是否成功。

◆ 平台承諾顧客的送達時間

(e.g. 某顧客訂餐時間為11:33)



圖 1.5 平台承諾顧客的即刻送訂單送達時窗(空腹熊貓)

在空腹熊貓的營業時間內 (10:30~21:30) 顧客皆可以點餐，平台會承諾顧客在顧客訂餐後的 60 分鐘內送達餐點。

■ 誠實蜜蜂

◆ 利害關係人

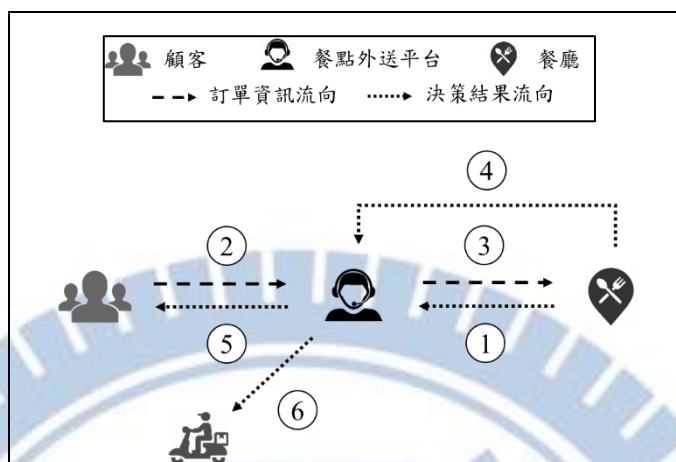
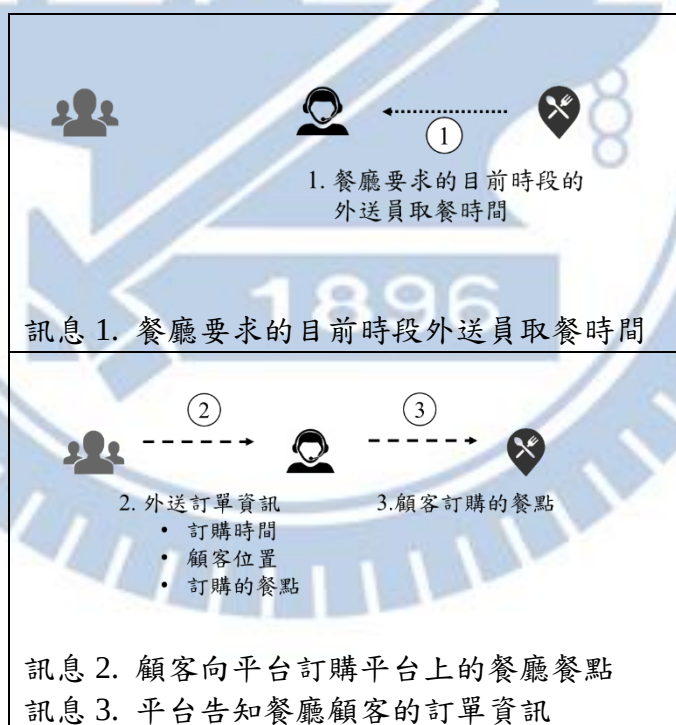


圖 1.6 利害關係人間的即刻送訂單資訊流向圖(誠實蜜蜂)



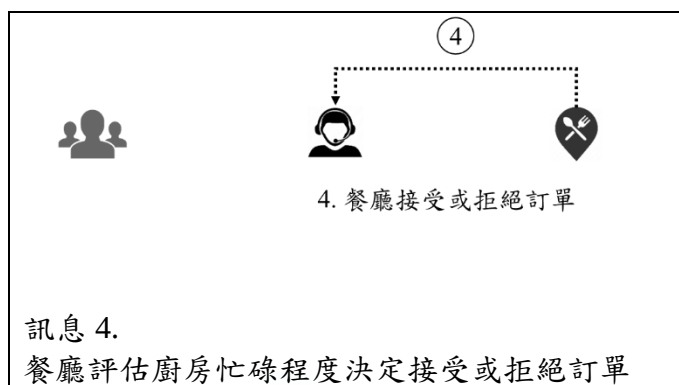


圖 1.7 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(誠實蜜蜂)

不同於空腹熊貓，誠實蜜蜂提供給餐廳決策取餐時間的權利。平台可以在誠實蜜蜂提供的平板電腦上輸入餐廳想要的外送員取餐時間(e.g. 顧客點餐後 20 分鐘)。但餐廳必須在接到顧客訂單前，決策目前時段所接到的顧客訂單的取餐時間。餐廳可以依照目前廚房情形，決策取餐時間。

但若餐廳在接到訂單之後評估無法在餐廳之前設立的取餐時間之前完成訂單裡的餐點，餐廳只能拒絕訂單，不能要求平台更改取餐時間。



圖 1.8 若餐廳接受即刻送訂單-平台告知顧客與外送員之資訊(誠實蜜蜂)

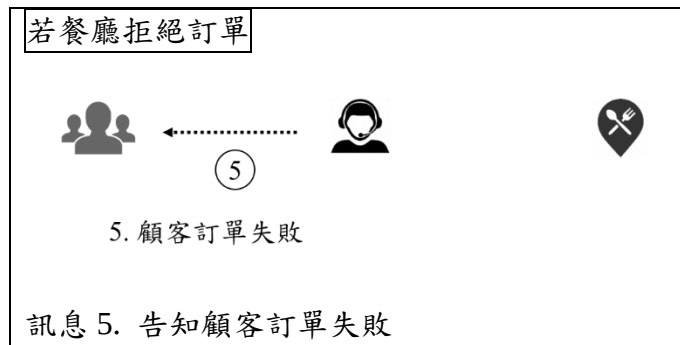


圖 1.9 若餐廳拒絕即刻送訂單-平台告知顧客之資訊(誠實蜜蜂)

若餐廳接受訂單，平台將會告知顧客訂單成功，並告知預估送達時間。同時，調度外送員進行配送。反之，若餐廳拒絕訂單，則告知顧客訂購失敗。一般而言，雖然誠實蜜蜂並無承諾在一定的時間內通知顧客訂購結果，但顧客可以隨時在誠實蜜蜂的手機 app 上查詢訂單處理狀況。

◆ 平台承諾顧客的送達時間

(e.g. 某顧客訂餐時間為11:33)

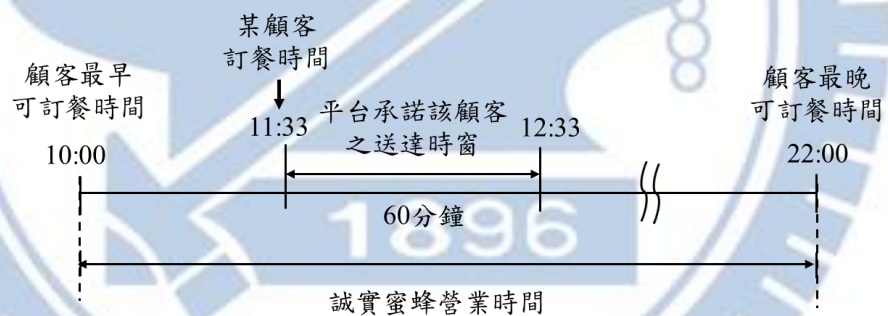


圖 1.10 平台承諾顧客的即刻送訂單送達時窗(誠實蜜蜂)

在誠實蜜蜂的營業時間內 (10:00~22:00) 顧客皆可以點餐，平台會承諾顧客在顧客訂餐後的 60 分鐘內送達餐點。

■ 優步快送

◆ 利害關係人

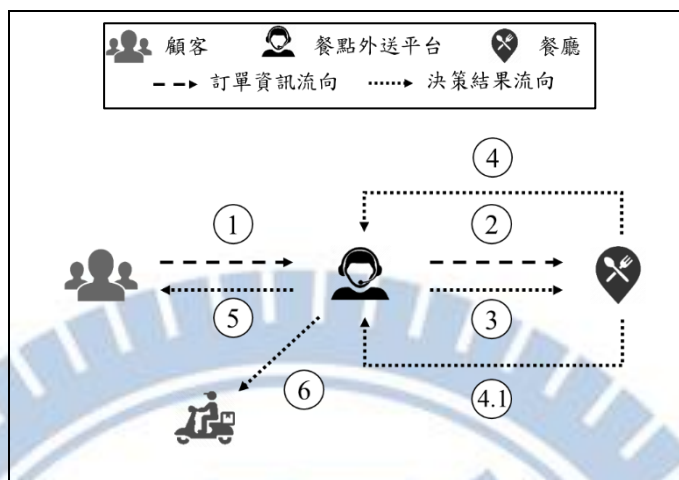


圖 1.11 利害關係人間的即刻送訂單資訊流向圖(優步快送)



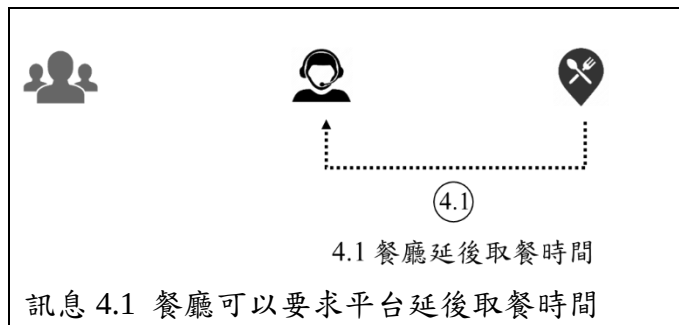
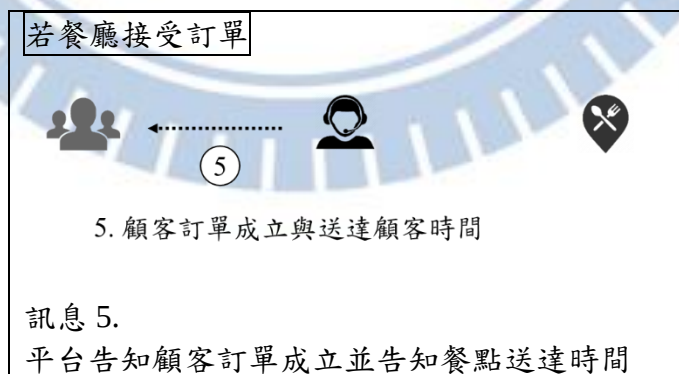


圖 1.12 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(優步快送)

當顧客向優步快送點餐後，平台便會將訂單資訊告知餐廳，同時告知外送員到餐廳取餐的時間。不同於空腹熊貓，優步快送餐廳在決策取餐時間時，提供餐廳修改取餐時間的機會。雖然同樣平台在告知餐廳取餐時間時，不會將餐廳廚房情況納入考慮。並且假設訂單可以馬上開始製作，取訂單內製作時間最長的餐點作為該訂單的預估完成時間。並依據各個外送員的位置與外送員的車廂剩餘容量指派離該餐廳最近的外送員去取餐。取外送員預估抵達時間與訂單餐點的預估完成時間的最大值作為取餐時間。

但是當餐廳評估無法在平台決策的取餐時間前完成這張訂單裡的餐點，餐廳可要求延後取餐時間。一旦餐廳要求平台延後取餐時間，平台只能接受更改後的取餐時間。



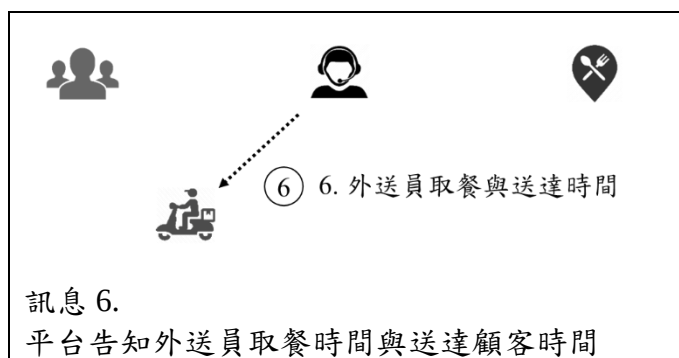


圖 1.13 若餐廳接受即刻送訂單-平台告知顧客與外送員之資訊(優步快送)



圖 1.14 若餐廳接受即刻送訂單-平台告知顧客之資訊(優步快送)

若餐廳接受訂單，平台將會告知顧客訂單成功，並告知預估送達時間。同時，調度外送員進行配送。反之，若餐廳拒絕訂單，則告知顧客訂購失敗。一般而言，雖然優步快送並無承諾在一定的時間內通知顧客訂購結果，但顧客可以隨時在優步快送的手機 app 上查詢訂單處理狀況。

◆ 平台承諾顧客的送達時間

(e.g. 某顧客訂餐時間為11:33)

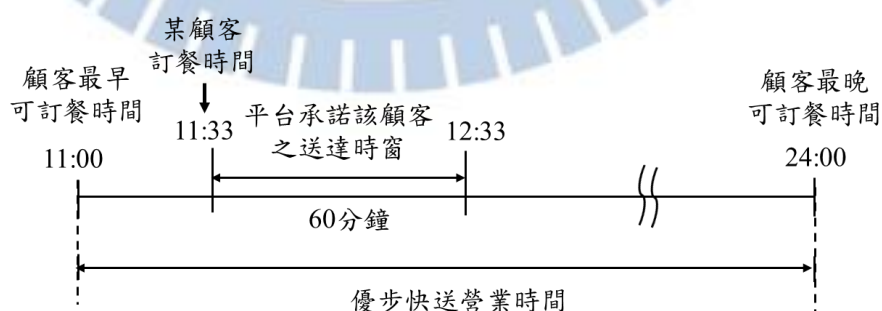


圖 1.15 平台承諾顧客的即刻送訂單送達時窗(優步快送)

在優步快送的營業時間內(11:00~24:00)顧客皆可以點餐，平台會承諾顧客在顧客訂餐後的 60 分鐘內送達餐點。

■ 饕客送

◆ 利害關係人

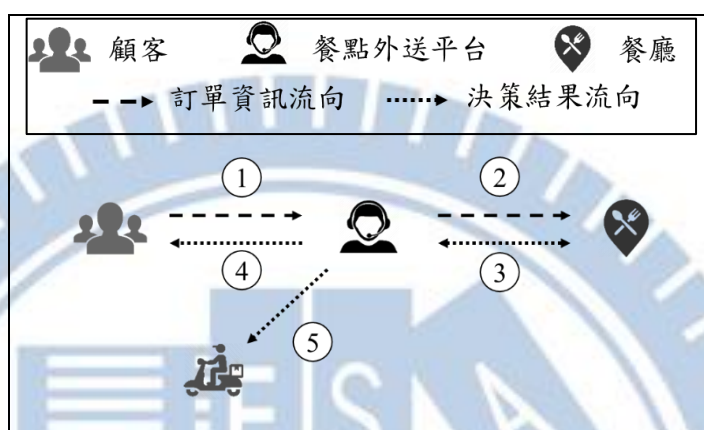


圖 1.16 利害關係人間的即刻送訂單資訊流向圖(饕客送)

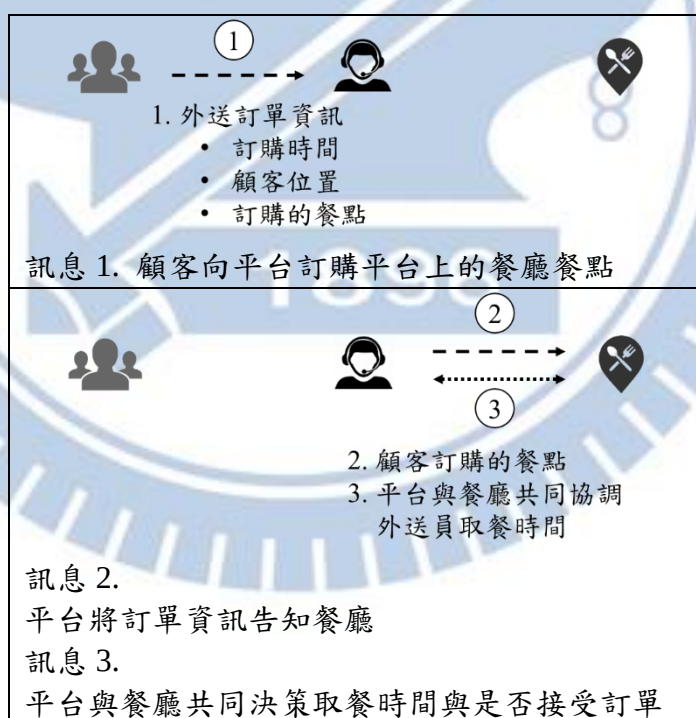


圖 1.17 即刻送訂單資訊和餐廳與平台決策結果傳遞方向(饕客送)

當顧客向饕客送點餐後，平台便會將訂單資訊告知餐廳。不同於上述的平台，饕客送與餐廳充分交換內部資訊，餐廳可以依照餐廳內廚房

情形告知平台預估訂單餐點完成時間，平台也可以依照外送員的位置與剩餘容量預估外送員抵達時間，而要求餐廳更改訂單的完成時間。雙方充分討論後再決定外送員的取餐時間。

若無法協調出雙方皆同意的取餐時間，餐廳會選擇拒絕訂單。在協調過程中，餐廳仍是佔有較強勢的地位，若餐廳願意接單，平台仍是會盡量配合餐廳的訂單完成時間，並不會拒絕訂單。雖然這個方式會花較多的時間，但是卻較能決策出雙方皆滿意的取餐時間。



圖 1.18 若餐廳接受即刻訂單-平台告知顧客與外送員之資訊(饕客送)

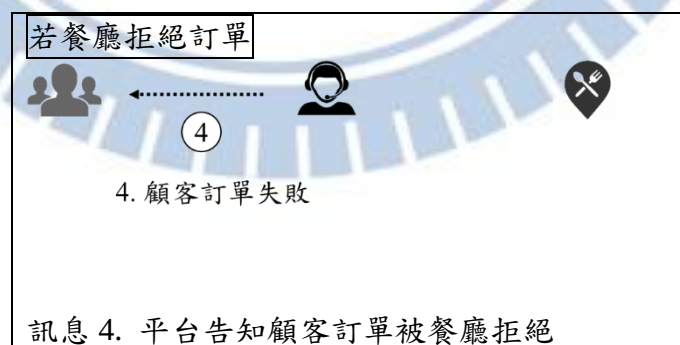


圖 1.19 若餐廳拒絕即刻訂單-平台告知顧客之資訊(饕客送)

若餐廳接受訂單，平台將會告知顧客訂單成功，並告知預估送達時間。同時，調度外送員進行配送。反之，若餐廳拒絕訂單，則告知顧客訂購失敗。一般而言，饗客送並無承諾顧客在一定的時間內通知訂購是否成功，一旦平台與顧客協調完畢，便會馬上以電話或簡訊通知顧客訂購結果。

◆ 平台承諾顧客的送達時間

(c.g. 某顧客訂餐時間為11:33)

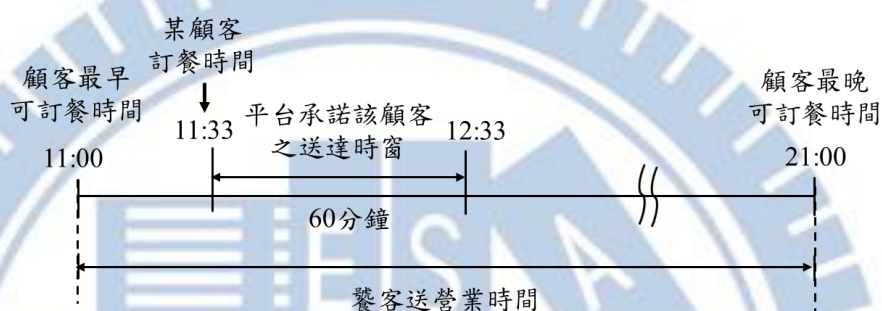


圖 1.20 平台承諾顧客的即刻送訂單送達時窗(饗客送)

在饗客送的營業時間內 (11:00~21:00) 顧客皆可以點餐，平台會承諾顧客在顧客訂餐後的 60 分鐘內送達餐點。

➤ 預訂單

■ 空腹熊貓

◆ 利害關係人

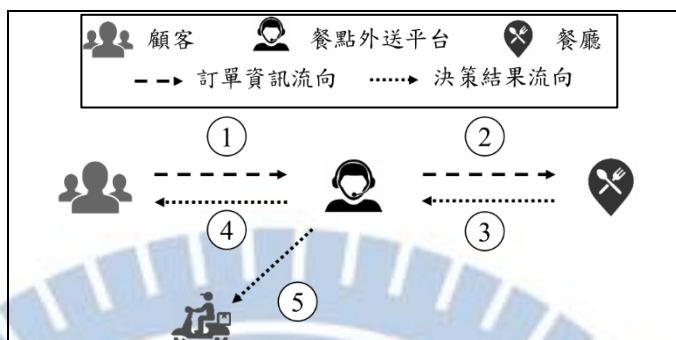


圖 1.21 利害關係人間的預訂單資訊流向圖(空腹熊貓)

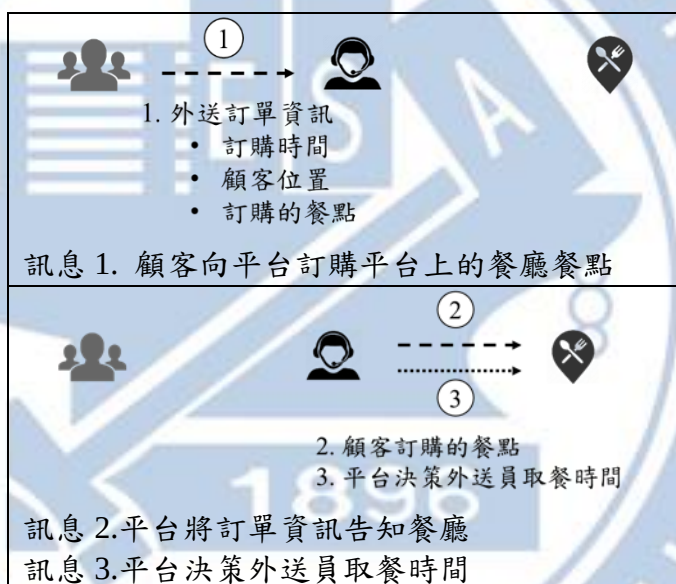


圖 1.22 預訂單訊息和餐廳與平台決策結果傳遞方向(空腹熊貓)

當顧客向空腹熊貓點餐後，平台便會告知餐廳訂單資訊與平台所決策的取餐時間。決策的方式是依照顧客選擇的送達時間點，減去顧客與餐廳間的最短旅行時間，作為外送員的取餐時間並指派任一外送員執行配送任務。決策完畢後再告知餐廳外送員的取餐時間。

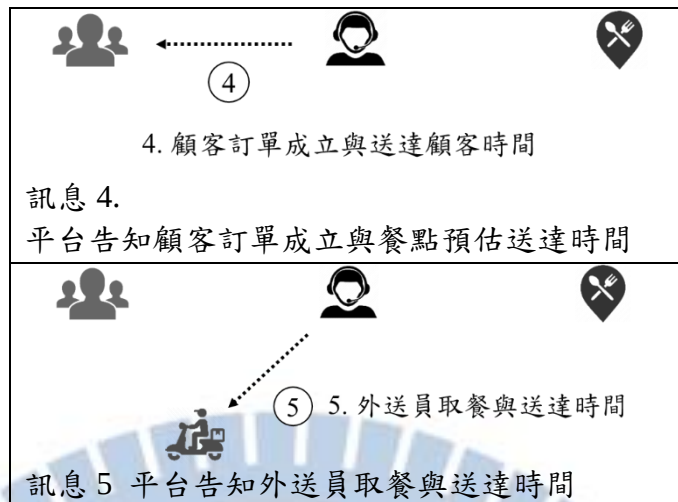


圖 1.23 平台告知顧客與外送員之資訊(空腹熊貓)

由於預訂單為顧客提前預訂的訂單，一般而言餐廳並不會拒絕該訂單。

◆ 顧客可選擇之送達時間

將平台營業時間 (10:30~21:30)，每 30 分鐘設定一個可供選擇的送達時間點。

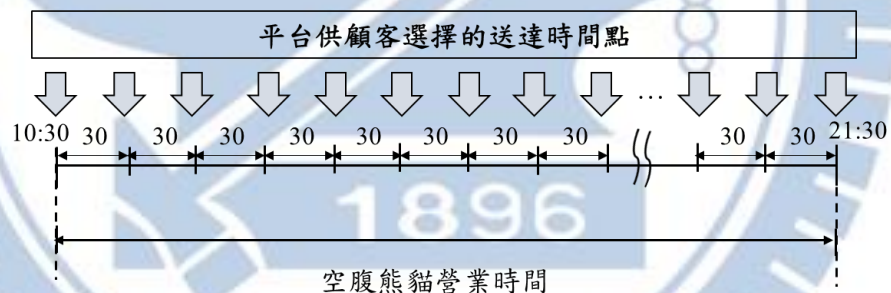


圖 1.24 空腹熊貓-平台提供顧客選擇的預訂單送達時間點

但是為了避免顧客選擇距離訂餐時間太近的送達時間點，導致平台在決策取餐時間與送達時間上的不便。空腹熊貓限制顧客僅能選擇訂餐時間 2 小時之後的送達時間點。

(e.g. 某顧客訂餐時間為11:33)

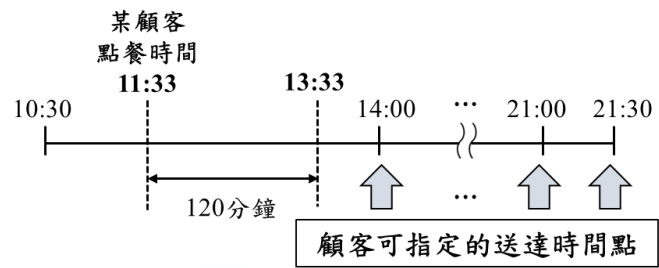


圖 1.25 空腹熊貓-顧客可指定的預訂單送達時間點

■ 誠實蜜蜂

◆ 利害關係人

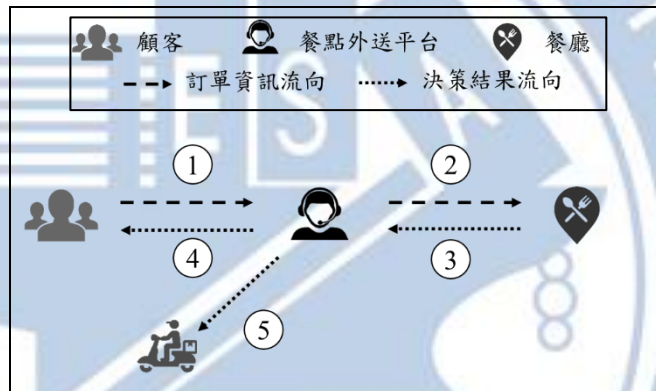


圖 1.26 利害關係人間的預訂單資訊流向圖(誠實蜜蜂)

顧客向誠實蜜蜂訂購預訂單之流程皆與空腹熊貓相同。

◆ 顧客可選擇之送達時間

將平台營業時間 (10:00~22:00)，每 15 分鐘設一個可供選擇的送達時窗，時窗的長度為 15 分鐘。



圖 1.27 誠實蜜蜂-平台供顧客選擇的預訂單送達時間窗

為了避免顧客選擇距離訂餐時間太近的送達時窗，導致平台在決策取餐時間與送達時間上的不便。優步快送限制顧客僅能選擇送達時窗下界(lower bound)位於訂餐時間 30 分鐘之後的送達時窗。

(e.g. 某顧客訂餐時間為11:33)



圖 1.28 誠實蜜蜂承諾預訂單的送達時間

■ 優步快送

◆ 利害關係人

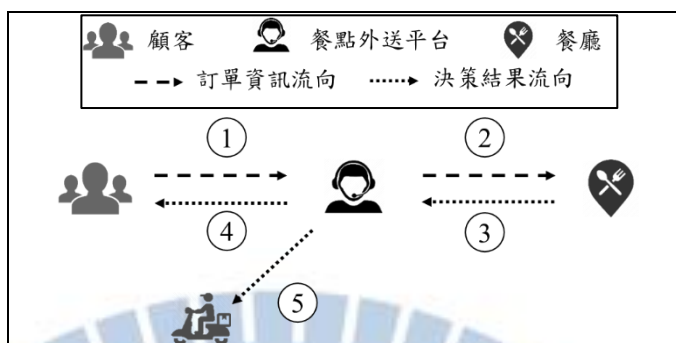


圖 1.29 利害關係人間的預訂單資訊流向圖 (優步快送)

顧客向優步快送訂購預訂單之流程皆與空腹熊貓相同。

◆ 顧客可選擇之送達時間

將平台營業時間 (11:00~00:00)，每 15 分鐘設一個可供選擇的送達時窗，時窗的長度為 30 分鐘。

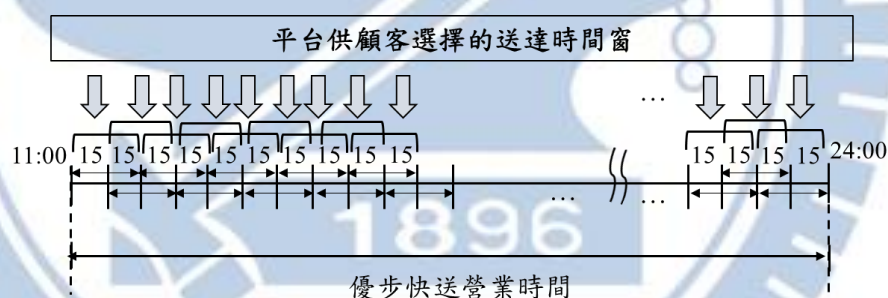


圖 1.30 優步快送-平台供顧客選擇的預訂單送達時窗

同樣為了避免顧客選擇距離訂餐時間太近的送達時窗，導致平台在決策取餐時間與送達時間上的不便。優步快送限制顧客僅能選擇送達時窗下界(lower bound)位於訂餐時間 1 小時 30 分鐘之後的送達時窗。

(e.g. 某顧客訂餐時間為11:33)

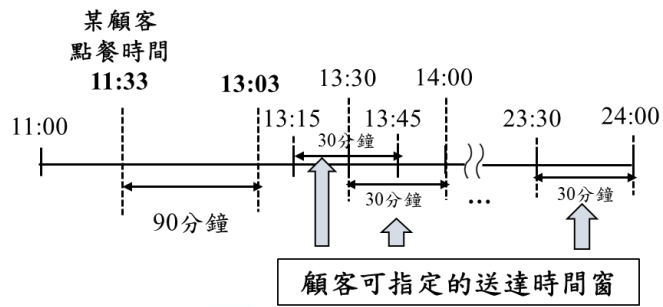


圖 1.31 優步快送-顧客可指定的預訂單送達時窗

■ 饕客送

◆ 利害關係人

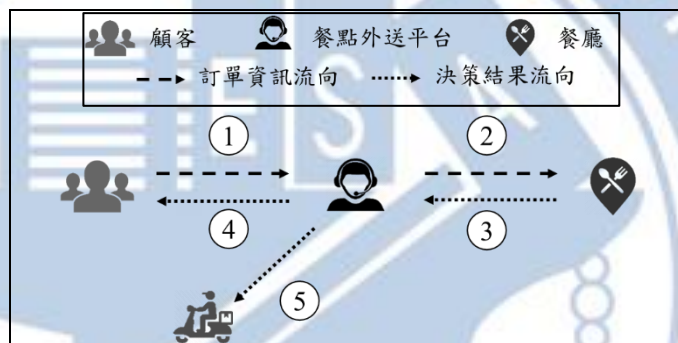


圖 1.32 利害關係人間的預訂單資訊流向圖 (饕客送)

顧客向饕客送訂購預訂單之流程皆與空腹熊貓相同。

◆ 顧客可選擇之送達時間

將平台營業時間 (11:00~24:00)，每 30 分鐘設定一個可供選擇的送達時間點。

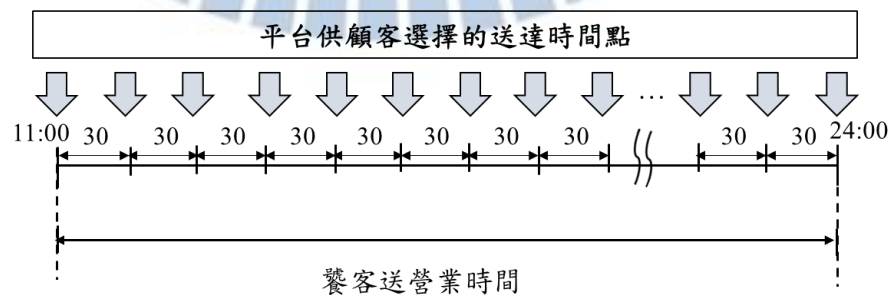


圖 1.33 饕客送-平台供顧客選擇的預訂單送達時間點

同樣為了避免顧客選擇距離訂餐時間太近的送達時間點，導致平台在決策取餐時間與送達時間上的不便。饕客送限制顧客僅能選擇訂餐時間 1 小時之後的送達時間點。

(e.g. 某顧客訂餐時間為11:33)

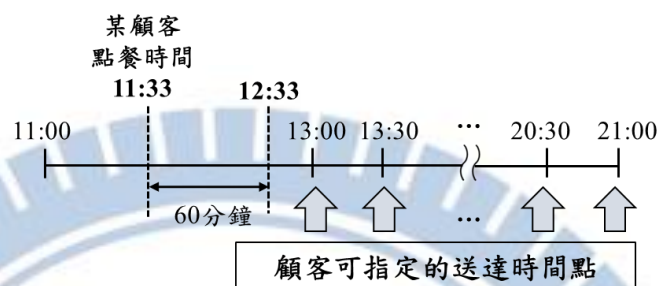


圖 1.34 饕客送-顧客可指定的預訂單送達時窗

綜合上述四個平台的資料，經由本研究整理，即刻送訂單與預訂單，平台與餐廳所需做的決策如表 1.3：

表 1.3 平台與餐廳針對即刻送與預訂單分別所需做的決策

	即刻送訂單	預訂單
餐廳	◆ 是否接受訂單 ◆ 若接受，訂單餐點的完成時間	-
平台	◆ 外送員到餐廳的取餐時間 ◆ 外送員將餐點送達顧客的時間	◆ 外送員到餐廳的取餐時間 ◆ 外送員將餐點送達顧客的時間

由上述介紹可知，平台主要面對三個利害關係人，分別是：顧客、外送員與餐廳。每一個利害關係人皆會影響平台在此商業模式的獲利，亦即若是餐點無法在平台承諾顧客的送達時間(窗)送達顧客手中，顧客有權拒絕訂單。另一方面，若是平台提供的取餐時間使餐廳無法在該時間前完成訂單內的餐點，餐廳有權拒絕訂單。而要如何適當的調度外送員，減少配送的成本，也是影響平台利潤的因素。

這類新興的 O2O 餐點外送平台所提供的服務與傳統的餐點外送服務（如：便當店或速食餐廳外送）不同。傳統的便當店或速食餐廳由於餐點種類較少、製作步驟簡單、餐點較不精緻且單價低，通常可以先製作一些餐點，放在保溫裝置中保存再販賣給顧客。對於傳統的餐點外送服務而言，僅需考慮如何在最少成本的路線規劃下，將餐點在承諾顧客的送達時間送達顧客手中。

而目前外送平台所合作的餐廳大多為中高價位的餐廳。這些餐廳的餐點種類多、製作步驟複雜、餐點精緻且單價高，無法使用與傳統餐點外送服務相同的方式，顧客對於送達時的餐點品質要求也相對高。餐廳在製作訂單內的餐點時會要求其完成時間與外送員取餐時間空隙不要過大，類似於工廠中的即時化生產方式 (Just in time, JIT)。

此外，對於中高價位的餐廳來說，內用顧客相當注重餐廳的服務品質，服務內用顧客比服務平台的外送顧客更為重要。一旦內用顧客點餐，餐廳便無法拒絕服務，並且不能讓內用顧客等候餐點的時間過長。

綜合上述，我們可以發現，為了減少餐廳拒絕外送訂單的機會，達到提升 O2O 餐點外送平台的利潤的效果，平台在決策取餐時間時必須考慮複雜的餐廳內部訂單排程，包含餐廳的內用訂單與已經接受但尚未完成的外送訂單。同時減少調度外送員與路線規劃的成本，並在平台承諾的送達時間點或送達時間段，將品質良好的餐點送達顧客手中。

於是，本研究希望透過提出新的平台與餐廳決策取餐時間的流程，在餐廳與平台合作的前提之下，站在平台的角色，由平台整合規劃，並向餐廳建議哪些依照目前餐廳廚房忙碌情況應該接受或拒絕那些外送訂單，同時規劃訂單的取餐時間和送達顧客時間。此外，本研究也將考慮由歷史資料所預估的未來可能發生的內用與外送訂單，藉此達到平台利潤最大化的目的。同時，平台告知顧客訂購結果的時間也不能距離顧客點餐時間太久，亦即平台計算規劃結果的時間，有一定的上限。

由平台整合規劃的方式，也獲得諸多與 O2O 餐點外送平台合作的餐廳認同。在本研究訪談多家餐廳時發現，在外送訂單量較多的情況下，餐廳是可以接受由平台協助餐廳決策應該接受或拒絕哪些外送訂單，來減少餐廳決策的時間。

1.2 研究流程

如 1.1 節所述，本研究探討的問題為 O2O 整合餐點外送之生產與配送問題，同時協助餐廳決策外送訂單裡餐點的生產排程與決策平台外送員的取送餐路徑。並將此問題建構成最佳化數學規劃問題求解，由於此複雜的最佳化問題為 NP-hard 問題，當考慮的資料量上升時，求解時間也會依指數上升。但是在實務上，本問題需要快速的求解時間。於是，本研究將整合生產與配送之最佳化模型分為兩個子問題，使用階層式最佳化 (Hierarchical Optimization) 的概念，利用啟發式解法分別求解。首先利用 IterILS (Iterated Local Search) 求解生產子問題再將其生產子問題的結果代入配送問題內，利用 ALNS (Alternative Large Neighborhood Search) 求解，希望在可接受時間上限內求得一個品質優良的可行解。接著對利用實務上的資料所產出的結果分析，並建議後續研究方向。

圖 1.35 為本研究之流程圖，依序為：

1. 文獻蒐集與回顧：針對與本問題最直接相關的整合易腐性商品的生產與配送問題 (Integrated production and distribution problem with perishable products) 進行文獻回顧，了解目前學術界對這類主題的相關研究。
2. 問題環境描述：定義整合餐廳生產排程與平台配送問題，並提出符合現實的假設。
3. 建立數學模型：將本問題建立成整數規劃最佳化模型。
4. 子問題分割：定義本問題的複雜限制式 (complicated constraints)，藉由改寫複雜限制式，將本問題分為生產子問題與配送子問題，並描述利用階層式演算法求解本問題之方法。
5. 設計啟發式解法：選擇適合本研究的生產子問題與配送子問題之演算法，並進行啟發式演算法的績效測試。
6. 實務案例分析：利用台北市內某餐點外送平台所提供的部分資料，將其擴充為適合本研究之測試例子，進行測試與分析。
7. 結論與建議：提出本研究之發現與結論，並建議後續研究方向。

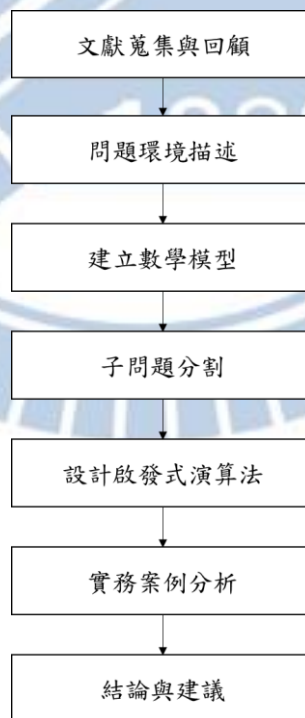


圖 1.35 研究流程圖

1.3 章節簡述

本研究接著分為以下幾個章節：第二章為文獻蒐集與回顧，第三章為問題描述，第四章為數學模型，第五章為啟發式演算法，包含子問題分割與設計啟發式演算法架構。第六章為實務例題分析，第七章為結論與建議。

2 文獻回顧

近年來，探討整合生產與配送的問題 (production routing problem) 的文獻日漸增多，該問題整合了生產批次規劃問題 (lot-sizing problem) 和車輛路線規劃問題 (vehicle routing problem)，通常目標為整個系統的成本最小化 (Adulyasak et al., 2015)。Adulyasak et al. (2015) 為此類問題的數學模型和演算法做了完善的回顧，並且為此類問題整理出一個一般化的數學模型。Gao et al. (2015) 則是在生產端考慮批量生產問題 (batch production problem)，提出不同以往的整合生產與配送問題，目標為最小化產品完成時間與配送開始時間的間隔，而非一般的最小化系統總成本。然而，上述的研究並無考慮本研究的重要特性，即產品的效期 (lifespans)。接著，本研究將會回顧與本研究直接相關的文獻，也就是考慮整合易腐性產品的生產與配送問題，該問題中，產品抵達顧客的時間必須滿足顧客的服務時窗與產品的效期。

Geismar et al. (2007) 考慮工業用黏著劑的整合生產與配送問題，一般而言，這是第一篇研究考慮配送時間與車輛路線規劃的整合生產與配送問題。在該篇前的研究往往為了簡化問題，忽略外送員旅行時間或是僅考慮點對點 (point-to-point) 的配送問題。此外，這也是第一篇將產品效期 (lifespans) 納入整合生產與配送問題的研究。Geismar et al. (2007) 考慮的問題環境為給定顧客點的一座工廠的生產排程問題與一台車輛的路徑規劃問題，但是一台車輛可以進行多趟次 (multi-trip) 的配送，即當車輛配送完車上的貨物後，可以回到工廠取貨，再進行下一趟次的配送。在生產端，本研究將顧客訂單指派至不同的生產週期中，並考慮固定之生產率。Naso et al. (2007) 考慮的是整合預拌混凝土的即時 (just-in-time) 生產問題，該研究考慮的問題環境為多座工廠，而每個工廠配有多個貨車碼頭 (dock)。因為每個貨車碼頭有固定的裝載率，與生產排程問題中的生產率相似。所以決策每個貨車抵達貨車碼頭的時間可視為於多個平行機台 (parallel machines) 的生產排程問題。此外，該問題考慮多車輛的車輛路線規劃問題，同時該研究允許車輛進行多趟次的配送。但為了簡化問題，該研究僅考慮每台

車一次只能配送一張訂單的環境。目標為最小化整個供應鏈的系統成本。Armstrong et al. (2008) 則研究一個特別的整合生產與配送問題，在該研究環境中，顧客的服務順序是固定的，如：先到先服務 (first come first serve)，問題環境為考慮一個工廠生產與一台車輛的配送問題，並考慮顧客的服務時窗、產品的效期、固定的工廠的產能與車輛的容量，目標為最大化顧客被滿足的需求。在該研究中提出一個有效的啟發式演算法，能夠快速的找到最佳解範圍的下界。Chen et al. (2009) 考慮一個工廠其中含有一條生產線的多易腐性產品的問題環境，為了簡化問題，僅考慮一台車輛的路線規劃問題。該研究考慮隨機性的顧客需求，顧客需求服從給定的機率分布，所以整個研究的目標為最大化整個系統的期望利潤。此外，產品的品質會隨著時間而腐敗，而最後的收益會受到顧客收到產品時的產品品質的影響。Farahani et al. (2012) 為了降低整合生產與配送問題的複雜度提出一個階層式求解的數學模型 (hierarchical modeling approach)。該研究考慮外燴產業的環境，在生產端考慮與生產順序相關的 (sequence-dependent) 設置時間與成本，並且考慮一個工廠中配有多個平行機台的環境，在配送端則考慮硬式時窗 (strict time window) 的配送問題。同時，該研究允許批量生產 (batching production)，特性相同的產品一起被生產可以減少機台的設置時間與成本，如：生產溫度相近，目標式為最小化設置成本、送達顧客的餐點品質與生產成本的權重和。Chang et al. (2016) 提出一個更為複雜的整合生產與配送問題，考慮的生產環境為一家餐廳內的彈性零工式排程問題 (flexible job shop scheduling problem)，即為完成每個產品必須經過餐廳內的不同步驟的工作站，每個工作站中有不同廚師數，每個產品的每個步驟所需的時間與順序也不全然相同。此外，該研究將此複雜的彈性零工問題轉為多輛車的旅行家問題 (m-TSP problem)。而在配送端則是考慮有順序的收送貨問題 (precedence-constrained pickup and delivery problem)，即每輛車一旦開始送貨，就無法再次收貨。

表 2.1 本研究相關文獻整理

作者	產品種類	生產者數	生產問題	車輛數	配送問題
Naso et al. (2007)	一種	一個	單一機台 排程問題	多台	有時窗限制的 多車輛路線規劃問題
Armstrong et al. (2008)	一種	一個	單一機台 排程問題	一台	有時窗限制的 單一車輛路線規劃問題
Geismar et al. (2008)	一種	一個	單一機台 排程問題	一台	單一車輛路線規劃問題
Chen et al. (2009)	一種	一個	單一機台 排程問題	多台	有時窗限制的 多車輛路線規劃問題
Farahani et al. (2012)	多種	多個	平行機台 排程問題	多台	有時窗限制的 多車輛路線規劃問題
Chang et al. (2016)	多種	一個	彈性零工 排程問題	多台	有時窗限制的 多車輛收送貨問題

由文獻回顧可知，本研究的主要貢獻如下：

- 本研究為第一篇研究 O2O 餐點外送平台的整合生產與配送問題。
- 本研究為第一篇擁有滾動時域架構 (rolling horizon) 的整合生產與配送問題。
- 本研究提出將彈性零工問題轉換成平行機台排程問題之新方法

3 問題描述

如同第一章所述，本研究提出新的平台與餐廳決策外送訂單是否接受與取餐時間的方式。即站在平台的角度，由平台整合規劃，考慮餐廳內部的訂單資訊，建議餐廳應該接受或拒絕那些外送訂單，並規劃外送員的配送路線。同時將由歷史資料所預估的未來可能發生的內用與外送訂單納入考慮，藉此達到平台利潤最大化的目的。

3.1 O2O 餐點外送模式之利害關係人

本研究針對所有外送訂單（包含即刻送訂單與預訂單），提出新的處理訂單的方式，其利害關係人間的訂單資訊與決策結果流向如下圖：

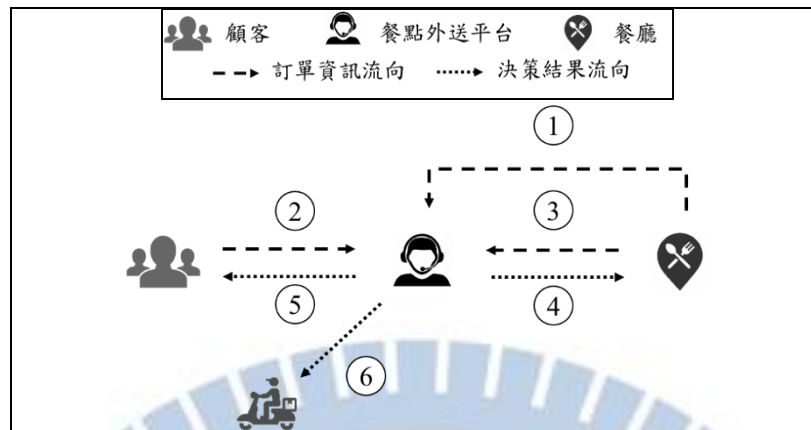


圖 3.1 利害關係人間的訂單資訊與流向圖 (本研究提出)



圖 3.2 餐廳內部資訊與訂單資訊流向圖 (本研究提出)

不同於目前四大平台的訊息傳遞方式，本研究提出的方法要求餐廳在每周開始時傳遞餐廳上周的內用訂單資訊（訊息 1），並且在平台準備決策時，傳遞餐廳內部資訊給平台（訊息 3）。本研究將未來可能出現的內用與外送訂單納入決策環境，有下列優點：

表 3.1 考慮未來可能出現的訂單之優點

	對生產端規劃之優點	對配送端規劃之優點
未來可能出現之內用訂單	避免目前決策時接受的外送訂單，在未來無法完成	—
未來可能出現之外送訂單	避免目前決策時接受的外送訂單，排擠未來利潤較高的外送訂單	避免目前決策時接受的外送訂單，排擠未來利潤較高的外送訂單

由於對於餐廳而言，內用訂單一定要接受，平台沒有建議餐廳拒絕的權利，並且不能讓內用訂單的顧客等候餐點過久。一般而言，內用顧客在點餐後 40 分鐘一定要拿到餐點。所以考慮未來可能出現的內用訂單，可以避免在目前決策時接受過多外送訂單，這些被接受的外送訂單在未來可能會因為內用訂單的排擠，而無法及時完成。另一方面，考慮未來可能出現的外送訂單，則是可以避免目前接受的外送訂單，排擠未來可能出現的利潤較高之外送訂單。

故我們要求餐廳傳遞歷史內用訂單資料給平台，平台便可以利用餐廳傳來的歷史內用訂單資料與平台本身有的歷史外送訂單資料來預測未來可能發生的內用與外送訂單。

此外，本研究也要求餐廳在平台決策時，傳遞其內部已接受但尚未完成的內用訂單與外送訂單裡的餐點資訊給平台，藉此了解餐廳內部廚房情形。

雖然外送訂單中包含即刻送訂單與預訂單，但是因為預訂單餐廳並不會拒絕，所以平台並不需評估餐廳與平台是否接受預訂單，預訂單恆為已被餐廳與平台同時接受的訂單。所以我們可以更進一步將外送訂單細分為即刻送訂單與預訂單。

綜合上述，在平台決策時，平台將擁有的訂單資訊可分為下列幾種狀態：

表 3.2 平台在決策時擁有的訂單資訊

已被餐廳與平台同時接受但尚未完成	已被餐廳與平台同時接受已完成但尚未送達	待平台評估餐廳與平台是否接受	預測未來可能出現
內用訂單 (圖 3.2 訊息 3)	-	-	內用訂單 (圖 3.2 訊息 1)
外送訂單 ◆ 即刻送訂單 ◆ 預訂單 (圖 3.2 訊息 3)	外送訂單 ◆ 即刻送訂單 ◆ 預訂單 (平台本身擁有之資訊)	外送訂單 ◆ 即刻送訂單 (圖 3.2 訊息 2)	外送訂單 ◆ 即刻送訂單 ◆ 預訂單 (平台本身擁有之資訊)

而每一張內用與外送訂單含有下列資訊供平台決策時使用：

表 3.3 內用與外送訂單詳細資訊

內用訂單	外送訂單-	
	即刻送訂單	預訂單
◆ 點餐時間 ◆ 餐點 ◆ 餐點製作時間 ◆ 餐廳位置	◆ 點餐時間 ◆ 餐點 ◆ 餐點製作時間 ◆ 餐廳位置 ◆ 顧客位置	◆ 點餐時間 ◆ 顧客選擇的送達時間 ◆ 餐點 ◆ 餐點製作時間 ◆ 餐廳位置 ◆ 顧客位置

平台利用這些資訊進行決策，來建議餐廳是否接受訂單，若接受則告知餐廳外送員取餐時間(訊息 4)、告知顧客訂單成立與餐點送達時間(訊息 5)與告知外送員到餐廳取餐時間和送達顧客時間(訊息 6)。



圖 3.3 平台決策結果傳遞方向-平台建議餐廳接受訂單 (本研究提出)

反之，若平台決策結果為建議餐廳拒絕訂單，則平台將告知餐廳此訊息 (訊息 4)，並告知顧客訂購失敗 (訊息 5)。

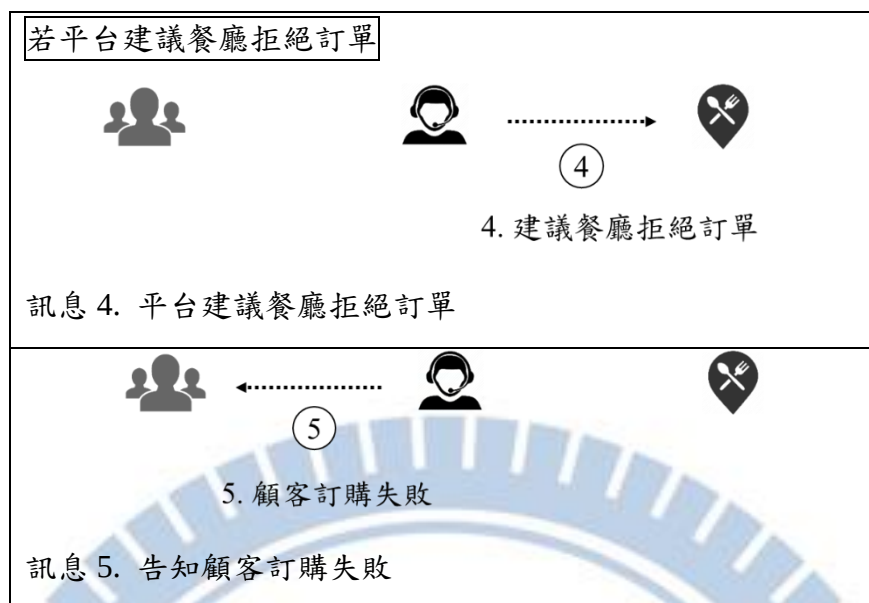


圖 3.4 平台決策結果傳遞方向-平台建議餐廳拒絕訂單 (本研究提出)

平台在決策時會藉由上述訂單資訊，利用數學模型同時評估餐廳與平台是否有能力接受目前接到的外送訂單，平台利用數學模型評估平台與餐廳的接受訂單結果可能有下列四種情境：

表 3.4 平台評估餐廳與平台接受或拒絕訂單的可能情境

	平台	餐廳
情境 1	接受	接受
情境 2	接受	拒絕
情境 3	拒絕	接受
情境 4	拒絕	拒絕

僅有當情境 1 發生時，平台才會建議餐廳接受該外送訂單，並且告知取餐時間。其餘的情境，平台皆會建議餐廳拒絕訂單。此外，當情境 3 發生時，也就是平台評估餐廳可以接受訂單，但是由於平台調度外送員上的問題，平台無法服務該訂單時，平台必須支付餐廳與訂單金額相等的補償金。因為在實務上，當餐廳可以接受訂單時，平台就一定要服務訂單，所以本研究利用補償金的方式來處理情境 3 的問題。

3.2 規劃長度與決策時段定義

接下來，本小節將會定義介紹平台每日必須待規劃的外送訂單，這些外送訂單的點餐時間分布區間為何，該時間區間稱之為規劃長度 (planning horizon)。而在每日平台的規劃長度內，平台將會進行多次決策，在每個決策中納入規劃的外送訂單，其訂單的點餐時間分布區間稱之為決策時段 (decision interval)。

■ 規劃長度 (planning horizon)

由於大部分平台合作的餐廳營業時間為上午 11 點到下午 2 點，在下午 2 點到 5 點會打烊休息，再從下午 5 點營業到晚上 8 點，且在餐廳休息時，顧客無法從平台訂購該餐廳的餐點，餐廳休息時間的示意圖如圖 3.5。



圖 3.5 餐廳營業時間

因此，為了簡化問題，我們可以將餐廳的營業時間，上午 11 點到下午 2 點與下午 5 點到晚上 8 點，視為兩個獨立的區間，每段區間的長度為 180 分鐘。而平台每日僅需規劃點餐時間落在這兩個規劃長度內的外送訂單。

■ 決策時段 (decision interval)

而在每日的規劃長度中，平台會進行多次針對外送訂單的決策。除了第一章提到本研究將過往已被餐廳與平台同時接受但尚未完成或送達的內用與外送訂單與未來可能出現的內用或外送訂單納入規劃協助決策。我們還發現若是能累積越多待決策外送訂單，再一同決策，能夠提升平台利潤的機會也越大。

於是本研究提出的方法為，平台每隔一段相同的時間區間便針對該時間區間顧客下訂的外送訂單進行決策，同時將平台已同時接受但尚未完成或送達的內用與外送訂單、未來可能出現的內用與外送訂單納入規劃協助決策。上述這些訂單所對應的點餐時間區間，即為本研究的決策時段。規劃長度、決策時間點與決策時段的關係如圖 3.6。

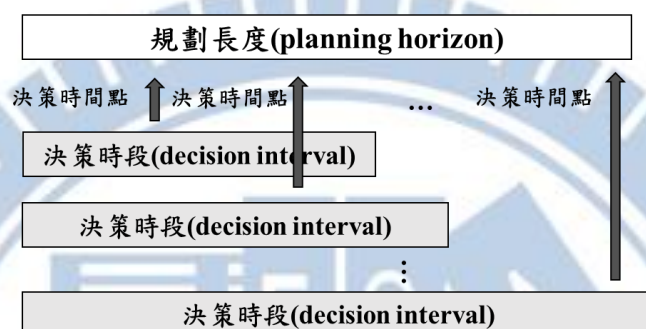


圖 3.6 規劃長度與決策時段之關係圖

更進一步的說，上述的訂單點餐時間分布的時間區間可以分為三個時段，即每個決策時段由以下三個時段組成：已規劃時段、目前規劃時段與預測時段。這三個時段與表 3.2 平台在決策時擁有的訂單資訊的訂單資訊對應關係如表 3.5 已規劃時段、目前規劃時段與預測時段所包含的訂單。如同前述，平台僅需評估目前接到的即刻送訂單餐廳是否該接受，以及該訂單的取餐與送達時間。內用訂單與已被餐廳和平台同時接受的即刻送訂單與預訂單，平台則是無法再建議餐廳拒絕該訂單。

表 3.5 已規劃時段、目前規劃時段與預測時段所包含的訂單

已規劃時段	目前規劃時段	預測時段
已被餐廳與平台同時接受但尚未完成的	已被餐廳與平台同時接受但尚未完成的	預測未來可能出現
內用訂單	內用訂單	內用訂單
外送訂單 ◆ 即刻送訂單 ◆ 預訂單	外送訂單 ◆ 預訂單	外送訂單 ◆ 即刻送訂單 ◆ 預訂單
已被餐廳與平台同時接受已完成但尚未送達的	待平台評估餐廳與平台是否接受的	
外送訂單 ◆ 即刻送訂單 ◆ 預訂單	外送訂單 ◆ 即刻送訂單	

由於已規劃時段與預測時段的時間長度是從目前規劃時段的時間長度推算而來，所以接下來會先介紹如何決定目前規劃時段的時間長度，再分別介紹如何決定已規劃時段與預測時段的時間長度。

◆ 目前規劃時段

目前空腹熊貓、誠實蜜蜂、優步快送與饕客送在接到每張外送訂單時，皆會進行該外送訂單的取餐時間與送達顧客時間之決策，如圖 3.7。

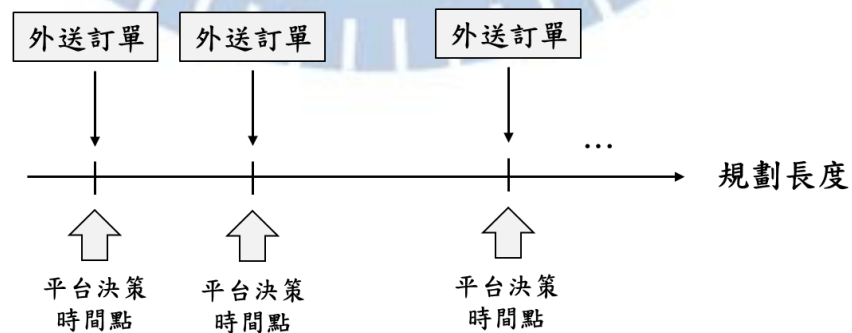


圖 3.7 目前外送平台的決策時間點

但若平台能夠累積越多訂單再利用最佳化數學模型進行決策，提升平台利潤的機會越高。所以本研究設立一個平台等候訂單時間區間，平台會將這個時間區間內的接到外送訂單一起做決策，藉此提高獲得更高利潤的機會。

而為了顧及平台的服務品質，平台不能讓顧客等候訂購是否成功的資訊過久，所以本研究將顧客等候結果的時間上限設為唯一有設置顧客等候結果上限的空腹熊貓，亦即顧客等候訂購結果的時間不能超過 5 分鐘。此外，本研究評估模型求解時間最快為 1 分鐘，演算法的設計也以此求解時間為目標。故可以得出規劃時段的長度為 4 分鐘，如此一來，在規劃時段內下訂單的顧客，其等候訂購結果的時間皆不會超過 5 分鐘。規劃時段時間長度與模型計算時間長度的示意圖如下圖 3.8。

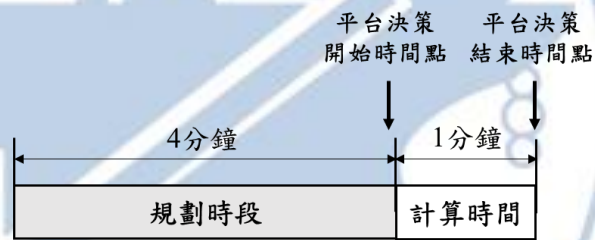


圖 3.8 規劃時段時間長度與模型計算時間長度

根據上述，我們便能知道平台在每個規劃長度 (planning horizon) 的 180 分鐘裡面，皆會包含 45 個時間長度為 4 分鐘的規劃時段，如圖 3.9。

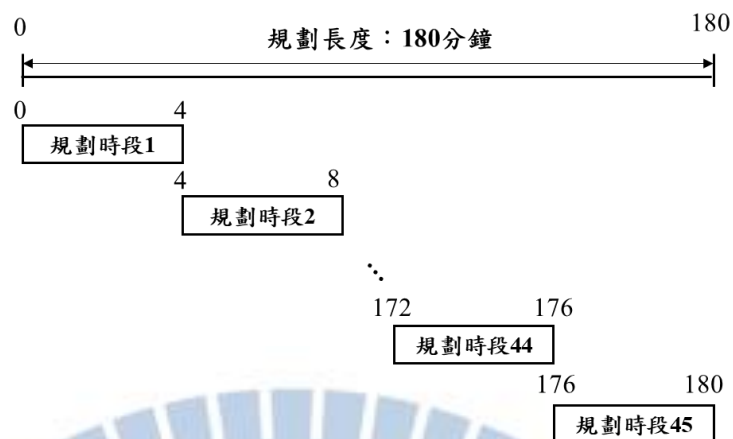


圖 3.9 規劃長度中的規劃時段劃分方式

◆ 已規劃時段

已經規劃過但餐廳或平台尚未完成或送達的訂單，如：已被餐廳與平台同時接受但尚未完成的內用訂單與外送訂單和已被餐廳與平台同時接受已完成但尚未送達的外送訂單，依然會影響平台對目前規劃時段的外送訂單決策。所以本研究必須將已規劃時段納入平台決策時考慮的決策時段。圖 3.10 為在規劃長度中的已規劃時段與規劃時段劃分方式。

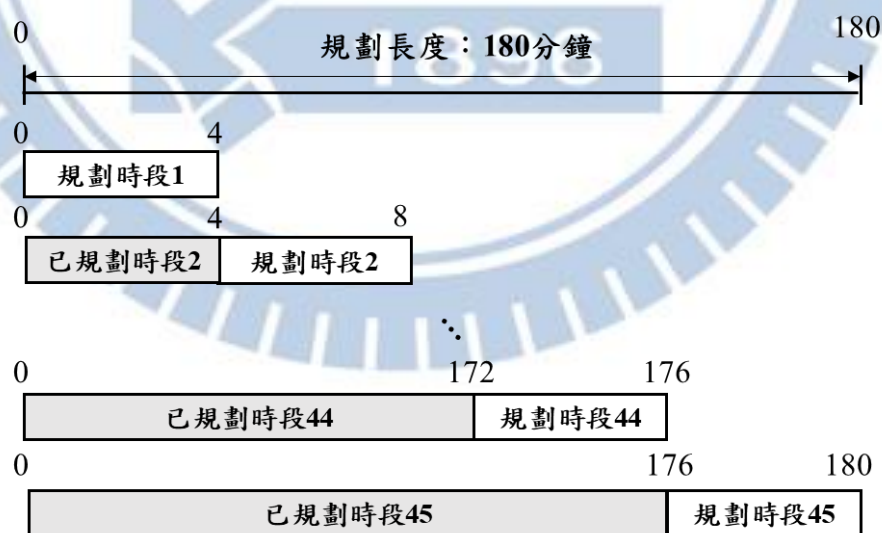


圖 3.10 已規劃時段與規劃時段劃分方式

◆ 預測時段

此外，本研究會將未來可能出現的訂單納入考慮。但是顧客點餐時間離目前規劃時段太遠的訂單，將其納入考慮並不會影響決策結果，考慮的訂單量過多，也會造成平台模型計算時間過久。

因此，我們必須決定適當的預測時段的長度，點餐時間在預測時段之後的內用與外送訂單皆不會對目前規劃時段決策造成影響。以規劃時段 1 為例，本研究考慮最壞情況，亦即有哪些預測時段內的訂單可能影響到目前規劃時段的決策時間點（時間點 4）所接到的外送訂單。

雖然本研究將外送訂單分為即刻送訂單與預訂單，但由於預訂單的訂單數不大，經由訪談餐點外送平台，預訂單僅占每日外送訂單的 1/43。且由於預訂單為顧客提前約兩個小時下訂的訂單，所以餐廳有充足的時間提前準備，減少對其他訂單的排程影響。所以本研究在決定預測時段時，不將可能影響預訂單的訂單納入考慮。

一般而言，餐廳必須等候平台的數學模型計算結束後，才會得知平台的決策結果，所以這張訂單的最早可開始製作時間為時間點 5。此外，平台承諾即刻送訂單必須在顧客點餐後 60 分鐘內送達，故這張訂單的最晚可送達時間為時間點 64。換言之，餐廳與平台針對這張訂單餐點的生產與配送行為只可能發生在時間點 5 到時間點 65 之間。亦即，只有從該訂單餐點的最早可開始製作時間到該訂單餐點的最晚可送達時間，這個區間內的內用與外送訂單才可能影響該即刻送訂單的生產與配送決策。預測時段便可由此得出，為從目前規劃時段決策開始時間點（時間點 4）起算 60 分鐘，見圖 3.11。

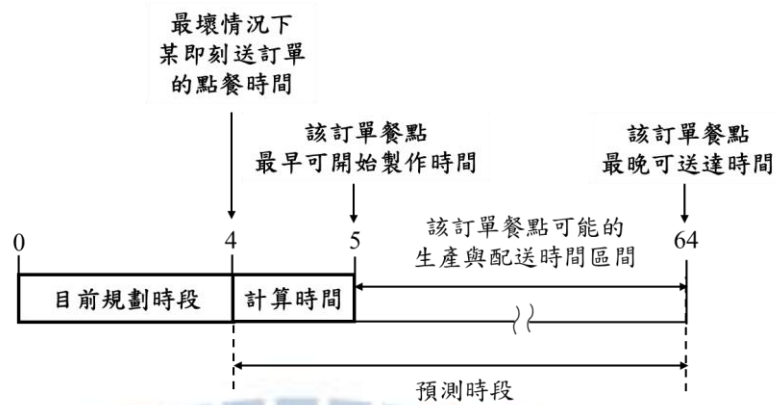


圖 3.11 預測時段時間區間

接著，我們可以將本研究所有會考慮的規劃時段與其對應的已規劃時段和預測時段，依照上述做劃分。下圖 3.12 為規劃長度中的已規劃時段、規劃時段和預測時段的劃分方式。預測時段 1 到預測時段 30 的時間長度皆為 60 分鐘。但是從預測時段 31 開始，由於受到規劃長度的時間上限 180 分鐘影響，每一個預測時段皆會縮短 4 分鐘。依此類推，預測時段 44 的時間長度僅剩 4 分鐘，並且不存在預測時段 45。

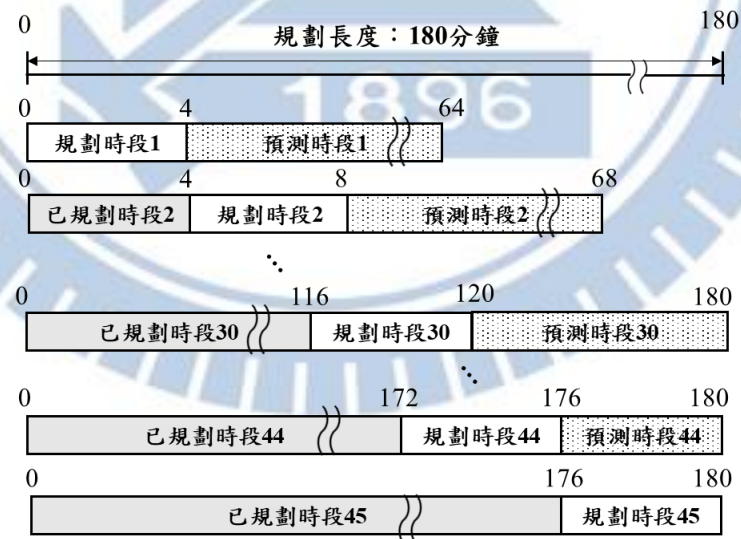


圖 3.12 規劃長度中的已規劃時段、規劃時段與預測時段劃分方式

3.3 餐點完成時間與取餐和送達時間之關係

當平台在決策時，會將決策時段內尚未服務的內用與外送訂單納入規劃。在規劃時必須注意到這些內用與外送訂單在餐點完成時間的限制，以及外送訂單在取餐與送達時間上的限制。

不同的訂單的限制也有所不同。承襲表 3.2，本研究將其依餐點完成時間、取餐時間或送達時間的不同分為 4 大類，分別是：

- 內用訂單
- 待平台評估餐廳與平台是否接受的外送訂單
- 已被餐廳與平台同時接受但尚未完成或送達的外送訂單
- 預測時段的外送訂單

■ 內用訂單

◆ 餐點完成時間

內用訂單的餐點完成時間有下列 2 個限制：

1. 餐點最早可完成時間為點餐時間+餐點製作時間。

由於內用訂單餐點必須點餐後才能製作，故有此限制。

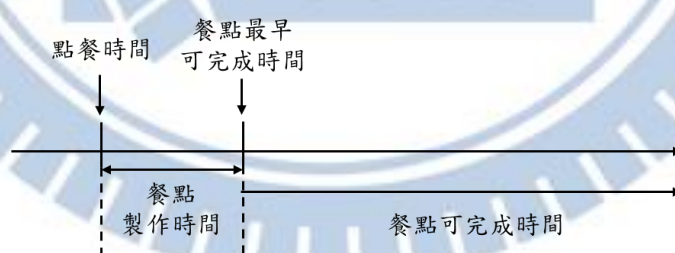


圖 3.13 內用訂單餐點最早可完成時間

2. 餐點完成時間必須符合內用訂單的服務時窗。

在 1.1 節提及，餐廳不能讓內用顧客點餐後等候過久才拿到餐點。一般而言，餐廳必須在內用顧客點餐後 40 分鐘內服務該顧客，故本研究將內用訂單的服務時窗訂為該訂單的點餐時間到該訂單的點餐時間 + 40 分鐘的時間區間。

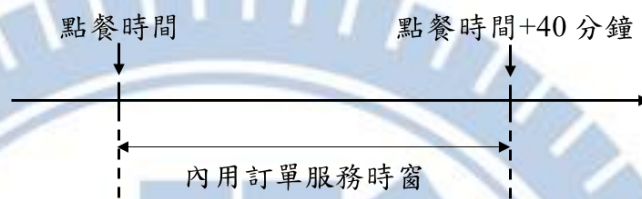


圖 3.14 內用訂單的服務時窗

■ 待平台評估餐廳與平台是否接受的外送訂單

◆ 餐點完成時間

待平台評估餐廳與平台是否接受的外送訂單的餐點完成時間必須比目前規劃時段決策完成時間點晚。因為餐廳必須知道平台的決策結果後才能開始製作餐點，所以餐點最早開始製作時間點為目前規劃時段決策完成時間點，藉由餐點最早開始製作時間便可推算出餐點最早可完成時間點，與餐點可完成的時間區間。

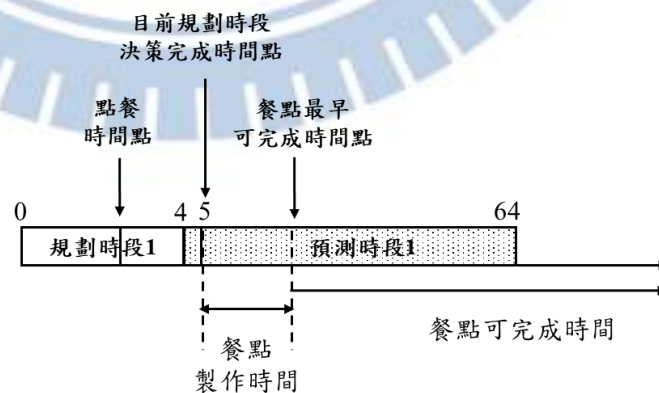


圖 3.15 待平台評估餐廳與平台是否接受的外送訂單餐點可完成時間

◆ 取餐時間

為了顧及餐點送達的品質，餐廳不希望平台取餐時間和餐點完成時間相隔太久，餐廳會根據平台提供的取餐時間倒推其餐點完成時間。因此，本研究在計算取餐時間時，假設在餐點製作完 10 分鐘內，外送員一定要取餐。此外，同一張訂單裡的所有餐點必須完成後，外送員才能取餐。

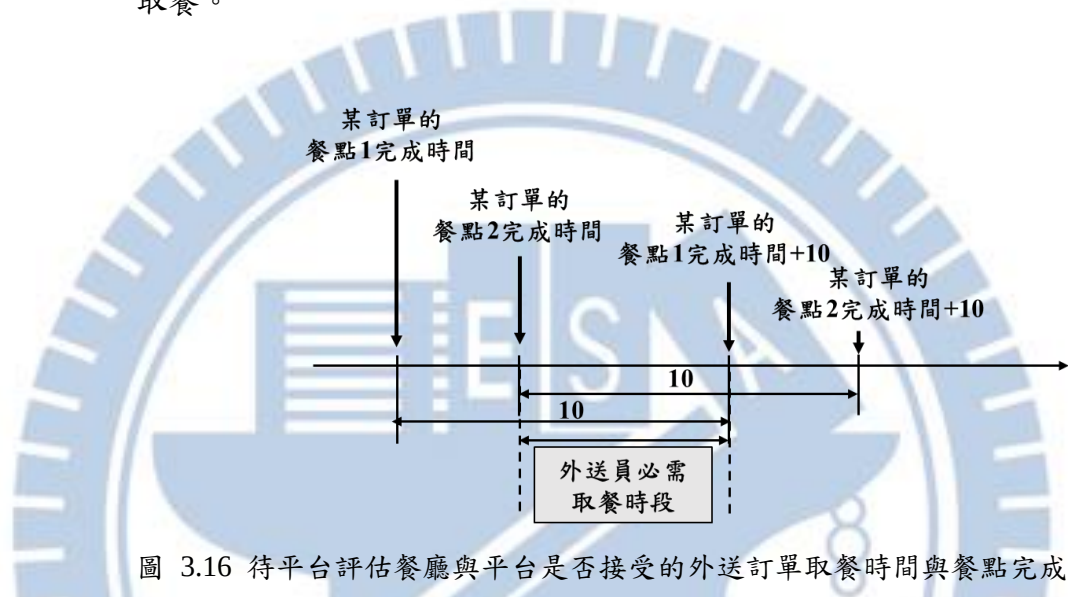


圖 3.16 待平台評估餐廳與平台是否接受的外送訂單取餐時間與餐點完成時間之關係

◆ 送達時間

送達時間則有下列 3 個限制：

1. 送達時間必須在平台承諾顧客的服務時窗內。

如同表 3.2，待平台評估餐廳與平台是否接受的外送訂單即為即刻送訂單。平台承諾顧客的即刻送訂單在顧客點餐時間後 60 分鐘內會送達，所以，即刻送訂單的服務時窗為顧客點餐時間與顧客點餐時間 + 60 分鐘的區間。

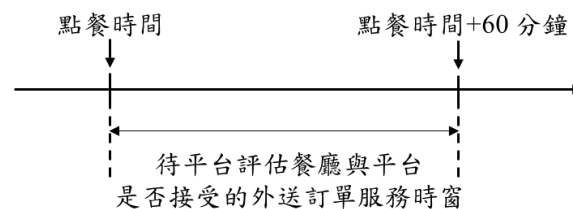


圖 3.17 待平台評估餐廳與平台是否接受的外送訂單的服務時窗

2. 送達時間必須在訂單餐點的賞味期限內

由於餐點送達顧客手中時必須保持可食用的狀態，故送達時間必須落在餐點的賞味期限內。賞味期限表示餐點在完成之後，多久的時間內顧客仍可食用。

3. 送達時間必須晚於取餐時間

外送員一定要先取該訂單裡的餐點，才能配送該訂單。

綜合第 2 點與第 3 點我們可以得出外送員的必須送達時段圖，如圖 3.18 所示。

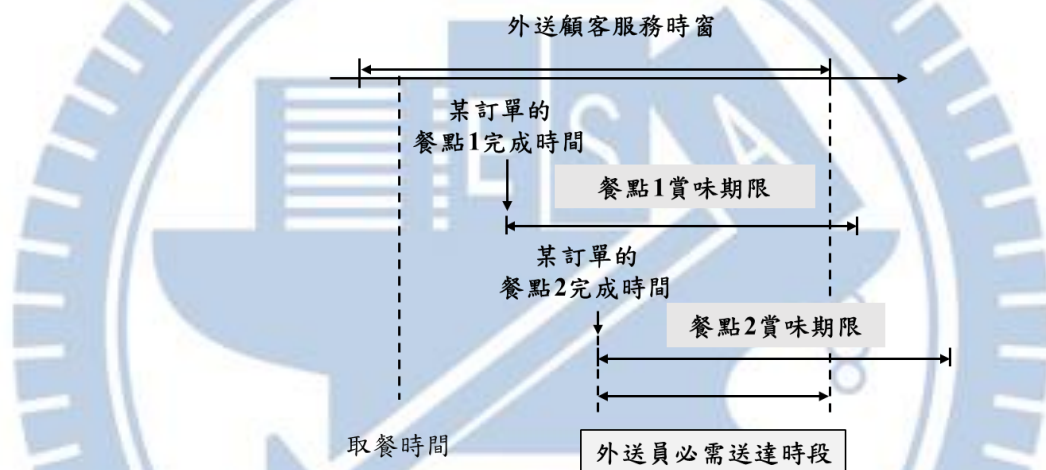


圖 3.18 目前規劃時段的外送訂單外送員的必須送達時段

■ 已被餐廳與平台同時接受但尚未完成或送達的外送訂單

見表 3.2，這類訂單可包含已被餐廳與平台同時接受但尚未完成或送達的即刻送訂單與預訂單，以下將會就即刻送訂單與預訂單來分別介紹。

◆ 即刻送訂單

● 餐點完成時間

已被餐廳與平台同時接受但尚未完成或送達的即刻送訂單表示平台已建議餐廳與平台皆該接受該訂單，估算過該訂單餐點的完成時間、取餐時間與送達時間，並且已經告知餐廳該訂單的取餐時

間。由此得知，餐廳完成餐點時間並不會離取餐時間太遠，依照前面的假設為 10 分鐘，所以我們便可以得到已被餐廳與平台同時接受但尚未完成的訂單餐點之預估完成時間段。

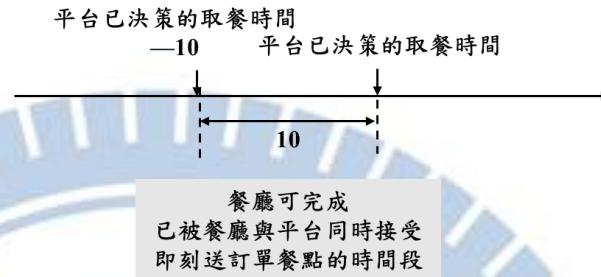


圖 3.19 餐廳可完成已被餐廳與平台同時接受即刻送訂單餐點的時間段

● 取餐時間

為了讓平台在調度外送員時有彈性，本研究假設外送員可以比平台已決策的取餐時間晚取餐，但是不能超過訂單可取餐時間段的上界 (upper bound)。這裡的可取餐時間段表示該訂單仍待平台評估餐廳與平台是否接受時，我們推估的平台必須取餐時間段。

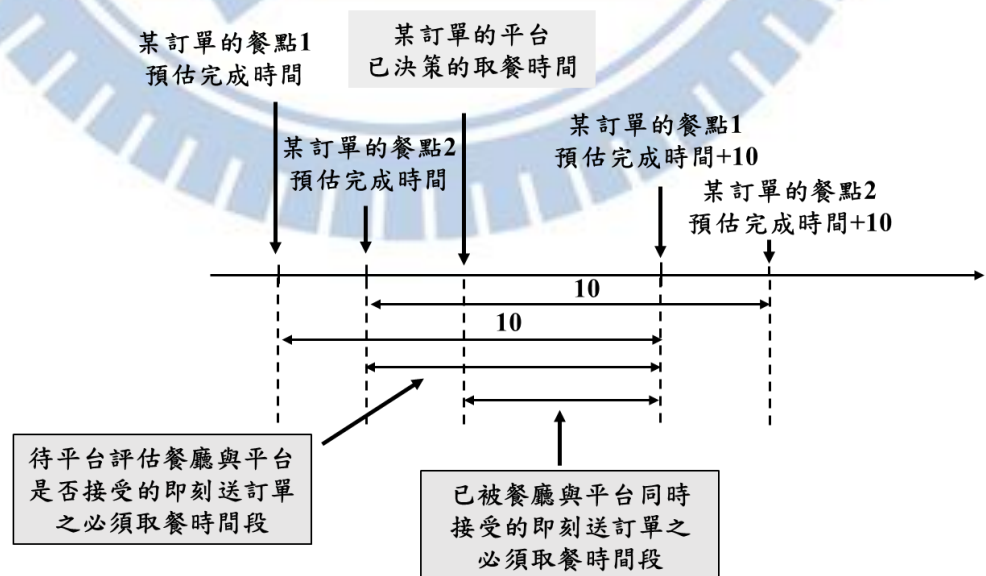


圖 3.20 已被餐廳與平台同時接受的即刻送訂單之必須取餐時間段

- 送達時間

由於已被餐廳與平台同時接受的即刻送訂單已經告知顧客該訂單的預計送達時間，透過訪談平台得知，若是顧客的訂單遲到，不論遲到多久顧客皆有可能與客服抱怨並拒絕訂單。反之，顧客比較能忍受外送員將訂單提前送達。為了讓平台在調度外送員時有彈性，本研究假設外送員最早可以將訂單提前 10 分鐘送達。

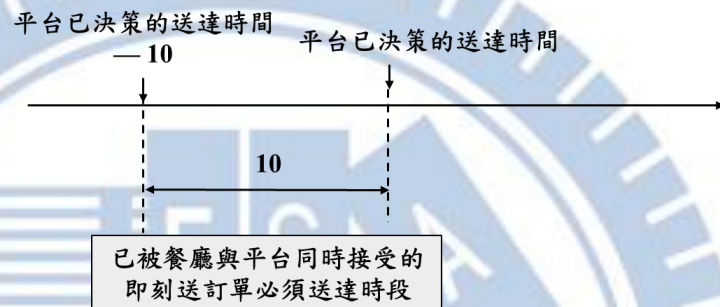


圖 3.21 已被餐廳與平台同時接受的即刻送訂單的必須送達時間段

- ◆ 預訂單

- 餐點完成時間

預訂單的送達時間段上界減去賞味期限，為餐廳最早可以完成該餐點的時間，這樣一來才可以保證餐點送達時的仍是可以食用的情況。一般而言，只要在餐點最早完成時間之後皆是該餐點的可行完成時間，但是如同前述，外送員必須要在餐廳完成餐點的 10 分鐘之內取餐。為了省去餐廳回報預訂單餐點的完成時間給平台外送員的步驟，本研究假設餐廳會盡早完成預訂單餐點，而預訂單的餐點完成時間段為餐點最早可完成時間再後推 10 分鐘區間的時間段。

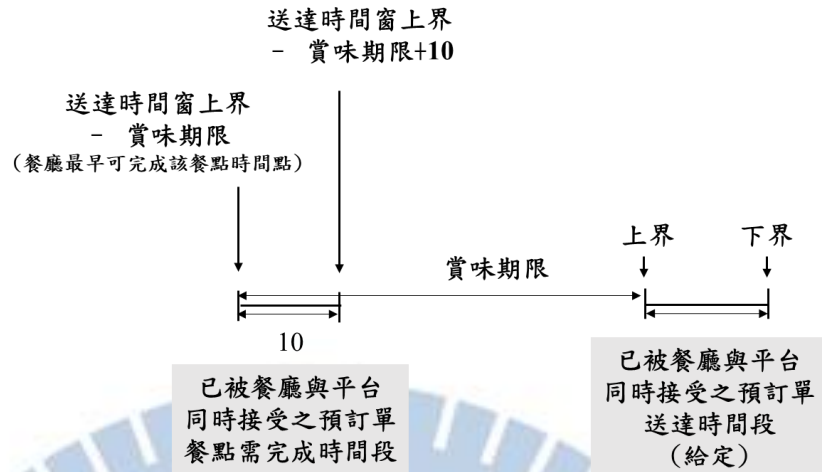


圖 3.22 已被餐廳與平台同時接受的預訂單的餐點須完成時間段

● 取餐時間

承上述，由於預訂單的餐點完成時間已知，為了使外送員能在餐點完成 10 分鐘之內取餐，該訂單所有的餐點可完成時間段交集即為該訂單的外送員可取餐時間段。

● 送達時間

如同第一章所介紹，不同外送平台提供給顧客選擇的送達時間段皆不相同。本研究參考目前在台灣規模最大之外送平台 - 空腹熊貓所提供給顧客選擇之送達時間點。並如同前面的假設，外送員最早可以將訂單提前 10 分鐘送達。

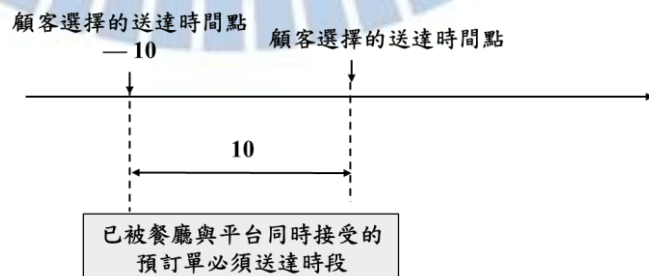


圖 3.23 已被餐廳與平台同時接受的預訂單的送達時間段

■ 預測時段的外送訂單

預測時段的外送訂單同樣分成即刻送訂單與預訂單介紹。

◆ 即刻送訂單

● 餐點完成時間

與待平台評估餐廳與平台是否接受之訂單的決策方式相同。

● 取餐時間

與待平台評估餐廳與平台是否接受之訂單的決策方式相同。

● 送達時間

與待平台評估餐廳與平台是否接受之訂單的決策方式相同。

◆ 預訂單

● 餐點完成時間

與已被餐廳與平台同時接受之預訂單的決策方式相同。

● 取餐時間

與已被餐廳與平台同時接受之預訂單的決策方式相同。

● 送達時間

與已被餐廳與平台同時接受之預訂單的決策方式相同。

3.4 生產排程問題

在平台估算外送訂單的餐點完成時間時，平台必須知道訂單的餐點資訊與餐廳的生產系統資訊，來估算訂單的餐點預計何時完成。在本章節中，不同的訂單平台獲得的資訊方式不同，在決策中使用到的資訊也可能有所不同。所以本研究將平台擁有的訂單資訊所包含的餐點對應到不同的餐點狀態。敘述完餐點狀態後，將會接著定義餐廳的生產系統。

■ 餐點狀態

承表 3.5，表 3.6 為訂單狀態與餐點狀態之對照。此外，在生產問題中，不需考慮已被餐廳與平台同時接受已完成但尚未送達的訂單。

表 3.6 已規劃時段、目前規劃時段與預測時段所包含的訂單狀態與餐點狀態之對應

已規劃時段			目前規劃時段		預測時段	
已被餐廳與平台 同時接受但尚未完成的			已被餐廳與平台 同時接受但尚未完成的		預測未來可能出現	
①	2	內用訂單	2	內用訂單	5	內用訂單
	3	外送訂單 (即刻送訂單)	3	外送訂單(預訂單)	6	外送訂單 (即刻送訂單)
		外送訂單(預訂單)			3	外送訂單 (預訂單)
已被餐廳與平台同時接受 已完成但尚未送達的			待平台評估餐廳與平台 是否接受的			
外送訂單(即刻送訂單)			4	外送訂單 (即刻送訂單)		
外送訂單(預訂單)						

表 3.7 餐點狀態之說明

餐點狀態編號	餐點狀態
1	已被餐廳與平台同時接受且正在製作的內用或外送餐點
2	已被餐廳與平台同時接受但尚未開始製作的內用餐點
3	已被餐廳與平台同時接受但尚未開始製作的外送餐點
4	待平台評估餐廳與平台是否接受的外送餐點
5	預測未來在餐廳可能接到的內用餐點
6	預測未來在餐廳可能接到的外送餐點

首先，已被餐廳與平台同時接受的訂單可分為已被餐廳與平台同時接受的內用訂單與外送訂單，其中外送訂單包含即刻送訂單與預訂單。這些訂單的餐點可以分為已被餐廳與平台同時接受且正在製作的內用或外送餐點（狀態 1）、已被餐廳與平台同時接受但尚未開始製作的內用餐點（狀態 2）和已被餐廳與平台同時接受但尚未開始製作的外送餐點（狀態 3）。

首先，正在製作的內用與外送訂單餐點，因為餐廳無法改變其排程，所以在建立數學模型的時候不需要特別將內用與外送訂單分類。另外，由於已規劃時段與目前規劃時段裡接到的內用訂單，資訊來源都是來自餐廳，且平台並不會建議餐廳接受或拒絕，所以不需要將其分類。

目前規劃時段內接到的待平台評估餐廳與平台是否接受的外送訂單（即刻送訂單），則是對應到待平台評估餐廳與平台是否接受的外送餐點（狀態 4）。而目前規劃時段內接到的已被餐廳與平台同時接受的外送訂單（預訂單），由於餐廳和平台皆不會拒絕預訂單，所以該訂單對應到已被餐廳與平台同時接受但尚未開始製作的外送餐點（狀態 3）。

預測未來在餐廳可能接到的內用餐點（狀態 5）與預測未來在餐廳可能接到的外送餐點（狀態 6），則分別對應目前規劃時段接到的外送訂單，預測時段可能接到的內用與外送訂單（即刻送訂單）。為了使分類不要過於複雜，我們將預測時段可能接到的外送訂單（預訂單）對應到已被餐廳與平台同時接受但尚未開始製作的外送餐點（狀態 3），原因如同前述，餐廳與平台並不會拒絕預訂單。

■ 餐廳生產系統

一般而言，餐廳的廚房是由負責不同餐點製作步驟的工作站組成。而內用與外送訂單的餐點的製作順序也有一些限制。所以在建構餐廳的生產系統時，本研究將其分為三個部分：餐點製作步驟、工作站與餐點製作順序。

◆ 餐點製作步驟

每一個餐點是由一個或多個製作步驟組成，每個步驟會由餐廳廚房的工作站所負責。圖 3.24 餐廳的餐點可能的製作步驟(e.g. 漢堡與薯條)表示某餐廳的餐點可能的製作步驟的範例，該餐廳提供漢堡與薯條兩種餐點，漢堡含有 3 個步驟，炸 2 分鐘、炒 7 分鐘與煎 8 分鐘)，其中炸跟炒可以同時進行，煎則是要等其他材料炸或炒之後，才能執行。薯條僅需炸 10 分鐘。

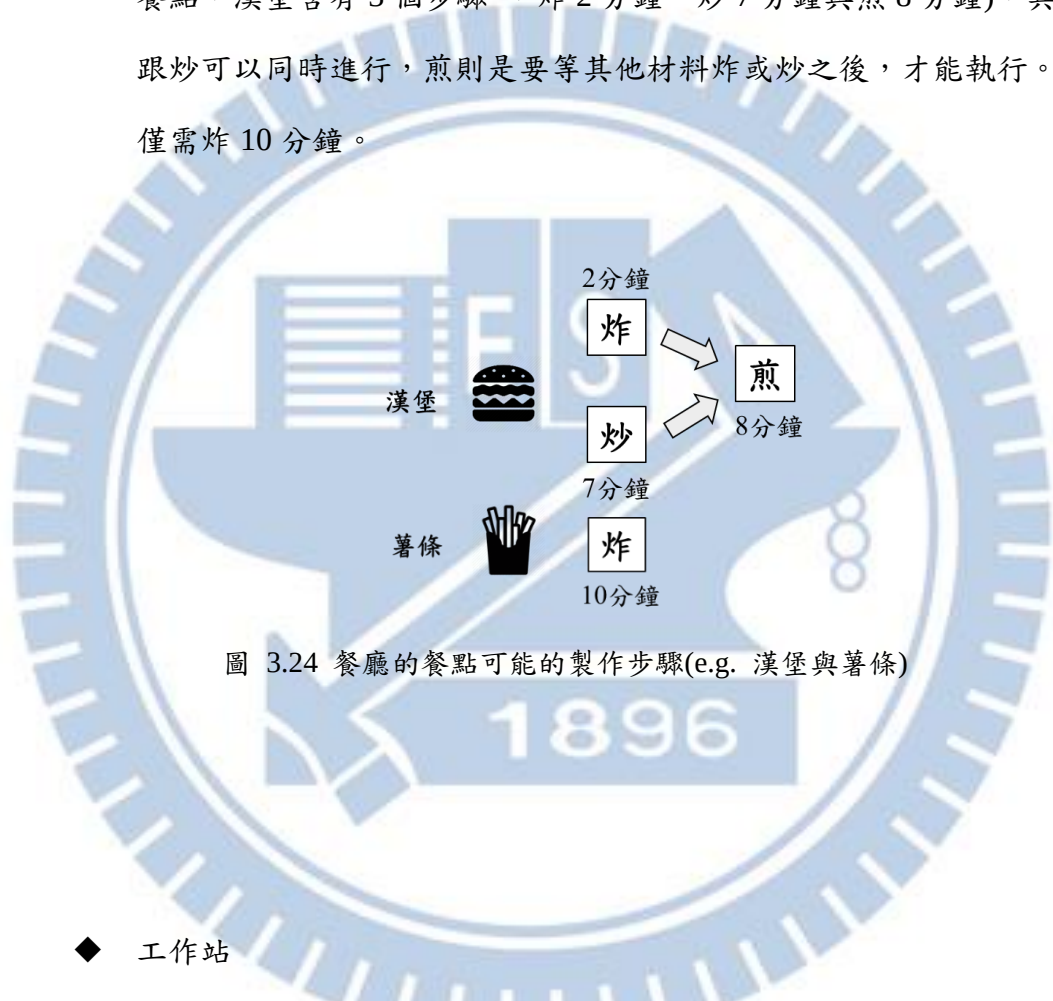


圖 3.24 餐廳的餐點可能的製作步驟(e.g. 漢堡與薯條)

◆ 工作站

通常，餐廳廚房裡會有一個或多個工作站，每個工作站負責不同且不重複的餐點製作步驟，例如：煎、煮、炒或炸，每個工作站有一位或多位的廚師執行餐點製作步驟。為了簡化問題，本研究假設：

- 每個工作站的產能僅受限於工作站的廚師數量。
- 每位廚師在某時間點最多製作一道餐點。
- 餐點製作步驟執行到一半不能中斷。

圖 3.25 為某餐廳廚房的工作站組成範例。

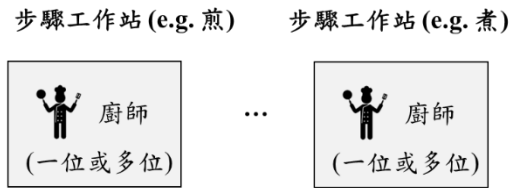


圖 3.25 餐廳廚房的工作站組成範例

◆ 餐點製作順序限制

此外，餐廳在製作內用訂單餐點的時候，必須注意內用訂單的餐點，先點餐的訂單餐點要先完成，否則會引起內用顧客的不滿。而外送訂單的餐點則無此限制。圖 3.26 表示內用餐點與外送餐點的可行餐點完成時間。此外，由於工作站產能的限制，每道餐點未必能在點餐後就馬上製作。因此，餐點完成時間不一定為點餐時間加上餐點製作步驟的時間長度。

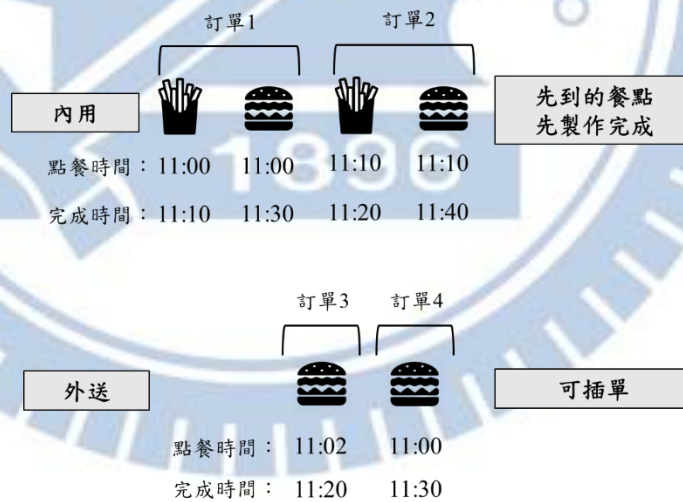


圖 3.26 內用餐點與外送餐點的可能的完成時間

由上述的問題環境可知，本研究問題類似 Chang et al. (2016) 所提出的有同時進行的餐點製作步驟之彈性零工排程問題 (the flexible job shop problem with simultaneous operations, FJSPSO)。

■ 有同時進行的餐點製作步驟之彈性零工排程問題

Chang et al. (2016) 將年菜外送問題的生產部分建立成有同時進行的餐點製作步驟之彈性零工排程問題，該問題參考自 Pinedo (2009) 與 Brucker et al. (2011) 的彈性零工問題 (the flexible job shop problem)。該問題將餐廳的廚房建構成數個工作站，餐廳裡面的每個餐點有不同的製作步驟，同一餐點的某些製作步驟可以同時進行，如：圖 3.24 漢堡的炸跟炒步驟。與本研究不同的是，在年菜外送的環境中，製作年菜的廚房通常不會同時服務內用顧客餐點，所以並不需要將內用餐點納入生產排程問題。

一般而言，FJSPSO 問題通常以分離節線圖 (disjunctive arc graph) 的方式描述與求解。分離節線圖的概念 Pinedo (2009) 與 Brucker et al. (2011) 皆有提及。在分離節線圖中分別有 2 種節線連接節線 (conjunctive arc) 與分離節線 (disjunctive arc)，下面將舉一個 FJSPSO 問題的例子，並建構成分離節線圖來做介紹。

表 3.8 FJSPSO 問題之範例

訂單編號	內用/外送	點餐時間	餐點	餐點製作步驟
1	內用	11:00	漢堡	① 炸2分鐘 → ③ 煎8分鐘 ② 炒7分鐘 →
			薯條	④ 炸10分鐘
2	內用	11:05	薯條	⑤ 炸10分鐘
3	外送	11:03	漢堡	⑥ 炸2分鐘 → ⑧ 煎8分鐘 ⑦ 炒7分鐘 →

在這個小例子中，我們先不把表 3.7 的餐點狀態納入，因為不同的餐點狀態不會影響餐點在分離節線圖的表示方式。餐廳待製作的訂單有 3 張，訂

單 1 與 2 為內用訂單，訂單 3 為外送訂單。點餐時間分別為 11:00、11:05 與 11:03，訂單 1 有漢堡和薯條，訂單 2 有薯條，訂單 3 只有漢堡。

漢堡的製作步驟如圖 3.24 所述：分別將某些材料炒 7 分鐘、某些材料炸 2 分鐘，再將所有材料一起在煎工作站煎 8 分鐘，即可完成一個漢堡。而薯條只需要炸 10 分鐘，即可完成。每道餐點的每個製作步驟皆以不同號碼編號，如上表 3.8 的步驟編號 1 到 8，即表示不同訂單餐點的製作步驟編號。另外，餐廳內部有 3 個工作站，煎、炒與炸，分別由 1、2 和 2 位廚師負責。以此例子所建構的分離節線圖如圖 3.27。

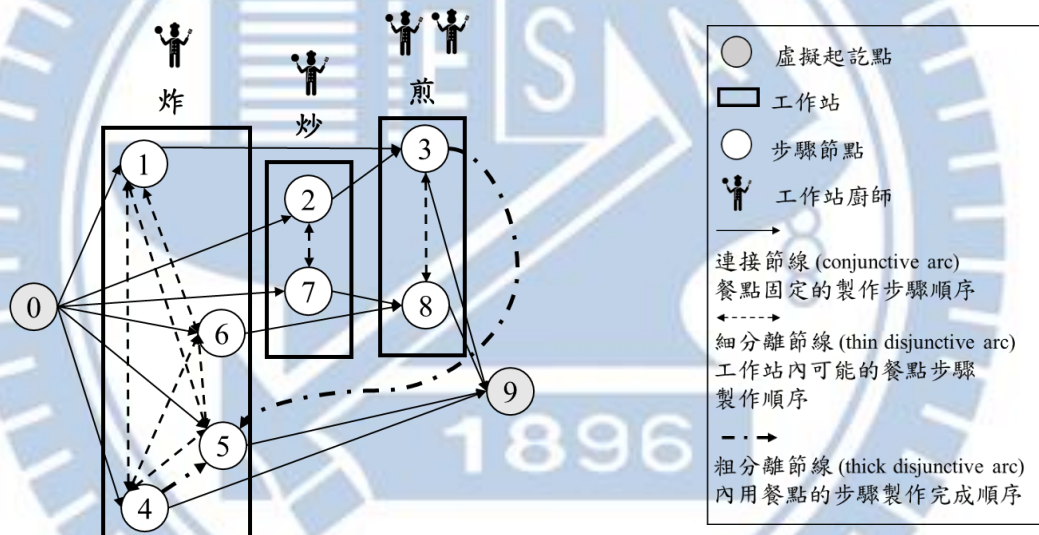


圖 3.27 FJSPSO 問題的分離節線圖

在分離節線圖中，虛擬起訖點 0 與 9 表示餐點開始製作與製作完成，步驟節點則是對應到待規劃餐點的製作步驟。節線分 3 種：連接節線 (conjunctive arc)、細分離節線 (thin disjunctive arc) 與粗分離節線 (thick disjunctive arc)。連接節線表示每道餐點的製作步驟，例如：步驟 1、2 與 3 代表的是訂單 1 漢堡的製作步驟，其中步驟 1 與 2 可以同時製作，所以節線 (0, 1)、(0, 2)、(1, 3)、(2, 3) 與 (3, 9) 連結該餐點的步驟節點，來表示

該餐點固定的製作順序。而細分離節線表示，工作站中可能的製作步驟，例如：炸工作站中待製作的步驟節點 1、4、5 與 6 並沒有先後順序，所以用雙向的細分離節線連結所有節點。最後，粗分離節線的概念參考自 Chang et al. (2016) 是用來表示內用餐點的先到先服務限制，也就是先點的餐點，餐點一定要先完成。於是，圖中將訂單 1 的漢堡與薯條的最後一個步驟節點 3 與 4，與訂單 2 薯條的最後一個步驟節點 5 連結，來表示訂單 2 的薯條完成時間一定不能比訂單 1 的漢堡與薯條早。

如此一來，FJSPSO 問題的分離節線圖便建立完成，由於 FJSPSO 問題的數學模型是相當複雜的，所以在本章節中並不會詳細列出其數學模型。

■ 有資源限制的平行機台排程問題

FJSPSO 問題的建立方式雖然可以充分表示餐廳的製作排程，但是由於在本問題中，餐點外送平台只需要告知餐廳外送訂單的取餐時間，交由餐廳利用取餐時間的資訊推算餐點的完成時間，平台並不需要建議餐廳該餐點每個步驟的完成時間。此外，平台擁有的餐點資訊僅有每個餐點的總製作時間長度，並不知道該餐點的詳細製作步驟。所以，若是考慮到每個步驟的製作排程，事實上只是徒增模型的複雜度。

因此，本研究提出一個將有同時進行的餐點製作步驟之彈性零工排程問題轉換成有資源限制的平行機台排程問題 (parallel machine scheduling problem with resources constraint, PMSPRS) 的方式。我們選擇平行機台問題做為 FJSPSO 的簡化是由於在餐廳中相同餐點種類與不同餐點種類的餐點會互相影響，類似於相同餐點種類與不同餐點種類在平行機台上的排程問題。並且利用在生產排程問題中的資源限制，來表示整個生產系統的產能限制。

在 PMSPRS 問題中，餐點的細部步驟將被省略，僅關心整個餐點開始製作時間與完成時間。在平行機台生產問題中，我們假設每台機器的生產效率

相同，而不論餐點種類的同質與異質餐點可以在所有平行機台上進行排程。此外，在轉換過程中生產系統的產能與餐點的製作時間長度必須保持不變。

◆ 生產系統產能轉換

產能是指一個設備在某一特定時間內，能夠接收、儲存或生產的產品數量（林則孟、2012）。一般而言，一個生產系統產能是由瓶頸(bottleneck) 工作站產能來定義。而不同的餐點製作流程在生產系統中對應的瓶頸工作站未必相同，所以我們將生產系統的產能表示為，餐廳在樂觀且長期穩定的狀態下，即餐廳僅製作同質餐點，沒有受到其他種類餐點干擾下的瓶頸工作站產能。例如：某餐廳的生產系統產能可定義為，該餐廳平均每分鐘可製作完成 $1/7$ 個漢堡，或是在平均每分鐘可製作完成 $1/10$ 份薯條。

下圖的例子承襲圖 3.24 與圖 3.25，表餐廳廚房工作站的廚師與餐點製作步驟的網路圖。

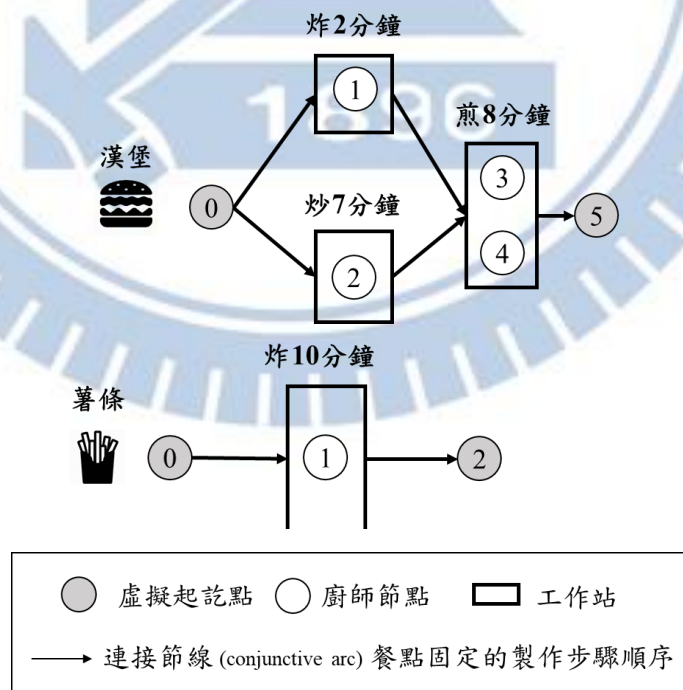


圖 3.28 餐點製作步驟與該步驟對應的工作站

我們參考林則孟 (2012) 的描述，工作站的產能 = 資源數量 (如：廚師數目) / 該工作站的步驟執行時間 = 每單位時間該工作站可以完成該餐點步驟數目。以漢堡為例，在本生產系統平均每分鐘可以完成 1/2 個炸工作，平均每分鐘可完成 1/7 炒工作，平均每分鐘可完成 2/8 煎工作。瓶頸工作站即為炒工作站，因為 $\text{Min}\{1/2, 1/7, 2/8\} = 1/7$ ，表示在樂觀且長期穩定的狀況下，餐廳製作完成的漢堡數量會受制於炒工作站。亦即表示在樂觀且長期穩定的狀況下，餐廳平均每分鐘可以完成 1/7 個漢堡。

上述計算可以透過下圖為證，每個不同顏色的方格代表不同工作站的工作，每個工作的編號為對應的漢堡編號。在樂觀且長期穩定的狀態下，即該餐廳僅製作漢堡，在第 15 分鐘之後，該餐廳平均每 7 分鐘可以產出 1 個漢堡，也就是平均每分鐘可完成 1/7 個漢堡，該數值亦等於瓶頸工作站的產能數值。

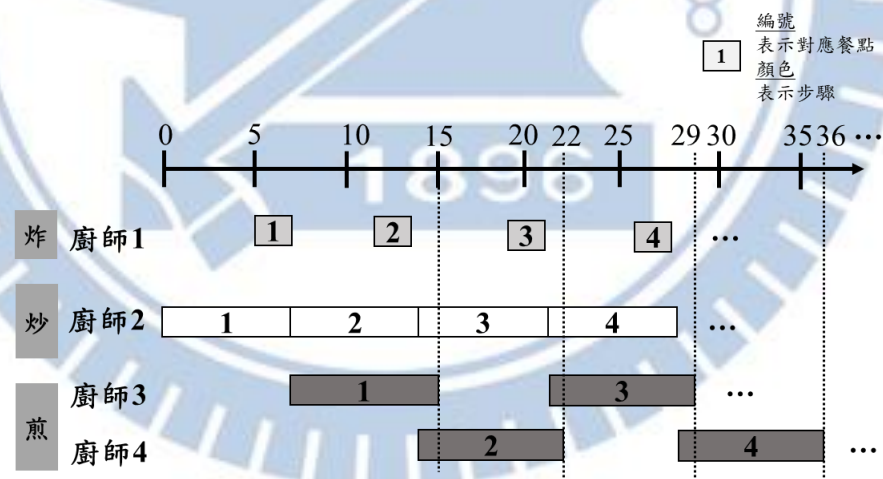


圖 3.29 某餐廳在樂觀且長期穩定狀態下製作漢堡的情形

於是，我們就可以將該餐廳生產系統的產能定義為在樂觀且長期穩定的狀態下，餐廳平均每分鐘可完成 1/7 個漢堡或是平均每分鐘完成 1/10 份薯條。

◆ 餐點製作時間長度

在 PMSPRS 問題中，僅關心整個餐點開始製作時間與完成時間，為了簡化問題，我們承襲 FJSPSO 問題的假設並加以延伸，假設每個餐點一旦開始製作，製作的途中不能暫停。根據實地訪談，在餐廳中通常會備有一位排菜員，掌控餐點製作的順暢度，盡量不會讓各個步驟間有時間空隙。

所以本研究將餐點製作時間長度的轉換方式如下：

1. 若有同時進行的步驟，則僅保留製作時間最長步驟的製作時間
2. 若有先後順序之步驟，則加總各步驟的製作時間

下圖以漢堡製作時間長度的轉換為例：

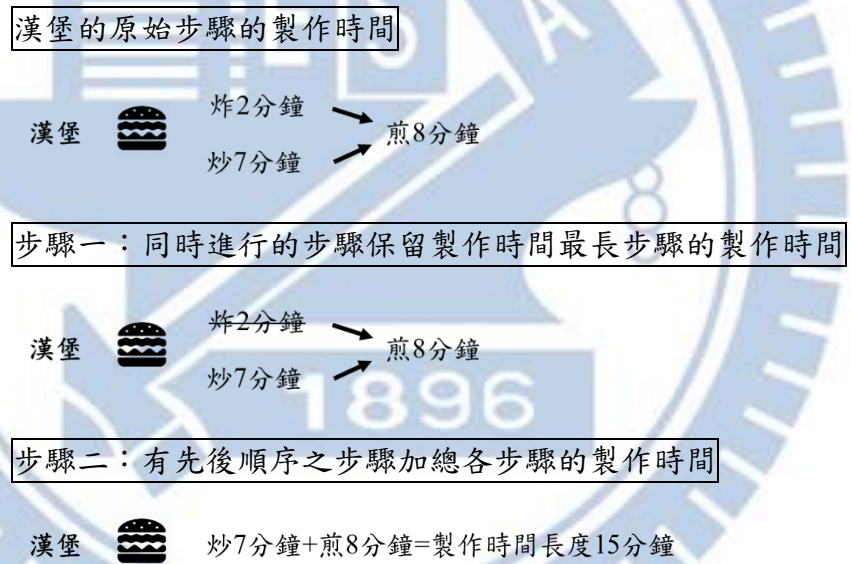


圖 3.30 漢堡製作時間長度計算方式

如此一來，FJSPSO 問題的餐點製作時間長度 (如：漢堡製作時間長度為炒 7 分鐘+煎 8 分鐘，假設步驟間沒有等候時間)，在轉換為 PMSPRS 問題中，每個餐點從開始製作到完成的製作時間長度不會被高估或低估 (如：漢堡製作時間長度為 15 分鐘)。

◆ 平行機台數量

在將 FJSPSO 問題轉換成 PMSPRS 問題後，生產系統的產能必須保持相同。而生產系統的產能則會受到餐點在每個時間點製作時能同時使用的平行機台數量影響，以漢堡為例，在 PMSPRS 問題中，漢堡的製作時間長度為 15 分鐘。為了要保持在樂觀且長期穩定的狀態下，生產系統產能依然為平均每分鐘完成 $1/7$ 漢堡，我們將漢堡設定為每分鐘僅能在 2 個平行機台上同時製作。

若漢堡每分鐘僅能在 2 台平行機台上同時製作，在樂觀且長期穩定的狀態下，該 PMSPRS 問題的生產系統產能為平均每分鐘可以完成 $2/15$ 個漢堡，如下圖所示，將 $2/15$ 四捨五入後約等於 $1/7$ 。

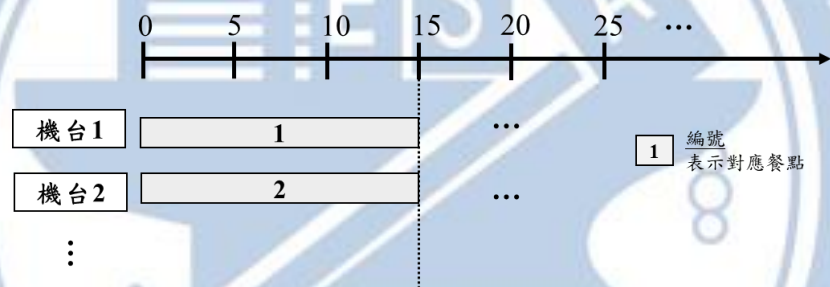


圖 3.31 漢堡每分鐘只能同時在 2 台平行機台上生產的情形

所以我們計算餐點在單位時間同時最多能使用的平行機台數的公式為：某餐點的瓶頸工作站產能 = 某餐點在單位時間同時最多使用的平行機台數/某餐點的製作時間長度。

同樣的，我們將薯條設定為每分鐘僅能在 1 個平行機台上同時製作，以保持 FJSPSO 問題與 PMSPRS 問題的生產系統產能相同，皆為在樂觀且長期穩定的狀態下，平均每分鐘可以完成 $1/10$ 份薯條。

因此，在 PMSPRS 問題中的平行機台數目會設定為，各個餐點種類的餐點在單位時間，能使用的平行機台數目的最大值。在此例中，漢堡在每分鐘最多可以在 2 台平行機台上同時製作，而薯條在每分鐘最多僅

能在 1 台平行機台上同時製作。於是在本 PMSPRS 問題的生產系統中
 平行機台數便會設為 2 台，並限制薯條在每個時間點時，僅能選擇其中
 1 台平行機台製作。

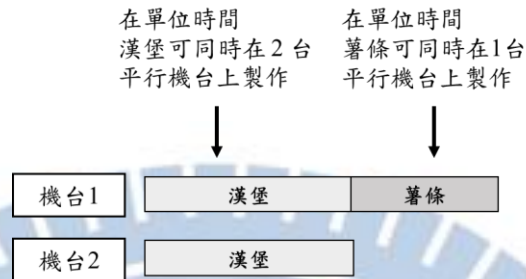


圖 3.32 PMSPRS 問題之示意圖

■ 生產排程問題之範例

在本章節中，將舉 3 個生產排程問題的例子，代表 3 種決策時段。3 種
 決策時段為：包含目前規劃時段與預測時段，包含已規劃時段、目前規劃時
 段與預測時段，包含已規劃時段與目前規劃時段的型態。3 種決策時段如圖
 3.33 所示：

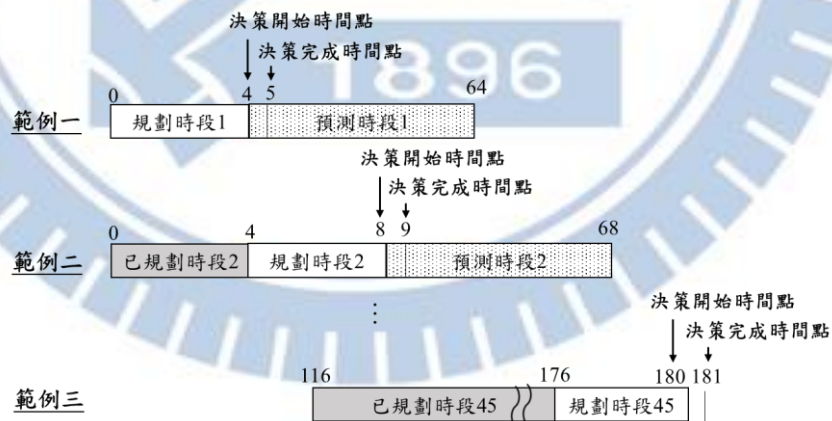


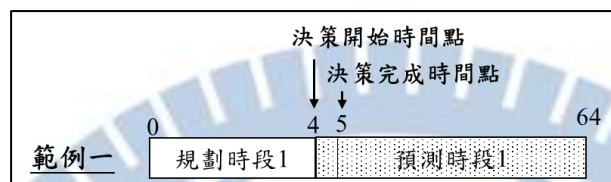
圖 3.33 範例一、二和三對應的決策時段圖

但由於範例二與範例三所考慮的限制皆與範例一相同，差別僅在範例二
 比範例一多了已規劃時段的訂單資料，範例三比範例一多了已規劃時段少了

預測時段的訂單資料。所以，接下來只介紹範例一與範例二，並介紹前一規劃時段轉變為下一決策時段已規劃時段時的訂單變化。

◆ 範例一

考慮決策時段



平行機台與餐點資訊

平行機台數量	餐點種類	製作時間	賞味期限	餐點體積 (以便當盒大小為單位)	每分鐘餐廳最多可製作數量
2	漢堡	15	20	3	2
	薯條	10	10	2	1

目前規劃時段決策時考慮的顧客訂單範例

訂單編號	點餐時間	旅行時間 (顧客到餐廳)	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間	圖示
1	0	0	1	1	2	漢堡	15	
2	25	0	1	2	5	薯條	10	
3	3	3	2	3	4	薯條	10	
				4	4	漢堡	15	
4	4	8	1	5	4	薯條	10	
5	15	5	1	6	6	薯條	10	

表 3.9 生產排程問題範例一的餐點狀態

餐點狀態編號	餐點狀態
2	已被餐廳與平台同時接受但尚未開始製作的內用餐點
4	待平台評估餐廳與平台是否接受的外送餐點
5	預測未來在餐廳可能接到的內用餐點
6	預測未來在餐廳可能接到的外送餐點

範例一待規劃的訂單共有 5 張。訂單 1 與 2 為內用訂單，點餐時間分別為時間點 0 與 25，分別含有 1 個餐點，餐點狀態為已被餐廳與平台同時接受但尚未開始製作的內用餐點以及預測未來在餐廳可能接到的內用餐點，訂單 1 的餐點為 1 份漢堡，訂單 2 的餐點為 1 份漢堡。

訂單 3、4 與 5 為外送訂單，點餐時間分別為時間點 3、4 與 15，分別含有 1、2、1 個餐點，訂單 3 與 4 的餐點狀態為待平台評估餐廳與平台是否接受的外送餐點，訂單 5 的餐點狀態為預測未來在餐廳可能接到的外送餐點，訂單 3 內含 1 份薯條，訂單 4 的餐點為 1 份漢堡與薯條，訂單 5 內含 1 份漢堡。

範例一的可行生產排程甘特圖如下圖：

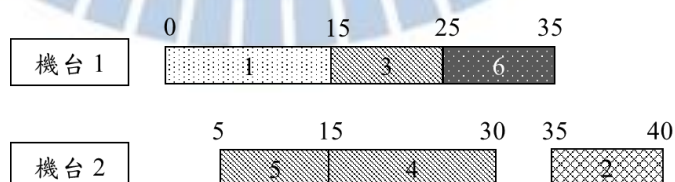
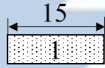


圖 3.34 範例一的可行生產排程甘特圖

接著，我們將會檢驗這樣的生產排程是否真的可行，先介紹內用訂單，再介紹外送訂單。如 3.3 節所述，內用訂單的餐點必須符合內用訂單服務時窗與先到先服務限制，外送訂單的餐點則必須符合餐點完成時間、以及與取餐時間與送達時間限制。

表 3.10 範例一內用餐點完成時間的可行性檢驗

訂單編號	點餐時間	服務時窗	餐點數量	餐點編號	餐點種類	製作時間	賞味期限
1	0	[0, 40]	1	1	漢堡	15	20
2	25	[25, 65]	1	2	薯條	10	10

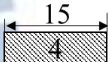
最早可完成時間	平台估算的完成時間	圖示
$0 (\text{點餐時間}) + 15 = 15$	15	
$25 (\text{點餐時間}) + 10 = 35$	40	

訂單 1 的餐點 1 平台估算的完成時間為第 15 分鐘，比最早可完成時間晚，並且落於內用訂單的服務時窗。

訂單 2 的餐點 2 平台估算的完成時間為第 40 分鐘，比最早可完成時間晚，並且落於內用訂單的服務時窗。此外，餐點 1 與餐點 2 彼此也滿足先到先服務限制 ($40 > 15$)。

表 3.11 範例一外送餐點完成時間的可行性檢驗

訂單編號	點餐時間	服務時窗	旅行時間 (顧客到餐廳)	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間	賞味期限
3	3	[3, 63]	3	2	3	4	薯條	10	10
					4	4	漢堡	15	20
4	4	[4, 64]	8	1	5	4	薯條	10	10
5	15	[25, 85]	5	1	6	6	薯條	10	10

最早可完成時間	平台估算的 完成時間	平台決策的		圖示
		取餐時間	送達時間	
5 (目前決策完成時間點) +10 = 15	25	31	35	
5 (目前決策完成時間點) +15 = 20	30			
5 (目前決策完成時間點) +10 = 15	15	17	25	
15 (目前決策完成時間點) +10 = 25	35	38	43	

由於外送訂單餐點完成時間是否可行，與外送訂單的取餐與送達時間相關，所以在表 3.11 中，我們列出該訂單平台以模型求解的取餐與送達時間，以供參考。

訂單 3 的餐點 3 與 4 完成時間分別為第 25 分鐘與第 30 分鐘，皆比最早可完成時間晚。接著，這 2 個餐點的賞味期限分別為 20 分鐘與 10 分鐘，該訂單的送達時間也滿足賞味期限限制 ($20 + 25 \geq 35$ 、 $30 + 10 \geq 35$)。

訂單 4 的餐點 5 完成時間為第 15 分鐘，恰好為最早可完成時間。
接著，該餐點的賞味期限為 10 分鐘，該訂單的送達時間也滿足賞味期限限制 ($15 + 10 \geq 25$)。

訂單 5 的餐點 6 完成時間為第 35 分鐘，比最早可完成時間晚。該餐點的賞味期限為 10 分鐘，因此，該訂單的送達時間也滿足賞味期限限制 ($35 + 10 \geq 43$)。

最後，必須檢查生產排程是否符合資源 (產能) 限制，由下方的餐點種類與生產排程對照圖可知，該生產排程符合每分鐘最多製作 2 個漢堡、1 分薯條的限制為可行的生產排程。

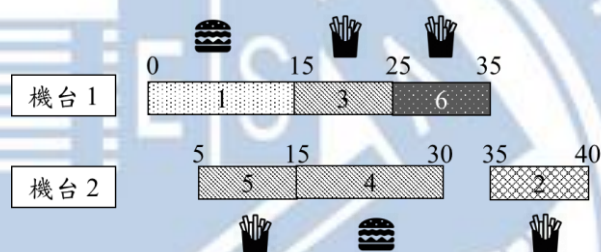
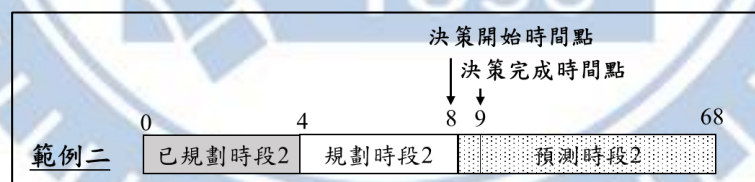


圖 3.35 範例一餐點種類與生產排程對照圖

◆ 範例二

考慮決策時段



平行機台與餐點資訊

平行 機台數量	餐點 種類	製作 時間	賞味 期限	餐點體積 (以便當盒大小為單位)	每分鐘餐廳最多 可製作數量
2	漢堡	15	20	3	2
	薯條	10	10	2	1

目前規劃時段決策時考慮的顧客訂單範例

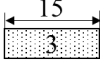
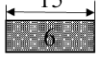
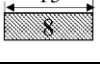
訂單 編號	點餐 時間	旅行 時間 (顧客 到餐廳)	餐點 數量	餐點 編號	餐點 狀態	餐點 種類	製作 時間	圖示
1 (原訂單 1)	0	0	1	1 (原餐點 1)	1	漢堡	7	
2 (原訂單 4)	4	8	1	2 (原餐點 5)	1	薯條	7	
3	6	0	1	3	2	漢堡	15	
4	25	0	1	4	5	薯條	10	
5 (原訂單 3)	3	3	2	5 (原餐點 3)	3	薯條	10	
				6 (原餐點 4)	3	漢堡	15	
6	7	5	2	7	4	薯條	10	
				8	4	漢堡	15	
7	25	5	1	9	6	漢堡	15	

表 3.12 範例二的餐點狀態

餐點狀態編號	餐點狀態
1	餐廳與平台已接受且正在製作的內用或外送餐點
2	餐廳與平台已接受但尚未開始製作的內用餐點
3	餐廳與平台已接受但尚未開始製作的外送餐點
4	待平台評估餐廳與平台是否接受的外送餐點
5	預測未來在餐廳可能接到的內用餐點
6	預測未來在餐廳可能接到的外送餐點

範例二待規劃的訂單共有 7 張。訂單 1 為在規劃時段 1 已規劃過的訂單 1 餐點 1，在規劃時段 2 時，由餐廳回報給平台，該訂單的餐點正在製作，並告知平台剩餘製作時間為 7 分鐘，至此，該餐點狀態轉變為狀態 1。同樣的，訂單 2 為規劃時段 1 所規劃的訂單 4 餐點 5，如圖 3.36 所示。

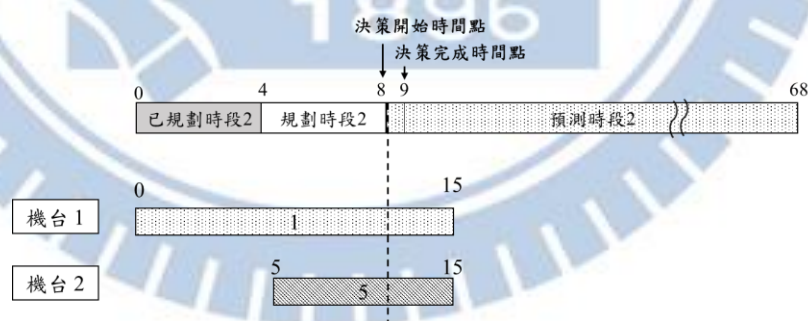


圖 3.36 規劃時段 2 決策開始時正在製作的餐點

訂單 3 與 4 為內用訂單，點餐時間分別為時間點 6 與 25，分別含有 1 個餐點，餐點狀態為已被餐廳與平台同時接受但尚未開始製作的內用

餐點以及預測未來在餐廳可能接到的內用餐點，訂單 1 的餐點為 1 份漢堡，訂單 2 的餐點為 1 份薯條。

訂單 5 為在規劃時段 1 已經規劃過的訂單 3 餐點 3 與 4，在規劃時段 2 時，由餐廳回報給平台，該訂單的餐點尚未開始製作，至此，餐點狀態轉變為狀態 3，如圖 3.37。

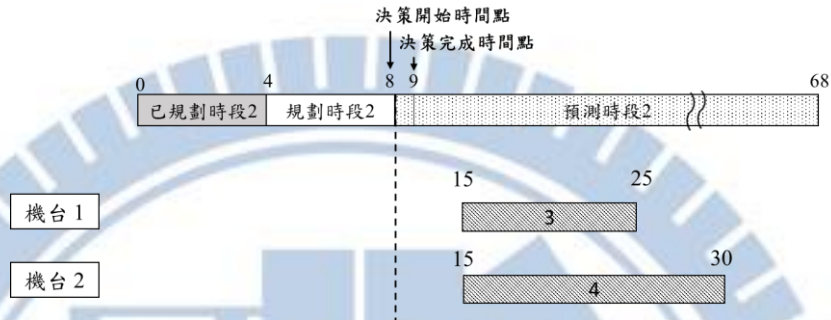


圖 3.37 規劃時段 2 決策開始時尚未製作的外送餐點

訂單 6 與 7 為外送訂單，點餐時間分別為時間點 7 與 25，分別含有 2 與 1 個餐點，訂單 6 的餐點狀態為待平台評估餐廳與平台是否接受的外送餐點，訂單 7 的餐點狀態為預測未來在餐廳可能接到的外送餐點，訂單 6 的餐點為 1 份薯條與 1 份漢堡，訂單 5 內含 1 份漢堡。

範例二的可行生產排程甘特圖如圖 3.38：

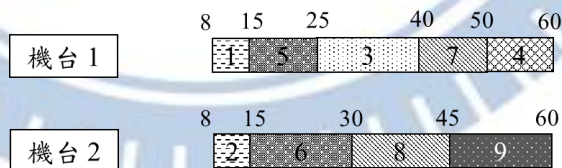


圖 3.38 範例二的可行生產排程甘特圖

接著，我們將會檢驗這樣的生產排程是否真的可行。這邊僅介紹訂單 1、2 與 5 的餐點 1、2、5 與 6 的檢測方式，也就是正在製作的餐點與已被餐廳與平台同時接受但尚未開始製作的外送餐點，因為其餘訂單餐點狀態的檢測方式皆與範例一相同。

表 3.13 範例二正在製作的餐點完成時間的可行性檢驗

訂單編號	點餐時間	服務時窗	旅行時間 (顧客到餐廳)	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間	賞味期限
1	0	[0, 40]	0	1	1	1	漢堡	7	20
2	4	[4, 64]	8	1	2	1	薯條	7	20

餐廳回報的 完成時間	平台估算的 完成時間	圖示
15	15	
15	15	

由於訂單 1 與 2 的餐點 1 與 2，在目前規劃時段決策時，餐廳正在製作且完成時間為第 15 分鐘，故目前規劃時段所決策的餐點完成時間須等於餐廳回報的預估完成時間。

表 3.14 範例二已被餐廳與平台同時接受但尚未開始製作外送餐點完成時間的可行性檢驗

訂單編號	點餐時間	服務時窗	旅行時間 (顧客到餐廳)	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間	賞味期限
5	3	[3, 63]	3	2	5	3	薯條	10	10
					6	3	漢堡	15	20

平台估算的 完成時間	規劃時段 1 平台決策的		圖示
	取餐時間	送達時間	
25	31	35	
30			

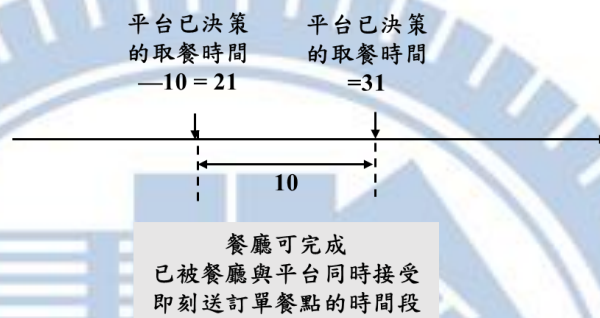


圖 3.39 餐廳可完成範例二餐點 5 與 6 的時間段

訂單 5 的餐點 5 和 6 為在目前規劃時段決策時，餐廳與平台已接受，但尚未完成的外送餐點。由上圖 3.41 的計算方式，便可以求出餐廳必須完成已接受但尚未開始製作的外送餐點時間段，並可發現餐點 5 與 6 的完成時間皆落在此區間中。

最後，必須檢查生產排程是否符合資源（產能）限制，該生產排程符合每分鐘最多製作 2 個漢堡、1 分薯條的限制為可行的生產排程。

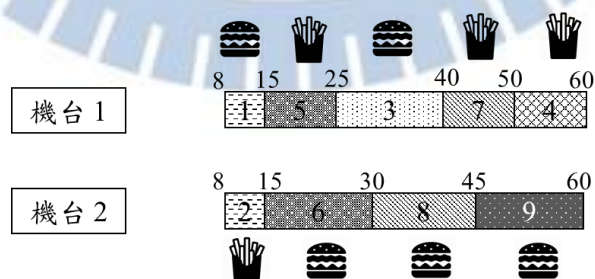


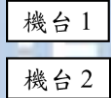
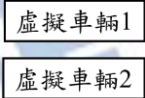
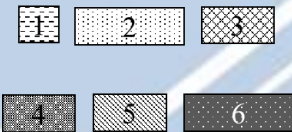
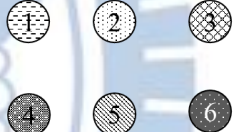
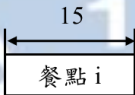
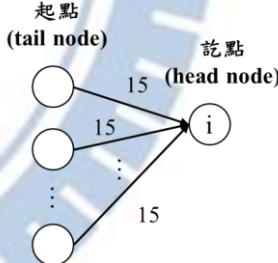
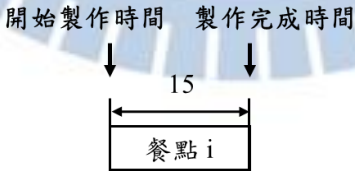
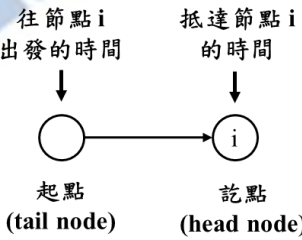
圖 3.40 範例二餐點種類與生產排程對照圖

■ 有時窗與資源限制的團隊尋寶問題

為了建立整合生產與配送數學模型，本研究參考 Kouki et al. (2007)與 Chang et al. (2016)的方法，將有資源限制的平行機台排程問題轉換成路線規劃問題，這類問題本研究稱之為有時窗與資源限制的團隊尋寶問題 (team orienteering problem with time window and resources constraint, TOPTWRC)。

轉換方式如表 3.15：

表 3.15 轉換成路徑規劃問題之方式

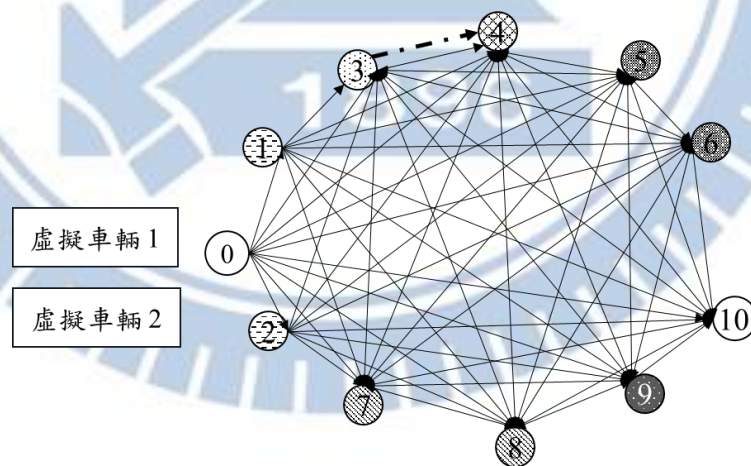
	有資源限制的平行機台排程問題	有時窗與資源限制的團隊尋寶問題
生產系統		
餐點		
餐點製作時間		
餐點開始製作與完成時間		

本研究將有資源限制的平行機台排程問題中的平行機台，轉換成虛擬車輛。每個餐點皆對應到一個節點，節點間以節線連結，表餐點間的可能製作

順序，除了內用餐點的製作順序有先到先服務的限制，以及由虛擬起點為起點的節線與以虛擬訖點的節線為單向之外，其餘的節點皆以雙向的節線連結。餐點的製作時間長度轉換為以該節點為訖點的節線的旅行時間。所有以某節點為訖點的節線，其旅行時間皆相同。每個餐點的開始製作時間與製作完成時間，表示往該節點出發時間與抵達該節點的時間。

如此一來，每一個餐廳的有資源限制的生產排程問題皆可以轉換成有時窗資源限制的團隊尋寶問題。這類問題為有時窗限制的團隊尋寶問題 (Team orienteering problem with time window, TOPTW) 之變形，一般而言團隊尋寶問題為旅行家問題 (Travel salesman problem, TSP) 的延伸。不同的是，每個節點有對應的服務時窗與收益，但旅行員不需要服務每個節點。整個問題的目標為追求整體的收益最大化。關於 TOPTW 問題與 TOPTWRC 問題之問題環境的差別可以參考本研究之 5.1 節。

我們以範例二為例，因為範例二包含本問題可能出現的餐點狀態，範例二對應的網路圖如下：



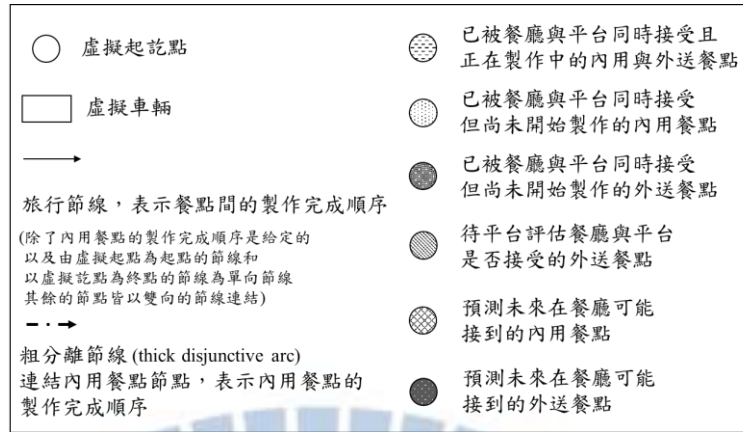
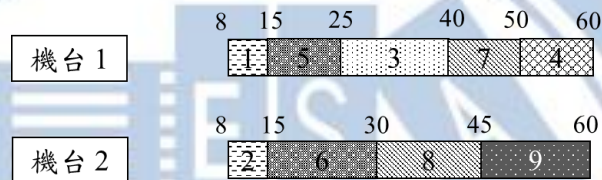


圖 3.41 範例二對應的 TOPTWRC 問題之網路圖

可行生產排程甘特圖



可行路線規劃圖

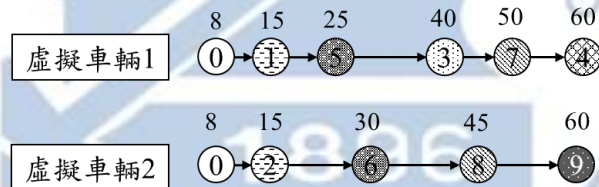


圖 3.42 範例二的可行生產排程甘特圖與可行路線規劃圖的對照圖

3.5 配送路線規劃問題

在平台決策外送訂單的取餐與送達時間時，平台必須知道外送員出發位置、車廂剩餘容量與訂單對應的取餐餐廳、顧客位置，來估算訂單的取餐與送達完成時間。接下來，將會介紹每個規劃時段決策時外送員的出發位置與車廂剩餘容量，並敘述不同狀態的餐點所對應的虛擬取餐節點與顧客位置節點狀態。

■ 外送員出發位置

在決策時段 1 決策時，所有外送員皆會從平台場站出發。為了簡化問題，本研究假設平台場站為平台辦公室所在地。圖 3.43 為平台在決策時段 1 所規劃的外送員 1 與 2 的路線範例。餐廳與顧客上的數字表示外送員到餐廳取餐的時間或抵達顧客的時間。

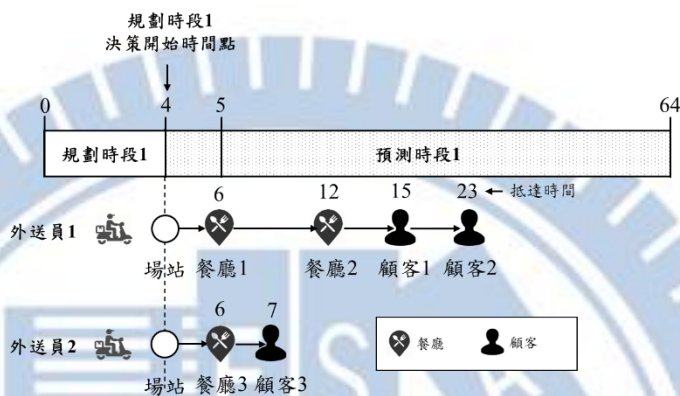


圖 3.43 平台在規劃時段 1 決策的外送員 1 與 2 的路線範例

承上圖的路線規劃範例，平台在規劃時段 2 決策時，會發現外送員 1 正在前往餐廳 2 的路上，而外送員 2 已經服務完他所負責的訂單，停留在顧客 2 的位置，等待下一個任務。為了符合實務並簡化問題，本研究假設外送員的配送過程不能中斷，也就是當外送員 1 正在前往餐廳 2 路上時，平台不能指派新的任務給他，只能等到外送員 1 抵達餐廳 2 時，外送員才能接受平台指派的新任務。如下圖 3.46 所示，我們會訂定外送員 1 在規劃時段 2 決策開始時的出發位置為餐廳 2，外送員 1 在決策時段 2 決策時的出發位置為顧客 3。

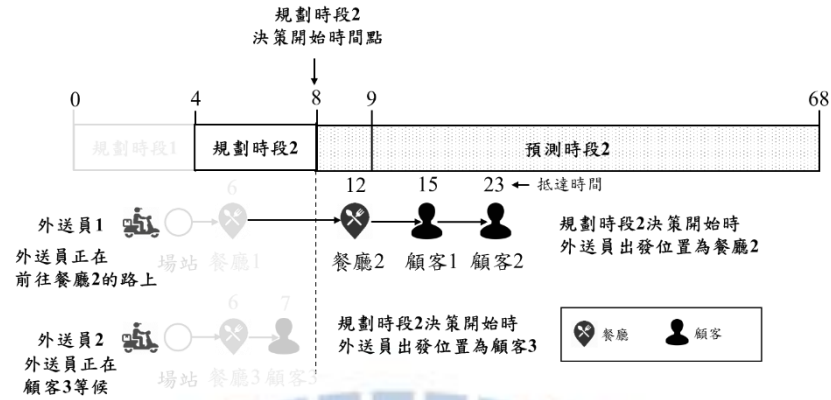


圖 3.44 外送員 1 與 2 在規劃時段 2 決策開始時的出發位置

■ 外送員車廂剩餘容量

接著，我們必須知道在每個規劃時段決策時，外送員車廂剩餘容量為何。經過訪談各家外送平台，外送員的車廂容量大約為 50 個便當盒大小。承襲圖 3.43 的範例，圖 3.45 為外送員所負責配送的訂單對應餐點體積。

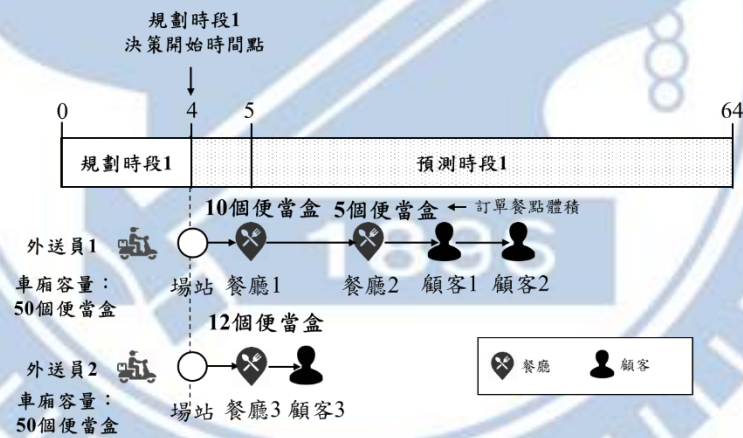


圖 3.45 平台在規劃時段 1 所規劃的外送員 1 與 2 的路線範例

此外，如同前述，外送員的配送過程不能中斷，所以外送員 1 在規劃時段 2 決策時的車廂剩餘容量為初始容量 50 個便當盒大小減去餐廳 1 與 2 的訂單餐點體積 15 個便當盒大小， $50 - 15 = 35$ ，車廂剩餘容量為 35 個便當盒大小。外送員 2 由於已經配送完他所被指派的訂單，所以車廂剩餘容量為初始車廂容量 50 個便當盒大小。

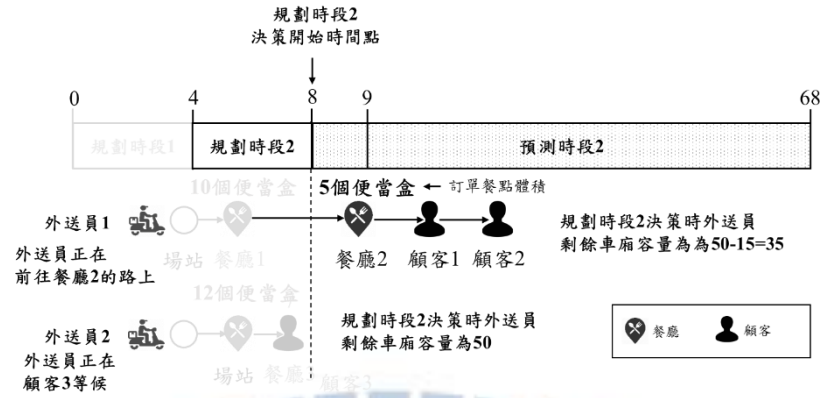


圖 3.46 外送員 1 與 2 在決策時段 2 決策時的車廂剩餘容量

此外，在計算外送員在的車廂剩餘容量時，不會將預測時段的訂單納入計算，因為預測時段的訂單只是協助平台決策，而非真實發生的訂單，以下圖為例。圖 3.47 為平台在某規劃時段所規劃的外送員路線圖，其中餐廳 1 與顧客 1 為在該規劃時段內接到的訂單，而餐廳 2 與顧客 2 為預測的訂單。在決定下一決策時段決策時的外送員車廂剩餘容量時，並不需考慮預測的訂單，所以外送員在下一決策時段決策時的外送員車廂剩餘容量為 $50 - 10 = 40$ 。



圖 3.47 在某決策時段規劃的有預測訂單之外送員路線圖

■ 虛擬取餐節點狀態

在本研究中，每張外送訂單皆會對應到一個虛擬取餐節點。該節點並非真實存在於餐廳中的一個位置，而是在建立模型時，表示該訂單的所有餐點必須先在餐廳完成，外送員才能去該餐廳取餐。承表 3.5，表 3.16 已規劃時段、目前規劃時段與預測時段所包含的訂單與取餐節點狀態為訂單狀態對應到的虛擬取餐節點狀態。我們將表 3.5 的已被餐廳與平台同時接受已完成但尚未送達的外送訂單，細分為已取餐與尚未取餐的訂單，以便介紹虛擬取餐節點狀態。

因為僅有尚未取餐的外送訂單有對應的虛擬取餐節點，所以我們僅介紹這類訂單的虛擬取餐節點狀態。

表 3.16 已規劃時段、目前規劃時段與預測時段所包含的訂單與取餐節點狀態

已規劃時段		目前規劃時段		預測時段	
已被餐廳與平台同時接受但尚未完成的		已被餐廳與平台同時接受但尚未完成的		預測未來可能出現	
內用訂單		內用訂單		內用訂單	
1	外送訂單(即刻送訂單)	1	外送訂單(預訂單)	3	外送訂單(即刻送訂單)
	外送訂單(預訂單)				外送訂單(預訂單)
已被餐廳與平台同時接受已完成但尚未取餐與送達的		待平台評估餐廳與平台是否接受的			
1	外送訂單(即刻送訂單)	2	外送訂單(即刻送訂單)		
	外送訂單(預訂單)				
已被餐廳與平台同時接受已完成已取餐但尚未送達的					
外送訂單(即刻送訂單)					
外送訂單(預訂單)					

表 3.17 虛擬取餐節點狀態之說明

虛擬取餐節點 狀態編號	虛擬取餐節點狀態
1	已被餐廳與平台同時接受但平台尚未取餐之訂單對應的虛擬取餐節點
2	待平台評估平台與餐廳是否接受之訂單對應的虛擬取餐節點
3	預測在未來可能出現的訂單對應的虛擬取餐節點

■ 顧客節點狀態

同樣的，每張外送訂單皆會對應到一個顧客節點，表示該訂單的所有餐點必須送達的顧客。表 3.18 為訂單狀態對應的顧客節點狀態。承表 3.5，下表 3.18 為訂單狀態對應到的虛擬取餐節點狀態。我們將表 3.5 的已被餐廳與平台同時接受已完成但尚未送達的外送訂單，細分為已取餐與尚未取餐的訂單，以便介紹顧客節點狀態。

表 3.18 已規劃時段、目前規劃時段與預測時段所包含的訂單與顧客節點狀態

已規劃時段		目前規劃時段		預測時段	
已被餐廳與平台 同時接受但尚未完成的		已被餐廳與平台 同時接受但尚未完成的		預測未來可能出現	
內用訂單		內用訂單		內用訂單	
2	外送訂單(即刻送訂單)	2	外送訂單(預訂單)	4	外送訂單 (即刻送訂單)
	外送訂單(預訂單)				外送訂單 (預訂單)
已被餐廳與平台同時接受 已完成但尚未取餐與送達的		待平台評估餐廳與平台 是否接受的			
2	外送訂單(即刻送訂單)	3	外送訂單 (即刻送訂單)		
	外送訂單(預訂單)				
已被餐廳與平台同時接受 已完成已取餐但尚未送達的					
1	外送訂單(即刻送訂單)				
	外送訂單(預訂單)				

表 3.19 顧客節點狀態之說明

顧客位置 節點狀態編號	顧客位置節點狀態
1	已被餐廳與平台同時接受且平台已取餐但尚未送達之訂單對應的顧客節點
2	已被餐廳與平台同時接受但平台尚未取餐與送達之訂單對應的顧客節點
3	待平台評估平台與餐廳是否接受之訂單對應的顧客節點
4	預測在未來可能出現的訂單對應的顧客節點

■ 多起點且有時窗限制與訂單可拒絕的收送貨問題

本研究的配送路線規劃問題類似於 Li et al. (2016) 所提出的考慮保留需求且有時窗限制的收送貨問題 (pickup and delivery problem with time windows, profits,

and reserved requests, PDPTWSR)。不同的地方是，在本研究中每個外送員在決策時皆有其對應的出發位置。此外，由於本問題考慮已被餐廳與平台同時接受且已完成已取餐但尚未送達的訂單，有顧客節點無對應的虛擬取餐節點的情形。所以我們將本研究的配送路線規劃問題稱之為多起點且有時窗限制與訂單可拒絕的收送貨問題 (multiple depots pickup and delivery problem with time window and selective requests, MDPDPTWSR)。關於兩個問題環境的差別可參考本研究之 5.1 節。本問題的示意圖如下，外送員有其對應的起點，並且在虛擬取餐節點取餐並配送至對應的顧客位置節點，此外，只要外送員收取的餐點體積沒有超過外送員車廂容量的上限，外送員可以取完多張訂單，再進行配送。

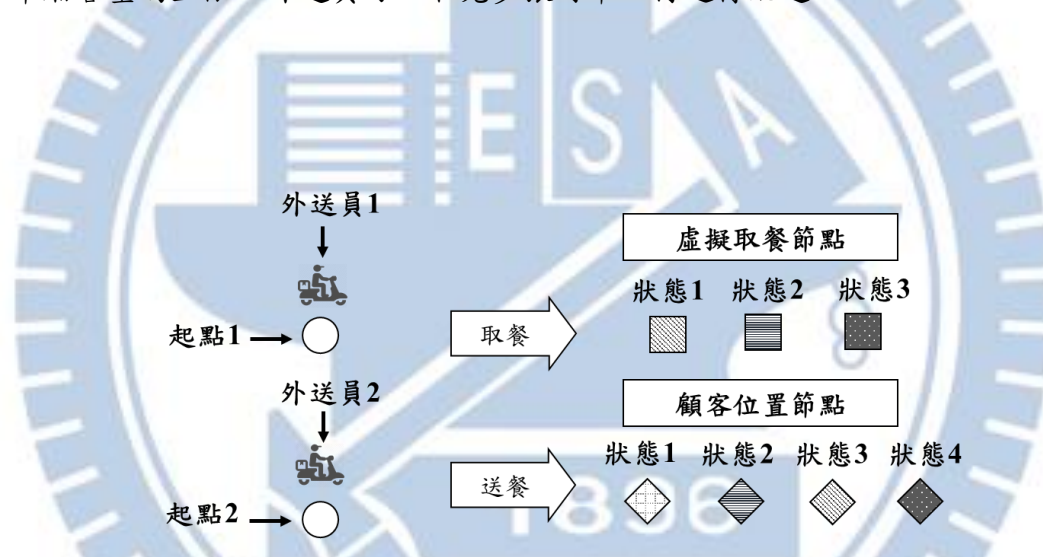


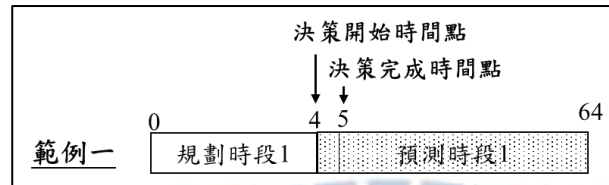
圖 3.48 MDPDPTWSR 問題之示意圖

■ 配送路線規劃問題之範例

承生產排程問題的顧客範例，配送路線規劃問題的範例同樣分為範例一、二與三。由於範例三的訂單狀態除了沒有預測時段的訂單外，其餘皆與範例二相同，下方範例僅介紹範例一與範例二。

◆ 範例一

考慮決策時段



平行機台與餐點資訊

平行機台數量	餐點種類	製作時間	賞味期限	餐點體積 (以便當盒大小為單位)	每分鐘餐廳最多可製作數量
2	漢堡	15	20	3	2
	薯條	10	10	2	1

外送員資訊

外送員	出發位置	最早可出發時間	剩餘車廂容量
1	場站	5	10
2	場站	5	10

目前規劃時段決策時考慮的顧客訂單範例

訂單編號	點餐時間	旅行時間 (顧客到餐廳)	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間	圖示
3	3	3	2	3	4	薯條	10	
				4	4	漢堡	15	
4	4	8	1	5	4	薯條	10	
5	15	5	1	6	6	薯條	10	

表 3.20 配送路線規劃範例一的餐點狀態

餐點狀態編號	餐點狀態
4	待平台評估餐廳與平台是否接受的外送餐點
6	預測未來在餐廳可能接到的外送餐點

目前規劃時段考慮的顧客訂單對應的虛擬取餐與顧客節點

訂單編號	對應的虛擬取餐節點		對應的顧客節點編號	
	編號	狀態	編號	狀態
3	2	2	5	3
4	3	2	6	3
5	4	3	7	4

表 3.21 配送路線規劃範例一的虛擬取餐與顧客節點狀態

虛擬取餐節點 狀態編號	虛擬取餐節點狀態
2	待平台評估平台與餐廳是否接受之訂單對應的虛擬取餐節點
3	預測未來在餐廳可能接到的外送訂單對應的虛擬取餐節點
顧客節點 狀態編號	顧客節點狀態
3	待平台評估平台與餐廳是否接受之訂單對應的顧客節點
4	預測未來在餐廳可能接到的外送訂單對應的顧客節點

訂單 3、4 與 5 為外送訂單，點餐時間分別為時間點 3、4 與 15，分別含有 1、2、1 個餐點，訂單 3 與 4 的餐點狀態為待平台評估餐廳與平台是否接受的外送餐點，訂單 5 的餐點狀態為預測未來在餐廳可能接到的外送餐點，訂單 3 內含 1 份薯條，訂單 4 的餐點為 1 份漢堡與薯條，訂單 5 內含 1 份漢堡。

訂單 3、4 與 5 分別對應到虛擬取餐節點 2、3 與 4，與顧客節點 5、6 與 7。

訂單 3 與 4 的虛擬取餐節點狀態為待平台評估平台與餐廳是否接受之訂單對應的虛擬取餐節點，顧客節點狀態為待平台評估平台與餐廳是否接受之訂單對應的顧客節點。

訂單 5 的虛擬取餐節點狀態為預測未來在餐廳可能接到的外送訂單對應的虛擬取餐節點，顧客節點狀態為預測未來餐廳可能接到的外送訂單對應的顧客節點。

在決策時段 1 決策時的配送網路圖如下：



圖 3.49 配送路線規劃問題範例一的配送網路圖

每個外送員皆對應一個虛擬起點，各節點間以節線連結，節點以節線連結表示節點間有可行的行進路線，成本表示節點間的空間距離。由虛擬起點為起點的節線和以虛擬迄點為終點的節線為單向節線。由同一

張訂單的虛擬取餐節點為起點，顧客節點為終點的節線為單向節線。其餘的節線皆為雙向節線。此外，範例一的可行配送路線圖如下：

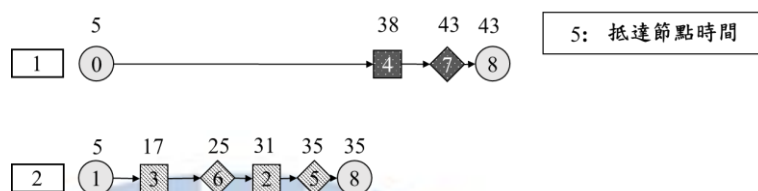


圖 3.50 範例一的可行配送路線圖

接著，我們將會檢驗這樣的配送路線是否真的可行。如 3.3 節所述，待平台評估餐廳與平台是否接受的訂單必須在餐點完成 10 分鐘內取餐，並且在對應的服務時窗內送達。

表 3.22 範例一的外送訂單配送可行性檢驗

訂單編號	點餐時間	服務時窗	旅行時間 (顧客到餐廳)	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間	賞味期限
3	3	[3, 63]	3	2	3	4	薯條	10	10
					4	4	漢堡	15	20
4	4	[4, 64]	8	1	5	4	薯條	10	10
5	15	[25, 85]	5	1	6	6	薯條	10	10

平台估算的完成時間	取餐時間	送達時間	對應的 虛擬取餐節點	對應的 顧客節點
25	31	35	2	5
30				
15	17	25	3	6
35	38	43	4	7

訂單 3 的取餐時間為 31，在餐點 3 完成時間 25 與餐點 4 完成時間 30 之後的 10 分鐘內。送達時間為 35，滿足服務時窗[3, 63]，並且符合餐點 3 賞味期限限制 ($25+10 \geq 35$) 與餐點 4 賞味期限限制 ($30+20 \geq 35$)。

訂單 4 的取餐時間為 16，在餐點 5 完成時間 15 之後的 10 分鐘內。送達時間為 25，滿足服務時窗[4, 64]，並且符合餐點 5 的賞味期限限制($25+10 \geq 25$)。

訂單 5 的取餐時間為 38，在餐點 6 完成時間 35 之後的 10 分鐘內。送達時間為 43，滿足服務時窗[25, 85]，並且符合餐點 6 的賞味期限限制 ($35+10 \geq 43$)。

並且該路線在取餐送餐的過程中並不會違反外送員的車廂容量限制，如下圖，下圖節點下方的底色白色的數字代表外送員服務完該虛擬取餐節點後，車廂體積會被該訂單餐點占用多少，而底色灰色的數字則代表外送員服務完該顧客節點後，車廂會因為訂單餐點送達釋放多少體積。

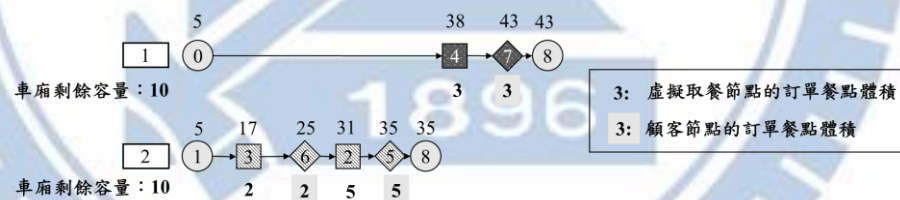
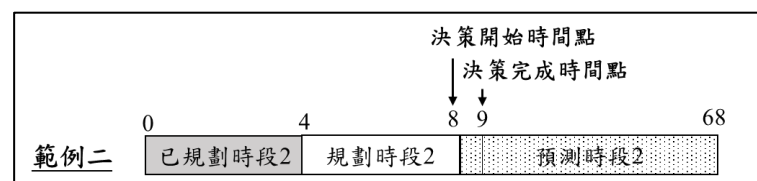


圖 3.51 範例一路線圖上虛擬取餐與顧客節點的訂單餐點體積

◆ 範例二

考慮決策時段



平行機台與餐點資訊

平行 機台數量	餐點 種類	製作 時間	賞味 期限	餐點體積 (以便當盒大小為單位)	每分鐘餐廳最多 可製作數量
2	漢堡	15	20	3	2
	薯條	10	10	2	1

外送員資訊

外送員	出發位置	最早可出發時間	剩餘車廂容量
1	規劃時段 1 的 節點 4 對應餐廳	38	10
2	規劃時段 1 的 節點 3 對應餐廳	16	8 (被訂單 4 的餐點占用)

目前規劃時段決策時考慮的顧客訂單範例

訂單 編號	點餐 時間	旅行 時間	餐點 數量	餐點 編號	餐點 狀態	餐點 種類	製作 時間	圖示
2 (原訂單 4)	4	8	1	2 (原餐點 5)	1	薯條	7	
5 (原訂單 3)	3	3	2	5 (原餐點 3)	3	薯條	10	
				6 (原餐點 4)	3	漢堡	15	
6	7	5	2	7	4	薯條	10	
				8	4	漢堡	15	
7	25	5	1	9	6	漢堡	15	

表 3.23 配送路線規劃範例二出現的餐點狀態

餐點狀態編號	餐點狀態
1	餐廳與平台已接受且正在製作的內用或外送餐點
3	餐廳與平台已接受但尚未開始製作的外送餐點
4	待平台評估餐廳與平台是否接受的外送餐點
6	預測未來在餐廳可能接到的外送餐點

顧客訂單對應的虛擬取餐與顧客節點

訂單編號	對應的虛擬取餐節點		對應的顧客節點編號	
	編號	狀態	編號	狀態
2	無	無	2	2
5	3	1	6	1
6	4	2	7	3
7	5	3	8	4

表 3.24 配送路線規劃範例二出現的虛擬取餐與顧客節點狀態

虛擬取餐節點 狀態編號	虛擬取餐節點狀態
1	已被餐廳與平台同時接受但平台尚未取餐之訂單對應的虛擬取餐節點
2	待平台評估平台與餐廳是否接受之訂單對應的虛擬取餐節點
3	預測未來在餐廳可能接到的外送訂單對應的虛擬取餐節點
顧客節點 狀態編號	顧客節點狀態
1	已被餐廳與平台同時接受且平台已取餐但尚未送達之訂單對應的顧客節點
2	餐廳與平台已接受但平台尚未取餐與送達之訂單對應的顧客節點

3	待平台評估平台與餐廳是否接受之訂單對應的顧客節點
4	預測未來在餐廳可能接到的外送訂單對應的顧客節點

根據範例一的規劃結果，可以發現在目前規劃時段（規劃時段2）決策完成時（時間點9），外送員1在前往節點4的路上，外送員2在前往節點3的路上。由於我們假設外送員的配送過程不能中斷，所以在規劃時段2決策時，外送員1的出發位置為節點4對應的餐廳，最早可出發時間為時間點38，剩餘車廂容量為10個便當盒大小，因為預測訂單並不會算入占用容量中。外送員2的出發位置為節點3對應的餐廳，最早可出發時間為時間點16，剩餘車廂容量為8個便當盒大小，被節點3的訂單餐點占用2個便當盒大小的空間。

在目前規劃時段決策時須考慮的外送訂單有訂單2、5、6和7。其中，訂單2與5分別是由上一規劃時段的訂單4與3轉變而來，訂單2對應到顧客節點2，訂單5對應到虛擬取餐節點3與顧客節點6。

由於外送員取餐與送餐過程不能中斷的假設，雖然在規劃時段2決策時間點時，負責訂單2的外送員實際上尚未取餐，但我們必須等到該外送員取完訂單2的餐點後才能指派下一個任務。所以訂單2被視為已取餐的訂單。因此，訂單2的顧客節點2狀態為已被餐廳與平台同時接受且平台已取餐但尚未送達之訂單對應的顧客節點。

訂單4的虛擬取餐節點狀態為已被餐廳與平台同時接受但平台尚未取餐之訂單對應的虛擬取餐節點，顧客節點狀態為已被餐廳與平台同時接受但平台尚未取餐與送達之訂單對應的顧客節點。

訂單6的虛擬取餐節點狀態為待平台評估平台與餐廳是否接受之訂單對應的虛擬取餐節點，顧客節點狀態為待平台評估平台與餐廳是否接受之訂單對應的顧客節點。

訂單 7 的虛擬取餐節點狀態為預測未來在餐廳可能接到的外送訂單對應的虛擬取餐節點，顧客節點狀態為預測未來餐廳可能接到的外送訂單對應的顧客節點。

在規劃時段 2 決策時的配送網路圖如下：



圖 3.52 配送路線規劃問題範例二的配送網路圖

此外，範例二的可行配送路線圖如下：

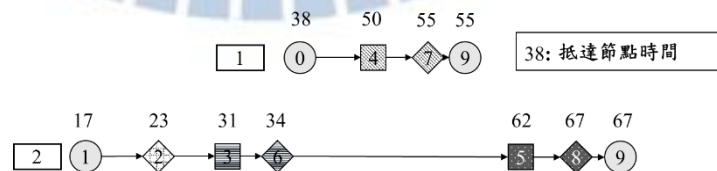


圖 3.53 範例二的可行配送路線圖

接著，我們必須檢驗該路線圖是否可行，下面僅列出訂單 2 與 5 的檢驗方式，因為訂單 6 與 7 的檢驗方式與範例一待平台評估餐廳與平台是否接受的訂單和預測餐廳未來接到的外送訂單相同。

表 3.25 範例二的外送訂單配送可行性檢驗

訂單編號	點餐時間	服務時窗	旅行時間	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間	賞味期限
2	4	[4, 64]	8	1	2	1	薯條	7	10
5	3	[3, 63]	3	2	5	3	薯條	10	10
					6	3	漢堡	15	20

平台估算的完成時間	平台已決策過的取餐時間	平台已決策過的送達時間	取餐時間	送達時間	對應的虛擬取餐節點	對應的顧客節點
15	17	25	-	23	-	
25	31	35	31	34		
30						

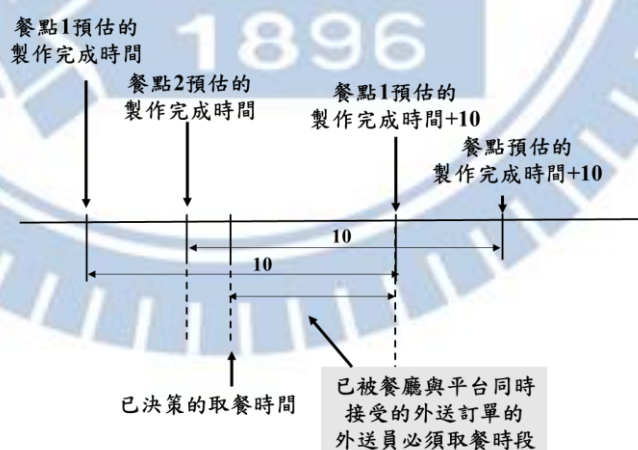


圖 3.54 已被餐廳與平台同時接受的外送訂單(即刻送訂單)的外送員必須取餐時段

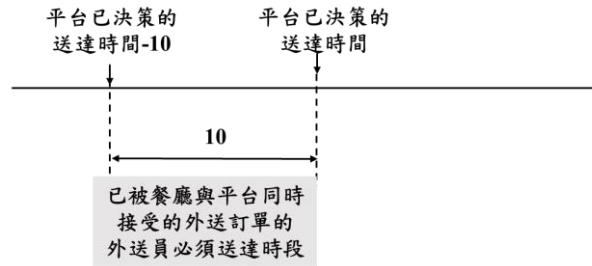


圖 3.55 已被餐廳與平台同時接受的外送訂單的外送員必須送達時段

根據上圖，我們可以推得訂單 2 在目前規劃時段決策時的送達時段為 $[15, 25]$ ，而送達時間 23 落在此時段中。訂單 5 的必須取餐時段必須晚於已決策的取餐時間 31 並且必須落在訂單餐點的預估完成時間與預估完成時間+10 的區間中 $[25, 35] \cap [30, 40] = [30, 35]$ ，取餐時間 31 符合此限制。訂單的必須送達時段為 $[25, 35]$ ，而送達時間 34 在此時段中。

並且該路線在取餐送餐的過程中並不會違反外送員的車廂容量限制，如下圖，下圖節點下方的底色白色的數字代表外送員服務完該虛擬取餐節點後，車廂體積會被該訂單餐點占用多少，而底色灰色的數字則代表外送員服務完該顧客節點後，車廂會因為訂單餐點送達釋放多少體積。

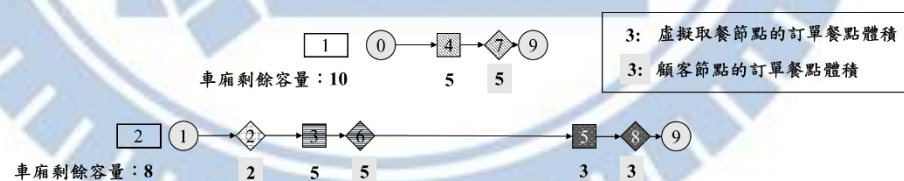
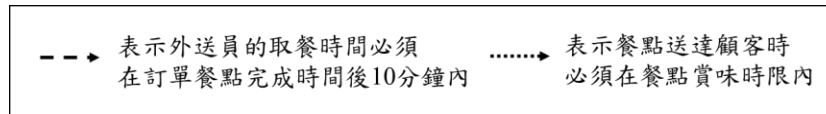


圖 3.56 範例二路線圖上虛擬取餐與顧客節點的訂單餐點體積

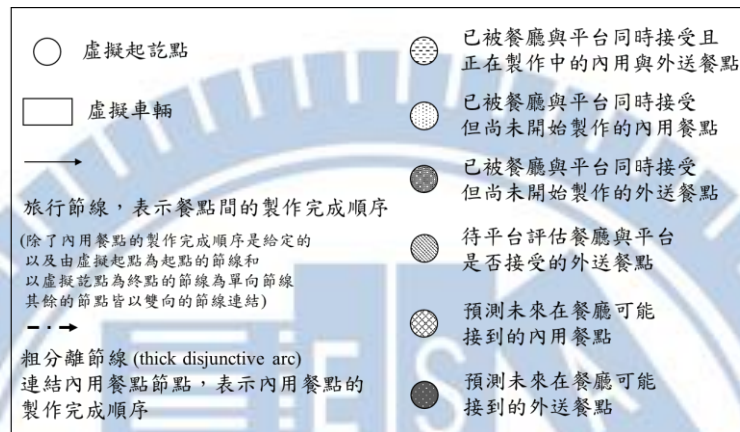
3.6 整合生產排程與配送路線規劃問題

本研究利用外送訂單餐點的取餐時間以及賞味期限限制來連接兩個問題，因為取餐時間與賞味期限，使得這兩個問題相互連結。我們兩種節線分別連接待平台評估平

整合生產排程問題與配送路線規劃問題之節線



生產網路圖例



配送網路圖例

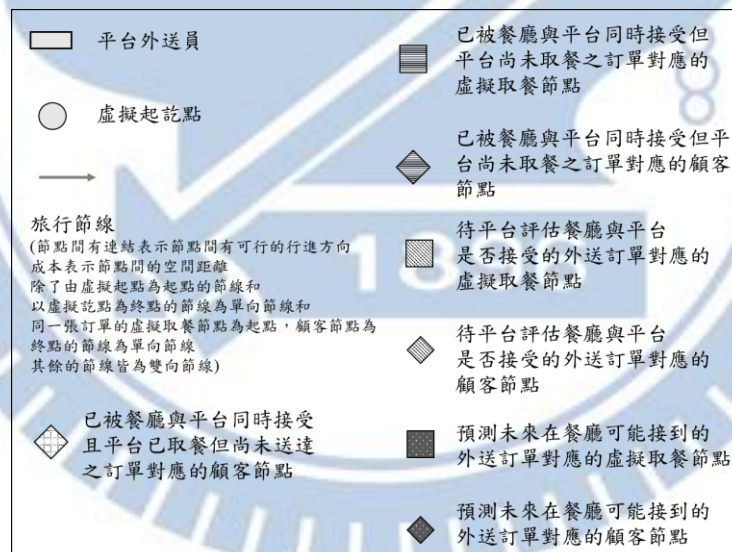


圖 3.58 整合生產排程問題與配送路線規劃問題之網路圖

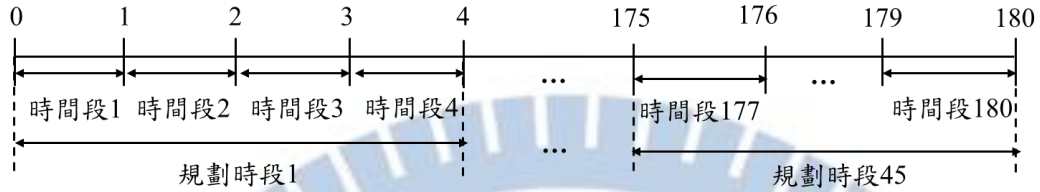
4 數學模型

在每個規劃時段決策時，皆會執行以下數學模型一次。為了敘述上的方便我們稱在規劃時段 h 的決策為第 h 決策。

集合

T ：規劃長度 (planning horizon) 中的時間段 (time interval) 集合， $T=\{1,...,180\}$ 。

H ：規劃時段集合，將規劃長度依每 4 分鐘劃分為一規劃時段， $H=\{1,...,45\}$



生產排程問題

R ：餐廳的集合。

Ψ_r ：餐廳 r 的餐點種類 (如：漢堡、薯條) 集合， $r \in R$ 。

B_{br}^h ：在第 h 決策開始時，餐廳 r 裡屬於餐點種類 b 的餐點集合， $b \in \Psi_r$ ， $r \in R$ ， $h \in H$ 。

O_r^h ：在第 h 決策開始時，已被餐廳 r 與平台同時接受且正在製作中的內用或外送餐點所對應的生產網路節點集合， $r \in R$ ， $h \in H$ 。

Γ_r^h ：在第 h 決策開始時，已被餐廳 r 與平台同時接受但尚未開始製作的內用餐點所對應的生產網路節點集合， $r \in R$ ， $h \in H$ 。

C_r^h ：在第 h 決策開始時，已被餐廳 r 與平台同時接受但尚未開始製作的外送餐點所對應的生產網路節點集合， $r \in R$ ， $h \in H$ 。

Π_r^h ：在第 h 決策開始時，待平台評估餐廳 r 是否接受的外送餐點所對應的生產網路節點集合， $r \in R$ ， $h \in H$ 。

X_r^h ：預測在第 h 決策開始後，餐廳 r 可能接到的預測內用餐點所對應的生產網路節點集合， $r \in R$ ， $h \in H$ 。

Θ_r^h ：預測在第 h 決策開始後，餐廳 r 可能接到的預測外送餐點所對應的生產網路節點集合， $r \in R$ ， $h \in H$ 。

編號	餐點狀態	集合
1	已被餐廳與平台同時接受且正在製作中的內用與外送餐點	O_r^h
2	已被餐廳與平台同時接受但尚未開始製作的內用餐點	Γ_r^h
3	已被餐廳與平台同時接受但尚未開始製作的外送餐點	C_r^h
4	待平台評估餐廳與平台是否接受的外送餐點	Π_r^h
5	預測未來在餐廳可能接到的內用餐點	X_r^h
6	預測未來在餐廳可能接到的外送餐點	Θ_r^h

Q_r^h ：為 O_r^h 、 Γ_r^h 、 C_r^h 和 X_r^h 的聯集， $Q_r^h = O_r^h \cup \Gamma_r^h \cup C_r^h \cup X_r^h$ ， $r \in R$ ， $h \in H$ 。

F_r^h ：為 Π_r^h 、 Θ_r^h 的聯集， $F_r^h = \Pi_r^h \cup \Theta_r^h$ ， $r \in R$ ， $h \in H$ 。

S_r^h ：為 Q_r^h 與 F_r^h 的聯集， $r \in R$ ， $h \in H$ ， $S_r^h = Q_r^h \cup F_r^h = \{1, 2, \dots, |S_r^h|\}$ 。

Z_r^h ： S_r^h 與虛擬起點 0 和虛擬訖點 $|S_r^h|+1$ 的聯集， $Z_r^h = S_r^h \cup \{0, |S_r^h|+1\}$ ， $r \in R$ ， $h \in H$ 。

K_r ：餐廳 r 由平行機台轉換而成的虛擬車輛集合， $r \in R$ 。餐廳的每一個平行機台皆會對應到一輛生產網路中的虛擬車輛。

J_r^h ：在第 h 決策開始時，餐廳 r 的節線集合， $r \in R$ ， $h \in H$ 。

L_r^h ：在第 h 決策開始時，連接集合 Γ_r^h 與 X_r^h 的先到先服務順序節線 (precedence arc) 集合， $r \in R$ ， $h \in H$ 。

配送路線規劃問題

V ：配送網路的外送員集合。

E ：配送網路虛擬起點集合，虛擬起點總數為外送員數目， $E = \{0, 1, \dots, |V|-1\}$ 。

I^h ：在第 h 決策開始時，已被餐廳與平台同時接受但平台尚未取餐之訂單對應的虛擬取餐節點集合， $h \in H$ 。

Ω^h ：在第 h 決策開始時，已被餐廳與平台同時接受但平台尚未取餐之訂單對應的顧客節點集合， $h \in H$ 。

G_v^h ：在第 h 決策開始時，已被餐廳與平台同時接受且某外送員 v 已經取餐但尚未送達之訂單所對應的顧客節點集合， $h \in H$ ， $v \in V$ 。

G^h ：在第 h 決策開始時，已被餐廳與平台同時接受且平台外送員已經取餐但尚未送達之訂單所對應的顧客節點集合， $h \in H$ ， $G^h = \bigcup_{v \in V} G_v^h$ 。

Φ^h ：在第 h 決策開始時，待平台評估平台與餐廳是否接受之訂單對應的虛擬取餐節點集合， $h \in H$ 。

Λ^h ：在第 h 決策開始時，待平台評估平台與餐廳是否接受之訂單對應的顧客節點集合， $h \in H$ 。

Ξ^h ：預測在第 h 決策開始後，可能出現的外送訂單對應的虛擬取餐節點集合， $h \in H$ 。

Υ^h ：預測在第 h 決策開始後，可能出現的外送訂單對應的顧客節點集合， $h \in H$ 。

編號	餐點狀態	集合
取餐節點狀態 1	已被餐廳與平台同時接受但平台尚未取餐與送達之訂單對應的虛擬取餐節點	I^h
顧客節點狀態 1	已被餐廳與平台同時接受但平台尚未取餐與送達之訂單對應的顧客節點	Ω^h

顧客節點 狀態 2	已被餐廳與平台同時接受且平台已取餐但尚未送達之訂單對應的顧客節點	G^h
取餐節點 狀態 2	待平台評估平台與餐廳是否接受之訂單對應的虛擬取餐節點	Φ^h
顧客節點 狀態 3	待平台評估平台與餐廳是否接受之訂單對應的顧客節點	Λ^h
取餐節點 狀態 3	預測在未來可能出現的訂單對應的虛擬取餐節點	Ξ^h
顧客節點 狀態 4	預測在未來可能出現的訂單對應的顧客節點	Υ^h

P^h ：在第 h 決策開始時，所有虛擬取餐節點集合， $h \in H$ ， $P^h = I^h \cup \Phi^h \cup \Xi^h$ 。

D^h ：在第 h 決策開始時，所有顧客節點集合， $h \in H$ ， $D^h = \Omega^h \cup G^h \cup \Lambda^h \cup \Upsilon^h$ 。

N^h ：在第 h 決策開始時，所有虛擬取餐節點及顧客節點集合， $h \in H$ ，

$N^h = P^h \cup D^h$ ， $N^h = \{|V|, \dots, |V| - 1 + |G^h| + 2|P^h|\}$ 。

M^h ：在第 h 決策開始時，包含配送網路虛擬起訖點、虛擬取餐節點及顧客節點集合， $M^h = N^h \cup E \cup \{|V| + |G^h| + 2|P^h|\} = \{0, 1, \dots, |V| + |G^h| + 2|P^h|\}$ 。

A^h ：在第 h 決策開始時，配送網路的節線集合， $h \in H$ 。

連結生產與配送問題

Y_r^h ：在第 h 決策開始時，連接集合 Π_r^h 與集合 Φ^h 和集合 Θ_r^h 與集合 Ξ^h 的順序節線 (precedence arc) 集合， $h \in H$ ， $r \in R$ 。表示同一顧客的外送訂單餐點，必須先匯集於虛擬取餐節點才能由外送員取餐配送。

W_r^h ：在第 h 決策開始時，連接集合 Π_r^h 與集合 Λ^h 和集合 Θ_r^h 與集合 Υ^h 的順序節線 (precedence arc) 集合， $h \in H$ ， $r \in R$ 。表示餐點抵達顧客時必須在其賞味期間內。

參數

m ：一個極大的正數。

生產排程問題

θ_{ir}^h ：餐廳接受某外送餐點所獲得的收益。如果平台在第 h 決策結束時，評估餐廳 r 可以接受餐廳 r 生產網路餐點 i ，餐廳將會得到相當於餐點售價的收入， $i \in F_r^h$ ， $h \in H$ ， $r \in R$ 。本研究將此值設為 0.5。

α_{ir}^h ：平台送達餐廳接受的某外送餐點所獲得的收益。如果平台在第 h 決策結束時，評估餐廳 r 與外送員皆可以接受餐廳 r 餐點 i ，平台將會得到餐廳所付給平台手續費之收入，通常為餐點售價之 30%， $i \in F_r^h$ ， $h \in H$ ， $r \in R$ 。

β_{ir}^h ：平台所必須補償餐廳的金額。如果平台在第 h 決策結束時，評估餐廳 r 可以接受餐廳 r 餐點 i ，但平台評估外送員無法送達餐點時，平台必須支付餐點售價扣除手續費的補償金額給餐廳， $i \in F_r^h$ ， $h \in H$ ， $r \in R$ 。

d_{ijr}^h ：餐廳製作某餐點的時間。在第 h 決策開始時，餐廳 r 製作餐點 j 的時間。亦即在餐廳 r 生產網路中節線 (i, j) 的旅行時間， i 為生產網路中非節點 j 的任一點，每一個 i 到節點 j 的旅行時間皆相同， $(i, j) \in J_r^h$ ， $h \in H$ ， $r \in R$ 。

$[a_{ir}^h, b_{ir}^h]$ ：某餐點的服務時窗。在第 h 決策開始時，餐廳 r 製作餐點 i 的完成時間窗。亦即在餐廳 r 生產網路中節點 i 的服務時窗， $i \in \Gamma_r^h \cup X_r^h \cup C_r^h \cup \Pi_r^h \cup \Theta_r^h$ ， $h \in H$ ， $r \in R$ 。

ε_{ir}^h ：某餐點的賞味期限。在第 h 決策開始時，餐廳 r 所製作的餐點 i 之賞味期限，亦即在餐廳 r 生產網路中節點 i 被服務後，必須在多少時間內服務對應的顧客節點， $i \in S_r^h$ ， $h \in H$ ， $r \in R$ 。

w ：平台所預測的外送訂單對應平台目前接到外送訂單的收益權重， $i \in \Theta_r^h$ ， $h \in H$ ， $r \in R$ ， $0 \leq w \leq 1$ 。

e_{br} ：餐廳在任一時間點製作屬於某餐點種類的餐點之數量上限。餐廳 r 在任一時間點最多可同時製作餐點種類 b 的餐點數量， $b \in \Psi_r$ ， $r \in R$ 。

配送路線規劃問題

t_{ij}^h ：旅行時間。在第 h 決策開始時，配送網路節線 (i, j) 的旅行時間， $(i, j) \in A^h$ ， $h \in H$ 。

q_i^h ：服務完某節點後訂單餐點占用的車廂容量。在第 h 決策開始時，服務完節點 i 所占用的外送員車廂容量，容量以 1 便當盒大小為單位， $i \in N^h$ $h \in H$ 。

ω_i^h ：服務完某節點後，因訂單餐點送達而釋放的车廂容量。在第 h 決策開始時，服務完節點 i 所釋放的外送員車廂容量，容量以 1 便當盒大小為單位， $i \in N^h$ $h \in H$ 。

$[f_i^h, g_i^h]$ ：某節點的服務時間窗。在第 h 決策開始時，節點 i 的服務時間窗， $i \in I^h \cup \Omega^h \cup G^h \cup \Xi^h \cup \Upsilon^h$ ， $h \in H$ 。

u_{iv}^h ：平台最早可以指派某外送員下一個工作的時間，由於外送員的配送過程不能中斷，所以平台必須等待外送員抵達某節點時，才能指派下個工作。此參數即表示外送員抵達該節點時間，該時間也同時為外送員在下一個規劃時段決策時，外送員最早可以由虛擬起點出發的時間。所以此參數為在第 h 決策開始時，外送員 v 最早可以由虛擬起點 i 出發的時間， $i \in E$ ， $v \in V$ ， $h \in H$ 。

δ_{iv}^h ：某節點的訂單餐點佔用的某外送員車廂容量。在 h 決策開始時，外送員 v 由虛擬起點 i 出發時，已被佔用的外送員 v 車廂容量， $i \in E$ ， $v \in V$ ， $h \in H$ 。

π ：運輸成本。外送員 v 行駛每分鐘所需的運輸成本。

c ：外送員的車廂容量。以 1 便當盒大小為單位。

決策變數

生產排程問題

x_{ijrk}^h ： $x_{ijrk}^h=1$ ，表示如果平台在第 h 決策的決策結果為，在餐廳 r 中某平行機台 k 上，餐點 i 完成後會接著製作餐點 j ，亦即虛擬車輛 k 經過餐廳 r 生產網路節線 $(i, j) \in J_r^h$ ；反之則等於 0， $k \in K_r$ ， $h \in H$ ， $r \in R$ 。

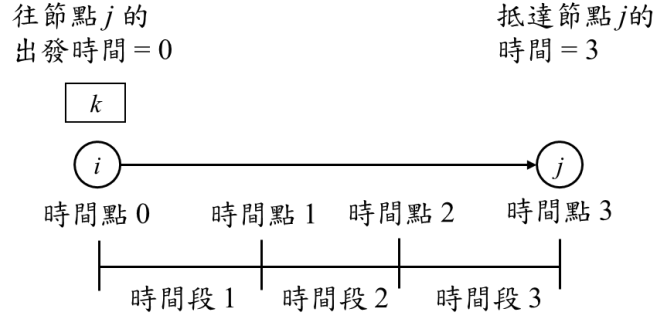
r_{ir}^h ： $r_{ir}^h=1$ ，表示如果平台在第 h 決策的評估結果為，餐廳 r 無法完成餐點 i ，亦即餐廳 r 生產網路的節點 i 無法被服務；反之則等於 0， $i \in F_r^h$ ， $h \in H$ ， $r \in R$ 。

s_{irk}^h ：平台在第 h 決策所評估之餐廳 r 完成餐點 i 的時間，亦即虛擬車輛 k 抵達餐廳 r 生產網路節點 i 的時間， $k \in K_r$ ， $i \in S_r^h \cup \{|S_r^h|+1\}$ ， $h \in H$ ， $r \in R$ 。

γ_{irkt}^h ： $\gamma_{irkt}^h=1$ ，表示如果平台在第 h 決策的評估結果為，餐廳 r 的餐點 i 在時間段 t 正在平行機台 k 上製作；反之則等於 0， $i \in S_r^h$ ， $h \in H$ ， $t \in T$ ， $k \in K_r$ ， $r \in R$ 。見下圖，若虛擬車輛 k 由虛擬起點 i 出發的時間點為 0，抵達節點 j 的時間點為 3。表示餐點 j 在平行機台 k 製作的時間為時間段 1 至時間段 3。

σ_{irkt}^h ： $\sigma_{irkt}^h=1$ ，表示如果平台在第 h 決策的評估結果為，餐廳 r 的餐點 i 在時間段 t 正開始在平行機台 k 上製作；反之則等於 0， $i \in S_r^h$ ， $h \in H$ ， $t \in T$ ， $k \in K_r$ ， $r \in R$ 。見下圖，平行機台 k 開始製作餐點 i 的時間段為時間段 1。

ϕ_{irkt}^h ： $\phi_{irkt}^h=1$ ，表示如果平台在第 h 決策的評估結果為，餐廳 r 的餐點 i 在時間段 t 正在平行機台 k 上製作完成，等於 1；反之則等於 0， $i \in S_r^h$ ， $h \in H$ ， $t \in T$ ， $k \in K_r$ ， $r \in R$ 。見下圖，平行機台 k 結束製作餐點 i 的時間段為時間段 3。



$$\sigma_{jrk1}^h = 1 \quad \phi_{jrk3}^h = 1$$

$$\gamma_{jrk1}^h = 1 \quad \gamma_{jrk2}^h = 1 \quad \gamma_{jrk3}^h = 1$$

配送路線規劃問題

y_{ijv}^h : $y_{ijv}^h = 1$, 表示如果平台在 h 決策的決策結果為, 外送員 v 經過配送網路節線 $(i, j) \in A^h$; 反之則等於 0, $v \in V$, $h \in H$ 。

o_i^h : $o_i^h = 1$, 表示如果平台在 h 決策的決策結果為, 平台無法服務節點 i , 等於 1; 反之則等於 0, $i \in N^h$, $h \in H$ 。

u_{iv}^h : 平台在 h 決策所決策的外送員 v 抵達節點 i 的時間, $i \in M^h \setminus E$, $v \in V$, $h \in H$ 。

δ_{iv}^h : 平台在 h 決策所決策的外送員 v 服務完節點 i 時, 餐點所占用的車廂容量 $v \in V$, $i \in M^h \setminus E$, $h \in H$ 。

連結生產與配送問題

λ_{ir}^h : $\lambda_{ir}^h = 1$, 表示如果平台在 h 決策之評估結果為, 餐廳 r 可以完成餐點 i 且平台能夠服務餐點; 其餘的情況則等於 0, $i \in F_r^h$, $h \in H$, $r \in R$ 。

ρ_{ir}^h : $\rho_{ir}^h = 1$, 表示如果平台在 h 決策之評估結果為, 平台無法服務餐點 i , 但餐廳 r 可以完成餐點 i 時, 等於 1; 其餘的情況則等於 0, $i \in F_r^h$, $h \in H$, $r \in R$ 。

在第 $h \in H$ 決策開始時，皆會執行下述數學模型一次。

Model

目標式

$$\begin{aligned} \max \quad & m^* \left[\sum_{r \in R} \sum_{i \in \Pi_r^h} \theta_{ir}^h * (1 - r_{ir}^h) + \sum_{r \in R} \sum_{i \in \Theta_r^h} w^* \theta_{ir}^h * (1 - r_{ir}^h) \right] \\ & + \sum_{r \in R} \sum_{i \in \Pi_r^h} [\alpha_{ir}^h * \lambda_{ir}^h - \beta_{ir}^h * \rho_{ir}^h] + \sum_{r \in R} \sum_{i \in \Theta_r^h} w^* [\alpha_{ir}^h * \lambda_{ir}^h - \beta_{ir}^h * \rho_{ir}^h] - \sum_{v \in V} \sum_{(i,j) \in A^h} \pi^* t_{ij}^h * y_{ijv}^h \end{aligned} \quad (1)$$

由於本研究中，平台站在餐廳的角度，評估目前接到的外送訂單餐點餐廳是否該接受，只要餐廳有能力完成該訂單，餐廳就會接受。此外，由於內用訂單餐點餐廳一定接受，並不需考慮餐廳的內用訂單收益，所以目標式的第一部分為餐廳的外送餐點收益。平台同時也站在平台的角度，評估該外送餐點平台是否該接受。所以目標式的第二部分為平台利潤。平台利潤為目前接到的外送訂單與預測訂單的利潤減去外送員運輸成本。外送訂單利潤計算方式為餐廳與平台同時接受的外送訂單餐點收益，減去餐廳接受但平台無法接受外送餐點時，平台必須支付給餐廳的補償金。

s.t.

生產網路限制式

生產網路中哪些節點可拒絕服務的相關限制式

$$\sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (i,j) \in J_r^h} \sum_{k \in K_r} x_{ijrk}^h = 1 \quad \forall i \in Q_r^h, r \in R \quad (2)$$

除了外送餐點所對應的生產網路節點外，其他生產網路節點一定要服務。

$$\sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (i,j) \in J_r^h} \sum_{k \in K_r} x_{ijrk}^h + r_{ir}^h = 1 \quad \forall i \in F_r^h, r \in R \quad (3)$$

僅有外送餐點所對應的生產網路節點能夠拒絕服務

流量守恆的相關限制式

$$\sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (0,j) \in J_r^h} x_{0jrk}^h = 1 \quad \forall k \in K_r, r \in R \quad (4)$$

生產網路的虛擬車輛皆必須由虛擬起點出發

$$\sum_{i \in S_r^h \cup \{0\}; (i,s) \in J_r^h} x_{isrk}^h - \sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (s,j) \in J_r^h} x_{sjrk}^h = 0 \quad \forall s \in S_r^h, k \in K_r, r \in R \quad (5)$$

除了虛擬起點與迄點外，虛擬車輛抵達某點後必會由該點出發。

$$\sum_{i \in S_r^h \cup \{0\}; (i,|S_r^h|+1) \in J_r^h} x_{i(|S_r^h|+1)rk}^h = 1 \quad \forall k \in K_r, r \in R \quad (6)$$

生產網路的虛擬車輛皆必須抵達虛擬迄點出發

虛擬車輛抵達節點時間的相關限制式

$$s_{irk}^h = h * 4 + \sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (i,j) \in J_r^h} x_{ijrk}^h * d_{0ir}^h \quad \forall i \in O_r^h, k \in K_r, r \in R \quad (7)$$

已被餐廳 r 與平台同時接受且正在製作中的內用或外送餐點所對應的生產網路節點的虛擬車輛抵達時間，必須等於目前規劃時段結束時間點加上餐廳回報平台的該節點對應餐點剩餘製作時間。

$$\sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (i,j) \in J_r^h} x_{ijrk}^h * b_{ir}^h \geq s_{irk}^h \geq \sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (i,j) \in J_r^h} x_{ijrk}^h * a_{ir}^h \quad \forall i \in S_r^h \setminus O_r^h, k \in K_r, r \in R \quad (8)$$

除了已被餐廳 r 與平台同時接受且正在製作中的內用或外送餐點所對應的生產網路節點外，其餘節點的抵達時間必須落在該節點的服務時窗內。

$$s_{jrk}^h \geq s_{irk}^h + d_{ijr}^h - m * (1 - x_{ijrk}^h) \quad \forall (i,j) \in J_r^h, k \in K_r, r \in R \quad (9)$$

若虛擬車輛經過節線 (i,j) 服務節點 i 與 j ，抵達節點 j 的時間必須大於等於抵達節點 i 的時間加節線 (i,j) 的旅行時間。

$$m * (1 - \sum_{s \in S_r^h \cup \{|S_r^h|+1\}} x_{jsrk}^h) + s_{jrk}^h \geq s_{ilr}^h - m * (1 - \sum_{s \in S_r^h \cup \{|S_r^h|+1\}} x_{isrk}^h) \quad \forall (i,j) \in L_r^h, k \in K_r, l \in K_r, r \in R \quad (10)$$

內用餐點所對應的節點的抵達時間，先點餐餐點所對應節點的抵達時間必須小於等於後點餐餐點所對應節點的抵達時間。

餐點屬於同一訂單相關限制式

$$r_{ir}^h = r_{sr}^h \quad \forall (i,j) \in Y_r^h, (s,j) \in Y_r^h, j \in P^h, i \neq s, r \in R \quad (11)$$

同一張訂單的外送餐點所對應的節點必須全部接受或全部拒絕。

生產網路與平行機台系統的相關限制式

$$s_{irk}^h - (d_{0ir} - 1) - m^* (1 - \sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (j,i) \in J_r^h} x_{jirk}^h) \leq \sum_{t \in T} t^* \sigma_{irkt}^h \leq s_{irk}^h - (d_{0ir} - 1) + m^* (1 - \sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (j,i) \in J_r^h} x_{jirk}^h) \quad \forall i \in S_r^h, k \in K_r, r \in R \quad (12)$$

開始往某生產網路節點的出發時間等於該節點對應餐點的開始製作時間段

$$s_{irk}^h = \sum_{t \in T} t^* \phi_{irkt}^h \quad \forall i \in S_r^h, k \in K_r, r \in R \quad (13)$$

抵達某生產網路節點的時間等於該節點對應餐點的製作完成時間段

$$\sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (j,i) \in J_r^h} x_{jirk}^h = \sum_{t \in T} \sigma_{irkt}^h \quad \forall i \in S_r^h, k \in K_r, r \in R \quad (14)$$

若某餐點沒有被服務，必無對應的開始製作時間段；反之亦然。

$$\sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (j,i) \in J_r^h} x_{jirk}^h = \sum_{t \in T} \phi_{irkt}^h \quad \forall i \in S_r^h, k \in K_r, r \in R \quad (15)$$

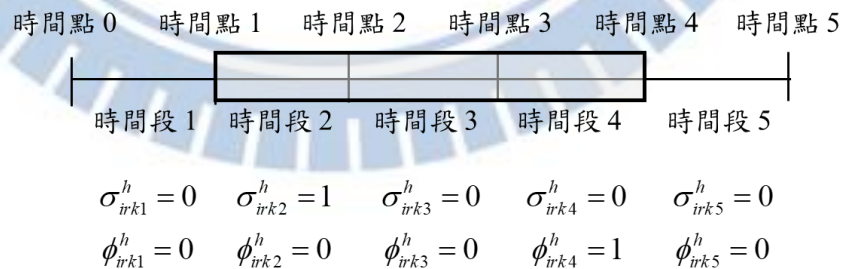
若某餐點沒有被服務，必無對應的製作完成時間段；反之亦然。

$$m^* \sum_{j \in S_r^h \cup \{|S_r^h|+1\}; (j,i) \in J_r^h} x_{jirk}^h \geq \sum_{t \in T} \gamma_{irkt}^h \quad \forall i \in S_r^h, k \in K_r, r \in R \quad (16)$$

若某餐點沒有被服務，必無對應的製作時間段。

$$\gamma_{irkt}^h - \gamma_{irk,t-1}^h = \sigma_{irkt}^h - \phi_{irk,t-1}^h \quad \forall i \in S_r^h, k \in K_r, r \in R, t \in T, t: \{t \geq 2\} \quad (17)$$

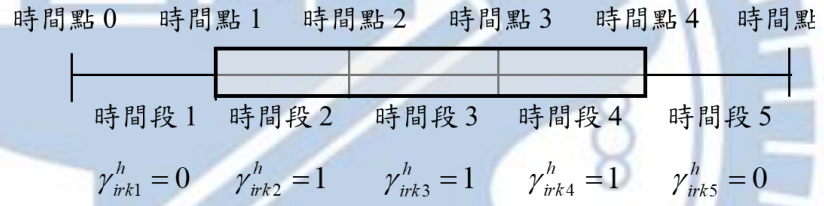
計算某餐點正在製作時間段的限制式。詳見下圖的範例：



假設 $T=\{1,2,3,4,5\}$ ，且餐點 i 在時間段 2 正在餐廳 r 平行機台 k 開始製作，在時間段 4 完成，其對應的 σ_{irkt}^h 和 ϕ_{irkt}^h 如上。利用限制式(17)，便可計算出該時間段餐點 i 是否正在製作的 γ_{irkt}^h 值。

T	限制式(17)	γ_{irkt}^h
t=2	$\gamma_{irk2}^h - \gamma_{irk1}^h = \sigma_{irk2}^h - \phi_{irk1}^h$ $\rightarrow \gamma_{irk2}^h - \gamma_{irk1}^h = 1 - 0$	$\gamma_{irk2}^h = 1$ 、 $\gamma_{irk1}^h = 0$
t=3	$\gamma_{irk3}^h - \gamma_{irk2}^h = \sigma_{irk3}^h - \phi_{irk2}^h$ $\rightarrow \gamma_{irk3}^h - 1 = 0 - 0$	$\gamma_{irk3}^h = 1$
t=4	$\gamma_{irk4}^h - \gamma_{irk3}^h = \sigma_{irk4}^h - \phi_{irk3}^h$ $\rightarrow \gamma_{irk4}^h - 1 = 0 - 0$	$\gamma_{irk4}^h = 1$
t=5	$\gamma_{irk5}^h - \gamma_{irk4}^h = \sigma_{irk5}^h - \phi_{irk4}^h$ $\rightarrow \gamma_{irk5}^h - 1 = 0 - 1$	$\gamma_{irk5}^h = 0$

所以，計算出的 γ_{irkt}^h 值如下圖：



產能相關限制式

$$\sum_{i \in B_{br}} \sum_{k \in K_r} \gamma_{irkt}^h \leq e_{rb} \quad \forall t \in T, b \in \Psi_r, r \in R \quad (18)$$

在每個時間段，某餐點種類在所有平行機台上同時製作的數量不能大於對應的產能上限。

配送網路限制式

配送網路中哪些節點可拒絕服務的相關限制式

$$\sum_{j \in N^h \cup \{|V|+|G^h|+2|P^h|\}; (i,j) \in A^h} y_{ijv}^h + o_i^h = 1 \quad \forall i \in N^h \setminus \{G^h \cup I^h \cup \Omega^h\} \quad (19)$$

除了已被餐廳與平台同時接受的訂單所對應的節點外，其餘的節點平台可以選擇服務或拒絕。

$$\sum_{j \in N^h \cup \{|V|+|G^h|+2|P^h|\}; (i,j) \in A^h} y_{ijv}^h = 1 \quad \forall i \in G_v^h, v \in V \quad (20)$$

已被餐廳與平台同時接受且某外送員 v 已經取餐但尚未送達之訂單所對應的顧客節點，必須由該外送員 v 服務。

$$\sum_{j \in N^h \cup \{|V|+|G^h|+2|P^h|\}; (i,j) \in A^h} \sum_{v \in V} y_{ijv}^h = 1 \quad \forall i \in I^h \cup \Omega^h \quad (21)$$

已被餐廳與平台同時接受但平台尚未取餐之訂單對應的虛擬取餐與顧客節點一定要服務。

流量守恆的相關限制式

$$\sum_{j \in P^h \cup G_v^h \cup \{|V|+|G^h|+2|P^h|\}} y_{(v-1)jv}^h = 1 \quad \forall v \in V \quad (22)$$

外送員一定由其對應的虛擬起點出發

$$\sum_{i \in N^h \cup E; (i,j) \in A^h} y_{isv}^h - \sum_{i \in N^h \cup \{|V|+|G^h|+2|P^h|\}; (i,j) \in A^h} y_{siv}^h = 0 \quad \forall s \in N^h, v \in V \quad (23)$$

除了虛擬起點與迄點外，外送員抵達某點後必會由該點出發。

$$\sum_{i \in D^h \cup E} y_{i|V|+|G^h|+2|P^h|v}^h = 1 \quad \forall v \in V \quad (24)$$

外送員一定會抵達虛擬迄點。

虛擬取餐節點與顧客節點的服務順序限制式

$$\sum_{j \in N^h} y_{ijv}^h - \sum_{j \in M^h} y_{|P^h|+i,jv}^h = 0 \quad \forall i \in P^h, v \in V \quad (25)$$

外送員若取餐則一定要送達。

$$u_{iv}^h \leq u_{|P^h|+i,v}^h \quad \forall i \in P^h, v \in V \quad (26)$$

外送員必須先取餐才能送餐。

外送員車廂容量的相關限制式

$$\delta_{iv}^h + q_j^h - \omega_j^h + m^*(1 - y_{ijv}^h) \geq \delta_{jv}^h \geq \delta_{iv}^h + q_j^h - \omega_j^h - m^*(1 - y_{ijv}^h) \quad \forall (i,j) \in A^h, v \in V \quad (27)$$

外送員服務完某節點後的車廂容量變化。

$$\delta_{iv}^h \leq \sum_{j \in N^h \cup \{|V|+|G^h|+2|P^h|\}; (i,j) \in A^h} c * y_{ijv}^h \quad \forall i \in N^h \cup (v-1), v \in V \quad (28)$$

外送員車廂容量限制。

外送員抵達節點時間的相關限制式

$$u_{jv}^h + m^*(1 - y_{ijv}^h) \geq u_{iv}^h + t_{ij} \quad \forall (i, j) \in A^h, v \in V \quad (29)$$

若外送員經過節線 (i, j) 服務節點 i 與 j ，抵達節點 j 的時間必須大於等於抵達節點 i 的時間加節線 (i, j) 的旅行時間。

$$\sum_{j \in M^h: (i, j) \in A^h} y_{ijv}^h * f_i^h \leq u_{iv}^h \leq \sum_{j \in M^h: (i, j) \in A^h} y_{ijv}^h * g_i^h \quad \forall i \in I^h \cup D^h, v \in V \quad (30)$$

外送員抵達已被餐廳與平台同時接受但平台尚未取餐之訂單對應的虛擬取餐節點以及所有顧客節點時必須滿足其服務時窗。

連結生產與配送網路之限制式

餐廳與平台接受餐點的情境

$$o_j^h \leq 1 - \lambda_{ir}^h \quad \forall (i, j) \in Y_r^h, r \in R \quad (31)$$

o_j^h	λ_{ir}^h
1	0
0	1 或 0

若餐點 i 對應的虛擬取餐節點 j 平台無法服務($o_j^h = 1$)，餐廳可以完成且平台可以服務該餐點 i 的情況就不會發生($\lambda_{ir}^h = 0$)。反之，若餐點 i 對應的虛擬取餐節點 j 被平台服務($o_j^h = 0$)，餐廳可以完成且平台可以服務該餐點 i 的情況才有可能發生($\lambda_{ir}^h = 1$ 或 $\lambda_{ir}^h = 0$)。

$$r_{ir}^h \leq 1 - \lambda_{ir}^h \quad \forall i \in F_r^h, r \in R \quad (32)$$

r_{ir}^h	λ_{ir}^h
1	0
0	1 或 0

若餐廳無法完成餐點 i ($r_{ir}^h=1$)，餐廳可以完成且平台可以服務該餐點 i 的情況就不會發生 ($\lambda_{ir}^h=0$)。反之，若餐廳可以完成餐點 i ($r_{ir}^h=0$)，餐廳可以完成且平台可以服務該餐點 i 的情況才有可能發生 ($\lambda_{ir}^h=1$ 或 $\lambda_{ir}^h=0$)。

$$o_j^h + r_{ir}^h \geq (1 - \lambda_{ir}^h) \quad \forall (i, j) \in Y_r^h, r \in R \quad (33)$$

o_j^h	r_{ir}^h	λ_{ir}^h
1	1	0
1	0	0
0	1	0
0	0	1

搭配限制式 (32) 與 (33)，若餐點 i 對應的虛擬取餐節點 j 平台無法服務 ($o_j^h=1$) 或餐廳無法完成餐點 i ($r_{ir}^h=1$) 時，餐廳可以完成且平台可以服務該餐點 i 的情況就不會發生 ($\lambda_{ir}^h=0$)。唯有餐點 i 對應的虛擬取餐節點 j 被平台服務 ($o_j^h=0$) 且餐點 i 被餐廳完成 ($r_{ir}^h=0$)，才是餐廳可以完成且平台可以服務該餐點 i 的情況 ($\lambda_{ir}^h=1$)。

$$o_j^h \geq \rho_{ir}^h \quad \forall (i, j) \in Y_r^h, r \in R \quad (34)$$

o_j^h	ρ_{ir}^h
0	0
1	1 或 0

若餐點 i 對應的虛擬取餐節點 j 被平台服務 ($o_j^h = 0$)，餐廳可以完成但平台無法服務餐點的情況就不會發生 ($\rho_{ir}^h = 0$)。反之，若餐點 i 對應的虛擬取餐節點 j 平台無法服務 ($o_j^h = 1$)，餐廳可以完成但平台無法服務該餐點 i 的情況才可能發生 ($\rho_{ir}^h = 1$ 或 $\rho_{ir}^h = 0$)。

$$r_{ir}^h \leq 1 - \rho_{ir}^h \quad \forall i \in F_r^h, r \in R \quad (35)$$

r_{ir}^h	ρ_{ir}^h
1	0
0	1 或 0

若餐廳無法完成餐點 i ($r_{ir}^h = 1$)，餐廳可以完成但平台無法服務餐點 i 的情況就不會發生 ($\rho_{ir}^h = 0$)。反之，若餐廳可以完成餐點 i ($r_{ir}^h = 0$)，餐廳可以完成但平台無法服務該餐點 i 的情況才可能發生 ($\rho_{ir}^h = 1$ 或 $\rho_{ir}^h = 0$)。

$$o_j^h - r_{ir}^h \leq \rho_{ir}^h \quad \forall (i, j) \in Y_r^h, r \in R \quad (36)$$

o_j^h	r_{ir}^h	λ_{ir}^h
1	1	0
0	1	0
0	0	0
1	0	1

搭配限制式 (35) 與 (36) ，若餐點 i 對應的虛擬取餐節點 j 平台可以服務 ($o_j^h = 0$) 或餐廳無法完成餐點 i ($r_{ir}^h = 1$) 時，餐廳可以完成但平台無法服務該餐點 i 的情況就不會發生 ($\rho_{ir}^h = 0$) 。唯有餐點 i 對應的虛擬取餐節點 j 平台無法服務 ($o_j^h = 1$) 且餐廳可以完成餐點 i ，才是餐廳可以完成但平台無法服務餐點 i 的情況 ($\rho_{ir}^h = 1$) 。

平台外送員取餐時間

$$\begin{aligned} s_{irk}^h + 10 + m^*(1 - \sum_{s \in S_r^h \cup \{|S_r^h|+1\}; (i,s) \in J_r^h} x_{irsk}^h) &\geq m^*(1 - \sum_{s \in N^h \cup \{|V|+|G^h|+2|P^h|\}; (j,s) \in A^h} y_{jsv}^h) + u_{jv}^h \\ &\geq s_{irk}^h - m^*(1 - \sum_{s \in S_r^h \cup \{|S_r^h|+1\}; (i,s) \in J_r^h} x_{irsk}^h) \end{aligned}$$

$$\forall (i, j) \in Y_r^h, v \in V, k \in K_r, r \in R \quad (37)$$

表示外送員必須在外送訂單的餐點完成後的 10 分鐘內取餐。

餐點賞味期限

$$\begin{aligned} -m^*(1 - \sum_{s \in N^h \cup \{|V|+|G^h|+2|P^h|\}; (j,s) \in A^h} y_{jsv}^h) - m^*(1 - \sum_{s \in S_r^h \cup \{|S_r^h|+1\}; (i,s) \in J_r^h} x_{irsk}^h) + u_{jv}^h - s_{irk}^h &\leq \varepsilon_{ir}^h \end{aligned}$$

$$\forall (i, j) \in W_r^h, v \in V, k \in K_r, r \in R \quad (38)$$

表示外送訂單餐點送達顧客節點時，必須在其賞味期限內。

變數範圍限制式

$$x_{ijrk}^h \in \{0,1\} \quad \forall (i, j) \in J_r^h, k \in K_r, r \in R \quad (39)$$

$$r_{ir}^h \in \{0,1\} \quad \forall i \in F_r^h, h \in H, r \in R \quad (40)$$

$$s_{irk}^h \geq 0 \quad \forall i \in S_r^h \cup \{|S_r^h|+1\}, t \in T, k \in K_r, r \in R \quad (41)$$

$$\sigma_{irkt}^h \in \{0,1\} \quad \forall i \in S_r^h, t \in T, k \in K_r, r \in R \quad (42)$$

$$\gamma_{irkt}^h \in \{0,1\} \quad \forall i \in S_r^h, t \in T, k \in K_r, r \in R \quad (43)$$

$$\phi_{irkt}^h \in \{0,1\} \quad \forall i \in S_r^h, t \in T, k \in K_r, r \in R \quad (44)$$

$$y_{ijv}^h \in \{0,1\} \quad \forall (i, j) \in A^h, v \in V \quad (45)$$

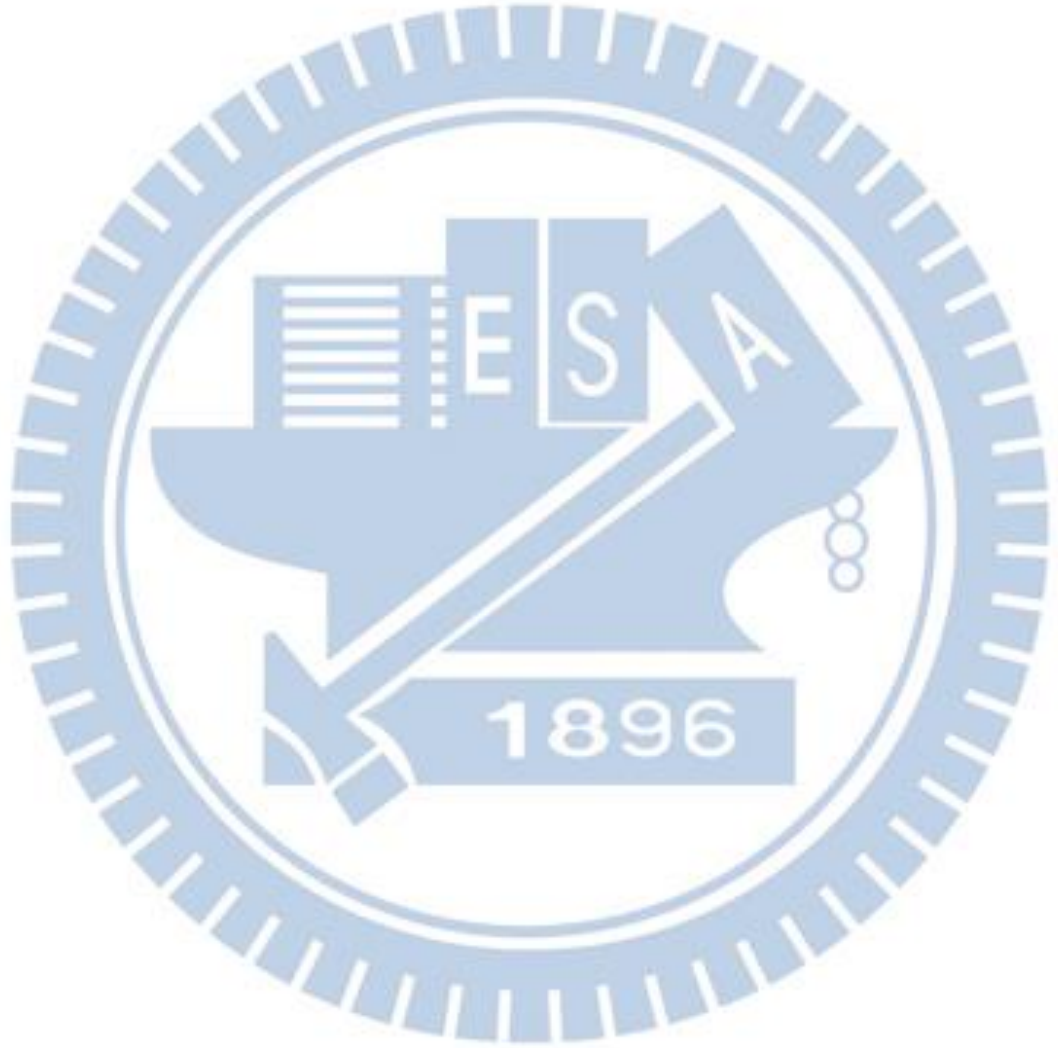
$$o_i^h \in \{0,1\} \quad \forall i \in N^h \quad (46)$$

$$u_{iv}^h \geq 0 \quad \forall i \in M^h \setminus E, v \in V \quad (47)$$

$$\delta_{iv}^h \geq 0 \quad \forall i \in M^h \setminus E, v \in V \quad (48)$$

$$\lambda_{ir}^h \in \{0,1\} \quad \forall i \in F_r^h, r \in R \quad (49)$$

$$\rho_{ir}^h \in \{0,1\} \quad \forall i \in F_r^h, r \in R \quad (50)$$



5 演算法

一般而言，本研究的整合生產與配送規劃之最佳化問題在求解複雜度上屬於 *NP-Hard* 的問題。因為問題過於複雜，目前的精確解法在求得最佳解的時間上，會隨著問題的規模呈現指數成長，無法在多項式時間 (polynomial time) 內求得最佳解。

為了使本問題能夠在決策者可以接受的時間內得到品質不錯的解，即 1 分鐘，本研究將切割問題再解決 (divide and conquer) 的技巧，將問題切割成多個獨立的子問題再分別設計適合的啟發式演算法 (heuristic algorithm) 求解。一般而言，Bradley et al. (1977) 提及，若能將大規模 (large scale) 的問題劃分成 k 個獨立的子問題再求解，其計算時間僅為原問題的 $1/k^2$ 。

所以在本章裡，將會介紹如何將本問題分割成 2 個子問題的方式 (decomposition)，分別是有時窗與資源限制的團隊尋寶問題 (TOPTWRC) 和有時窗限制與訂單可拒絕的收送貨問題 (MDPDPTWRS)，並且利用 Osyczka (1985) 提出的分層最佳化 (hierarchical optimization) 的概念，依序求解 2 個子問題。再介紹求解 TOPTWRC 問題的啟發式演算法與 MDPDPTWRS 問題的啟發式演算法。

5.1 子問題分割與分層最佳化架構

為了將本研究之整合生產與配送規劃之最佳化問題劃分為多個獨立的子問題，我們參考 Bradley et al. (1977) 提及的方式，首先分析哪幾個限制式為本問題之複雜限制式 (complicating constraints)，各個子問題裡的變數會因為這些限制式，在求解時互相影響。若移除這些限制式，本問題就可以簡化為多個可獨立計算的子問題。接著，使用分層分層最佳化 (hierarchical optimization) 的概念，依序求解 2 個子問題，以期在 1 分鐘內求得品質不錯的解。

於是我們將整合生產與配送規劃之最佳化問題的目標式和限制式中的決策變數列出，並將其分為生產問題之決策變數與配送問題之決策變數。如圖 5.1。

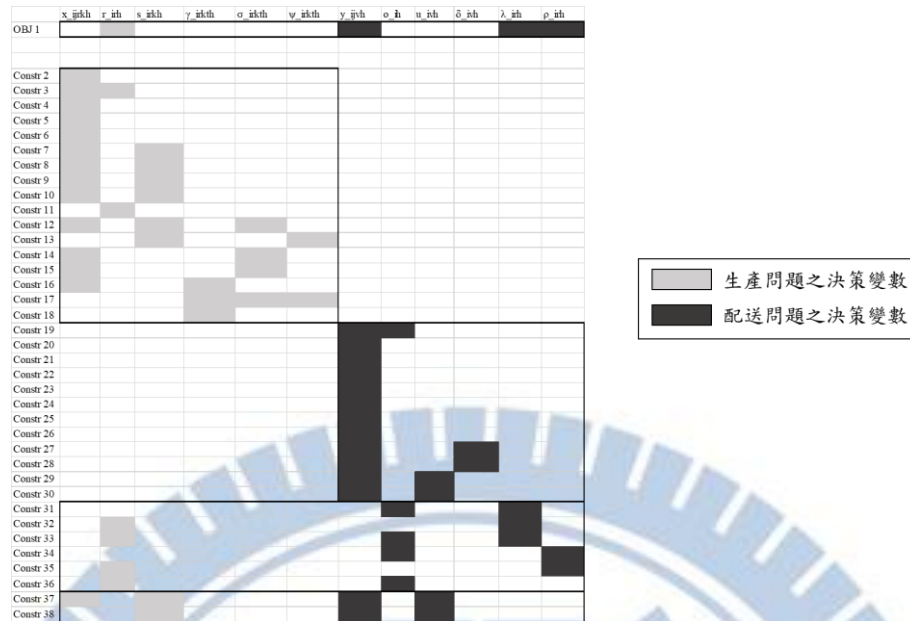


圖 5.1 數學模型目標式與限制式的決策變數

根據圖 5.1 可以發現，目標式的第一部分與限制式 2 至限制式 18 為生產排程問題，目標式的第二部分與限制式 19 至限制式 30 為配送路線規劃問題。此外，因為限制式 31 至 38，即餐廳與平台同時接受餐點的限制式、餐廳接受但平台拒絕餐點的相關限制式與取餐時間限制式和賞味時間限制式，使生產排程問題的變數與配送路線規劃問題的變數在模型計算時相互影響，所以限制式 31 至 38 即為複雜限制式。

若將複雜限制式拿掉，本研究可將整合生產排程與配送路線規劃問題切割為子問題 1，生產排程問題即有時窗與資源限制的團隊尋寶問題 (team orienteering problem with time window and resource constraint, TOPTWRC) 與子問題 2，配送路線規劃問題即多起點且有時窗限制與訂單可拒絕的收送貨問題 (multiple depots pickup and delivery problem with time window and selective requests, MDPDPTWSR)。其中，子問題 1 內的每個餐廳生產排程問題接彼此獨立，所以子問題 1 又可劃分為 R 個獨立生產排程問題， R 為決策時的餐廳總數。

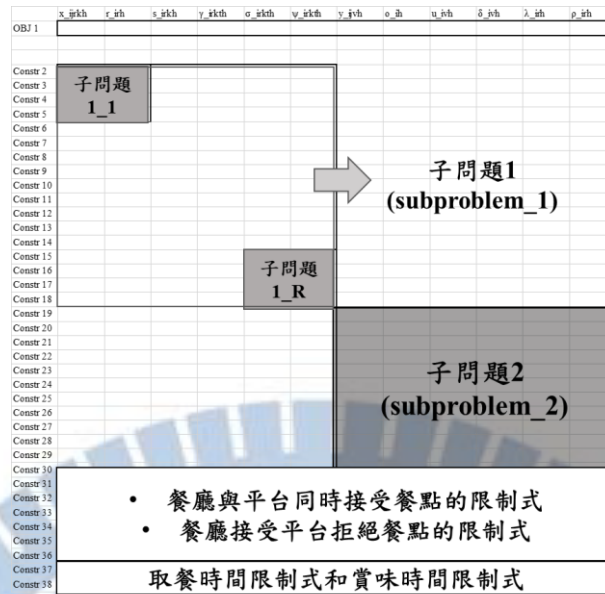


圖 5.2 子問題 1、2 與複雜限制式

■ TOPTWRC 問題 (子問題 1)

子問題 1 為有時窗與資源限制的團隊尋寶問題 (team orienteering problem with time window and resource constraint, TOPTWRC) 為有時窗限制的團隊尋寶問題之變形。關於有時窗限制的團隊尋寶問題 (team orienteering problem with time window, TOPTW) 之相關文獻可參考 Vansteenwegen et al. (2011) 的整理。TOPTW 問題環境給定若干個旅行員，每個節點有對應的服務時窗與收益，每個節點僅能被旅行員服務一次，但旅行員不需要服務每個節點。整個問題的目標為追求整體的收益 (revenue) 最大化。

表 5.1 為 TOPTW 與 TOPTWRC 的問題環境之整理，TOPTWRC 問題為比 TOPTW 多了內用餐點之節點的先到先服務限制、同一外送訂單餐點之節點必須同時接受或同時拒絕限制與資源限制的變形問題，分別對應到數學模型之限制式 (10)、限制式 (11)、限制式 (18)。

表 5.1 為 TOPTW 與 TOPTWRC 的問題環境之整理

	TOPTW	TOPTWRC
起點	1	1
旅行員人數	≥ 1	≥ 1
節點是否 一定要服務	否	否
節點最多被 服務之次數	1	1
目標式	最大化收益	最大化收益
限制式	◆ 流量守恆限制 ◆ 服務時窗限制	◆ 流量守恆限制 ◆ 服務時窗限制 ◆ 內用餐點之節點的先到先服務限制 ◆ 資源限制 ◆ 同一外送訂單餐點之節點必須同時接受或同時拒絕

■ MDPDPTWSR 問題 (子問題 2)

子問題 2 為多起點且有時窗限制與訂單可拒絕的收送貨問題 (multiple depots pickup and delivery problem with time window and selective requests, MDPDPTWSR)。類似於 Li et al. (2016) 所提出的考慮保留需求且有時窗限制的收送貨問題 (pickup and delivery problem with time windows, profits, and reserved requests, PDPTWSR)。

Li et al. (2016) 為第一篇提出 PDPTWSR 問題之文獻，該問題為傳統有時窗限制的收送貨問題 (PDPTW) 之變形。在 PDPTWSR 問題中，有多個外送員，待外送員服務的需求分為一定要服務的需求 (reserved request) 與不一定要服務的需求 (selective request)。每個需求有其貨物體積與若外送員服務該需求公司可獲得的收益，並且每個需求分別對應到一個收貨點與顧客點，每個收貨點與顧客點有對應的服務時窗。決策者必須服務所有一定要服務的需求，並選擇服務哪些不一定要服務的需求，達到最大化利潤 (profit) 目的。

下表 5.2 為 PDPTWSR 與 MDPDPTWSR 的問題環境之整理。本研究之 MDPDPTWSR 問題較 PDPTWSR 問題環境上不同的地方為，如 3.5 節所述，本研究的 MDPDPTWSR 問題中每個外送員都對應到一個起點，為多起點問題。並且每個需求未必都具備取餐節點與顧客節點，參見表 3.17 與表 3.19，若需求的顧客節點狀態為已被餐廳與平台同時接受且平台已取餐但尚未送達之訂單對應的顧客節點，則無對應的取餐節點。

表 5.2 PDPTWSR 與 MDPDPTWSR 的問題環境之整理

	PDPTWSR	MDPDPTWSR
外送員出發起點	1	≥ 1
旅行員人數	≥ 1	≥ 1
需求是否 一定要服務	否	否
需求是否皆有 取餐點與顧客點	是	否
目標式	最大化利潤	最大化利潤
限制式	◆ 不一定要服務的需求之相關限制 ◆ 流量守恒限制 ◆ 服務時窗限制 ◆ 收送貨順序限制	◆ 不一定要服務的需求之相關限制 ◆ 流量守恒限制 ◆ 服務時窗限制 ◆ 收送貨順序限制

■ 分層最佳化架構

由於在決策餐廳與平台接受的訂單時，一旦餐廳能接受待決策的外送訂單，平台就必須配送該外送訂單，在實務上平台決策的取餐時間與送達時間並不會影響餐廳是否接受該訂單。所以本研究依照此特性設計分層最佳化之架構，先求解子問題 1 與目標式的第一部分，最大化餐廳接受的外送訂單收益。再求解子問題 2 與目標式的第二部分，將決策出的餐點完成時間用來更新子問題 2 取餐節點的取餐時間與顧客節點的送達時間。關於餐點完成時間、

取餐時間與送達時間的關係可參見 3.3 節。分層最佳化求解架構如圖 5.3 所示。

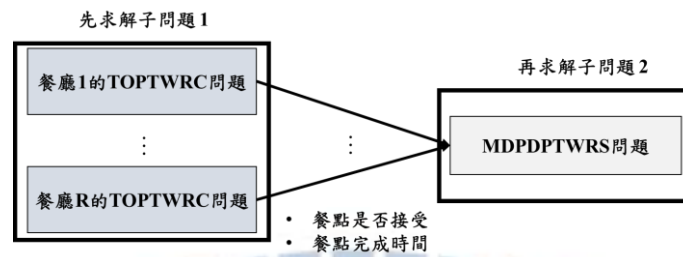


圖 5.3 分層最佳化求解架構圖

5.2 求解 TOPTWRC 問題之演算法

由於 TOPTWRC 為 TOPTW 之變形問題，於是我們參考 Gunawan et al. (2015) 所整理的近年針對 TOPTW 的求解演算法之求解績效之比較。我們可以發現 Vansteenwegen et al. (2009) 提出的 IterILS (Iterated local search) 演算法每題求解時間僅需 2.8 秒，並且以 IterILS 演算法找到的最佳解平均與目前所知 TOPTW 可找到的最佳目標值的差距僅為 3.5%。而由於本研究之求解時間緊迫，所以該演算法最適合處理本研究的 TOPTWRC 問題。接著，我們將會參考 IterILS 演算法，並修改成符合本研究需求的演算法。參見表 5.1 由於本問題較 TOPTW 問題多內用餐點之節點的先到先服務限制、資源限制與同一外送訂單餐點之節點必須同時接受或同時拒絕，所以在選擇節點插入車輛路線建構與改善可行解時，必須考慮到以上 3 個限制。以下我們會先列出本研究提出之 IterILS 流程圖，並一一介紹流程圖中的步驟。

● IterILS 演算法
演算法流程圖

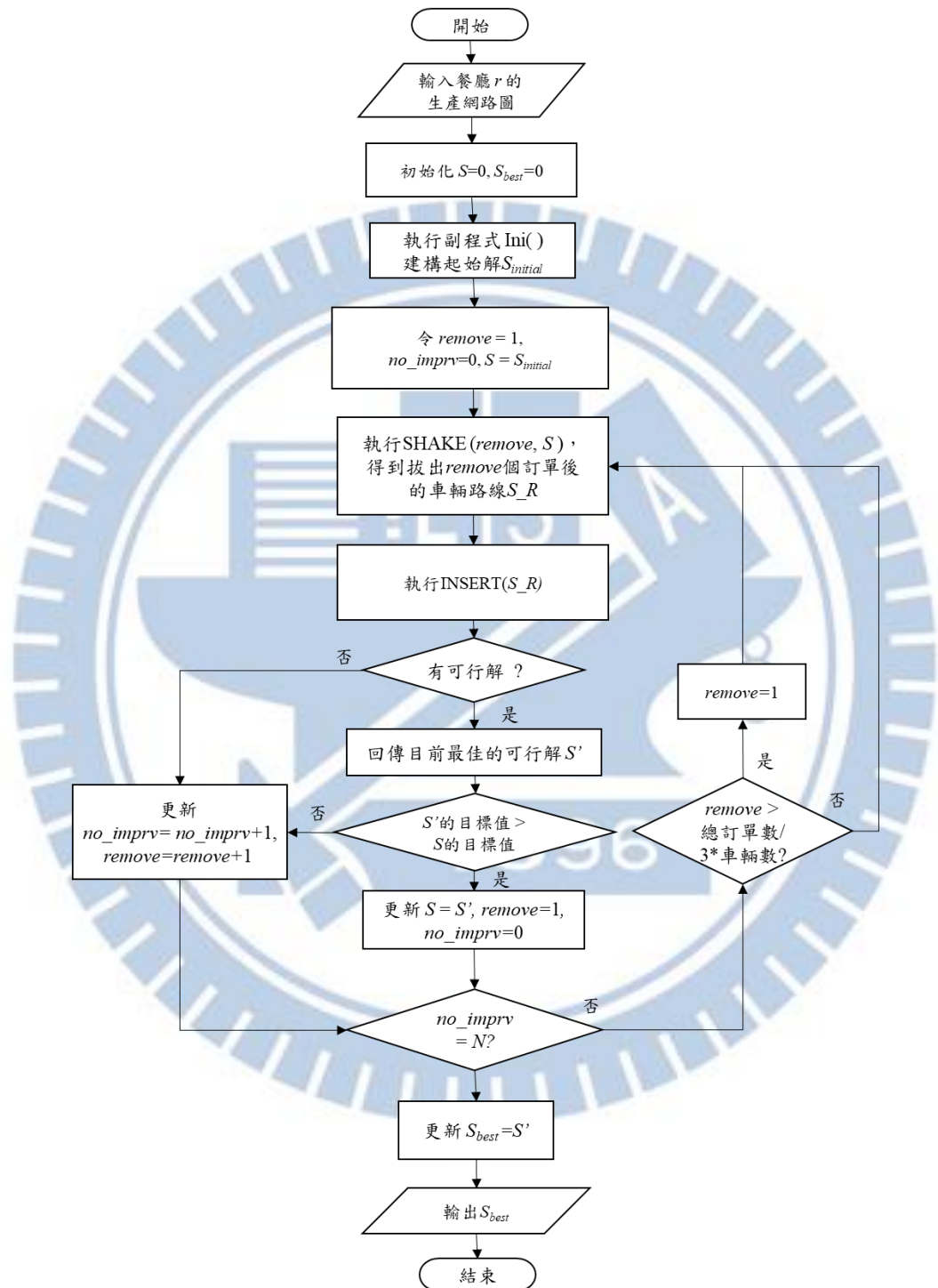


圖 5.4 本研究提出之 IterILS 演算法流程圖

為了演算法設計上的方便，本研究將已被餐廳與平台同時接受且正在製作的內用或外送餐點與已被餐廳與平台同時接受但尚未開始製作的外送餐點對應的訂單重新編號，每一張訂單只含有一個餐點。在本問題中，必須考慮每張外送訂單的餐點必須同時接受或拒絕，但因為已被餐廳與平台同時接受的外送餐點餐廳一定要完成，所以並不需要特別區分這些餐點是否屬於同一張訂單。重新編號的方式如下：

原始訂單：

訂單編號	點餐時間	旅行時間	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間長度
1	4	8	2	1	1	漢堡	7
				2	1	薯條	7
2	3	4	2	3	3	薯條	10
				4	3	漢堡	15

重新編號後的訂單：

訂單編號	點餐時間	旅行時間	餐點數量	餐點編號	餐點狀態	餐點種類	製作時間長度
1	4	8	1	1	1	漢堡	7
2	4	8	1	2	1	薯條	7
3	3	4	1	3	3	薯條	10
4	3	4	1	4	3	漢堡	15

餐點狀態編號	餐點狀態
1	已被餐廳與平台同時接受且正在製作的內用或外送餐點
3	已被餐廳與平台同時接受但尚未開始製作的外送餐點

接著我們將會介紹在 IterILS 演算法的虛擬程式碼 (pseudo code) 以及其中提及之函式，在函式說明中使用到的參數如下所示：

p_{gk} : 車輛 k 往節點 g 出發的時間。

s_{gk} : 車輛 k 抵達節點 g 的時間。

$wait_{gk}$: 車輛 k 在節點 g 的等候時間。

d_{gi} : 節點 g 與節點 i 間的距離。

$[a_i, b_i]$: 節點 i 的服務時窗。

此外，若參數右上加註逗號 (comma)，如 p'_{gk} ，則表示在某節點嘗試插入時，車輛 k 往節點 g 出發的時間。無加註逗號，如 p_{gk} ，則表示該節點已插入最佳可行位置，插入後，車輛 k 往節點 g 出發的時間。

主程式

IterILS()

```
1   $S \leftarrow S_{best} \leftarrow 0$ 
2  call Ini( ) return  $S_{initial}$ 
3   $remove \leftarrow 1, no\_imprv \leftarrow 0, S \leftarrow S_{best} \leftarrow S_{initial},$ 
4  while  $no\_imprv < N$ 
5       $P_1 \leftarrow 1 - no\_imprv / N, P_2 \leftarrow 1 - P_1$ 
6      call SHAKE (remove,  $S$ )
7      call INSERT (  $S_R$  )
8      if  $f(S') > f(S)$ 
9           $S \leftarrow S', remove \leftarrow 1, no\_imprv \leftarrow 0$ 
10     else
11          $no\_imprv \leftarrow no\_imprv + 1, remove \leftarrow remove + 1$ 
```

```

12   end if
13   if  $remove > \text{number of the orders} / 3 * \text{number of the vehicles}$ 
14        $remove \leftarrow 1$ 
15   end if
16 end while

```

列 1：

初始化

- 現有解 $S (x_{ijrk}^h, r_{ir}^h, s_{irk}^h, \gamma_{irkt}^h, \sigma_{irkt}^h, \phi_{irkt}^h)$
- 最佳解 $S_{best} (x_{ijrk}^{h*}, r_{ir}^{h*}, s_{irk}^{h*}, \gamma_{irkt}^{h*}, \sigma_{irkt}^{h*}, \phi_{irkt}^{h*})$

列 2：

執行 $Ini()$ ，建構起始解 $S_{initial}$

列 3：

初始化

- 令連續未更新次數 no_imprv 為 0，表示在進行 SHAKE ($remove, S$) 和 INSERT (S_R) 的搜尋架構下， S_{best} 連續未更新次數。
- 令初始移除訂單數量 $remove$ 為 1。

列 4：

若連續未更新次數小於連續未更新次數上限 N ，執行下列步驟：

列 5：

P_1 和 P_2 分別代表選擇較注重搜尋廣度或較注重搜尋深度的拔除策略和插入策略機率，利用目前 no_imprv 與 N 的比例做計算。一旦該機率選定，在下述迴圈內 SHAKE

(*remove*, *S*) 和 INSERT (*S_R*) 的也會依照該機率選擇較注重搜尋廣度或較注重搜尋深度的拔除策略和插入策略。

列 6-16 :

執行 SHAKE (*remove*, *S*) 和 INSERT (*S_R*) 進行從現有解 *S* 拔除節點在重新插入進行改善，若更新後的解 *S'* 的目標值較現有解 *S* 好，則更新 *S* 為 *S'*，並初始化 *remove* = 1 和 *no_imprv* = 0。若沒有比現有解之目標值來好或是找不到可行解則更新 *no_imprv* 為 *no_imprv* + 1，*remove* 為 *remove* + 1。最後檢查 *remove* 是否大於總訂單數 / 3 * 總車輛數，若大於則初始化 *remove* = 1。

函式介紹

Ini ()

```

1  create a list O to store the accepted but semi-finished for-here, platform orders
   and the responding nodes in the orders
2  while O ≠ { $\phi$ } do
3      select an order i in the list O with the smallest order number
4      randomly select an empty vehicle to insert the node in the order i
5      remove order i from the list O
6  end while
7  create a list B to store the accepted but unstarted and unfinished for-here orders
   and the responding in the orders
8  while B ≠ { $\phi$ } do
9      select an order i in the list B with the smallest order number
10     insert the node a in order i in a feasible position which has the smallest  $s_{ak}$ 
11     remove order i from the list B
12 end while
13 create a list C to store the accepted but unstarted and unfinished platform orders
   and the nodes in the orders
14 while C ≠ { $\phi$ } do
15     select an order i in the list C randomly

```

```

16   insert the node  $a$  in the order  $i$  in a feasible position which has the smallest
       $shift_{ak}$ 
17   remove order  $i$  from the list  $C$ 
18   end while
19   create a list  $F$  to store platform orders that haven't been decided yet and the nodes
      in the orders
20   while  $F \neq \{\emptyset\}$  do
21     select an order  $i$  with the highest price in the list  $F$ 
22     randomly sort the sequence of nodes in order  $i$ 
23     for node  $a \leftarrow 1$  to  $|order_i|$  do
24       insert the node  $a$  in the order  $i$  in a feasible position which has the smallest
           $shift_{ak}$ 
25       if a feasible position cannot be found check next order
26     end for
27     remove the order  $i$  from the list  $F$ 
28   end while

```

列 1-6：

建立一個 list O 儲存已被餐廳與平台同時接受且正在製作中的內用與外送餐點所對應訂單，並在每張訂單內儲存該訂單餐點對應的生產網路節點。每張訂單內僅含一道餐點並對應一個節點。因此，節點編號與訂單編號相同。此外，訂單編號不同的節點（餐點）並不會由同一個虛擬車輛（平行機台）服務（製作）。所以，我們可以依照訂單編號大小選擇編號最小的那張訂單，將該訂單內的生產網路節點 a 隨機插入一台空的虛擬車輛（平行機台）中。

列 7-12：

建立一個 list B 儲存已被餐廳與平台同時接受但尚未開始製作的內用餐點和預測未來可能出現的內用餐點所對應訂單，並在每張訂單內儲存該訂單餐點對應的生產網路節點。同樣的，由於這類訂單，每張訂單內僅含有一個生產網路節點，所以節點編

號與訂單編號相同。此外，先點餐的內用餐點必須先服務。而本研究在編內用訂單的訂單編號時，是依照點餐時間來編號。所以，我們可以依照訂單編號大小選擇編號最小的那張訂單，將該訂單內的生產網路節點 a 插入該節點抵達時間(餐點完成時間)值最小的可行插入位置。因為先到先服務的限制，若內用餐點的完成時間越小，在該餐點之後點餐的內用餐點，其對應節點的可行插入位置越多。

而該插入位置是否可行，則須檢查在將該節點插入某位置後，車輛路線上的節點抵達時間是否符合先到先服務限制、時窗限制與資源限制詳細的步驟介紹如下：

步驟一：檢查先到先服務限制與插入節點的時窗限制

為了滿足先到先服務限制與插入節點 i 的時窗限制，本研究對於某候選插入位置 (k, g, j) ；表示車輛 k 路線上的節點 g 與 j 間的位置。首先，由於節點 i 必須符合先到先服務限制，而內用訂單餐點之對應節點編號是由點餐時間順序而得。所以我們必須比較車輛 k 最早可以抵達節點 i 的時間 $s_{gk} + d_{gi}$ ；亦即車輛 k 在節點 g 不等候，馬上前往節點 i ，跟抵達編號在節點 i 前一點的節點 $i-1$ 的時間 $s_{i-1,k'}$ 的大小 (k 與 k' 可以為同一台車輛或是不同台車輛)。

可能情形如下圖：

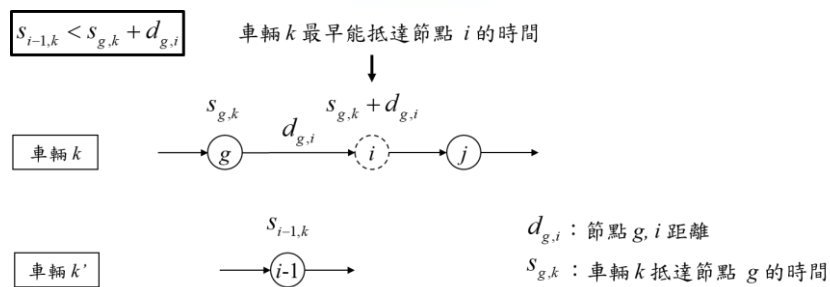


圖 5.5 車輛 k' 抵達節點 $i-1$ 的時間小於車輛 k 最早能抵達節點 i 的時間

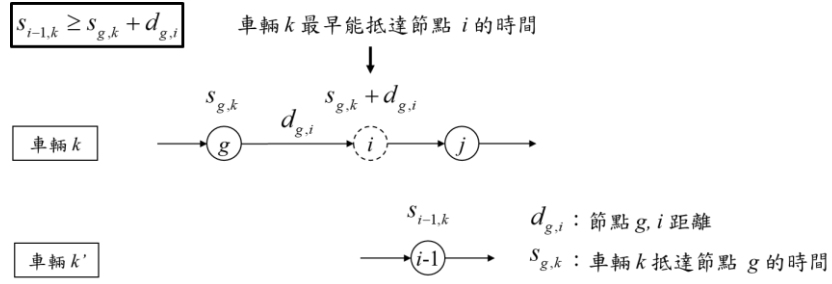


圖 5.6 車輛 k' 抵達節點 $i-1$ 的時間大於或等於車輛 k 最早能抵達節點 i 的時間

最早抵達節點 i 的時間不符合先到先服務限制或是不符合節點 i 的服務時窗，則必須在節點 g 等待 $wait'_{gk}$ 。比較結果與對應的計算方式如下：

- $s_{gk} + d_{gi} \geq s_{i-1,k'}$ ，表示符合先到先服務限制，並記錄 $p'_{ik} = s_{gk}$ 、 $s'_{ik} = p'_{ik} + d_{gi}$ ，並且比較 b_i 和 s'_{ik} 的大小。 p'_{ik} 表示不在節點 g 停等，直接往節點 i 出發的時間， s'_{ik} 表示不在節點 g 停等，直接往節點 i 出發抵達節點 i 時間。由於該時間可能會違反節點 i 的服務時窗，故經由下面的方式調整抵達節點 i 的時間。

■ 若 $b_i \geq s'_{ik}$

- ◆ $s'_{ik} \geq a_i$ ，表示抵達節點 i 的時間在時窗內。此時在節點 g 的等待時間 $wait'_{gk}$ 的值為 0。

- ◆ $s'_{ik} < a_i$ ，則表示 s'_{ik} 較節點 i 服務時窗上界 a_i 早。而車輛 k 在節點 g 的等待時間 $wait_{gk}$ 的值為 $a_i - s'_{ik}$ ，表示車輛 k 必須在節點 g 等待。

此時將抵達節點 i 的時間為節點 i 的時窗上界 a_i ， $s'_{ik} = a_i$ ，並倒推往節點 i 的出發時間 $p'_{ik} = s'_{ik} - d_{gi}$ 。如下圖：

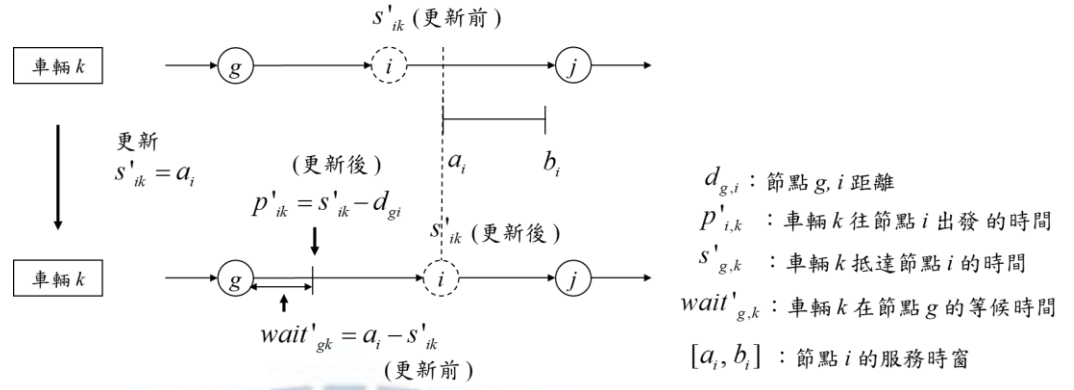


圖 5.7 若插入節點 i 後其 s'_{ik} 較節點 i 服務時窗上界 a_i 早所需的調整

■ 若 $b_i < s'_{ik}$ ，表示抵達節點 i 的時間違反時窗限制，此位置不可行。

➤ $s_{gk} + d_{gi} < s_{i-1,k'}$ ，在這個情況下，車輛 k 若不在節點 g 停等，抵達節點 i 的時間將會違反先到先服務限制。所以我們利用節點 $i-1$ 的抵達時間 $s_{i-1,k'}$ 倒推節點 i 的出發時間，所以 $p'_{ik} = s_{i-1,k'} - d_{gi}$ 、 $s'_{ik} = s_{i-1,k'}$ ，如下圖：

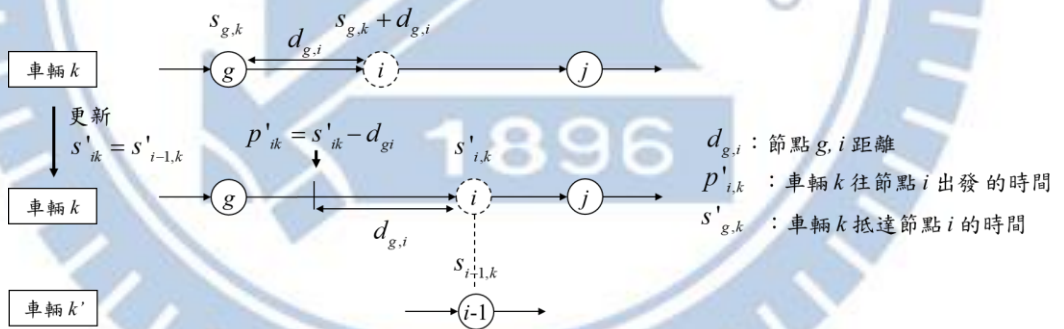


圖 5.8 車輛 k' 抵達節點 $i-1$ 的時間大於或等於車輛 k 最早能抵達節點 i 所需的調整

接著比較 b_i 和 s'_{ik} 的大小。由於 s'_{ik} 可能會違反時窗限制，所以經由下列步驟進行調整。

■ 若 $b_i \geq s'_{ik}$

◆ $s'_{ik} \geq a_i$ ，表示抵達節點 i 的時間在時窗內。此時在節點 g 的等待時

間 $wait'_{gk}$ 的值為 $p'_{ik} - s_{gk}$ ，即因為先到先服務而必須在節點 g 等待的時間。

- ◆ $s'_{ik} < a_i$ ，則表示 s'_{ik} 較節點 i 服務時窗上界 a_i 早。而車輛 k 在節點 g 的等待時間 $wait'_{gk}$ 的值為 $p'_{ik} - s_{gk} + (a_i - s'_{ik})$ ，見圖 5.7 和 5.8，因為先到先服務限制車輛 k 必須在節點 g 等候的時間為 $p'_{ik} - s_{gk}$ ，與服務時窗限制車輛 k 必須在節點 g 等候的時間為 $a_i - s'_{ik}$ ，所以總等待時間為 $p'_{ik} - s_{gk} + (a_i - s'_{ik})$ 。並令 $s'_{ik} = a_i$ 、 $p'_{ik} = s'_{ik} - d_{gi}$ 。

- 若 $b_i < s'_{ik}$ ，則此位置不可行。

步驟二：檢查在該插入節點之後的其他節點的時窗限制

若此位置在通過步驟一的檢驗，則計算車輛 k 在節點 i 的等候時間 $wait'_{ik}$ ， $wait'_{ik} = \max[0, p_{jk} - s'_{ik}]$ 與插入節點 i 造成車輛 k 抵達節點 i 後面的節點（節點 j ）時間向後延遲的幅度， $shift'_{ik}$ ， $shift'_{ik} = d_{gi} + wait'_{gk} + d_{ij} + wait'_{ik} - d_{gj} - wait_{gk}$ 。 $shift'_{ik}$ 之示意圖如下。

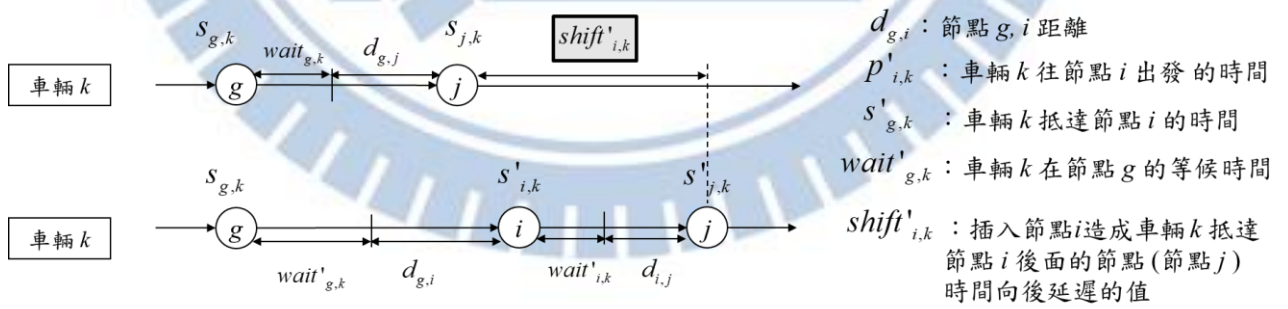


圖 5.9 因車輛 k 插入節點 i 造成節點 i 之後的節點 j 的抵達時間向後推遲之示意圖

而車輛 k 抵達節點 j 的時間最多能被往後推遲的幅度為 m_shift_{jk} ，所以若 $shift'_{ik} \leq m_shift_{jk}$ ，則表示此插入位置並不會造成該車輛上節點 j 與其後的節點違反時窗限制。

節點 j 與同台車輛在節點 j 之後的節點 m_shift_{jk} 的值會在步驟四更新，但若節點 j 為該車輛 k 路線的除了虛擬起訖點外的唯一一個節點，其計算方式為 $m_shift_{jk} = b_j - s'_{jk}$ 。為節點時窗上界 b_j 與車輛 k 抵達節點 j 的時間 s'_{jk} 之差。

步驟三：更新在該插入節點後的其他節點的抵達時間

若該插入位置通過前步驟一與二的檢查，則更新同一車輛上該節點後每個節點 (e.g. 節點 $j, l, \dots$) 的 s'_{jk} 、 $wait'_{jk}$ 、 $shift'_{jk}$ 、 m_shift_{jk} ，更新方法為：

$$s'_{jk} = s_{jk} + shift'_{ik}，參見圖 5.9。$$

$$wait'_{jk} = \max[0, wait_{jk} - shift'_{ik}]，見下圖 5.10。$$

$$shift'_{jk} = \max[0, shift'_{ik} - wait_{jk}]，參見圖 5.10。$$

若 $wait_{jk} < shift'_{ik}$ ，會造成原本車輛 k 在節點 j 的等候時間 $wait_{jk}$ 完全被抵銷，所以 $wait'_{jk} = 0$ ，而因為節點 i 的插入造成抵達節點 j 的下一點時間的被推遲的幅度會被節點 j 本身的 $wait_{jk}$ 減少一部分，所以 $shift'_{jk} = shift'_{ik} - wait_{jk}$ 。反之， $wait_{jk} \geq shift'_{ik}$ ，會造成原本車輛 k 在節點 j 的等候時間 $wait_{jk}$ 僅被抵銷部分， $wait'_{jk} = wait_{jk} - shift'_{ik}$ 。反之，則會讓因為節點 i 的插入造成抵達節點 j 的下一點時間的被推遲的幅度完全被節點 j 本身的 $wait_{jk}$ 抵銷。

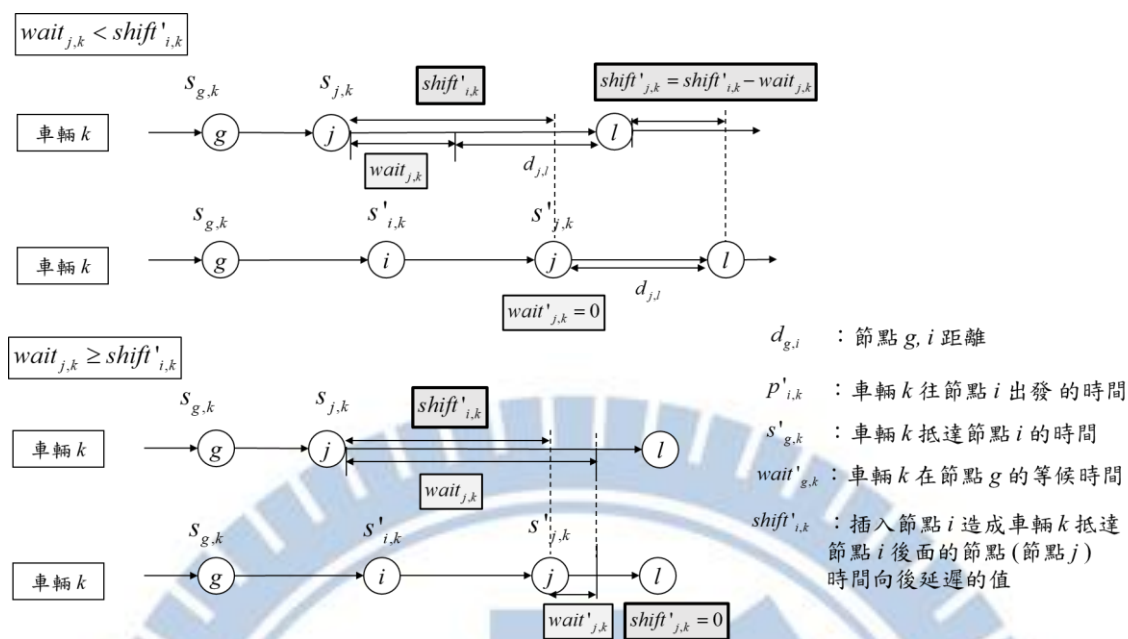


圖 5.10 $wait'_{jk}$ 與 $shift'_{ik}$ 的更新方式示意圖

$m_shift'_{jk} = m_shift_{jk} - shift'_{ik}$ ，而車輛 k 抵達節點 j 的時間最多能被往後推遲的幅度，則會因為節點 i 的插入而減少 $shift'_{ik}$ 。

若更新至其中某一節點 l ，其 $shift'_{lk} = 0$ ，則停止。 s_{jk} 、 $wait_{jk}$ 、 m_shift_{jk} 表示在插入節點 i 前，車輛 k 抵達節點 j 的時間、車輛 k 在節點 j 等待時間，車輛 k 抵達節點 j 的時間最多能被推遲多少時間。

步驟四：檢查先到先服務與資源限制

經過步驟三更新 $shift'_{jk}$ 後，檢查各台車輛有被服務的內用訂單餐點對應的生產網路節點是否符合先到先服務限制。若皆有符合，則繼續檢查資源限制。檢查在插入節

點 i 之後所有節點 j 所對應的餐點，在平行機台上的占用時間段，是否小於等於在每個時間段同時製作的餐點種類 ψ 的數量上限 e_ψ 。

- 若符合資源限制，則記錄此位置 s'_{ik} 值。如同前述，因為先到先服務的限制，若內用餐點的完成時間越小，在該餐點之後點餐的內用餐點，其對應節點的可行插入位置越多。

- 若不符合資源限制，則跳至下一個候選位置，重新進行步驟一~四。

直到所有位置皆被搜尋過。

步驟五：選擇 s'_{ik} 值最小的位置插入

首先，令 s'_{ik} 值最小的位置為 (k^*, g^*, j^*) 。更新對應 (k^*, g^*, j^*) 的 p'_{ik} 、 s'_{ik} 、 $wait'_{gk}$ 、 $wait'_{ik}$ 、 $shift'_{ik}$ 為 p_{ik} 、 s_{ik} 、 $wait_{gk}$ 、 $wait_{ik}$ 、 $shift_{ik}$ 。接著，更新車輛 k 在插入節點 i 之後的每一個節點 j 的 s'_{jk} 、 $wait'_{jk}$ 、 $shift'_{jk}$ 和 $m_shift'_{jk}$ 為 s_{jk} 、 $wait_{jk}$ 、 $shift_{jk}$ 和 m_shift_{jk} 。再計算節點 i 的 m_shift_{ik} 。利用公式 $\min[b_i - s_{ik}, wait_{ik} + m_shift_{jk}]$ 。車輛 k 抵達節點 i 的時間最多被推遲的幅度不能超過本身的時窗限制，與不能造成節點 i 的下一點節點 j 被推遲的幅度超過 m_shift_{jk} ，如同圖 5.10 所示， $wait_{ik}$ 能抵銷節點 i 對車輛 k 抵達節點 j 的推遲，所以公式即為 $\min[b_i - s_{ik}, wait_{ik} + m_shift_{jk}]$ 。最後，更新位置在節點 i 之前所有節點 g 的 m_shift_{gk} 。利用公式 $\min[b_g - s_{gk}, wait_{gk} + m_shift_{ik}]$ 。

列 13-18：

建立一個 list C 儲存已被餐廳與平台同時接受但尚未開始製作的外送餐點所對應的訂單，並在每張訂單內儲存該訂單餐點對應的生產網路節點。同樣的，由於這類訂

單，每張訂單內僅含有一個生產網路節點，所以節點編號與訂單編號相同。由於本問題的服務時窗限制、先到先服務限制與資源限制(在每個時間段同時能製作的同一種類餐點數量限制)。同樣的，所以我們選擇 $shift_{ak}$ 最小的可行位置插入節點。由於這類插入節點本身沒有先到先服務的限制，所以我們將檢驗可行位置的步驟一修改為：

步驟一：檢查插入節點的時窗限制

- 記錄 $p'_{ik} = s_{gk}$ 、 $s'_{ik} = p'_{ik} + d_{gi}$ ，並且比較 b_i 和 s'_{ik} 的大小。 p'_{ik} 與 s'_{ik} 表示不在節點 g 停等，直接往節點 i 出發的時間，與抵達節點 i 的時間。由於該時間可能會違反節點 i 的服務時窗，故經由下面的方式調整抵達節點 i 的時間。
 - 若 $b_i \geq s'_{ik}$
 - ◆ $s'_{ik} \geq a_i$ ，表示抵達節點 i 的時間在時窗內。此時在節點 g 的等待時間 $wait'_{gk}$ 的值為 0。
 - ◆ $s'_{ik} < a_i$ ，則表示在節點 g 的等待時間 $wait_{gk}$ 的值為 $a_i - s'_{ik}$ ，表示車輛 k 必須在節點 g 等待。此時將抵達節點 i 的時間為節點 i 的時窗上界 a_i ， $s'_{ik} = a_i$ ，並倒推往節點 i 的出發時間 $p'_{ik} = s'_{ik} - d_{gi}$ 。
 - 若 $b_i < s'_{ik}$ ，表示抵達節點 i 的時間違反時窗限制，此位置不可行。

即在選擇插入位置的步驟一時不需考慮車輛 k 抵達節點 $i-1$ 的時間 s'_{ik} 。

列 19-28：

建立一個集合 F 儲存待平台評估餐廳與平台是否接受的外送餐點所對應的訂單，並在每張訂單內儲存該訂單餐點對應的生產網路節點。由於這類外送餐點，在決策是否接受時必須同時接受或拒絕同一張訂單內的所有外送餐點。所以在選擇插入節點 a 時，我們會先選擇一張價值最高的訂單，再隨機選擇一個該訂單內的餐點所對應的生產網路節點進行插入。雖然同一張訂單內節點的插入順序會影響解的好壞，但由於求解時間的關係，我們不會嘗試所有插入順序，而是採用隨機的方式決定同一張訂單內節點的插入順序。接著，選擇該節點 a 插入位置 $shift_{ak}$ 值最小的位置插入。由於這類插入節點本身沒有先到先服務的限制，所以我們檢測該位置可行性的方式與 list C 內的節點相同。

SHAKE(remove, S)

```

1  randomly generate a number  $random$ ,  $0 \leq random \leq 1$ 
2  if  $random \leq P_1$ 
3      if  $remove > \text{orders in the } S$ , exit
4      randomly select  $remove$  the orders, including 4 types
        (1. the accepted but semi-finished for-here orders and platform orders
         2. the accepted but unstarted and unfinished for-here orders
         3. the accepted but unstarted and unfinished platform orders
         4. the platform orders that haven't been decided yet)
        in the  $S$ 
5  else
6      if  $remove > \text{platform orders that haven't been decided yet in the } S$ , exit
7      randomly select  $remove$  platform orders that haven't been decided yet in the  $S$ 
8  end if
9  remove the nodes w.r.t.  $remove$  orders from  $S$ 
10 for each vehicle  $k$  do
11     update  $m\_shift$  and  $wait$ 
12 end for
13 return  $S\_R$ 

```

列 1：

在主程式列 5 生成的 P_1 和 P_2 表示選擇移除策略 1 與選擇移除策略 2 的機率。移除策略 1 表示針對所有被虛擬車輛服務的訂單移除，其中包含已被餐廳與平台同時接受的訂單與待平台評估餐廳與平台是否接受的訂單，而移除策略 2 僅針對被虛擬車輛服務的待平台評估餐廳與平台是否接受的訂單移除。並隨機產生一個介於 0 到 1 間的亂數。利用該亂數與 P_1 和 P_2 的大小關係，來決定該選擇移除策略 1 或是移除策略 2 的節點移除策略。這個方式參考自 Li et al. (2016) 的動態調整選擇插入與拔除策略之方式 (Dynamic Adjustment of Operator Behavior, DAOB)，在當連續未更新數較小時，演算法會偏向選擇移除策略 1，選擇所有被服務的訂單做移除，在之後重新將訂單插入車輛時，能夠有較多樣可能的解。亦即能搜尋到較多的解空間 (solution space)，增加搜尋廣度。但該方法的缺點為某些已被餐廳與平台同時接受的訂單，在重新插入車輛路線時無法插入，造成不可行解。所以在連續未更新數增加時，演算法會偏向選擇移除策略 2，僅選擇被服務的待平台評估餐廳與平台是否接受的訂單移除，該方式雖然能搜尋的解空間較小，但每次的訂單插入皆為可行解，增加搜尋深度。

列 2-4：

若此亂數小於等於選擇移除策略 1 的機率，則選擇移除策略 1。

➤ **移除策略 1：**

1. 檢查 *remove* 是否大於解 *S* 中的訂單總數。
 - ◆ 若大於，則跳出 SHAKE 函式。
 - ◆ 若小於等於，*remove* 保持不變。
2. 從解 *S* 中隨機選擇 *remove* 個訂單移除。

列 5-8：

反之，則選擇移除策略 2。

➤ 移除策略 2：

1. 檢查 $remove$ 是否大於解 S 中待平台評估餐廳與平台是否接受的訂單數。
 - ◆ 若大於，則跳出 SHAKE 函式。
 - ◆ 若小於等於， $remove$ 則保持不變。
2. 從解 S 中移除 $remove$ 個待平台評估餐廳與平台是否接受的訂單。

列 9：

將選定訂單內的生產網路節點從解 S 中拔出。

列 10-12：

更新拔出 $remove$ 個訂單後的解 S_R 節點的 $wait$ 與 m_shift 。由於某些節點已被拔除，所以針對尚在車輛路線內的節點更新其 $wait$ 、與 m_shift 的值。如下圖，假設尚在車輛 k 路線中的節點有節點 g 、節點 j 、節點 l 。所以必須更新節點 g 、節點 j 、節點 l 之 $wait$ 、與 m_shift 的值。計算方式以節點 j 的等候時間為例， $wait_{g,k} = s_{j,k} - d_{g,j} - s_{i,k}$ 。而 $m_shift_{j,k}$ 則是與節點 j 在同一車輛上後一點節點 l 的 m_shift 值有關，如同前述， $m_shift_{j,k} = \min[b_j - s_{jk}, wait_{j,k} + m_shift_{l,k}]$ 。

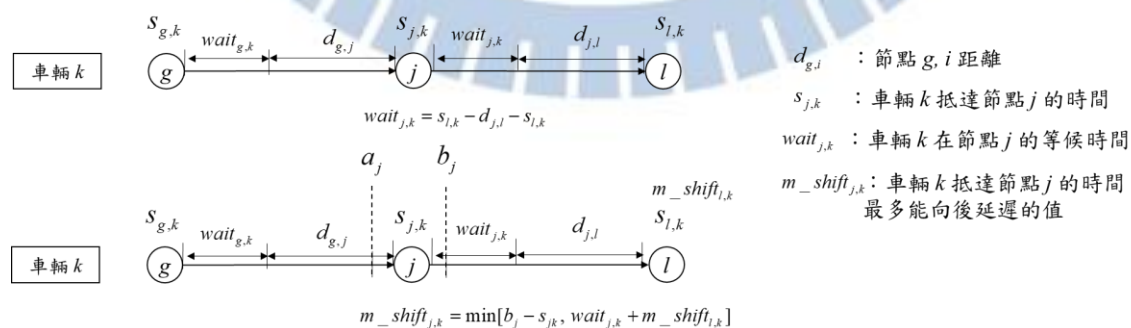


圖 5.11 拔出某些節點後剩餘節點之 $wait$ 與 m_shift 之計算方式

INSERT (S_R)

```

1  randomly generate a number random,  $0 \leq \text{random} \leq 1$ 
2  if  $\text{random} \leq P_1$ 
3      create a list A to store all the orders, including 4 order types
        (1. the accepted but semi-finished for-here orders and platform orders
        2. the accepted but unstarted and unfinished for-here orders
        3. the accepted but unstarted and unfinished platform orders
        4. the platform orders that haven't been decided yet)
4      sort list A randomly
5      for order  $i \leftarrow 1$  to  $|A|$ 
6          if the order  $i$  is the accepted but semi-finished for-here orders and platform
            order
7              do line 10 in the Ini( )
8              if feasible position cannot be found then exit
9          end if
10         if the order  $i$  is the accepted but unstarted and unfinished for-here orders
11             do line 10 in the Ini( )
12             if feasible position cannot be found then exit
13         end if
14         if the order  $i$  is the accepted but unstarted and unfinished platform orders
15             do line 16 in the Ini( )
16             if feasible position cannot be found then exit
17         end if
18         if the order  $i$  is the platform orders that haven't been decided yet
19             do line 21~25 in the Ini( )
20         end if
21     end for
22 else
23     create a list B to store all the orders, including 4 types
        (1. the accepted but semi-finished for-here orders and platform orders
        2. the accepted but unstarted and unfinished for-here orders
        3. the accepted but unstarted and unfinished platform orders
        4. the platform orders that haven't been decided yet)
24     sort list B by order type
        (1. the accepted but semi-finished for-here orders and platform orders

```

```

2. the accepted but unstarted and unfinished for-here orders
3. the accepted but unstarted and unfinished platform orders
4. the platform orders that haven't been decided yet)
25 for order  $i \leftarrow 1$  to  $|B|$ 
26   if order  $i$  is the accepted but semi-finished for-here orders and platform
      orders
27     do line 10 in the Ini( )
28     if feasible position cannot be found then exit
29   end if
30   if the order  $i$  is the accepted but unstarted and unfinished for-here orders
31     do line 10 in the Ini( )
32     if feasible position cannot be found then exit
33   end if
34   if the order  $i$  is the accepted but unstarted and unfinished platform orders
35     do line 16 in the Ini( )
36     if feasible position cannot be found then exit
37   end if
38   if the order is the platform orders that haven't been decided yet
39     do line 21~26 in the Ini( )
40   end if
41 end if

```

列 1-2 :

同樣利用 DAOB 之方法，隨機產生一個介於 0 到 1 間的亂數，在主程式列 5 生成的 P_1 和 P_2 表示選擇插入策略 1 與選擇插入策略 2 的機率。插入策略 1 表示隨機挑選一個尚未服務的已被餐廳與平台同時接受的訂單或待平台評估餐廳與平台是否接受的訂單插入目前車輛路線中，而插入策略 2 則是先插入已被餐廳與平台同時接受的訂單，插入順序為正在製作中的內用與外送訂單、已被餐廳與平台同時接受但尚未製作的內用訂單與已被餐廳與平台同時接受但尚未製作的外送訂單，再插入待平台評估餐廳與平台是否接受的訂單。並隨機產生一個介於 0 到 1 間的亂數。利用該亂數與 P_1 和 P_2 的大小關係，來決定該選擇移除策略 1 或是移除策略 2 的節點移除策略。在當連續

未更新數較小時，演算法會偏向選擇插入策略 1，該方式能夠有較多樣可能的解。亦即能搜尋到較多的解空間 (solution space)，增加搜尋廣度。但該方法的缺點為某些已被餐廳與平台同時接受的訂單，在重新插入車輛路線時可能無法插入，造成不可行解。所以在連續未更新數增加時，演算法會偏向選擇插入策略 2，先插入已被餐廳與平台同時接受的訂單，插入順序為正在製作中的內用與外送訂單、已被餐廳與平台同時接受但尚未製作的內用訂單與已被餐廳與平台同時接受但尚未製作的外送訂單，再插入待平台評估餐廳與平台是否接受的訂單。該方式雖然能搜尋的解空間較小，但較能確保找到可行解，增加搜尋深度。

列 3：

若此亂數小於等於選擇移除策略 1 的機率，則選擇插入策略 1。

列 4-18：

插入策略 1：

建立一個 list A 紀錄尚未服務的訂單，並將該陣列隨機排序。接著從該陣列的第一張訂單開始做下列的檢驗。

- 若此訂單為已被餐廳與平台同時接受但正在製作中的內用與外送訂單，則執行 Ini() 中的列 10。若其中一個訂單無法插入，則跳出 INSERT(S_R)。
- 若此訂單為已被餐廳與平台同時接受但尚未製作的內用訂單，則執行 Ini() 中的列 10。若其中一個訂單無法插入，則跳出 INSERT(S_R)。
- 若此訂單為已被餐廳與平台同時接受但尚未製作的外送訂單，則執行 Ini() 中的列 16。若其中一個訂單無法插入，則跳出 INSERT(S_R)。
- 若此訂單為待平台評估餐廳與平台是否接受的外送訂單，則執行 Ini() 中的列 21-26。

直到每張訂單都被嘗試插入過。

列 19：

若此亂數大於選擇插入策略 1 的機率，則選擇插入策略 2。執行下列步驟：

列 20-34：

插入策略 2：

建立一個 list *B* 紀錄尚未在目前車輛路線中所有的訂單，並將該陣列依訂單種類與訂單編號排序，先排已被餐廳與平台同時接受但正在製作中的內用或外送訂單，接著排已被餐廳與平台同時接受但尚未製作的內用訂單，再排已被餐廳與平台同時接受但尚未製作的外送訂單，最後為待平台評估餐廳與平台是否接受的外送訂單。接著從該陣列的第一張訂單開始做下列的檢驗。

- 若此訂單為已被餐廳與平台同時接受但正在製作中的內用與外送訂單，則執行 *Ini()* 中的列 10。若其中一個訂單無法插入，則跳出 *INSERT(S_R)*。
- 若此訂單為餐廳與平台已同時接受但尚未製作的內用訂單，則執行 *Ini()* 中的列 10。若其中一個訂單無法插入，則跳出 *INSERT(S_R)*。
- 若此訂單為餐廳與平台已同時接受但尚未製作的外送訂單，則執行 *Ini()* 中的列 16。若其中一個訂單無法插入，則跳出 *INSERT(S_R)*。
- 若此訂單為待平台評估餐廳與平台是否接受的外送訂單，則執行 *Ini()* 中的列 21-26。

直到每張訂單都被嘗試插入過。

5.3 求解 MDPDPTWPR 問題之演算法

由於 MDPDPTWPR 為 PDPTWPR 之變形問題，而目前 PDPTWPR 之研究僅有 Li et al. (2016) 之文獻，所以本研究針對 MDPDPTWPR 問題所使用之演算法參考自 Li et al. (2016) 所使用的 ALNS 演算法。參見表 5.2，MDPDPTWSR 問題不同於 PDPTWPR 問題的環境，每個外送員有其對應的出發起點，並且每個需求未必皆有對應的取餐節點，所以在選擇節點插入車輛路線建構與改善可行解時，必須考慮將這兩個因素納入考慮。以下我們會先列出本研究提出之 ALNS 之流程圖，並一一介紹流程圖中的步驟。



● ALNS 演算法
演算法流程圖

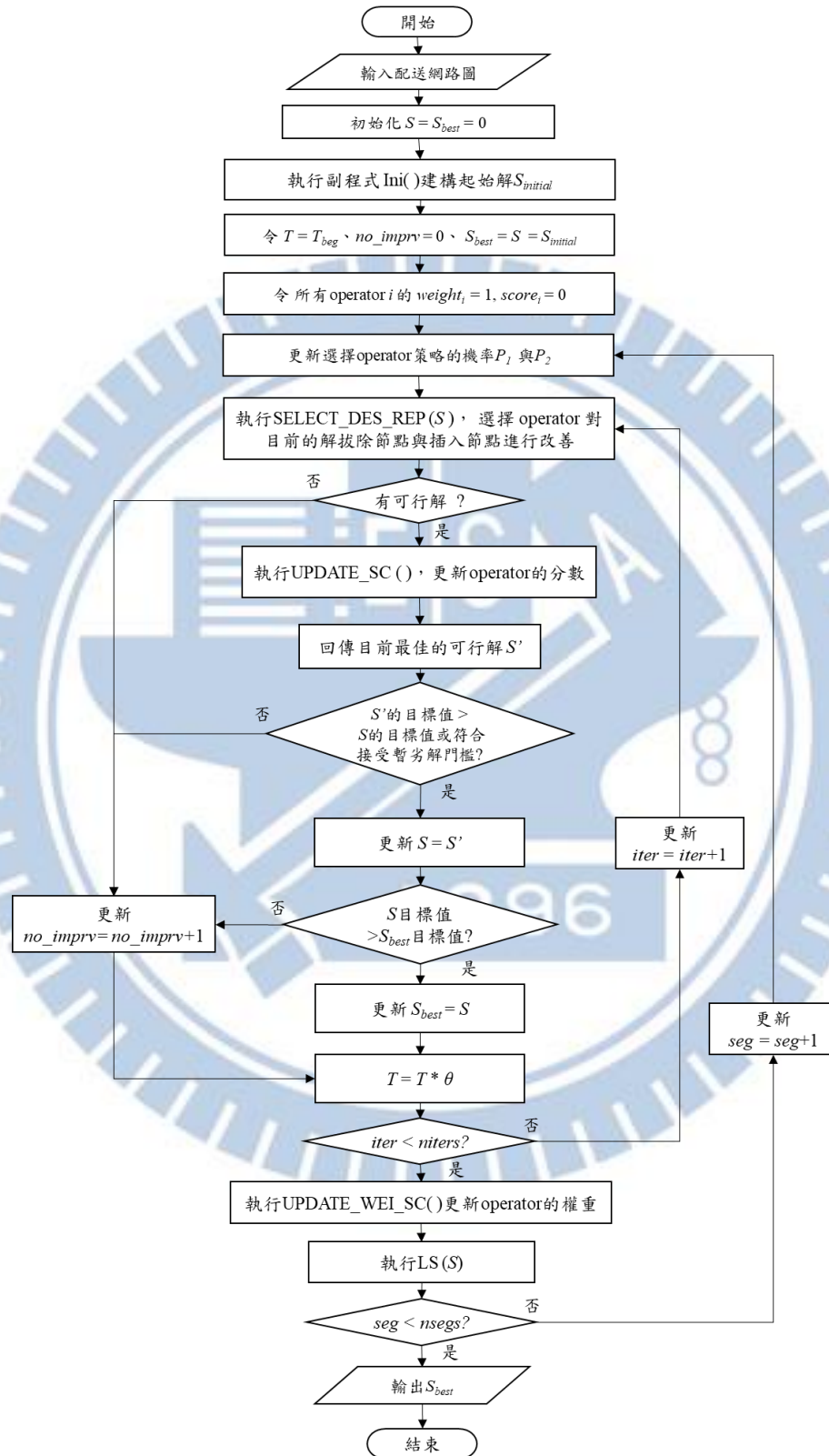


圖 5.12 本研究提出之 ALNS 演算法流程圖

主程式

ALNS()

```
1   $S \leftarrow S_{best} \leftarrow 0$ 

2  call Ini( ) return  $S_{initial}$ 

3   $S \leftarrow S_{best} \leftarrow S_{initial}$ ,  $T \leftarrow T_{beg}$ ,  $no\_imprv \leftarrow 0$ 

4  for each operator  $i$  do
5       $weight_i \leftarrow 1$ ,  $score_i \leftarrow 0$ 
6  end for
7  for  $seg \leftarrow 1$  to  $nsegs$ 
8       $P_1 \leftarrow 1 - seg / nsegs$ ,  $P_2 \leftarrow 1 - P_1$ 
9      for  $iter \leftarrow 1$  to  $niters$  do;
10         call SELECT_DES_REP(  $S$  )
11         if  $f(S') > f(S)$  or  $RANDOM(0,1) < e^{(f(S') - f(S))/T}$ 
12              $S \leftarrow S'$ 
13             if  $f(S) > f(S_{best})$ 
14                  $S_{best} \leftarrow S$ 
15             else
16                  $no\_imprv \leftarrow no\_imprv + 1$ 
17             end if
18         else
19              $no\_imprv \leftarrow no\_imprv + 1$ 
20         end if
21         call UPDATE_SC( )
22          $T \leftarrow T * \theta$ 
23     end for
24     call UPDATE_WEL_SC( )
25     call LS(  $S$  )
26 end for
```

列 1：

初始化

- 現有解 S (y_{ijv}^h 、 o_i^h 、 u_{iv}^h 、 δ_{iv}^h)
- 最佳解 S_{best} (y_{ijv}^{h*} 、 o_i^{h*} 、 u_{iv}^{h*} 、 δ_{iv}^{h*})

列 2：

執行 $\text{Ini}()$ ，建構起始解 $S_{initial}$

列 3-6：

初始化

- 令模擬退火溫度 T 為 T_{beg} ，本研究利用模擬退火法 (simulated annealing) 之架構，設定接受暫劣解之條件。
- 令連續未更新次數 no_imprv 為 0，在進行破壞運算元 (destroy operator) 與重建運算元 (repair operator) 的搜尋架構下， S_{best} 連續未更新次數。
- 破壞運算元與重建運算元 i ，令其初始權重 $weight_i$ 為 1，初始分數 $score_i$ 為 0。

列 7：

執行 $nseg$ 個主迴圈，每個 seg 迴圈選擇不同的破壞運算元和重建運算元，不同的破壞運算元和重建運算元會影響搜尋的深度與廣度。

由於在配送問題中有下列 3 種訂單狀態：

- ◆ 已被餐廳與平台同時接受且已取餐但尚未送達的訂單 (顧客節點集合： G_v^h)。
- ◆ 已被餐廳與平台同時接受但尚未取餐但尚未送達的訂單 (虛擬取餐節點集合： I^h 與顧客節點集合： Ω^h)。
- ◆ 待評估餐廳與平台是否接受的訂單 (虛擬取餐節點集合 Φ^h 和 Ξ^h 與顧客節點集合 Λ^h 和 Υ^h)。

注重搜尋廣度的策略在選擇訂單節點做拔除時，所有訂單皆為候選拔除的節點。此策略雖然可能導致後續插入節點時，因為某些已被餐廳與平台同時接受的訂單節點無法插入而無法搜尋到可行解，但能夠搜尋到較為多樣的解。而注重搜尋深度的策略僅選擇待評估餐廳與平台是否接受的訂單節點做拔除。此策略在後續插入節點時，一定可以搜尋到可行解，但搜尋到的解較少。

一般而言在較前面的 *seg* 會選擇較注重搜尋廣度的破壞運算元和重建運算元；而在較後面的 *seg* 則會選擇較注重搜尋深度的破壞運算元和重建運算元。

列 8：

P_1 和 P_2 分別代表選擇較注重搜尋廣度或較注重搜尋深度的候選破壞運算元和重建運算元之機率，利用目前 *seg* 與 *nseg* 的比例做計算。一旦該機率選定，在 *iter* 迴圈內也會依照該機率選擇破壞運算元和重建運算元。

列 9-23：

首先呼叫函數 SELECT_DES_REP(*S*)，對現有解 *S* 進行改善，利用破壞運算元與重建運算元搜尋到的解為 S' 。

若搜尋出來的解 S' 可行並較現有解 *S* 好，或是滿足模擬退火架構下之更新條件 $\text{RANDOM}(0,1) < e^{(f(S')-f(S))/T}$ ，則更新 *S*。反之則更新 *no_imprv* 為 *no_imprv* + 1。此外若更新後的解 *S* 較 S_{best} 好，則更新 S_{best} 。反之，則更新 *no_imprv* 為 *no_imprv* + 1。

更新完畢後，執行函式 UPDATE_SC()更新運算元之分數 $score_i$ 。RANDOM(0, 1)代表再 0 與 1 之間隨機產生一個實數。

列 24-25：

在每一個 *seg* 迴圈內，執行完 *niters* 個迴圈後，便會更新模擬退火溫度 *T*。並且執行 UPDATE_WEI_SC()函式，更新運算元的 $weight_i$ 與初始化 $score_i$ ，再將現有解 *S* 再執行 LS()進行鄰域搜尋 (local search) 更新。

函式說明：

Ini()

```
1  sort the accepted and picked but unarrived platform orders (set  $G_v^h$ ) in decreasing price
   order in a list  $L$ 
2  sort the accepted but unpicked and unarrived platform orders (set  $I^h$  and  $\Omega^h$ ) in
   decreasing price order in a list  $Q$ 
3  sort the platform orders that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and  $\Lambda^h$ ,  $\Upsilon^h$ ) in
   decreasing price order in a list  $U$ 
4  repeat
5    for  $i \leftarrow 1$  to  $|L|$  in the  $L$ 
6      select the particular vehicle  $v$ 
7      if feasible position existed, find a feasible position with the highest profit to insert
8      else exit
9      end if
10     end for
11  until all the accepted and picked but unarrived platform orders (set  $G_v^h$ ) are served
12  repeat
13    for  $i \leftarrow 1$  to  $|Q|$  in the  $Q$ 
14      if feasible position existed, find a feasible position with the highest profit to insert
15      else exit the program
16      end if
17    end for
18  until all the accepted but unpicked and unarrived platform orders (set  $I^h$  and  $\Omega^h$ ) are
   served
19  repeat
20    for  $i \leftarrow 1$  to  $|U|$  in the  $U$ 
21      if feasible position existed, find a feasible position with the highest profit (profit > 0)
        to insert
22      else check next request
23      end if
24    end for
```

```

25  until all the platform orders that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and  $\Lambda^h$ ,  $\Upsilon^h$ ) are
    tried to insert
26  return  $S_{initial}$ 

```

首先，將所有已被餐廳與平台同時接受且已取餐但尚未送達的訂單（顧客節點集合： G_v^h ）依訂單編號由低到高排列。再將已被餐廳與平台同時接受但尚未取餐但尚未送達的訂單（虛擬取餐節點集合： I^h 與顧客節點集合： Ω^h ）依訂單編號由低到高排列。由於上述兩類訂單為不能拒絕服務之訂單，所以我們將該訂單的價值訂為 0，故插入順序僅以訂單編號作為插入順序。接著，待評估餐廳與平台是否接受的訂單（虛擬取餐節點集合 Φ^h 和 Ξ^h 與顧客節點集合 Λ^h 和 Υ^h ）依**訂單價值 (price)** 由高到低排列。

由訂單編號低的已被餐廳與平台同時接受且已取餐但尚未送達的訂單開始，將其插入指定車輛的可行且利潤最高的位置。再來，將已被餐廳與平台同時接受但尚未取餐但尚未送達的訂單，由訂單編號較低訂單開始，將其插入可行且利潤最高之位置。接著，將待評估餐廳與平台是否接受的訂單，由價格最高之訂單開始，插入可行且利潤最高且利潤大於 0 之位置，若該可拒絕服務的需求沒有可行的插入位置或是可行插入位置之利潤皆小於 0，則繼續檢查下一個待評估餐廳與平台是否接受的訂單，直到所有可拒絕服務的需求都被嘗試插入過。檢查插入位置的時窗是否可行之方式與子問題 1 的 IterILS 演算法相同。

SELECT_DES_REP (S)

```

1   $prob \leftarrow \text{RANDOM}(0,1)$ 
2  if  $(prob - P_1) \leq 0$  //亂數落於  $P_1$  中，選擇注重搜尋廣度的搜尋策略
3  |  $policy \leftarrow 1$ 
4  else
5  |  $policy \leftarrow 2$ 

```



```

6  end if
7  for each operator  $i$  do
8       $p_i \leftarrow \text{weight}_i / \sum_{j=1}^n \text{weight}_j$ 
9  end for
10 if  $\text{no\_imprv} < \delta$ 
11     pick and conduct a destroy operator and a repair operator based on the  $p_i$ 
        on  $S$  and get  $S'$ 
12 else
13     pick and conduct two destroy operator and a repair operator based on the on
         $S$  and get  $S'$ 
14 end if
15 if feasible
16     return  $S'$ 
17 else
18      $\text{no\_imprv} \leftarrow \text{no\_imprv} + 1$ 
19 end if

```

列 1-5：

依機率 P_1 和 P_2 選擇重視廣度或深度的搜尋策略

列 6-8：

對每個運算元 i ，計算其被選擇到的機率 p_i 。 $\sum_{j=1}^n \text{weight}_j$ 表示所有運算元 weight 的和（包含運算元 i ）。

列 9-12：

若 no_imprv 小於本研究所設定之最大連續不更新次數 δ ，則依運算元之 p_i 選擇一個破壞運算元和一個重建運算元對現有解 S 進行搜尋，完成搜尋後的解為 S' 。反之，則選擇兩個破壞運算元和一個重建運算元對現有解 S 進行搜尋，以期增加搜尋的廣度。

列 16-19：

若有可行解則回傳 S' ，反之則更新 $no_imprv \leftarrow no_imprv + 1$ 。

破壞運算元(destroy operator)

1. Random

```

1   $removed \leftarrow 0$ 
2  if the requests in the solution  $< \lceil \rho * num\_rq \rceil$ , exit
3  if  $policy = 1$ 
4      while  $removed < \lceil \rho * num\_rq \rceil$ 
5          remove 1 request in  $S$  randomly
6           $removed \leftarrow removed + 1$ 
7      end while
8  end if
9  if  $policy = 2$ 
10     while  $removed < \lceil \rho * num\_rq \rceil$ 
11         remove 1 platform order that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and  $\Lambda^h$ ,  $\Upsilon^h$ ) in  $S$  randomly
12          $removed \leftarrow removed + 1$ 
13     end while
14 end if

```

若解 S 中能夠移除的訂單數量小於欲移除訂單數量 $\lceil \rho * num_rq \rceil$ ，則跳出此函式，直接執行重建運算元插入尚未服務的訂單節點。

若 $policy$ 為 1，運算元將由現有解 S 隨機移出一個訂單（即所有虛擬取餐點集合： P^h 與所有顧客位置節點集合： D^h ）。若 $policy$ 為 2，運算元將由現有解 S 隨機移出待評估餐廳與平台是否接受的訂單（虛擬取餐節點集合： Φ^h 和 Ξ^h 與顧客節點集合： Λ^h 和 Υ^h ）。其中 ρ 代表欲移出訂單之比率， num_rq 表示現有解 S 內總訂單數量。

2. Least profit

```
1  removed  $\leftarrow$  0
2  if the orders in the solution  $< \lceil \rho * num\_rq \rceil$ , exit
3  if policy =1
4  | sort all the orders in  $S$  in the increasing profit as an list  $L$ 
5  if policy =2
6  | sort all the platform orders that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and
   |  $\Lambda^h$ ,  $\Upsilon^h$ ) in  $S$  in the increasing profit as an list  $L$ 
7  while removed  $< \lceil \rho * num\_rq \rceil$ 
8  |  $j \leftarrow \lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 
9  | remove request  $L_j$  from  $S$ 
10 | removed  $\leftarrow$  removed+1
11 end while
```

若解 S 中能夠移除的訂單數量小於欲移除訂單數量 $\lceil \rho * num_rq \rceil$ ，則跳出此函式，直接執行重建運算元插入尚未服務的訂單節點。

若 $policy$ 為 1，運算元將由現有解 S 移出一個在所有訂單中利潤第 j 小的訂單 (虛擬取餐節點集合： P^h 與顧客節點集合： D^h)。每個訂單的利潤定義為移出此訂單節點將會減少或增加整體利潤。本研究利用 $\lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 計算值 j ，避免每次移出之需求皆為該陣列之利潤最小之需求，以期增加搜尋之廣度。若 $policy$ 為 2，則是由現有解 S 移出一個利潤第 j 小之待評估餐廳與平台是否接受的訂單 (即虛擬取餐節點集合： Φ^h 、 Ξ^h 與顧客節點集合： Λ^h 、 Υ^h)。其中 ρ 代表將移出訂單之比率， num_rq 表示現有解 S 內總訂單數量。

3. Least paid

```

1   $removed \leftarrow 0$ 
2  if the orders in the solution  $< \lceil \rho * num\_rq \rceil$ , exit
3  if  $policy = 1$ 
4      sort all the orders in  $S$  in the increasing price in an array  $L$ 
5  if  $policy = 2$ 
6      sort all the platform orders that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and  $\Lambda^h$ ,  $\Upsilon^h$ ) in  $S$  in the increasing price in an array  $L$ 
7  while  $removed < \lceil \rho * num\_rq \rceil$ 
8       $j \leftarrow \lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 
9      remove order  $L_j$  from  $S$ 
10      $removed \leftarrow removed + 1$ 
11 end while

```

若解 S 中能夠移除的訂單數量小於欲移除訂單數量 $\lceil \rho * num_rq \rceil$ ，則跳出此函式，直接執行重建運算元插入尚未服務的訂單節點。若 $policy$ 為 1，運算元將由現有解 S 移出一個所有訂單中訂單價值 (price) 第 j 小的訂單 (虛擬取餐節點集合： P^h 與顧客節點集合： D^h)。本文利用 $\lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 計算值 j ，避免每次移出之需求皆為該陣列之利潤最小之需求，以期增加搜尋之廣度。若 $policy$ 為 2，則是由現有解 S 移出一個訂單價值第 j 小之待評估餐廳與平台是否接受的訂單 (即虛擬取餐節點集合： Φ^h 、 Ξ^h 與顧客節點集合： Λ^h 、 Υ^h)。其中 ρ 代表要移出訂單之比率， num_rq 表示現有解 S 中總訂單數量。

4. Most expensive

```

1   $removed \leftarrow 0$ 
2  if the requests in the solution  $< \lceil \rho * num\_rq \rceil$ , exit
3  if  $policy = 1$ 
4      sort all the orders in  $S$  in the decreasing cost in an array  $L$ 

```

```

5  if  $policy = 2$ 
6      sort all the platform orders that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and  $\Lambda^h$ ,  $\Upsilon^h$ ) in  $S$  in the decreasing cost in an array  $L$ 
7  while  $removed < \lceil \rho * num\_rq \rceil$ 
8       $j \leftarrow \lceil RANDOM(0,1)^{100*\rho} |L| \rceil$ 
9      remove request  $L_j$  from  $S$ 
10      $removed \leftarrow removed + 1$ 
11 end while

```

若解 S 中能夠移除的訂單數量小於欲移除訂單數量 $\lceil \rho * num_rq \rceil$ ，則跳出此函式，直接執行重建運算元插入尚未服務的訂單節點。

若 $policy$ 為 1，此運算元將由現有解 S 移出一個所有訂單中訂單成本 (cost) 第 j 大的訂單 (即虛擬取餐點集合： P^h 與顧客位置節點集合： D^h)。每個訂單的成本定義為移出此訂單將會減少或增加整體成本。本文利用 $\lceil RANDOM(0,1)^{100*\rho} |L| \rceil$ 計算值 j ，避免每次移出之需求皆為該陣列之利潤最小之需求，以期增加搜尋之廣度。

若 $policy$ 為 2，則是由現有解 S 移出一個訂單成本第 j 大之待評估餐廳與平台是否接受的訂單 (即虛擬取餐節點集合： Φ^h 、 Ξ^h 與顧客節點集合： Λ^h 、 Υ^h)。其中 ρ 代表要移出訂單之比率， num_rq 表示現有解 S 內總訂單數量。

5. Shaw

```

1  if the orders in the solution  $< \lceil \rho * num\_rq \rceil$ , exit
2  if  $policy = 1$ 
3      randomly select one order  $r$  from solution  $S$  and put in a set  $Z$ 
4      while  $|Z| < \lceil \rho * num\_rq \rceil$ 

```

```

5      sort the orders in the solution  $S$  as  $R(r,i) < R(r,j)$  in a list  $L$ , excluding the
      orders in set  $Z$ 
6       $j \leftarrow \lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 
7       $Z = Z \cup \{L_j\}$ 
8      remove  $\{L_j\}$  from list  $L$ 
9  end while
10 remove all orders of  $Z$  from  $S$ 
11 if  $policy = 2$ 
12     randomly select one platform order that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$ 
13     and  $\Lambda^h$ ,  $\Upsilon^h$ )  $r$  from solution  $S$  and put in a set  $Z$ 
14     while  $|Z| < \lceil \rho * num\_rq \rceil$ 
15         sort the orders in the solution  $S$  as  $R(r,i) < R(r,j)$  in a list  $L$ , excluding the
16         orders in set  $Z$ 
17          $j \leftarrow \lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 
18          $Z = Z \cup \{L_j\}$ 
19         remove  $\{L_j\}$  from list  $L$ 
20     end while
21 remove all orders of  $Z$  from  $S$ 

```

若解 S 中能夠移除的訂單數量小於欲移除訂單數量 $\lceil \rho * num_rq \rceil$ ，則跳出此函式，直接執行重建運算元插入尚未服務的訂單節點。

若 $policy$ 為 1，運算元將由現有解 S 隨機選擇一個與訂單 r 綜合相似度第 j 高之訂單（即虛擬取餐點集合： P^h 與顧客位置節點集合： D^h ）。若 $policy$ 為 2，運算元則由現有解 S 隨機選擇一個待平台評估餐廳與平台是否接受的訂單（即虛擬取餐點集合： Φ^h 、 Ξ^h 與顧客位置節點集合： Λ^h 、 Υ^h ）。同樣利用 $\lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 計算值 j ，移出綜合相似度第 j 高之訂單。

兩個訂單 r 與 i 間綜合相似度的定義為 $R(r,i)$ ，其公式為：

$$R(r,i) = \varphi(\text{dist}_{(pk_r, pk_i)} + \text{dist}_{(de_r, de_i)}) + \lambda(\text{arri}_{(pk_r, pk_i)} + \text{arri}_{(de_r, de_i)}) + \zeta(|\text{dem}_r - \text{dem}_i|)。$$

其中， $\varphi(\text{dist}_{(pk_r, pk_i)} + \text{dist}_{(de_r, de_i)})$ ，表示參數 φ 乘上兩個訂單虛擬取餐節點的距離與兩個訂單顧客節點的距離的總和 $\lambda(\text{arri}_{(pk_r, pk_i)} + \text{arri}_{(de_r, de_i)})$ ，表示參數 λ 乘上兩個訂單虛擬取餐節點的抵達時間之時間差與兩個訂單顧客節點的抵達時間之時間差的總和。由於本研究有已被餐廳與平台同時接受且平台已取餐但尚未送達之訂單對應的顧客節點，這類節點在目前規劃週期沒有對應的取餐節點，故會將該訂單的 $\text{dist}_{(pk_r, pk_i)}$ 與 $\text{arri}_{(pk_r, pk_i)}$ 設為 0。 $\zeta(|\text{dem}_r - \text{dem}_i|)$ 表示調整參數 ζ 乘上兩個訂單之餐點容量的差。其中 ρ 代表要移出訂單之比率， num_rq 表示現有解 S 中總訂單數量。

6. Price similarity

```

1  if the orders in the solution  $< \lceil \rho * \text{num\_rq} \rceil$ , exit
2  if policy =1
3      randomly select one order  $r$  from solution  $S$  and put in a set  $Z$ 
4      while  $|Z| < \lceil \rho * \text{num\_rq} \rceil$ 
5          sort the orders in the solution  $S$  as  $P(r,i) < P(r,j)$  in a list  $L$ , excluding the
            orders in set  $Z$ 
6           $j \leftarrow \lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 
7           $Z = Z \cup \{L_j\}$ 
8          remove  $\{L_j\}$  from list  $L$ 
9      end while
10     remove all orders of  $Z$  from  $S$ 
11 if policy =2
12     randomly select one platform order that haven't been decided yet (set  $\Phi^h, \Xi^h$ 
        and  $\Lambda^h, \Upsilon^h$ )  $r$  from solution  $S$  and put in a set  $Z$ 

```

```

13  while  $|Z| < \lceil \rho * num\_rq \rceil$ 
14      sort the orders in the solution  $S$  as  $P(r,i) < P(r,j)$  in a list  $L$ , excluding the
        orders in set  $Z$ 
15       $j \leftarrow \lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 
16       $Z = Z \cup \{L_j\}$ 
17      remove  $\{L_j\}$  from list  $L$ 
18  end while
19  remove all orders of  $Z$  from  $S$ 

```

若解 S 中能夠移除的訂單數量小於欲移除訂單數量 $\lceil \rho * num_rq \rceil$ ，則跳出此函式，直接執行重建運算元插入尚未服務的訂單節點。

若 *policy* 為 1，運算元將由現有解 S 隨機選擇一個與訂單 r 價格相似度第 j 高之訂單（即虛擬取餐點集合： P^h 與顧客位置節點集合： D^h ）。若 *policy* 為 2，運算元則由現有解 S 隨機選擇一個待平台評估餐廳與平台是否接受的訂單（即虛擬取餐點集合： Φ^h 、 Ξ^h 與顧客位置節點集合： Λ^h 、 Υ^h ）。同樣利用 $\lceil \text{RANDOM}(0,1)^{100*\rho} |L| \rceil$ 計算值 j ，移出價格相似度第 j 高之訂單。兩個訂單 r 與 i 間相似度的定義為 $P(r,i)$ ，公式為： $P(r,i) = |price_r - price_i|$ 。其中 ρ 代表移出訂單之比率， num_rq 表示現有解 S 中總訂單數量。

重建運算元(Repair operator)

Greedy ()

```

1  if policy =1
2      sort unserved orders in decreasing price as a list  $Y$ 
3      for  $i \leftarrow 1$  to  $num\_un\_rq$  in list  $Y$ 
4          if feasible position existed, find a feasible position with the highest profit
            to insert

```



```

5      else
6          if order  $i$  is the accepted order (set  $G^h$ ,  $I^h$  and  $\Omega^h$ )
7               $no\_imprv \leftarrow no\_imprv + 1$ 
8              exit (to line 22 in ALNS())
9          end if
10     end if
11 end for
12 end if
13 if  $policy = 2$ 
14     sort the accepted and picked but unrarried platform orders (set  $G_v^h$ ) in
        increasing order number as a list  $U$ 
15     for  $i \leftarrow 1$  to  $num\_p\_rq$ 
16         if feasible position existed, find a feasible position with the highest profit
            to insert
17         else
18              $no\_imprv \leftarrow no\_imprv + 1$ 
19             break (to line 22 in ALNS())
20         end if
21     end for
22     sort the accepted but unpicked and unrarried platform orders (set  $I^h$  and
         $\Omega^h$ ) in increasing order number as a list  $P$ 
23     for  $i \leftarrow 1$  to  $num\_a\_rq$ 
24         if feasible position existed, find a feasible position with the highest profit
            to insert
25         else
26              $no\_imprv \leftarrow no\_imprv + 1$ 
27             break (to line 22 in ALNS())
28         end if
29     end for
30     sort the platform orders that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and  $\Lambda^h$ ,
         $\Upsilon^h$ ) in decreasing price as a list  $Q$ 
31     for  $i \leftarrow 1$  to  $num\_sl\_rq$ 
32         if feasible position existed, find a feasible position with the highest profit
            to insert
33         else

```

```

34 | | | check next order
35 | | end if
36 | end for
37 end if

```

列 1-12：

policy 為 1，首先會將所有未服務之訂單（虛擬取餐節點集合： P^h 與顧客節點集合： D^h ）依**訂單價值 (price)** 由高至低排列。從**訂單價值**最高的開始，找出其中利潤最高的可行虛擬取餐節點與顧客節點插入位置，將虛擬取餐與顧客節點插入現有解中。檢查插入位置的時窗是否可行之方式與子問題 1 的 IterILS 演算法相同。若已被餐廳與平台同時接受的訂單（虛擬取餐節點集合： I^h 與顧客節點集合： Ω^h 、 G_v^h ），則更新未更新次數 $no_imprv \leftarrow no_imprv + 1$ 。並跳出函式，到 ALNS() 主架構的 22 列，更新模擬退火溫度 T 。其中 num_un_rq 表示所有尚未服務的訂單。

列 13-37：

policy 為 2，首先將尚未服務的已被餐廳與平台同時接受已取餐但尚未送達的訂單（即顧客節點集合： G_v^h ）依**訂單編號**由低至高排列。由於這類訂單一定要服務，所以我們將這類訂單的價格訂為 0。因此，這類訂單的插入順序為從**訂單編號**最低的開始，找出其中利潤最高的可行虛擬取餐點與顧客位置節點插入位置，將虛擬取餐與顧客位置節點插入現有解中。檢查插入位置的時窗是否可行之方式與子問題 1 的 IterILS 演算法相同。若沒有可行的插入位置，則更新未更新次數 $no_imprv \leftarrow no_imprv + 1$ 。並跳出函式，到 ALNS() 主架構的 22 列，更新模擬退火溫度 T 。

再將尚未服務的已被餐廳與平台同時接受但尚未取餐與送達的訂單（即虛擬取餐點集合： I^h 與顧客位置節點集合： Ω^h ）依**訂單編號**由低至高排列。如同上述，這類訂單的訂單價值為 0。於是，這類訂單的插入順序為從**訂單編號**最低的開始，找出其中利潤最高的可行虛擬取餐節點與顧客節點插入位置，將虛擬取餐與顧客節點插入

現有解中。檢查插入位置的時窗是否可行之方式與子問題 1 的 IterILS 演算法相同。若沒有可行的插入位置，則更新未更新次數 $no_imprv \leftarrow no_imprv + 1$ 。並跳出函式，到 ALNS() 主架構的 22 列，也就是更新模擬退火溫度 T 。

最後是將尚未服務的待平台評估餐廳與平台是否接受的訂單（即虛擬取餐點集合： Φ^h 、 Ξ^h 與顧客節點集合： Λ^h 、 Υ^h ）依**訂單價值**由高至低排列。從**訂單價值**最高的開始，找出其中利潤最高的可行虛擬取餐點與顧客位置節點插入位置，將虛擬取餐與顧客位置節點插入現有解中。檢查插入位置的時窗是否可行之方式與子問題 1 的 IterILS 演算法相同。若沒有可行的插入位置，則繼續嘗試插入下一個訂單。

其中 num_p_rq 、 num_a_rq 、 num_sl_rq 分別代表尚未服務的已被餐廳與平台同時接受已取餐但尚未送達的訂單（即顧客位置節點集合： G^h ）、尚未服務的已被餐廳與平台同時接受但尚未取餐與送達的訂單（即虛擬取餐點集合： I^h 與顧客位置節點集合： Ω^h ）與尚未服務的待平台評估餐廳與平台是否接受的訂單（即虛擬取餐點集合： Φ^h 、 Ξ^h 與顧客位置節點集合： Λ^h 、 Υ^h ）數量。

Regret ()

```

1  if  $policy = 1$ 
2      sort unserved orders in decreasing regret value as a list  $Y$ 
3      for  $i \leftarrow 1$  to  $num\_un\_rq$  in list  $Y$ 
4          if feasible position existed, find a feasible position with the highest profit
            to insert
5          else
6              if request  $i$  is the accepted order ( $G^h, I^h, \Omega^h$ )
7                   $no\_imprv \leftarrow no\_imprv + 1$ 
8                  break (to line 22 in ALNS( ))
9              end if
10         end if
11     end for
12 end if
13 if  $policy = 2$ 

```

```

14  sort the accepted and picked but unrarried platform orders ( $G_v^h$ ) in decreasing
    regret value as a list  $U$ 
15  for  $i \leftarrow 1$  to  $num\_p\_rq$ 
16      if feasible position existed, find a feasible position with the highest profit
        to insert
17      else
18           $no\_imprv \leftarrow no\_imprv + 1$ 
19          break (to line 22 in ALNS())
20      end if
21  end for

    sort the accepted but unpicked and unrarried platform orders ( $I^h$  and  $\Omega^h$ )
    in decreasing regret value as a list  $P$ 
23  for  $i \leftarrow 1$  to  $num\_a\_rq$ 
24      if feasible position existed, find a feasible position with the highest profit
        to insert
25      else
26           $no\_imprv \leftarrow no\_imprv + 1$ 
27          break (to line 22 in ALNS())
28      end if
29  end for

    sort the platform orders that haven't been decided yet (set  $\Phi^h$ ,  $\Xi^h$  and  $\Lambda^h$ ,
     $\Upsilon^h$ ) in decreasing regret value as a list  $Q$ 
31  for  $i \leftarrow 1$  to  $num\_sl\_rq$ 
32      if feasible position existed, find a feasible position with the highest profit
        to insert
33      else
34          check next order
35      end if
36  end for
37 end if

```

Regret() 插入訂單節點的方式與 Greedy()幾乎相同，唯一不同的地方是，在排列訂單的插入順序時，是依後悔值 (regret value) 由高至低排列。訂單後悔值表示該訂單節點的利潤最高插入位置與利潤次高插入位置之差。

UPDATE_SC ()

```
1  create list  $\Omega$ 
2  transform  $S$  and put in list  $\Omega$ 
3  if the condition  $f(S') > f(S_{best})$  is used in ALNS line 12
4  |   update selected destroy operator  $i$  and repair operator  $j$  score by  $v_1$ 
5  else if the condition  $f(S') > f(S)$  is used in ALNS line 12 and  $S$  is not in list  $\Omega$ 
6  |   update selected destructor  $i$  and repair  $j$  score by  $v_2$ 
7  |   else if condition  $f(S') > f(S)$  is used in ALNS line 12 and  $S$  is in list
8  |    $\Omega$ 
9  |   |   update selected destructor  $i$  and repair  $j$  score by  $v_3$ 
10 |   |   else if condition  $\text{RANDOM}(0,1) < e^{(f(S')-f(S))/T}$  is used in ALNS line 12
11 |   |   |   update selected destructor  $i$  and repair  $j$  score by  $v_4$ 
12 |   |   |   end if
13 |   |   end if
14 |   end if
15 end if
```

此函式表示更新破壞運算元 (destroy operator) 與重建運算元 (repair operator) 分數的方式。

列 2：

表示，陣列 Ω 將會存取每一個更新後的現有解 S ，做為替運算元加分的依據。但為了增加運算速度，僅存目標值，服務的顧客數與被使用的外送員數目。

列 3-15：

表示下列 4 種情形

- 若於 ALNS()列 10 利用 SELECT_DES_REP (S) 函式所搜尋到的解 S' 比目前所搜尋到的最佳解好； $f(S') > f(S_{best})$ ，則為此破壞運算元與重建運算元之分數加 v_1 分。
- 若於 ALNS()列 10 利用 SELECT_DES_REP (S) 函式所搜尋到的解 S' 較現有解 S 佳，且尚未存在於陣列 Ω 中，則為此破壞運算元與重建運算元之分數加 v_2 分。
- 若於 ALNS()列 10 利用 SELECT_DES_REP (S) 函式所搜尋到的解 S' 較現有解 S 佳，但是已存在於陣列 Ω 中，則為此破壞運算元與重建運算元之分數加 v_3 分。
- 若於 ALNS()列 10 利用 SELECT_DES_REP (S) 函式所搜尋到的解，符合模擬退火門檻的更新條件； $\text{RANDOM}(0,1) < e^{(f(S')-f(S))/T}$ ，則為此破壞運算元與重建運算元之分數加 v_4 分。

UPDATE_WEI_SC ()

$$\begin{array}{ll}
 1 & \text{Update } weight_{i, seg+1} = \begin{cases} weight_{i, seg}, & \text{if } \beta_{i, seg} = 0 \\ (1-\eta) * weight_{i, seg} + \eta * \beta_{i, seg} / \alpha_{i, seg}, & \text{otherwise} \end{cases} \\
 2 & score_i \leftarrow 0
 \end{array}$$

此函式表示更新下一個 seg 的運算元權重值，其中：

- η 為調整參數
- $\alpha_{i, seg}$ 為運算元 i 被呼叫的次數

- $\beta_{i, seg}$ 為在這個 seg ，運算元 i 相對應的 $score_i$

最後初始化 $score_i$ 。

LS()

路徑內重新插入法 (Intraroute relocate) :

同一條路徑內，將一個訂單的虛擬取餐節點與顧客節點拔出，再重新插入於該路徑中。

路徑間重新插入法 (Interoute relocate) :

不同條路徑間，將一個訂單的虛擬取餐節點與顧客節點拔出，再重新插入另一條路徑中。在選擇拔出的訂單時，由於已被餐廳與平台同時接受且已取餐但尚未送達之訂單必須被該取餐外送員服務，所以不會被選擇為候選拔出訂單。

路徑內交換法 (Intraroute exchange) :

同一條路徑內，兩個訂單彼此的虛擬取餐節點交換與彼此的顧客節點交換。

路徑間交換法 (Interoute exchange) :

不同條路徑間，兩個訂單彼此的虛擬取餐節點交換與彼此的顧客節點交換。在選擇交換的訂單時，必須被某個車輛服務的節點不會被選擇為候選交換節點。

可拒絕需求之拔出法 (Selective request removal) :

將一個可拒絕的訂單，即待平台評估餐廳與平台是否接受之訂單，從路線中拔出。

可拒絕需求之插入法 (Selective request insertion) :

將一個在可拒絕的訂單，待平台評估餐廳與平台是否接受之訂單，插入路線中。

此函式將會隨機選擇三個交換或插入法執行，其中必須包含一個**可拒絕訂單 (selective request)** 的交換或插入法。在每一個交換或插入法搜尋所有可行的交換或插入位置，找出最好的位置做交換或插入更新。待無法搜尋到更好的位置進行交換或插入更新後，則執行下一個交換或插入法。

6 實例分析

本研究利用 Gurobi 求解第 4 章的數學模型，並利用 c++ 編寫第 5 章提及的 IterILS 與 ALNS 演算法，而所有的例題是在配有 Intel (R) Core(TM) i7-8700 CPU 3.20 GHz、16 GB RAM 與 64 位元的 Window 10 電腦上執行。首先，本研究建立 6 組小例題，比較 Gurobi 求出的解與演算法求出的解，藉此分析演算法之效能。接著利用演算法求解規模近似於實務平台資料的例題，分析本研究應用於實務上的結果。

6.1 小例題分析

本研究建立 6 個小例題題組，根據實際資料，餐廳可以依照其座位數分為 3 種規模，每個餐廳規模所包含的平行機台數與餐點種類和餐點製作時間不盡相同。小例題參考實際資料的餐廳規模與其餐點種類和對應的製作時間，由於在小例題中生產網路節點數量不多，所以將各個規模對應的平行機台數縮小為 1、2 與 3，並隨機產生各個餐廳規模內每個餐點種類在每分鐘最多可製作的數量。

此外，目前各大外送平台的手續費約餐點價格的 30%，若餐廳與平台都接受該訂單餐點，餐廳必須付給平台餐點價格 30% 的手續費，亦即平台可以獲得餐點價格 30% 的收入，而餐廳則可獲得餐點價格 70% 的收入。如同前述，本研究設定若平台評估餐廳可以接受訂單，但平台無法配送該訂單，平台必須付給餐廳賠償金，賠償金表示餐廳損失的收入，所以本研究亦將賠償金設為餐點價格的 70%。

而在小例題中，假設平台僅有 4 位外送員進行配送，且每位外送員的車廂容量為 10 便當盒單位，運輸成本則是依中油公告 2018 年 1 月 95 無鉛汽油價格 26.4/公升，工研院車輛耗能研究網公告 125cc 機車在市區時速 40 公里的情況下平均油耗大約為 44 公里/公升計算而得，每行駛 1 分鐘成本為 0.4 元。餐廳與平台相關資訊如表 6.21 與表 6.2 所示。

表 6.1 小例題之餐廳與餐點資訊

規模	平行 機台數	餐點 種類	製作 時間	賞味 期限	每分鐘最多 可製作數量	價格	餐廳 收入	平台 賠償金	平台 收入
1	1	A	10	20	1	160	105	105	45
		B	6	20	1	150	133	133	57
		C	3	25	1	70	49	49	21
2	2	A	10	20	2	190	133	133	57
		B	15	20	2	230	161	161	69
		C	15	20	2	220	154	154	66
		D	15	20	2	200	140	140	60
		E	3	25	1	170	119	119	51
		F	6	20	1	90	63	63	27
3	3	A	10	20	3	200	140	140	60
		B	15	20	3	240	168	168	72
		C	15	20	3	270	189	189	81
		D	15	20	3	260	182	182	78
		E	3	25	2	200	140	140	60
		F	6	20	2	140	98	98	42
		G	15	20	3	450	315	315	135
		H	3	25	1	100	70	70	30

表 6.2 外送平台資訊

外送員數量	外送員車廂容量	運輸成本
4	10	0.4 / 分鐘

本研究設計 6 種不同的例題，以表示實務上的不同情境。例題 1~3 包含 2 個餐廳，例題 4~6 則是包含 3 家餐廳。餐廳的規模為隨機抽選，並且設計不同的已被餐廳與平台接受與待平台評估是否接受之訂單數量的組合，以測試 Gurobi 與 IterILS 和 ALNS 在不同情境下的績效表現。各例題的詳細資料如表 6.3 所示。

表 6.3 本研究設計之小例題資料

	餐廳	規模	已被接受 內用與外送訂單數量	待評估 外送訂單數量	生產網路 節點數量	已被接受 外送訂單數量	待評估 外送訂單數量	配送網路 節點數量
例題1	1	1	2	2	6	3	4	16
	2	2	2	2	7			
例題2	1	1	3	1	5	1	2	9
	2	2	3	1	6			
例題3	1	2	2	3	7	3	6	20
	2	3	2	3	6			
例題4	1	1	2	2	7	4	6	21
	2	1	2	3	6			
	3	3	2	3	7			
例題5	1	1	3	1	4	3	3	13
	2	1	3	2	5			
	3	1	3	2	5			
例題6	1	1	1	3	7	3	9	25
	2	3	1	3	7			
	3	3	1	3	6			

首先，本研究先分別利用 Gurobi 與演算法 IterILS 和 ALNS 獨立測試兩個子問題，以了解 IterILS 和 ALNS 演算法應用在單獨一個子問題的表現。演算法使用的參數詳見附錄。由於 IterILS 和 ALNS 演算法會隨機選擇拔除節點與插入節點，所以在使用 IterILS 演算法與 ALNS 演算法進行例題測試時，本研究會獨立測試每個例題 10 次，並取其平均求解時間與平均求解結果。

子問題 1 與 2 測試結果如下表，可發現，本研究之演算法在每個子問題 1 的例題皆可以求得與 Gurobi 求得之最佳解相同的目標值，並且求解時間皆較 Gurobi 求解時間短，如求解時間差異最大的例題 1，僅須 Gurobi 求解時間的 0.12% ($0.0084 / 7.2$)。

而利用本研究演算法求解子問題 2，求得的解與 Gurobi 求得的解皆可在 1.6 % 內，甚至在較小的例題，如例題 2，可求得最佳解。而在求較複雜的例題，如 Gurobi 在 1 小時內也求不到最佳解的例題 6，目標值與 Gurobi 在一小時內求得的解僅差 0.54 %，而演算法求解時間也僅為 Gurobi 求解的 0.03% ($0.0084 / 7.2$)。

表 6.4 子問題 1 的 Gurobi 求出的解與 IterILS 和 ALNS 演算法求出的解之比較

子問題1	餐廳	Gurobi			IterILS & ALNS			Gap (%)
		目標值	拒絕服務的節點	求解時間 (sec)	平均目標值	拒絕服務的節點	平均求解時間 (sec)	
例題1	1	187	無	7.2	187	無	0.0084	0%
	2	113	3, 4, 5		113	3, 4, 5		0%
例題2	1	106	無	0.16	106	無	0.008	0%
	2	0	4, 5, 6		0	4, 5, 6		0%
例題3	1	324	無	0.66	324	無	0.0153	0%
	2	351	無		351	無		0%
例題4	1	161	3, 4, 5	5.43	161	3, 4, 5	0.0145	0%
	2	131	無		131	無		0%
	3	379	無		379	無		0%
例題5	1	56	無	0.26	56	無	0.0055	0%
	2	112	無		112	無		0%
	3	109	無		109	無		0%
例題6	1	293	無	1.34	293	無	0.0226	0%
	2	621	無		621	無		0%
	3	488	無		488	無		0%

表 6.5 子問題 2 的 Gurobi 求出的解與 IterILS 和 ALNS 演算法求出的解比較

子問題2	Gurobi			IterILS & ALNS			Gap (%)
	目標值	拒絕服務的節點	求解時間 (sec)	平均目標值	拒絕服務的節點	平均求解時間 (sec)	
例題1	217	無	1.09	215.4	無	0.1931	0.737%
例題2	130.2	無	0.04	130.2	無	0.0718	0%
例題3	267.4	無	595.58	263.16	無	0.3658	1.586%
例題4	407.6	無	35.88	405.84	無	0.4297	0.432%
例題5	101.4	無	0.14	99.8	無	0.1117	1.578%
例題6	555.8* (Best bound: 559.4) (Best bound gap: 0.644%)	無	3600*	552.8	無	0.9544	0.54%

(*表示 Gurobi 在時間限制內求到的近似解，Best bound 表示 Gurobi 求得該例題解之上界，Best bound gap 則表示該近似解與 Best bound 的目標值差距)
(最右欄之 Gap 表示 Gurobi 求得的解與 IterILS & ALNS 演算法求得的解之目標值差距)

表 6.6 則為分別使用 Gurobi 與 IterILS 和 ALNS 演算法求解整合生產與配送問題的例題之結果。同樣的在使用 IterILS 演算法與 ALNS 演算法進行例題測試時，本研究會獨立測試每個例題 10 次，並取其平均求解時間與平均求解結果。

由於生產排程問題的餐點完成時間會影響後續配送路線規劃問題之取餐與送達時間，也同時影響整合問題的解品質。所以一般而言，我們會傾向讓訂單與訂單間的餐點完成時間越為分散，這樣的作法對於配送路線規劃排程越為有利。所以，在測試整合問題時，我們將子問題 1 IterILS 演算法的全部的節點插入策略更改為，外送餐點在決定插入位置時採隨機插入的方式。

此外，根據 Gurobi 求解結果發現，本問題的複雜度與餐廳數目和外送訂單呈正相關，如：例題 1 為 2 家餐廳，共 4 張外送訂單、例題 3 為 2 家餐廳共 6 張外送訂單、例題 4 為 3 家餐廳，共 8 張外送訂單、例題 6 為 3 家餐廳共 9 張外送訂單，一旦餐廳數目或外送訂單越多，求解時間越長，也越難找到最佳解。其中例題 3、4 和 6，Gurobi 在一小時內仍無法求到最佳解。

但本研究設計的 IterILS 和 ALNS 演算法在求解複雜度較高的例題，如上述的例題 3、4 和 6，平均僅使用 0.1722、0.2033 與 0.2535 秒，便可以求到與 Gurobi 在 1 小時內求得的解僅差 6.078、3.251%與 6.153%的解。同時也表示本研究在求解較為複雜的整合生產與配送問題時，同樣能保有不錯的求解品質。

表 6.6 整合問題的 Gurobi 求得的解與 IterILS 和 ALNS 演算法求得的解比較

	Gurobi					IterILS & ALNS					Gap (%)	
	子問題 1		子問題 2		求解時間 (sec)	子問題 1		子問題 2		平均 求解時間 (sec)		
	目標值	拒絕服務的 節點	目標值	拒絕服務的 節點		平均 目標值	拒絕服務的 節點	平均 目標值	拒絕服務的 節點		子問題 1	子問題 2
例題1	187	無	116	9, 14	364.21	187	無	113.6	9, 14	0.1299	0%	2.069%
	113	3, 4, 5				113	3, 4, 5					
例題2	106	無	32.4	6, 8	0.28	106	無	32.4	6, 8	0.0592	0%	0%
	0	4, 5, 6				0	4, 5, 6					
例題3	324	無	262.6* (Best bound: 266.6) (Gap: 1.5%)	無	3600*	324	無	246.64	無	0.1722	0%	6.078%
	351	無				351	無					
例題4	161	3, 4, 5	268.2* (Best bound: 271.2) (Gap: 1.106%)	8, 15	3600*	161	3, 4, 5	259.48	8, 15	0.2033	0%	3.251%
	131	無				131	無					
	379	無				379	無					
例題5	56	無	101.4	無	0.95	56	無	100.2	無	0.1017	0%	1.183%
	112	無				112	無					
	109	無				109	無					
例題6	293	無	565.6* (Best bound: 581.5) (Gap: 2.734%)	無	3600*	293	無	530.8	無	0.2535	0%	6.153%
	621	無				621	無					
	488	無				488	無					

(*表示 Gurobi 在時間限制內求到的近似解，Best bound 表示 Gurobi 求得該例題解之上界，Best bound gap 則表示該近似解與 Best bound 的目標值差距)
(最右欄之 Gap 表示 Gurobi 求得的解與 IterILS & ALNS 演算法求得的解之目標值差距)

6.2 實務例題分析

● 實務例題建立方式

T 公司為一家在台北經營餐點外送平台之公司。本研究利用 T 公司提供之實務訂單資料建立實務例題，經由與 T 公司的訪談可知，一般而言，餐點外送平台在晚上擁有較多訂單，中午訂單與晚上訂單比例約為 1:2。表 6.7 為 T 公司每日晚上的訂單資訊。

表 6.7 T 公司提供之實務資料

T公司訂單資料	星期一晚上	星期二晚上	星期三晚上	星期四晚上	星期五晚上	星期六晚上	星期日晚上
訂單數量	9	6	7	7	11	12	13
單日晚上訂單數量/ 一周晚上平均訂單數量	0.969	0.646	0.754	0.754	1.185	1.292	1.4

接著，由於實務上空腹熊貓與優步快送每日全台的訂單量可達到約 3000 張。為了表現出本研究應用於實務上的結果，我們利用 T 公司的訪談資訊，計算出空腹熊貓全台每日晚上的訂單量約為 2000 張。而空腹熊貓在全台合作餐廳約為 1300 家，在台北市的合作餐廳約為 600 家，經由計算後，可以得到空腹熊貓每日晚上在台北的訂單數平均約為 923 張訂單 ($3000 * 600 / 1300 * 2/3$)。而再由 T 公司的資訊可知每日晚上訂單數占一周晚上訂單總訂單數的比例皆不同，我們利用該係數估算空腹熊貓在每日晚上的訂單數目。結果如表 6.8。

表 6.8 本研究模擬之空腹熊貓實務資料

模擬空腹熊貓 訂單資料	星期一晚上	星期二晚上	星期三晚上	星期四晚上	星期五晚上	星期六晚上	星期日晚上
訂單數量	895	596	696	696	1093	1193	1292

而每張訂單對應的餐廳則是根據空腹熊貓的網站，在星期天晚上實際有營業的台北市餐廳為 589 家，分布位置如圖 6.1。

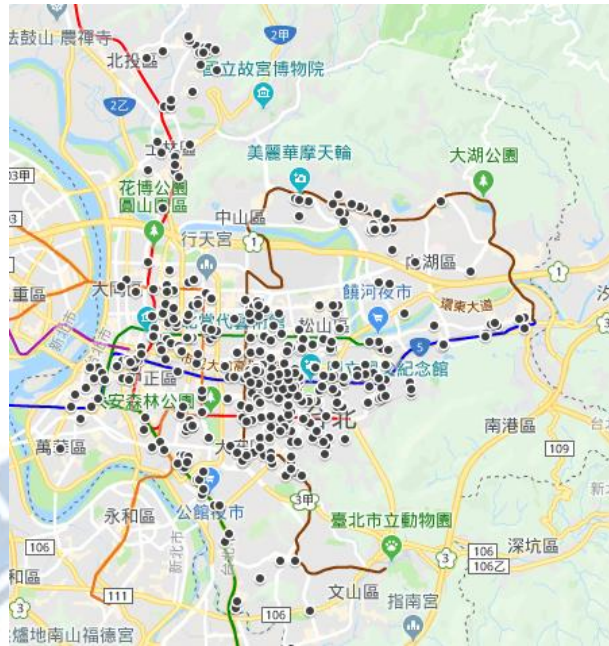


圖 6.1 空腹熊貓於台北市的合作餐廳分布圖

而由於空腹熊貓網站上並無顧客資訊，所以我們利用優步快送所公布的送餐熱區，將其設為本研究的顧客位置。共有 360 個顧客位置，分布位置如圖 6.2。

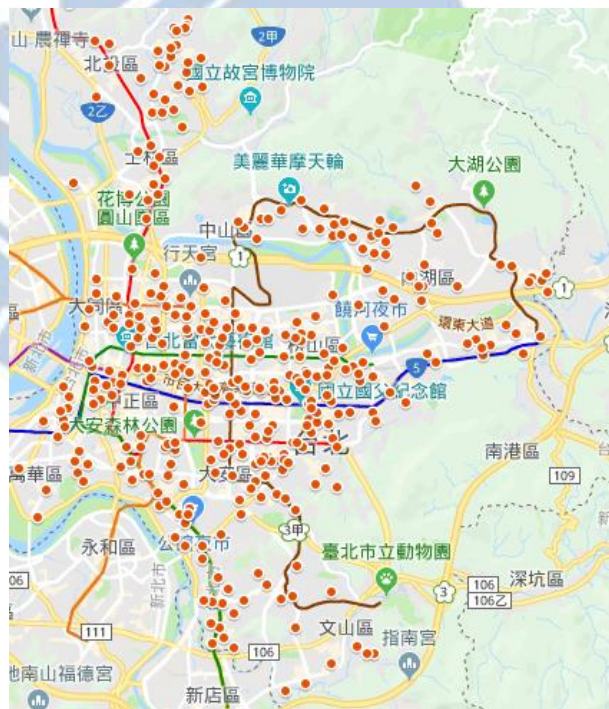


圖 6.2 優步快送於台北市的外送員送餐熱區分布圖

每張訂單對應的餐廳則是根據該餐廳的評價數占總餐廳評價數的比率，作為選擇的機率。而顧客則是根據該區的熱度作為選擇機率，優步快送將不同顧客區域依照其區域熱度，即顧客訂單數，分為 3 種顏色，即熱度 1、2 和 3，我們即以該顧客熱度占總顧客熱度的比率，作為選擇顧客的機率。此外，本研究假設顧客訂購行為服從卜松過程 (Poisson Process)，即事件發生的時間間隔 (time interval) 服從指數分布(exponential distribution)，事件發生的數量服從卜松分布 (poisson distribution)。一般而言，卜松過程可以用來描述等候理論 (queuing theory) 的行為 (Forbes et al., 2011)，而本研究的顧客向平台點餐的過程，也類似於等候理論內顧客到某櫃台等候服務的行為。

由表 6.8 可知，顧客在星期日晚上的訂單數量是一週內的高峰，於是我們選擇星期天晚上的訂單資料進行實例測試與分析。此外，由 T 公司的實務資料可知，顧客在星期天晚上點餐的時間間隔的平均值約為 13 分鐘，見表 6.9 T 公司星期天晚上之訂單點餐時間。

表 6.9 T 公司星期天晚上之訂單點餐時間

訂單編號	點餐時間	時間間隔 (min) (interarrvial time)
1	17:05	5
2	17:21	16
3	17:24	3
4	17:24	0
5	17:38	14
6	17:39	1
7	17:42	3
8	17:50	8
9	17:53	3
10	18:05	12
11	18:15	10
12	18:54	39
13	19:50	56
		Avg: 13

利用該值模擬出星期天晚上的顧客點餐時間與對應的訂單數量如表 6.10。

表 6.10 本研究模擬之空腹熊貓星期天晚上之顧客點餐時間與訂單數量

顧客點餐時間	17:21	17:36	17:52	18:17	18:25	18:42	19:10	19:20	19:25	19:49
訂單數量	128	144	121	133	139	123	121	131	128	130

此外，由於本研究必須考慮預測的訂單，所以利用相同的參數，模擬出一組預測外送訂單的資料，表 6.11。

表 6.11 本研究預測之空腹熊貓星期天晚上之顧客點餐時間與訂單數量

顧客點餐時間	17:14	17:29	17:40	18:34	19:02	19:21	19:43
訂單數量	150	171	178	155	159	149	152

根據 3.2 節的決策時段劃分方式，表 6.10 與表 6.11 的訂單對應的規劃時段可整理成 10 個題組。而事實上，每個題組在計算時均為獨立計算，所以本研究僅列出餐廳晚餐營業時間的第一個小時，也就是題組 1、2 與 3 的結果作為代表。

表 6.12 本研究之實務例題對應之題組

題組編號	對應已規劃時段	對應目前規劃時段	對應預測時段
1	17:00-17:20	17:20-17:24	17:24-18:24
2	17:00-17:36	17:36-17:40	17:40-18:40
3	17:00-17:52	17:52-17:56	17:56-18:56
4	17:00-18:16	18:16-18:20	18:20-19:20
5	17:00-18:24	18:24-18:28	18:28-19:28
6	17:00-18:40	18:40-18:44	18:44-19:44
7	17:00-19:08	19:08-19:12	19:12-20:00
8	17:00-19:20	19:20-19:24	19:24-20:00
9	17:00-19:24	19:24-19:28	19:28-20:00
10	17:00-19:48	19:48-19:52	19:52-20:00

此外，外送訂單的餐點則是根據訂單的餐廳而來，實務例題的餐廳資訊如下表：

表 6.13 實務例題之餐廳與餐點資訊

規模	平行 機台數	餐點 種類	製作 時間	賞味 期限	每分鐘最多 可製作數量	價格	餐廳 收入	平台 賠償金	平台 收入
1	1	A	10	20	8	160	105	105	45
		B	6	20	2	150	133	133	57
		C	3	25	3	70	49	49	21
2	2	A	10	20	8	190	133	133	57
		B	15	20	12	230	161	161	69
		C	15	20	9	220	154	154	66
		D	15	20	6	200	140	140	60
		E	3	25	4	170	119	119	51
		F	6	20	4	90	63	63	27
3	3	A	10	20	16	200	140	140	60
		B	15	20	18	240	168	168	72
		C	15	20	16	270	189	189	81
		D	15	20	12	260	182	182	78
		E	3	25	8	200	140	140	60
		F	6	20	6	140	98	98	42
		G	15	20	12	450	315	315	135
		H	3	25	4	100	70	70	30

我們依照餐廳座位數將與空腹熊貓合作的餐廳分為 3 種規模，大規模餐廳有約 50 個座位，中規模餐廳約有 25 個座位，小規模餐廳約有 15 個座位。每個餐廳規模的餐點種類依照餐廳的菜單大約可以分為大規模餐廳有 8 種餐點，中規模餐廳有 6 種餐點，而小規模餐廳有 3 種餐點。如：某大規模餐廳提供義大利麵、燉飯、焗烤、披薩、排餐、甜點、沙拉與炸物，8 種餐點。而餐點的每分鐘最多可製作數量，則是由各餐廳的工作站廚師數目與餐點的製作步驟計算而來，見附錄。

經過 T 公司提供的部分資料，每張訂單餐點數量最多可以達到 18 道餐點，所以每張外送訂單的餐點數量則是由 1 到 18 隨機抽樣，而每個餐點的餐點種類則是由該餐廳的餐點種類隨機抽一個種類作為該餐點的餐點種類。

每個餐廳的內用顧客點餐行為同樣假設服從顧客訂購行為服從卜松過程 (Poisson Process)，由於缺少餐廳的實際訂單資料，本研究假設內用與外送顧客點餐時間大致相同，所以我們利用 T 公司一周所有晚上訂單資料的點餐時間間

隔平均數作為內用顧客抵達餐廳點餐的時間間隔平均數，約為 2 分鐘。並且使用 Google 提供的到店用餐人數比率與餐廳座位數，模擬每個內用顧客的點餐時間。

表 6.14 T 公司一周晚上之訂單點餐時間

訂單編號	點餐時間	時間間隔 (min) (interarrvial time)
1	17:01	1
2	17:02	1
3	17:05	3
4	17:06	1
5	17:07	1
6	17:09	2
7	17:14	5
8	17:16	2
9	17:19	3
10	17:19	0
11	17:19	0
12	17:21	2
13	17:21	0
14	17:24	3
15	17:24	0
16	17:26	2
17	17:30	4
18	17:32	2
19	17:32	0
20	17:35	3
21	17:38	3
22	17:39	1
23	17:39	0
24	17:42	3
25	17:46	4
26	17:48	2
27	17:49	1
28	17:50	1
29	17:51	1
30	17:53	2
31	17:56	3
32	17:58	2
33	17:59	1
34	18:04	5
35	18:05	1
36	18:09	4
37	18:10	1
38	18:11	1
39	18:15	4
40	18:16	1
41	18:17	1
42	18:18	1
43	18:20	2
44	18:24	4
45	18:26	2
46	18:26	0
47	18:28	2
48	18:31	3
49	18:34	3
50	18:34	0
51	18:35	1
52	18:42	7
53	18:46	4
54	18:48	2
55	18:50	2
56	18:54	4
57	18:55	1
58	18:56	1
59	19:03	7
60	19:09	6
61	19:25	16
62	19:26	1
63	19:50	24
64	19:54	4
65	20:00	6
		Avg: 2

此外，一般而言，每個內用顧客一次僅會點一人份的餐點，所以假設每張內用訂單僅包含一道餐點。表 6.15 為每個題組包含的餐廳數、餐廳的生產節點

數目與配送網路節點數目。表 6.16 為每個餐廳的生產網路節點數分別包含多少內用餐點與外送餐點數目。詳細的生產網路資料與配送網路資料附於附錄。

表 6.15 實務例題之餐廳數、生產網路節點數與配送網路節點數

題組編號	餐廳數	外送訂單數	餐廳生產節點數			配送網路節點數
			最大值	最小值	平均數	
1	93	297	107	8	32	894
2	93	375	84	10	40	1070
3	91	387	94	14	39	1104

表 6.16 實務例題之每個餐廳的內用與外送餐點段應的生產網路節點數

題組編號	內用餐點			外送餐點		
	最大值	最小值	平均數	最大值	最小值	平均數
1	55	3	16	64	1	17
2	69	5	24	60	2	16
3	80	7	24	51	1	16

● 實務例題結果分析

在實務例題中，外送平台最多可使用的外送員人數設為訪談而得的 300 人，每個外送員的車廂容量為 50 個便當盒大小。此外，因為求解時間的壓力，本研究在測試實務例題時，子問題 2 的 ALNS 演算法僅建立起始解，並不再多做改善。並且，在插入訂單節點時，我們不選擇利潤最高的位置插入節點，而選擇第一個可行位置進行插入，以期加速求解時間。表 6.17 實務例題之求解結果為本研究之實務例題求解結果。

表 6.17 實務例題之求解結果

題組編號	餐廳收益		餐廳拒絕的訂單		平台利潤				平台拒絕的訂單		使用 外送員數目	子問題1 求解時間 (sec)	子問題2 求解時間 (sec)
	目前接到	預測	目前接到	預測	收益		成本		目前接到	預測			
					目前接到	預測	運輸成本	懲罰成本					
1	41230	34790	21	19	14124	11344	846.8	21218	16	40	42	9.673	12.81
2	53641	36985	24	29	19575	12390	1138.4	16071	17	33	55	15.904	28.97
3	52777	36143	21	8	15731	13211	1360	21422	19	20	58	17.333	34.735

根據表 6.15、表 6.16 與表 6.17，我們可以發現本研究提出的演算法，能夠在 52.067 秒內求解 91 家餐廳與 387 張外送訂單的實務例題，即例題 3。其中每間餐廳的生產網路平均有 39 個節點，配送路線網路有 1104 個節點，並得到不錯的結果。

在每間餐廳平均有 3 至 4 張外送訂單，平均每間餐廳共含有 16 至 17 道外送餐點的情況下。餐廳可以接受大部分的訂單，如：例題 1 共 297 張外送訂單，餐廳僅拒絕 40 張訂單，接受比率達到 87%。例題 2 共 375 張外送訂單，餐廳僅拒絕 53 張訂單，接受比率達到 86%。例題 3 共 387 張外送訂單，餐廳僅拒絕 29 張訂單，接受比率達到 93%。

而在平台方面，在分別使用 42、55 與 58 位外送員的情況下，平台也同樣可以接受大部分的外送訂單，如：例題 1 餐廳接受共 257 張外送訂單，平台僅拒絕其中 56 張訂單，接受比率達到 78%。例題 2 共 322 張外送訂單，平台僅拒絕其中 50 張訂單，接受比率達到 84%。例題 3 共 358 張外送訂單，平台僅拒絕其中 39 張訂單，接受比率達到 89%。

若更進一步分析平台的利潤組成，我們可以發現在每個例題中平台拒絕餐廳接受的訂單所造成的懲罰成本偏高。表 6.18 列出了平台拒絕餐廳接受訂單之訂單資訊，我們可以發現，平台拒絕的原因為該訂單的送達時窗下界減去取餐時間上界超過了該訂單的取餐餐廳與顧客距離，亦即就算外送員直接取完餐後就送給顧客，也無法在顧客送達時窗內抵達顧客位置。

表 6.18 例題 1 的平台拒絕餐廳接受訂單之訂單資訊

取餐節點	取餐時窗上界	取餐時窗下界	顧客節點	送達時窗上界	送達時窗下界	平台是否拒絕	節點距離
318	49	50	615	49	60	1	13
319	52	53	616	52	63	1	12
330	65	67	627	65	82	1	19
332	54	54	629	54	64	1	11
339	80	87	636	80	89	1	15
343	82	82	640	82	89	1	8
351	49	49	648	49	59	1	25
352	91	95	649	91	100	1	13
373	40	41	670	40	51	1	15
381	87	92	678	87	89	1	14
382	100	100	679	100	100	1	8
383	81	81	680	81	91	1	17
390	53	53	687	53	63	1	11
391	49	49	688	49	59	1	16
395	51	51	692	51	61	1	11
397	80	81	694	80	91	1	12
400	85	87	697	85	89	1	12
409	57	58	706	57	68	1	13
417	61	62	714	61	72	1	16
419	45	45	716	45	55	1	16
424	50	50	721	50	60	1	16
426	94	94	723	94	100	1	12
430	94	100	727	94	100	1	7
436	63	66	733	63	76	1	17
439	50	50	736	50	60	1	11
458	65	65	755	65	75	1	13
474	74	74	771	74	84	1	12
480	84	85	777	84	95	1	13
486	57	58	783	57	71	1	15
487	61	61	784	61	71	1	14
488	45	45	785	45	55	1	12
490	59	59	787	59	69	1	12
495	77	77	792	77	87	1	12
498	97	97	795	97	100	1	9
499	96	96	796	96	100	1	7
508	91	101	805	91	100	1	18
519	58	58	816	58	68	1	14
521	87	92	818	87	89	1	15
522	81	88	819	81	89	1	20
524	45	50	821	45	60	1	24
525	78	85	822	78	89	1	14
528	62	62	825	62	72	1	18
529	70	70	826	70	80	1	13
531	96	100	828	96	100	1	8
553	97	97	850	97	100	1	5
557	93	97	854	93	100	1	15
564	82	92	861	82	89	1	8
565	48	54	862	48	64	1	17
566	61	61	863	61	71	1	16
568	76	80	865	76	89	1	18
570	50	50	867	50	60	1	12
574	88	89	871	88	89	1	4
578	82	83	875	82	93	1	16
585	41	41	882	41	51	1	16
591	83	89	888	83	89	1	11
596	95	95	893	95	100	1	6

7 結論與建議

本研究為第一篇整合 O2O 餐點外送平台的生產與配送問題，由平台整合決策餐廳是否接受該訂單與該訂單的取餐與送達時間。此外，本研究也將未來可能出現的預測訂單也納入決策，這個方式使本研究的類似滾動時域架構 (rolling horizon approach)，並成為第一篇研究滾動時域架構下的整合生產與配送問題。

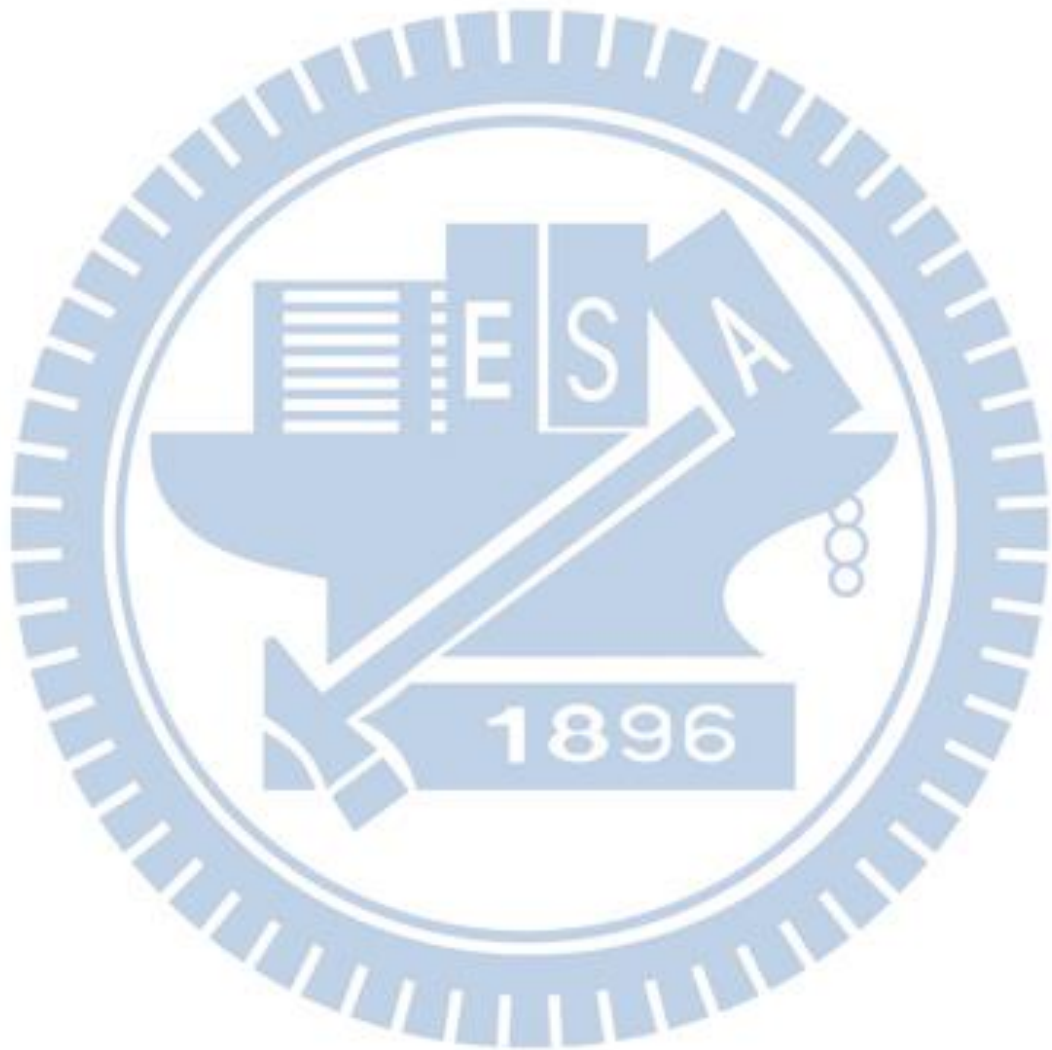
接著建構餐廳生產排程問題，提出一個新的方法將彈性零工問題轉換成平行機台排程問題，並轉換成與配送問題同樣形式的網路問題，即有時窗限制的團隊尋寶問題。再建構平台外送員配送路線規劃問題，即有時窗限制的收送貨問題 (pickup and delivery problem with time window)。針對整合生產與配送問題設計一個混合整數規劃最佳化模型。

為了減少求解時間，本研究設計一個分層最佳化架構 (hierarchical optimization)。利用 IterILS (Iterated Local Search) 求解子問題 1 生產問題再將其結果代入配送問題內，利用 ALNS (Alternative Large Neighborhood Search) 求解。這個方法使本研究在求解較為複雜的整合生產與配送問題時，同樣能保有不錯的求解品質，如小例題 3、4 和 6，平均僅使用 0.1722、0.2033 與 0.2535 秒，便可以求到與 Gurobi 在 1 小時內求得的解僅差 6.078、3.251%與 6.153%的解。

最後以 T 公司的實務資料建構出實務上空腹熊貓在台北的接單規模之例題，發現本研究能夠在 52.067 秒內求解 91 家餐廳與 387 張外送訂單的實務例題，即例題 3。其中每間餐廳的生產網路平均有 39 個節點，配送路線網路有 1104 個節點。而 3 個實務例題中餐廳分別能接受 87%、86% 與 93% 的外送訂單，同時平台分別可以服務 78%、84% 與 89% 的餐廳接受之外送訂單。

而在 6.2 節提及平台拒絕的原因為該訂單的送達時窗下界減去取餐時間上界超過了該訂單的取餐餐廳與顧客距離。該情形肇因於本研究必須在 1 分鐘內求解，使演算法計算完子問題 2 時，無法再將子問題 2 的結果回傳給子問題 1，當作子問題 1 的輸

入參數。若可以將子問題 2 的結果，在子問題 1 求解時用以改善生產排程問題之餐廳完成時間，使訂單的送達時窗下界減去取餐時間上界不要超過該訂單的取餐餐廳與顧客距離，如此便可以減少平台的懲罰成本，並有效的提升平台的利潤。



8 參考資料

1. Adulyasak, Y., Cordeau, J. F., & Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4), 851-867.
2. Armstrong, R., Gao, S. and Lei, L. (2008). A zero-inventory production and distribution problem with a fixed customer sequence, *Annals of Operations Research*, 159, pp. 395–414.
3. Forbes, C., Evans, M., Hastings, N., & Peacock, B. (2011). *Statistical distributions*. John Wiley & Sons.
4. Bradley, S. P., Hax, A. C., & Magnanti, T. L. (1977). *Applied mathematical programming*.
5. Brucker, P., & Knust, S. (2011). *Complex scheduling*. Springer Science & Business Media.
6. Chen, H.-K., Hsueh, C.-F. and Chang, M.-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products, *Computers & Operations Research*, 36, pp. 2311-2319.
7. Companies, B. T. (n.d.). *Venture Capital Database*. Retrieved October 30, 2017, from <https://www.cbinsights.com/>
8. Dai, H., Liu, P., and Liu, Y. (2017). Capacity Planning for O2O On-Demand Delivery Systems with Crowd-Sourcing.
9. Directorate General of Budget, Accounting and Statistics ... (n.d.). Retrieved October 30, 2016, from <http://eng.dgbas.gov.tw/>
10. Farahani, P., Grunow, M. and Günther, H.-O. (2012). Integrated production and distribution planning for perishable food products, *Flexible Services and Manufacturing Journal* 24:28–51

11. Gao, S., Qi, L., & Lei, L. (2015). Integrated batch production and distribution scheduling with limited vehicle capacity. *International Journal of Production Economics*, 160, 13-25.
12. Geismar, H.N., Laporte, G., Lei, L. and Sriskandarajah, C. (2008). The integrated production and transportation scheduling problem for a product with a short lifespan, *INFORMS Journal on Computing*, 20(1), pp. 21–33.
13. Gunawan, A., Lau, H. C., & Lu, K. (2015). SAILS: hybrid algorithm for the team orienteering problem with time windows.
14. Hvattum, L. M., Løkketangen, A., & Laporte, G. (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, 40(4), 421-438.
15. Kouki, Z., Chaar, B. F., Hammadi, S., & Ksouri, M. (2007, December). Analogies between flexible job shop scheduling and vehicle routing problems. In *Industrial Engineering and Engineering Management, 2007 IEEE International Conference on* (pp. 880-884). IEEE.
16. Li, Y., Chen, H., & Prins, C. (2016). Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252(1), 27-38.
17. Naso, D., Surico, M., Turchiano, B. and Kaymak, U. (2007). Genetic algorithms for supply-chain scheduling: a case study in the distribution of ready-mixed concrete, *European Journal of Operational Research*, 177(3), pp. 2069–2099.
18. Osyczka, A. (1985). Multicriteria optimization for engineering design. In *Design optimization* (pp. 193-227).

19. Peng, D., & Gatschke, J. (n.d.). • Statista - The Statistics Portal for Market Data, Market Research and Market Studies. Retrieved October 30, 2016, from <https://www.statista.com/>
20. Pinedo M.L. (2009) Machine Scheduling and Job Shop Scheduling. In: Planning and Scheduling in Manufacturing and Services. Springer, New York, NY
21. The Online Economy: Strategy and Entrepreneurship. (n.d.). Retrieved October 30, 2016, from <http://www.onlineeconomy.org/>
22. Chang, T.-S., Chang, C.-H. and Lin, Y.-C. (2016). Integrated Scheduling of Production and Distribution for Perishable Foods.
23. Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, 209(1), 1-10.
24. Vansteenwegen, P., Souffriau, W., Berghe, G. V., & Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, 36(12), 3281-3290.
25. 王柔雅 (2016)。Uber 搶攻外送 加速台灣餐飲業質變。商業週刊。第 1515 期
26. 吳家豪 (2017)。外送員月薪上看 6 萬 foodpanda 徵才搶破頭。中央社
27. 車輛耗能研究網站. (n.d.). Retrieved from <https://auto.itri.org.tw/Myenergy/>
28. 林則孟 (2012)。生產計畫與管理。華泰
29. 陳炳宏 (2018)。〈共享經濟外送篇〉美食外送 訂單成長百倍。自由時報

9 附錄

■ 餐廳資訊

規模 1 (小規模)

餐點種類	製作步驟
A	炒5分-沙1分-煎4分
B	沙3分
C	炸6分

工作站	廚師數
沙拉	2
炒台	4
煎台	4
炸鍋	4

規模 2 (中規模)

餐點種類	製作步驟
A	炒10分
B	炒5分-炒10分
C	煎5分-烤10分
D	烤15分
E	沙3分
F	炸6分

工作站	廚師數目
沙拉	4
炒台	8
煎台	8
炸鍋	4
烤爐	6

規模 3 (大規模)

餐點種類	製作步驟
A	炒10分
B	炒5分-炒10分
C	煎5分-烤10分
D	烤15分
E	沙3分
F	炸6分
G	烤15分
H	甜3分

工作站	廚師數目
沙拉	8
炒台	12
煎台	12
炸鍋	6
烤爐	12
甜品	4

■ 演算法參數表

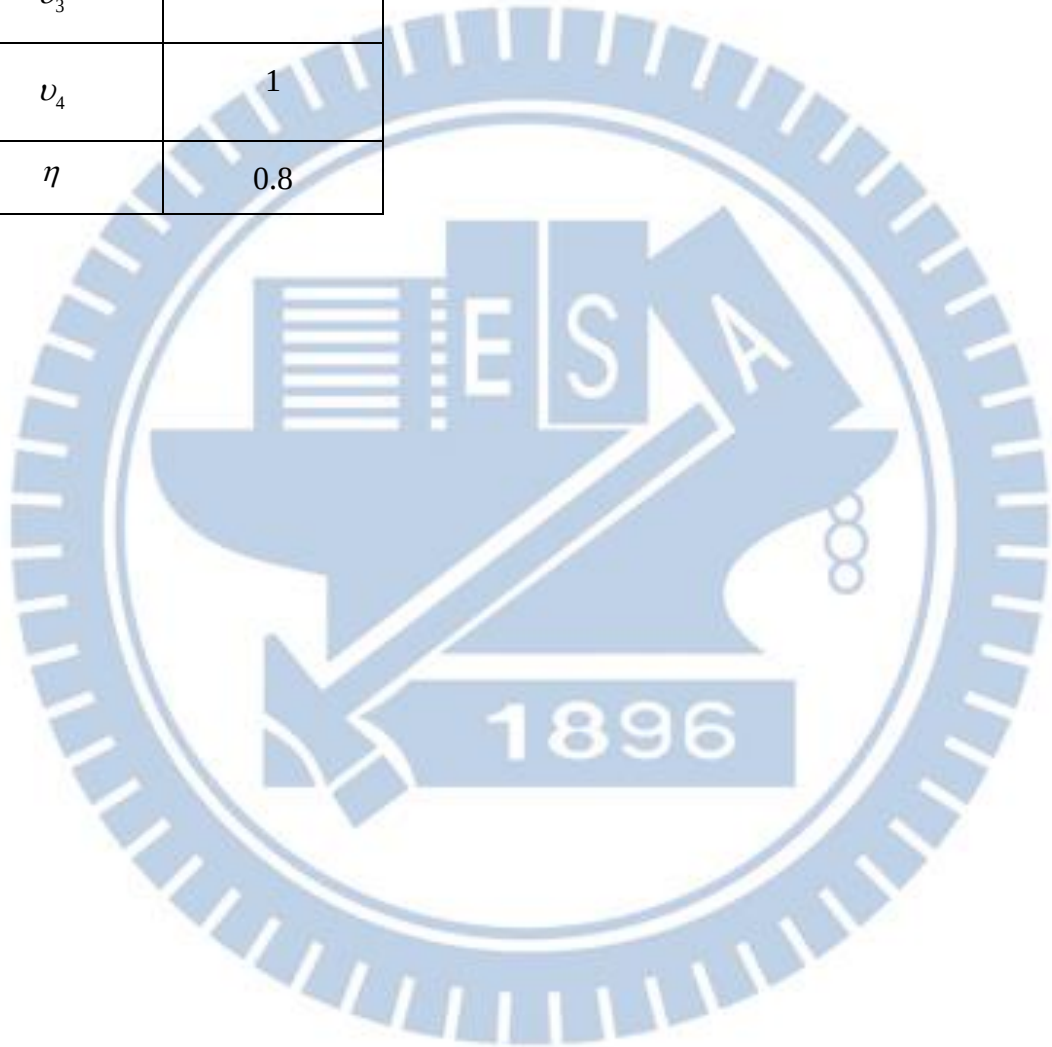
IterILS

N	20
-----	----

ALNS

$nsegs$	3
$Niters$	10
T_{beg}	0.3
θ	0.9995
φ	1
λ	1

ζ	1
ρ	0.2
ν_1	10
ν_2	5
ν_3	3
ν_4	1
η	0.8



■ 實務例題 (題組 1)

餐廳餐點資訊 (由於資料龐大餐廳餐點資訊僅節錄前 5 家餐廳)

餐廳 1

外送訂	點餐時	餐點種	訂單種	餐廳訂	餐點編	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
	59	B	1	2	1	59	99	0	25	3	62	99	150	1
	59	C	1	3	2	59	99	0	20	6	65	99	70	1
	65	C	1	4	3	65	105	0	20	6	71	105	70	1
40	21	A	0	1	4	21	81	6	20	10	35	75	160	1
40	21	C	0	1	5	21	81	6	20	6	31	75	70	1
40	21	C	0	1	6	21	81	6	20	6	31	75	70	1

餐廳 6

外送訂	點餐時	餐點種	訂單種	餐廳訂	餐點編	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
		13 F	1	5	1	13	53	0	20	6	19	53	140	6
		13 H	1	6	2	13	53	0	25	3	16	53	100	6
		13 G	1	7	3	13	53	0	20	15	28	53	450	6
		14 B	1	8	4	14	54	0	20	15	29	54	240	6
		14 E	1	9	5	14	54	0	25	3	17	54	200	6
		19 B	1	10	6	19	59	0	20	15	34	59	240	6
		19 F	1	11	7	19	59	0	20	6	25	59	140	6
		19 H	1	12	8	19	59	0	25	3	22	59	100	6
		25 A	1	13	9	25	65	0	20	10	35	65	200	6
		25 D	1	14	10	25	65	0	20	15	40	65	260	6
		25 F	1	15	11	25	65	0	20	6	31	65	140	6
		32 B	1	16	12	32	72	0	20	15	47	72	240	6
		32 H	1	17	13	32	72	0	25	3	35	72	100	6
		41 H	1	18	14	41	81	0	25	3	44	81	100	6
		41 F	1	19	15	41	81	0	20	6	47	81	140	6
		45 D	1	20	16	45	85	0	20	15	60	85	260	6
		45 E	1	21	17	45	85	0	25	3	48	85	200	6
		45 C	1	22	18	45	85	0	20	15	60	85	270	6
		45 F	1	23	19	45	85	0	20	6	51	85	140	6
		47 F	1	24	20	47	87	0	20	6	53	87	140	6
		47 A	1	25	21	47	87	0	20	10	57	87	200	6
		50 G	1	26	22	50	90	0	20	15	65	90	450	6
		50 C	1	27	23	50	90	0	20	15	65	90	270	6
		52 F	1	28	24	52	92	0	20	6	58	92	140	6
		52 D	1	29	25	52	92	0	20	15	67	92	260	6
		52 G	1	30	26	52	92	0	20	15	67	92	450	6
		54 C	1	31	27	54	94	0	20	15	69	94	270	6
		54 B	1	32	28	54	94	0	20	15	69	94	240	6
		62 E	1	33	29	62	102	0	25	3	65	102	200	6
		62 F	1	34	30	62	102	0	20	6	68	102	140	6
		62 A	1	35	31	62	102	0	20	10	72	102	200	6
		62 B	1	36	32	62	102	0	20	15	77	102	240	6
		62 H	1	37	33	62	102	0	25	3	65	102	100	6
		68 F	1	38	34	68	108	0	20	6	74	108	140	6
		72 C	1	39	35	72	112	0	20	15	87	112	270	6
		79 C	1	40	36	79	119	0	20	15	94	119	270	6
		79 E	1	41	37	79	119	0	25	3	82	119	200	6
		79 A	1	42	38	79	119	0	20	10	89	119	200	6
		79 G	1	43	39	79	119	0	20	15	94	119	450	6

3135	60 B	2	44	40	150	160	0	20	15	150	160	240	6
106	21 C	0	1	41	21	81	13	20	15	40	68	270	6
106	21 F	0	1	42	21	81	13	20	6	31	68	140	6
106	21 G	0	1	43	21	81	13	20	15	40	68	450	6
1857	29 A	0	2	44	29	89	12	20	10	43	77	200	6
1857	29 A	0	2	45	29	89	12	20	10	43	77	200	6
1857	29 E	0	2	46	29	89	12	25	3	36	77	200	6
1857	29 G	0	2	47	29	89	12	20	15	48	77	450	6
1857	29 H	0	2	48	29	89	12	25	3	36	77	100	6
2034	40 A	0	3	49	40	100	8	20	10	55	92	200	6
2034	40 B	0	3	50	40	100	8	20	15	60	92	240	6
2034	40 B	0	3	51	40	100	8	20	15	60	92	240	6
2034	40 C	0	3	52	40	100	8	20	15	60	92	270	6
2034	40 E	0	3	53	40	100	8	25	3	48	92	200	6
2034	40 F	0	3	54	40	100	8	20	6	51	92	140	6
2034	40 G	0	3	55	40	100	8	20	15	60	92	450	6
2034	40 H	0	3	56	40	100	8	25	3	48	92	100	6
2034	40 H	0	3	57	40	100	8	25	3	48	92	100	6
2175	40 A	0	4	58	40	100	3	20	10	55	97	200	6
2175	40 A	0	4	59	40	100	3	20	10	55	97	200	6
2175	40 B	0	4	60	40	100	3	20	15	60	97	240	6
2175	40 B	0	4	61	40	100	3	20	15	60	97	240	6
2175	40 D	0	4	62	40	100	3	20	15	60	97	260	6
2175	40 G	0	4	63	40	100	3	20	15	60	97	450	6
2175	40 G	0	4	64	40	100	3	20	15	60	97	450	6
2175	40 H	0	4	65	40	100	3	25	3	48	97	100	6

餐廳 7

外送訂	點餐時	餐點種	訂單種	餐廳訂	餐點編	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
	10 H		1	4	1	10	50	0	25	3	13	50	100	7
	10 E		1	5	2	10	50	0	25	3	13	50	200	7
	10 D		1	6	3	10	50	0	20	15	25	50	260	7
	11 G		1	7	4	11	51	0	20	15	26	51	450	7
	14 D		1	8	5	14	54	0	20	15	29	54	260	7
	14 A		1	9	6	14	54	0	20	10	24	54	200	7
	19 A		1	10	7	19	59	0	20	10	29	59	200	7
	21 G		1	11	8	21	61	0	20	15	36	61	450	7
	21 A		1	12	9	21	61	0	20	10	31	61	200	7
	28 B		1	13	10	28	68	0	20	15	43	68	240	7
	28 C		1	14	11	28	68	0	20	15	43	68	270	7
	28 E		1	15	12	28	68	0	25	3	31	68	200	7
	37 A		1	16	13	37	77	0	20	10	47	77	200	7
	37 G		1	17	14	37	77	0	20	15	52	77	450	7
	37 G		1	18	15	37	77	0	20	15	52	77	450	7
	37 A		1	19	16	37	77	0	20	10	47	77	200	7
	38 F		1	20	17	38	78	0	20	6	44	78	140	7
	43 B		1	21	18	43	83	0	20	15	58	83	240	7
	43 E		1	22	19	43	83	0	25	3	46	83	200	7
	43 G		1	23	20	43	83	0	20	15	58	83	450	7
	50 A		1	24	21	50	90	0	20	10	60	90	200	7
	63 B		1	25	22	63	103	0	20	15	78	103	240	7
	63 D		1	26	23	63	103	0	20	15	78	103	260	7
	63 E		1	27	24	63	103	0	25	3	66	103	200	7
	63 A		1	28	25	63	103	0	20	10	73	103	200	7
	67 G		1	29	26	67	107	0	20	15	82	107	450	7
	67 A		1	30	27	67	107	0	20	10	77	107	200	7
	67 F		1	31	28	67	107	0	20	6	73	107	140	7
	68 F		1	32	29	68	108	0	20	6	74	108	140	7
	68 H		1	33	30	68	108	0	25	3	71	108	100	7
	68 H		1	34	31	68	108	0	25	3	71	108	100	7
	68 C		1	35	32	68	108	0	20	15	83	108	270	7
33	21 E		0	1	33	21	81	3	25	3	28	78	200	7
33	21 F		0	1	34	21	81	3	20	6	31	78	140	7
33	21 H		0	1	35	21	81	3	25	3	28	78	100	7
2067	40 D		0	2	36	40	100	10	20	15	60	90	260	7
2067	40 F		0	2	37	40	100	10	20	6	51	90	140	7
2078	40 A		0	3	38	40	100	10	20	10	55	90	200	7
2078	40 C		0	3	39	40	100	10	20	15	60	90	270	7
2078	40 G		0	3	40	40	100	10	20	15	60	90	450	7

餐廳 8

外送訂	點餐時	餐點種	訂單種	餐廳訂	餐點編	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
	24	C	1	8	1	24	64	0	20	15	39	64	270	8
	28	H	1	9	2	28	68	0	25	3	31	68	100	8
	28	H	1	10	3	28	68	0	25	3	31	68	100	8
	34	G	1	11	4	34	74	0	20	15	49	74	450	8
	34	C	1	12	5	34	74	0	20	15	49	74	270	8
	34	D	1	13	6	34	74	0	20	15	49	74	260	8
	37	B	1	14	7	37	77	0	20	15	52	77	240	8
	46	H	1	15	8	46	86	0	25	3	49	86	100	8
	46	G	1	16	9	46	86	0	20	15	61	86	450	8
	46	F	1	17	10	46	86	0	20	6	52	86	140	8
	46	A	1	18	11	46	86	0	20	10	56	86	200	8
	46	C	1	19	12	46	86	0	20	15	61	86	270	8
	47	A	1	20	13	47	87	0	20	10	57	87	200	8
	47	E	1	21	14	47	87	0	25	3	50	87	200	8
	47	B	1	22	15	47	87	0	20	15	62	87	240	8
	59	H	1	23	16	59	99	0	25	3	62	99	100	8
	65	A	1	24	17	65	105	0	20	10	75	105	200	8
	67	H	1	25	18	67	107	0	25	3	70	107	100	8
	67	D	1	26	19	67	107	0	20	15	82	107	260	8
	68	A	1	27	20	68	108	0	20	10	78	108	200	8
	68	B	1	28	21	68	108	0	20	15	83	108	240	8
	68	E	1	29	22	68	108	0	25	3	71	108	200	8
	70	F	1	30	23	70	110	0	20	6	76	110	140	8
	70	H	1	31	24	70	110	0	25	3	73	110	100	8
	75	C	1	32	25	75	115	0	20	15	90	115	270	8
	75	G	1	33	26	75	115	0	20	15	90	115	450	8
	75	E	1	34	27	75	115	0	25	3	78	115	200	8
	76	C	1	35	28	76	116	0	20	15	91	116	270	8
	76	D	1	36	29	76	116	0	20	15	91	116	260	8
	76	A	1	37	30	76	116	0	20	10	86	116	200	8
	76	E	1	38	31	76	116	0	25	3	79	116	200	8
	83	A	1	39	32	83	123	0	20	10	93	123	200	8

81	21 A	0	1	33	21	81	11	20	10	35	70	200	8
81	21 A	0	1	34	21	81	11	20	10	35	70	200	8
81	21 A	0	1	35	21	81	11	20	10	35	70	200	8
81	21 B	0	1	36	21	81	11	20	15	40	70	240	8
81	21 B	0	1	37	21	81	11	20	15	40	70	240	8
81	21 B	0	1	38	21	81	11	20	15	40	70	240	8
81	21 C	0	1	39	21	81	11	20	15	40	70	270	8
81	21 D	0	1	40	21	81	11	20	15	40	70	260	8
81	21 D	0	1	41	21	81	11	20	15	40	70	260	8
81	21 D	0	1	42	21	81	11	20	15	40	70	260	8
81	21 E	0	1	43	21	81	11	25	3	28	70	200	8
81	21 F	0	1	44	21	81	11	20	6	31	70	140	8
81	21 F	0	1	45	21	81	11	20	6	31	70	140	8
81	21 F	0	1	46	21	81	11	20	6	31	70	140	8
81	21 G	0	1	47	21	81	11	20	15	40	70	450	8
81	21 G	0	1	48	21	81	11	20	15	40	70	450	8
81	21 H	0	1	49	21	81	11	25	3	28	70	100	8
81	21 H	0	1	50	21	81	11	25	3	28	70	100	8
1930	29 A	0	2	51	29	89	4	20	10	43	85	200	8
1930	29 B	0	2	52	29	89	4	20	15	48	85	240	8
1930	29 E	0	2	53	29	89	4	25	3	36	85	200	8
1930	29 E	0	2	54	29	89	4	25	3	36	85	200	8
1930	29 F	0	2	55	29	89	4	20	6	39	85	140	8
1930	29 F	0	2	56	29	89	4	20	6	39	85	140	8
1930	29 F	0	2	57	29	89	4	20	6	39	85	140	8
1930	29 F	0	2	58	29	89	4	20	6	39	85	140	8
1930	29 G	0	2	59	29	89	4	20	15	48	85	450	8
1936	29 A	0	3	60	29	89	7	20	10	43	82	200	8
1936	29 D	0	3	61	29	89	7	20	15	48	82	260	8
1936	29 F	0	3	62	29	89	7	20	6	39	82	140	8
1936	29 G	0	3	63	29	89	7	20	15	48	82	450	8
1936	29 H	0	3	64	29	89	7	25	3	36	82	100	8
1980	29 A	0	4	65	29	89	11	20	10	43	78	200	8
1980	29 A	0	4	66	29	89	11	20	10	43	78	200	8
1980	29 A	0	4	67	29	89	11	20	10	43	78	200	8
1980	29 A	0	4	68	29	89	11	20	10	43	78	200	8
1980	29 A	0	4	69	29	89	11	20	10	43	78	200	8
1980	29 B	0	4	70	29	89	11	20	15	48	78	240	8
1980	29 B	0	4	71	29	89	11	20	15	48	78	240	8
1980	29 B	0	4	72	29	89	11	20	15	48	78	240	8
1980	29 D	0	4	73	29	89	11	20	15	48	78	260	8
1980	29 D	0	4	74	29	89	11	20	15	48	78	260	8
1980	29 F	0	4	75	29	89	11	20	6	39	78	140	8
1980	29 F	0	4	76	29	89	11	20	6	39	78	140	8
1980	29 G	0	4	77	29	89	11	20	15	48	78	450	8
1980	29 G	0	4	78	29	89	11	20	15	48	78	450	8
1980	29 G	0	4	79	29	89	11	20	15	48	78	450	8
1980	29 H	0	4	80	29	89	11	25	3	36	78	100	8
1980	29 H	0	4	81	29	89	11	25	3	36	78	100	8
1980	29 H	0	4	82	29	89	11	25	3	36	78	100	8
2015	40 G	0	5	83	40	100	2	20	15	60	98	450	8
2015	40 H	0	5	84	40	100	2	25	3	48	98	100	8
2043	40 A	0	6	85	40	100	16	20	10	55	84	200	8
2043	40 D	0	6	86	40	100	16	20	15	60	84	260	8
2043	40 F	0	6	87	40	100	16	20	6	51	84	140	8
2043	40 H	0	6	88	40	100	16	25	3	48	84	100	8
2105	40 G	0	7	89	40	100	6	20	15	60	94	270	8

餐廳 9

外送訂	點餐時	餐點種	訂單種	餐廳訂	餐點編	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
	28 A		1	4	1	28	68	0	20	10	38	68	160	9
	39 C		1	5	2	39	79	0	20	6	45	79	70	9
	57 B		1	6	3	57	97	0	25	3	60	97	150	9
	72 A		1	7	4	72	112	0	20	10	82	112	160	9
	72 B		1	8	5	72	112	0	25	3	75	112	150	9
	78 C		1	9	6	78	118	0	20	6	84	118	70	9
71	21 B		0	1	7	21	81	11	25	3	28	70	150	9
71	21 B		0	1	8	21	81	11	25	3	28	70	150	9
71	21 C		0	1	9	21	81	11	20	6	31	70	70	9
71	21 C		0	1	10	21	81	11	20	6	31	70	70	9
71	21 C		0	1	11	21	81	11	20	6	31	70	70	9
2100	40 A		0	2	12	40	100	5	20	10	55	95	160	9
2100	40 A		0	2	13	40	100	5	20	10	55	95	160	9
2100	40 C		0	2	14	40	100	5	20	6	51	95	70	9
2147	40 C		0	3	15	40	100	4	20	6	51	96	70	9
2147	40 C		0	3	16	40	100	4	20	6	51	96	70	9

外送訂單資訊 (由於資料龐大外送訂單資訊僅節錄前 50 張訂單)

外送訂	餐廳地址	顧客地址	餐廳編	顧客編	取餐節	顧客節	取餐時	顧客時	送達時	送達時	容量	CF	PF
3135	台北市大安區永康街6巷6號號	台北市萬華區成都路88號	365	302	300	597	150	160	170	180	1	0	0
3115	台北市大安區敦化南路二段63巷42號	台北市士林區文昌路95號	420	282	301	598	150	155	170	180	5	0	0
3139	台北市松山區民生東路三段134號A棟1樓	台北市中正區濟南路二段37號	402	121	302	599	150	160	170	180	1	0	0
3110	台北市松山區民生東路三段134號A棟1樓	台北市大同區塔城街10號	402	10	303	600	120	125	140	150	7	0	0
3111	台北市大安區安和路二段69巷3號	台北市大安區大安路一段19巷20號	414	17	304	601	120	130	140	150	6	0	0
3117	台北市中山區長春路15號	台北市大安區延吉街129-85	382	35	305	602	150	160	170	180	4	0	0
3140	台北市中正區仁愛路二段108號	台北市中山區復興南路一段28號	439	103	306	603	150	155	170	180	2	0	0
3112	台北市中正區館前路59號	台北市松山區八德路二段346巷7弄15號	458	127	307	604	120	125	140	150	6	0	0
3141	台北市中山區中山北路一段121巷9號	台北市中山區新生北路三段82之5號	459	106	308	605	150	160	170	180	1	0	0
3142	台北市中山區中山北路一段105巷22號	台北市中山區新生北路二段38號	467	104	309	606	150	155	170	180	15	0	0
3113	台北市大同區南京西路18巷6弄8號	台北市士林區華齡街175號	498	290	310	607	120	125	140	150	8	0	0
3134	台北市中正區重慶南路一段5巷1號	台北市中正區汀州路三段230巷14弄2號	522	331	311	608	150	155	170	180	3	0	0
3136	台北市大安區忠孝東路四段126之13號	台北市中山區新生北路三段82之5號	526	106	312	609	150	155	170	180	5	0	0
3109	台北市信義區吳興街271號	台北市內湖區陽光街275號	529	216	313	610	150	160	170	180	1	0	0
3137	台北市大同區承德路一段1號	台北市萬華區萬大路346號	393	312	314	611	150	155	170	180	4	0	0
3114	台北市松山區八德路四段709巷2號	台北市大安區市民大道三段178號	577	22	315	612	120	125	140	150	7	0	0
3138	台北市大同區民權西路231號	台北市中山區龍江路62號	622	112	316	613	150	160	170	180	3	0	0
40	台北市中山區中山北路二段183巷4號	台北市松山區民生東路五段69巷2弄27號	538	141	317	614	25	999	21	81	3	90	210
106	台北市大同區承德路一段1號	台北市北投區東華街二段180號	393	256	318	615	25	999	21	81	3	258	602
1857	台北市大同區承德路一段1號	台北市內湖區內湖路一段360巷2號	393	209	319	616	25	999	29	89	5	173	403
2034	台北市大同區承德路一段1號	台北市中正區羅斯福路三段126之5號	393	122	320	617	25	999	40	100	9	291	679
2175	台北市大同區承德路一段1號	中山區大直街20巷18號	393	198	321	618	25	999	40	100	8	321	749
33	台北市大安區忠孝東路四段98號	台北市信義區基隆路一段350之41號	378	188	322	619	25	999	21	81	3	132	308
2067	台北市大安區忠孝東路四段98號	台北市內湖區文德路22巷44弄10號	378	218	323	620	25	999	40	100	2	60	140
2078	台北市大安區忠孝東路四段98號	台北市萬華區昆明街214-1號	378	303	324	621	25	999	40	100	3	138	322
81	台北市信義區松壽路20號	台北市中山區北安路608巷14弄1號	409	200	325	622	25	999	21	81	18	1227	2863
1930	台北市信義區松壽路20號	台北市內湖區瑞光路468號	409	210	326	623	25	999	29	89	9	278	648
1936	台北市信義區松壽路20號	台北市信義區松高路1號	409	179	327	624	25	999	29	89	5	173	403
1980	台北市信義區松壽路20號	台北市中山區林森北路369號	409	92	328	625	25	999	29	89	18	626	1460
2015	台北市信義區松壽路20號	台北市中山區新生北路二段38號	409	104	329	626	25	999	40	100	2	83	193
2043	台北市信義區松壽路20號	台北市士林區中山北路七段31號	409	271	330	627	25	999	40	100	4	105	245
2105	台北市信義區松壽路20號	台北市松山區南京東路五段291巷20弄3號	409	152	331	628	25	999	40	100	2	71	165
71	台北市松山區八德路二段449號	台北市士林區中正路223巷4號	547	284	332	629	25	999	21	81	5	153	357
2100	台北市松山區八德路二段449號	台北市大安區延吉街129-85	547	35	333	630	25	999	40	100	3	59	137
2147	台北市松山區八德路二段449號	台北市內湖區康寧路三段16巷41號	547	235	334	631	25	999	40	100	2	21	49
29	台北市大安區忠孝東路四段335號	台北市大安區仁愛路四段280號	448	20	335	632	25	999	21	81	3	111	259
1983	台北市大安區忠孝東路四段335號	台北市內湖區東湖路115號	448	236	336	633	25	999	29	89	8	161	375
82	台北市中山區長安東路二段36巷11號	台北市士林區中山北路六段470號	600	272	337	634	25	999	21	81	2	111	259
91	台北市松山區民生東路三段134號A棟1樓	台北市內湖區行忠路66巷2號	402	229	338	635	25	999	21	81	15	873	2037
1961	台北市松山區民生東路三段134號A棟1樓	台北市文山區指南路二段167號	402	354	339	636	25	999	29	89	2	60	140
128	台北市信義區松壽路22號2樓	中山區大直街20巷18號	377	198	340	637	25	999	21	81	1	78	182
10	台北市信義區松壽路22號2樓	台北市內湖區成功路二段325號	377	222	341	638	25	999	21	81	6	405	945
93	台北市信義區松壽路22號2樓	台北市大安區和平東路三段63號	377	34	342	639	25	999	21	81	9	444	1036
1898	台北市信義區松壽路22號2樓	台北市中山區林森北路62號	377	94	343	640	25	999	29	89	2	57	133
2097	台北市信義區松壽路22號2樓	台北市大同區重慶北路二段171號	377	7	344	641	25	999	40	100	4	129	301
2180	台北市信義區松壽路22號2樓	台北市中山區林森北路410號	377	93	345	642	25	999	40	100	3	93	217
7	台北市大安區安東街57-1號	台北市士林區中山北路七段190巷6號	469	265	346	643	25	999	21	81	1	45	105
123	台北市信義區市府路45號	台北市大安區大安路二段63號	373	18	347	644	25	999	21	81	7	270	630
2044	台北市信義區市府路45號	台北市文山區景福街225號	373	343	348	645	25	999	40	100	4	80	186
2082	台北市信義區市府路45號	台北市中山區撫順街16號	373	109	349	646	25	999	40	100	6	102	238

距離矩陣 (由於資料龐大距離矩陣僅節錄前 30 個節點組成之矩陣-30*30)

[illegible]

單位：分鐘

■ 實務例題 (題組 2)

餐廳餐點資訊 (由於資料龐大餐廳餐點資訊僅節錄前 5 家餐廳)

餐廳 4

外送訂單	點餐時間	餐點種類	訂單種類	餐廳訂單編號	餐點編號	時窗上界	時窗下界	與顧客距	賞味期限	製作時間	更新後時窗	更新後時窗	價格	餐廳編號
	31	F	1	6	1	31	71	0	20	6	37	71	90	4
	43	A	1	7	2	43	83	0	20	10	53	83	190	4
	43	A	1	8	3	43	83	0	20	10	53	83	190	4
	58	E	1	9	4	58	98	0	25	3	61	98	170	4
	76	A	1	10	5	76	116	0	20	10	86	116	190	4
	76	F	1	11	6	76	116	0	20	6	82	116	90	4
	96	A	1	12	7	96	136	0	20	10	106	136	190	4
213	36	F	0	1	8	36	96	9	20	6	47	87	90	4
2047	40	B	0	2	9	40	100	13	20	15	60	87	230	4
2047	40	D	0	2	10	40	100	13	20	15	60	87	200	4
2047	40	F	0	2	11	40	100	13	20	6	51	87	90	4
2099	40	A	0	3	12	40	100	15	20	10	55	85	190	4
2099	40	A	0	3	13	40	100	15	20	10	55	85	190	4
2099	40	D	0	3	14	40	100	15	20	15	60	85	200	4
2099	40	E	0	3	15	40	100	15	25	3	48	85	170	4
2253	94	D	0	4	16	94	154	6	20	15	112	148	200	4
2253	94	E	0	4	17	94	154	6	25	3	100	148	170	4
2255	94	B	0	5	18	94	154	9	20	15	112	145	230	4
2255	94	C	0	5	19	94	154	9	20	15	112	145	220	4
2255	94	D	0	5	20	94	154	9	20	15	112	145	200	4
2255	94	D	0	5	21	94	154	9	20	15	112	145	200	4
2255	94	E	0	5	22	94	154	9	25	3	100	145	170	4
2255	94	F	0	5	23	94	154	9	20	6	103	145	90	4

餐廳 6

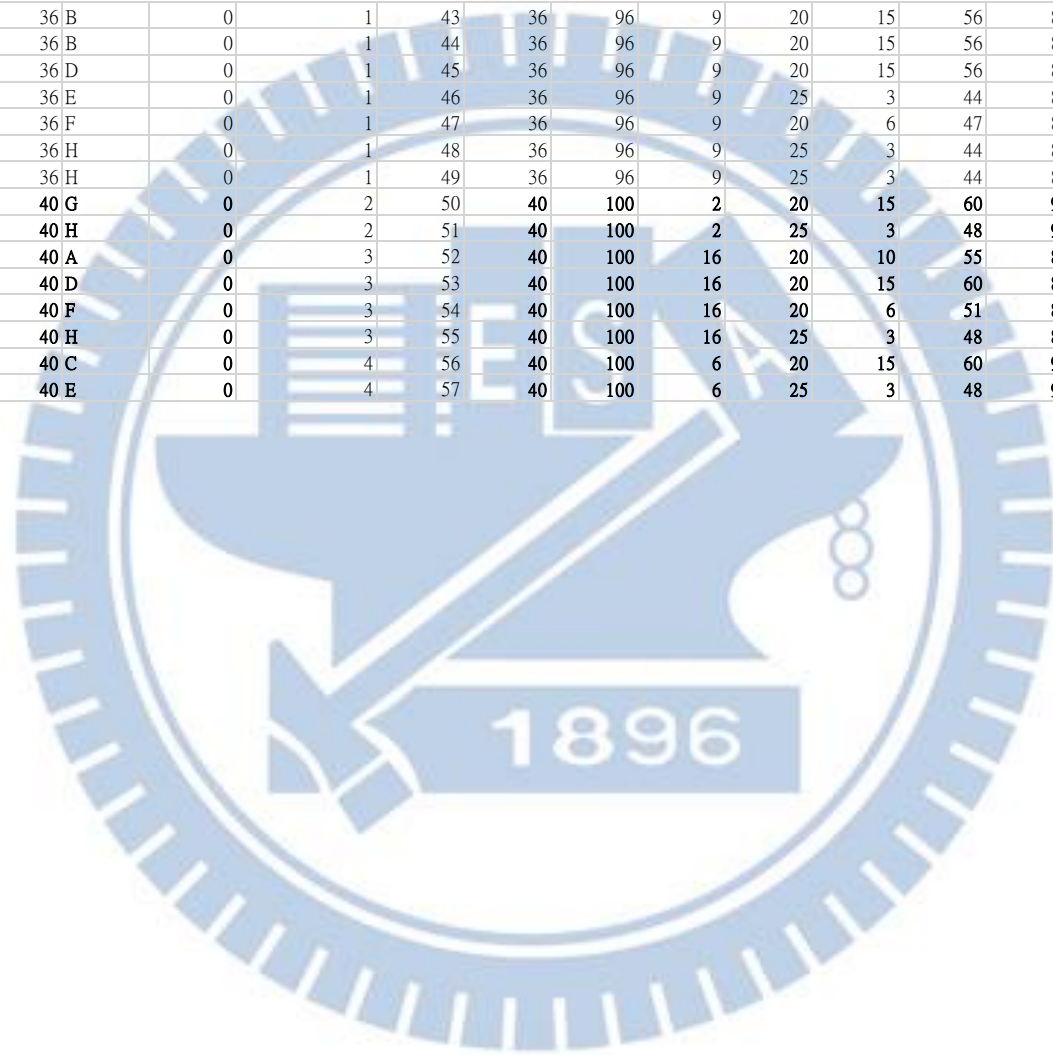
外送訂	點餐時	餐點種	訂單種	餐廳訂單編號	餐點編號	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
	13 F	1	5	1	13	53	0	20	6	19	53	140	6	
	13 H	1	6	2	13	53	0	25	3	16	53	100	6	
	13 G	1	7	3	13	53	0	20	15	28	53	450	6	
	14 B	1	8	4	14	54	0	20	15	29	54	240	6	
	14 E	1	9	5	14	54	0	25	3	17	54	200	6	
	19 B	1	10	6	19	59	0	20	15	34	59	240	6	
	19 F	1	11	7	19	59	0	20	6	25	59	140	6	
	19 H	1	12	8	19	59	0	25	3	22	59	100	6	
	41 H	1	13	9	41	81	0	25	3	44	81	100	6	
	41 F	1	14	10	41	81	0	20	6	47	81	140	6	
	45 D	1	15	11	45	85	0	20	15	60	85	260	6	
	45 E	1	16	12	45	85	0	25	3	48	85	200	6	
	45 C	1	17	13	45	85	0	20	15	60	85	270	6	
	45 F	1	18	14	45	85	0	20	6	51	85	140	6	
	47 F	1	19	15	47	87	0	20	6	53	87	140	6	
	47 A	1	20	16	47	87	0	20	10	57	87	200	6	
	50 G	1	21	17	50	90	0	20	15	65	90	450	6	
	50 C	1	22	18	50	90	0	20	15	65	90	270	6	
	52 F	1	23	19	52	92	0	20	6	58	92	140	6	
	52 D	1	24	20	52	92	0	20	15	67	92	260	6	
	52 G	1	25	21	52	92	0	20	15	67	92	450	6	
	54 C	1	26	22	54	94	0	20	15	69	94	270	6	
	54 B	1	27	23	54	94	0	20	15	69	94	240	6	
	62 E	1	28	24	62	102	0	25	3	65	102	200	6	
	62 F	1	29	25	62	102	0	20	6	68	102	140	6	
	62 A	1	30	26	62	102	0	20	10	72	102	200	6	
	62 B	1	31	27	62	102	0	20	15	77	102	240	6	
	62 H	1	32	28	62	102	0	25	3	65	102	100	6	
	68 F	1	33	29	68	108	0	20	6	74	108	140	6	
	72 C	1	34	30	72	112	0	20	15	87	112	270	6	
	79 C	1	35	31	79	119	0	20	15	94	119	270	6	
	79 E	1	36	32	79	119	0	25	3	82	119	200	6	
	79 A	1	37	33	79	119	0	20	10	89	119	200	6	
	79 G	1	38	34	79	119	0	20	15	94	119	450	6	
	84 G	1	39	35	84	124	0	20	15	99	124	450	6	
	84 G	1	40	36	84	124	0	20	15	99	124	450	6	
	84 E	1	41	37	84	124	0	25	3	87	124	200	6	
	92 H	1	42	38	92	132	0	25	3	95	132	100	6	
	92 A	1	43	39	92	132	0	20	10	102	132	200	6	
	92 D	1	44	40	92	132	0	20	15	107	132	260	6	
	92 C	1	45	41	92	132	0	20	15	107	132	270	6	
	96 E	1	46	42	96	136	0	25	3	99	136	200	6	
	96 D	1	47	43	96	136	0	20	15	111	136	260	6	
	96 G	1	48	44	96	136	0	20	15	111	136	450	6	
	96 C	1	49	45	96	136	0	20	15	111	136	270	6	

3123	25 D	2	50	46	120	130	0	20	15	120	130	260	6
3123	25 F	2	51	47	120	130	0	20	6	120	130	140	6
3135	60 B	2	52	48	150	160	0	20	15	150	160	240	6
173	36 G	0	1	49	36	96	1	20	15	56	95	450	6
2034	40 A	0	2	50	40	100	8	20	10	55	92	200	6
2034	40 B	0	2	51	40	100	8	20	15	60	92	240	6
2034	40 B	0	2	52	40	100	8	20	15	60	92	240	6
2034	40 C	0	2	53	40	100	8	20	15	60	92	270	6
2034	40 E	0	2	54	40	100	8	25	3	48	92	200	6
2034	40 F	0	2	55	40	100	8	20	6	51	92	140	6
2034	40 G	0	2	56	40	100	8	20	15	60	92	450	6
2034	40 H	0	2	57	40	100	8	25	3	48	92	100	6
2034	40 H	0	2	58	40	100	8	25	3	48	92	100	6
2175	40 A	0	3	59	40	100	3	20	10	55	97	200	6
2175	40 A	0	3	60	40	100	3	20	10	55	97	200	6
2175	40 B	0	3	61	40	100	3	20	15	60	97	240	6
2175	40 B	0	3	62	40	100	3	20	15	60	97	240	6
2175	40 D	0	3	63	40	100	3	20	15	60	97	260	6
2175	40 G	0	3	64	40	100	3	20	15	60	97	450	6
2175	40 G	0	3	65	40	100	3	20	15	60	97	450	6
2175	40 H	0	3	66	40	100	3	25	3	48	97	100	6
2259	94 A	0	4	67	94	154	12	20	10	107	142	200	6
2259	94 D	0	4	68	94	154	12	20	15	112	142	260	6
2259	94 D	0	4	69	94	154	12	20	15	112	142	260	6
2259	94 E	0	4	70	94	154	12	25	3	100	142	200	6
2259	94 E	0	4	71	94	154	12	25	3	100	142	200	6
2259	94 F	0	4	72	94	154	12	20	6	103	142	140	6
2259	94 F	0	4	73	94	154	12	20	6	103	142	140	6
2259	94 G	0	4	74	94	154	12	20	15	112	142	450	6
2259	94 H	0	4	75	94	154	12	25	3	100	142	100	6
2259	94 H	0	4	76	94	154	12	25	3	100	142	100	6

餐廳 8

外送訂	點餐時	餐點種	訂單種	餐廳訂單編號	餐點編號	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
		46 H	1	5	1	46	86	0	25	3	49	86	100	8
		46 G	1	6	2	46	86	0	20	15	61	86	450	8
		46 F	1	7	3	46	86	0	20	6	52	86	140	8
		46 A	1	8	4	46	86	0	20	10	56	86	200	8
		46 C	1	9	5	46	86	0	20	15	61	86	270	8
		47 A	1	10	6	47	87	0	20	10	57	87	200	8
		47 E	1	11	7	47	87	0	25	3	50	87	200	8
		47 B	1	12	8	47	87	0	20	15	62	87	240	8
		59 H	1	13	9	59	99	0	25	3	62	99	100	8
		65 A	1	14	10	65	105	0	20	10	75	105	200	8
		67 H	1	15	11	67	107	0	25	3	70	107	100	8
		67 D	1	16	12	67	107	0	20	15	82	107	260	8
		68 A	1	17	13	68	108	0	20	10	78	108	200	8
		68 B	1	18	14	68	108	0	20	15	83	108	240	8
		68 E	1	19	15	68	108	0	25	3	71	108	200	8
		70 F	1	20	16	70	110	0	20	6	76	110	140	8
		70 H	1	21	17	70	110	0	25	3	73	110	100	8
		75 C	1	22	18	75	115	0	20	15	90	115	270	8
		75 G	1	23	19	75	115	0	20	15	90	115	450	8
		75 E	1	24	20	75	115	0	25	3	78	115	200	8
		76 C	1	25	21	76	116	0	20	15	91	116	270	8
		76 D	1	26	22	76	116	0	20	15	91	116	260	8
		76 A	1	27	23	76	116	0	20	10	86	116	200	8
		76 E	1	28	24	76	116	0	25	3	79	116	200	8
		83 A	1	29	25	83	123	0	20	10	93	123	200	8
		88 E	1	30	26	88	128	0	25	3	91	128	200	8
		88 D	1	31	27	88	128	0	20	15	103	128	260	8
		88 F	1	32	28	88	128	0	20	6	94	128	140	8
		91 D	1	33	29	91	131	0	20	15	106	131	260	8
		91 E	1	34	30	91	131	0	25	3	94	131	200	8
		91 D	1	35	31	91	131	0	20	15	106	131	260	8
		91 D	1	36	32	91	131	0	20	15	106	131	260	8
		91 E	1	37	33	91	131	0	25	3	94	131	200	8
		93 B	1	38	34	93	133	0	20	15	108	133	240	8
		93 E	1	39	35	93	133	0	25	3	96	133	200	8
		97 H	1	40	36	97	137	0	25	3	100	137	100	8
		98 D	1	41	37	98	138	0	20	15	113	138	260	8
		98 F	1	42	38	98	138	0	20	6	104	138	140	8
		98 E	1	43	39	98	138	0	25	3	101	138	200	8
		98 A	1	44	40	98	138	0	20	10	108	138	200	8
		98 D	1	45	41	98	138	0	20	15	113	138	260	8
		99 C	1	46	42	99	139	0	20	15	114	139	270	8

151	36 B	0	1	43	36	96	9	20	15	56	87	240	8
151	36 B	0	1	44	36	96	9	20	15	56	87	240	8
151	36 D	0	1	45	36	96	9	20	15	56	87	260	8
151	36 E	0	1	46	36	96	9	25	3	44	87	200	8
151	36 F	0	1	47	36	96	9	20	6	47	87	140	8
151	36 H	0	1	48	36	96	9	25	3	44	87	100	8
151	36 H	0	1	49	36	96	9	25	3	44	87	100	8
2015	40 G	0	2	50	40	100	2	20	15	60	98	450	8
2015	40 H	0	2	51	40	100	2	25	3	48	98	100	8
2043	40 A	0	3	52	40	100	16	20	10	55	84	200	8
2043	40 D	0	3	53	40	100	16	20	15	60	84	260	8
2043	40 F	0	3	54	40	100	16	20	6	51	84	140	8
2043	40 H	0	3	55	40	100	16	25	3	48	84	100	8
2105	40 C	0	4	56	40	100	6	20	15	60	94	270	8
2105	40 E	0	4	57	40	100	6	25	3	48	94	200	8



餐廳 15

外送訂	點餐時	餐點種	訂單種	餐廳訂單編號	餐點編號	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
	26 A		1	6	1	26	66	0	20	10	36	66	200	15
	26 B		1	7	2	26	66	0	20	15	41	66	240	15
	26 B		1	8	3	26	66	0	20	15	41	66	240	15
	26 C		1	9	4	26	66	0	20	15	41	66	270	15
	38 A		1	10	5	38	78	0	20	10	48	78	200	15
	38 A		1	11	6	38	78	0	20	10	48	78	200	15
	38 B		1	12	7	38	78	0	20	15	53	78	240	15
	59 D		1	13	8	59	99	0	20	15	74	99	260	15
	59 H		1	14	9	59	99	0	25	3	62	99	100	15
	59 F		1	15	10	59	99	0	20	6	65	99	140	15
	59 G		1	16	11	59	99	0	20	15	74	99	450	15
	59 B		1	17	12	59	99	0	20	15	74	99	240	15
	59 A		1	18	13	59	99	0	20	10	69	99	200	15
	59 E		1	19	14	59	99	0	25	3	62	99	200	15
	59 G		1	20	15	59	99	0	20	15	74	99	450	15
	59 D		1	21	16	59	99	0	20	15	74	99	260	15
	59 G		1	22	17	59	99	0	20	15	74	99	450	15
	59 A		1	23	18	59	99	0	20	10	69	99	200	15
	59 E		1	24	19	59	99	0	25	3	62	99	200	15
	59 C		1	25	20	59	99	0	20	15	74	99	270	15
	59 A		1	26	21	59	99	0	20	10	69	99	200	15
	59 E		1	27	22	59	99	0	25	3	62	99	200	15
	59 D		1	28	23	59	99	0	20	15	74	99	260	15
	59 C		1	29	24	59	99	0	20	15	74	99	270	15
	59 D		1	30	25	59	99	0	20	15	74	99	260	15
	59 E		1	31	26	59	99	0	25	3	62	99	200	15
	59 A		1	32	27	59	99	0	20	10	69	99	200	15
	59 G		1	33	28	59	99	0	20	15	74	99	450	15
	59 A		1	34	29	59	99	0	20	10	69	99	200	15
	59 H		1	35	30	59	99	0	25	3	62	99	100	15
	65 H		1	36	31	65	105	0	25	3	68	105	100	15
	65 B		1	37	32	65	105	0	20	15	80	105	240	15
	65 A		1	38	33	65	105	0	20	10	75	105	200	15
	71 B		1	39	34	71	111	0	20	15	86	111	240	15
	71 E		1	40	35	71	111	0	25	3	74	111	200	15
	71 C		1	41	36	71	111	0	20	15	86	111	270	15
	85 B		1	42	37	85	125	0	20	15	100	125	240	15
	85 G		1	43	38	85	125	0	20	15	100	125	450	15
	85 E		1	44	39	85	125	0	25	3	88	125	200	15
	85 D		1	45	40	85	125	0	20	15	100	125	260	15
	88 D		1	46	41	88	128	0	20	15	103	128	260	15
	88 H		1	47	42	88	128	0	25	3	91	128	100	15
	88 A		1	48	43	88	128	0	20	10	98	128	200	15
	88 B		1	49	44	88	128	0	20	15	103	128	240	15
	88 D		1	50	45	88	128	0	20	15	103	128	260	15
	88 A		1	51	46	88	128	0	20	10	98	128	200	15
	88 B		1	52	47	88	128	0	20	15	103	128	240	15
	88 D		1	53	48	88	128	0	20	15	103	128	260	15
	88 B		1	54	49	88	128	0	20	15	103	128	240	15
	88 H		1	55	50	88	128	0	25	3	91	128	100	15
	89 H		1	56	51	89	129	0	25	3	92	129	100	15
	89 E		1	57	52	89	129	0	25	3	92	129	200	15
	95 H		1	58	53	95	135	0	25	3	98	135	100	15
	95 A		1	59	54	95	135	0	20	10	105	135	200	15
	95 B		1	60	55	95	135	0	20	15	110	135	240	15
	95 D		1	61	56	95	135	0	20	15	110	135	260	15

226	36 D	0	1	57	36	96	10	20	15	56	86	260	15
226	36 G	0	1	58	36	96	10	20	15	56	86	450	15
226	36 H	0	1	59	36	96	10	25	3	44	86	100	15
2125	40 A	0	2	60	40	100	7	20	10	55	93	200	15
2125	40 E	0	2	61	40	100	7	25	3	48	93	200	15
2125	40 E	0	2	62	40	100	7	25	3	48	93	200	15
2125	40 E	0	2	63	40	100	7	25	3	48	93	200	15
2125	40 E	0	2	64	40	100	7	25	3	48	93	200	15
2125	40 G	0	2	65	40	100	7	20	15	60	93	450	15
2125	40 G	0	2	66	40	100	7	20	15	60	93	450	15
2125	40 H	0	2	67	40	100	7	25	3	48	93	100	15
2125	40 H	0	2	68	40	100	7	25	3	48	93	100	15
2167	40 F	0	3	69	40	100	17	20	6	51	83	140	15
2167	40 G	0	3	70	40	100	17	20	15	60	83	450	15
2167	40 G	0	3	71	40	100	17	20	15	60	83	450	15
2176	40 B	0	4	72	40	100	8	20	15	60	92	240	15
2176	40 C	0	4	73	40	100	8	20	15	60	92	270	15
2176	40 D	0	4	74	40	100	8	20	15	60	92	260	15
2176	40 D	0	4	75	40	100	8	20	15	60	92	260	15
2176	40 D	0	4	76	40	100	8	20	15	60	92	260	15
2176	40 E	0	4	77	40	100	8	25	3	48	92	200	15
2176	40 H	0	4	78	40	100	8	25	3	48	92	100	15
2281	94 A	0	5	79	94	154	7	20	10	107	147	200	15
2281	94 B	0	5	80	94	154	7	20	15	112	147	240	15
2281	94 F	0	5	81	94	154	7	20	6	103	147	140	15

外送訂	點餐時	餐點種	訂單種	餐廳訂單編號	餐點編號	時窗上	時窗下	與顧客	賞味期	製作時	更新後時	更新後時	價格	餐廳編
		23 E	1	5	1	23	63	0	25	3	26	63	200	17
		23 F	1	6	2	23	63	0	20	6	29	63	140	17
		23 B	1	7	3	23	63	0	20	15	38	63	240	17
		23 A	1	8	4	23	63	0	20	10	33	63	200	17
		32 D	1	9	5	32	72	0	20	15	47	72	260	17
		32 E	1	10	6	32	72	0	25	3	35	72	200	17
		32 B	1	11	7	32	72	0	20	15	47	72	240	17
		32 A	1	12	8	32	72	0	20	10	42	72	200	17
		33 E	1	13	9	33	73	0	25	3	36	73	200	17
		33 F	1	14	10	33	73	0	20	6	39	73	140	17
		33 G	1	15	11	33	73	0	20	15	48	73	450	17
		34 H	1	16	12	34	74	0	25	3	37	74	100	17
		34 A	1	17	13	34	74	0	20	10	44	74	200	17
		34 D	1	18	14	34	74	0	20	15	49	74	260	17
		34 C	1	19	15	34	74	0	20	15	49	74	270	17
		49 B	1	20	16	49	89	0	20	15	64	89	240	17
		49 D	1	21	17	49	89	0	20	15	64	89	260	17
		49 B	1	22	18	49	89	0	20	15	64	89	240	17
		49 F	1	23	19	49	89	0	20	6	55	89	140	17
		49 G	1	24	20	49	89	0	20	15	64	89	450	17
		49 B	1	25	21	49	89	0	20	15	64	89	240	17
		49 G	1	26	22	49	89	0	20	15	64	89	450	17
		49 A	1	27	23	49	89	0	20	10	59	89	200	17
		49 E	1	28	24	49	89	0	25	3	52	89	200	17
		49 H	1	29	25	49	89	0	25	3	52	89	100	17
		51 D	1	30	26	51	91	0	20	15	66	91	260	17
		51 G	1	31	27	51	91	0	20	15	66	91	450	17
		51 D	1	32	28	51	91	0	20	15	66	91	260	17
		51 C	1	33	29	51	91	0	20	15	66	91	270	17
		51 E	1	34	30	51	91	0	25	3	54	91	200	17
		51 F	1	35	31	51	91	0	20	6	57	91	140	17
		51 H	1	36	32	51	91	0	25	3	54	91	100	17
		51 D	1	37	33	51	91	0	20	15	66	91	260	17
		51 E	1	38	34	51	91	0	25	3	54	91	200	17
		54 D	1	39	35	54	94	0	20	15	69	94	260	17
		54 E	1	40	36	54	94	0	25	3	57	94	200	17
		54 H	1	41	37	54	94	0	25	3	57	94	100	17
		54 C	1	42	38	54	94	0	20	15	69	94	270	17
		64 C	1	43	39	64	104	0	20	15	79	104	270	17
		64 E	1	44	40	64	104	0	25	3	67	104	200	17
		64 D	1	45	41	64	104	0	20	15	79	104	260	17
		64 E	1	46	42	64	104	0	25	3	67	104	200	17
		64 F	1	47	43	64	104	0	20	6	70	104	140	17
		64 A	1	48	44	64	104	0	20	10	74	104	200	17
		64 C	1	49	45	64	104	0	20	15	79	104	270	17
		65 E	1	50	46	65	105	0	25	3	68	105	200	17
		65 F	1	51	47	65	105	0	20	6	71	105	140	17
		65 F	1	52	48	65	105	0	20	6	71	105	140	17
		65 A	1	53	49	65	105	0	20	10	75	105	200	17
		65 F	1	54	50	65	105	0	20	6	71	105	140	17
		65 C	1	55	51	65	105	0	20	15	80	105	270	17
		69 A	1	56	52	69	109	0	20	10	79	109	200	17
		69 C	1	57	53	69	109	0	20	15	84	109	270	17
		69 A	1	58	54	69	109	0	20	10	79	109	200	17
		69 C	1	59	55	69	109	0	20	15	84	109	270	17
		69 F	1	60	56	69	109	0	20	6	75	109	140	17
		69 B	1	61	57	69	109	0	20	15	84	109	240	17
		69 F	1	62	58	69	109	0	20	6	75	109	140	17
		69 F	1	63	59	69	109	0	20	6	75	109	140	17
		79 D	1	64	60	79	119	0	20	15	94	119	260	17

餐廳 17

208	36 A	0	1	70	36	96	9	20	10	51	87	200	17
208	36 E	0	1	71	36	96	9	25	3	44	87	200	17
208	36 F	0	1	72	36	96	9	20	6	47	87	140	17
208	36 F	0	1	73	36	96	9	20	6	47	87	140	17
208	36 G	0	1	74	36	96	9	20	15	56	87	450	17
2097	40 A	0	2	75	40	100	5	20	10	55	95	200	17
2097	40 D	0	2	76	40	100	5	20	15	60	95	260	17
2097	40 D	0	2	77	40	100	5	20	15	60	95	260	17
2097	40 F	0	2	78	40	100	5	20	6	51	95	140	17
2180	40 B	0	3	79	40	100	7	20	15	60	93	240	17
2180	40 B	0	3	80	40	100	7	20	15	60	93	240	17
2180	40 F	0	3	81	40	100	7	20	6	51	93	140	17
2189	94 C	0	4	82	94	154	3	20	15	112	151	270	17
2189	94 D	0	4	83	94	154	3	20	15	112	151	260	17
2189	94 E	0	4	84	94	154	3	25	3	100	151	200	17
2189	94 E	0	4	85	94	154	3	25	3	100	151	200	17
2189	94 E	0	4	86	94	154	3	25	3	100	151	200	17
2189	94 H	0	4	87	94	154	3	25	3	100	151	100	17

外送訂單資訊 (由於資料龐大外送訂單資訊僅節錄前 50 張訂單

外送訂單	餐廳地址	顧客地址	餐廳編號	顧客編號	取餐節點	顧客節點	送達時間	送達時間	CF	PF	容量
3123	台北市大	台北市萬華區萬大路346號	365	312	300	626	140	150	0	0	2
3135	台北市大	台北市萬華區成都路88號	365	302	301	627	170	180	0	0	1
3120	台北市大	台北市內湖區東湖路106巷7弄3號	381	237	302	628	140	150	0	0	5
3139	台北市松	台北市中正區濟南路二段37號	402	121	303	629	170	180	0	0	1
3124	台北市萬	台北市信義區松廉路15號	436	182	304	630	140	150	0	0	2
3140	台北市中	台北市中山區復興南路一段28號	439	103	305	631	170	180	0	0	2
3141	台北市中	台北市中山區新生北路三段82之5號	459	106	306	632	170	180	0	0	1
3142	台北市中	台北市中山區新生北路二段38號	467	104	307	633	170	180	0	0	15
3126	台北市大	台北市內湖區瑞光路583巷24號	470	208	308	634	170	180	0	0	6
3122	台北市文	台北市大同區迪化街一段311號	608	6	309	635	170	180	0	0	4
3134	台北市中	台北市中正區汀州路三段230巷14弄2號	522	331	310	636	170	180	0	0	3
3136	台北市大	台北市中山區新生北路三段82之5號	526	106	311	637	170	180	0	0	5
3137	台北市大	台北市萬華區萬大路346號	393	312	312	638	170	180	0	0	4
3119	台北市松	台北市信義區市府路1號	548	164	313	639	170	180	0	0	2
3121	台北市南	台北市中山區新生北路二段38號	564	104	314	640	170	180	0	0	3
3125	台北市中	台北市松山區民生東路三段111號	517	139	315	641	170	180	0	0	3
3138	台北市大	台北市中山區龍江路62號	622	112	316	642	170	180	0	0	3
213	台北市大	台北市士林區中山北路六段77號	480	261	317	643	36	96	27	63	1
2047	台北市大	台北市南港區南港路三段47巷8號	480	252	318	644	40	100	78	182	3
2099	台北市大	台北市南港區南港路二段20巷5號	480	246	319	645	40	100	113	263	4
2253	台北市大	台北市松山區饒河街	480	161	320	646	94	154	56	130	2
2255	台北市大	台北市中山區農安街30號	480	108	321	647	94	154	167	389	6
173	台北市大	台北市中正區忠孝西路一段36號	393	294	322	648	36	96	135	315	1
2034	台北市大	台北市中正區羅斯福路三段126之5號	393	122	323	649	40	100	291	679	9
2175	台北市大	中山區大直街20巷18號	393	198	324	650	40	100	321	749	8
2259	台北市大	台北市士林區天母西路9號	393	280	325	651	94	154	308	718	10
151	台北市信	台北市中山區林森北路369號	409	92	326	652	36	96	384	896	7
2015	台北市信	台北市中山區新生北路二段38號	409	104	327	653	40	100	83	193	2
2043	台北市信	台北市士林區中山北路七段31號	409	271	328	654	40	100	105	245	4
2105	台北市信	台北市松山區南京東路五段291巷20弄3號	409	152	329	655	40	100	71	165	2
226	台北市土	台北市大同區大龍街51號	386	1	330	656	36	96	243	567	3
2125	台北市土	台北市士林區中山北路七段190巷6號	386	265	331	657	40	100	315	735	9
2167	台北市土	台北市士林區至誠路一段285號	386	264	332	658	40	100	156	364	3
2176	台北市土	台北市大安區忠孝東路三段217巷1弄6號	386	37	333	659	40	100	239	557	7
2281	台北市土	台北市大安區基隆路四段43號	386	329	334	660	94	154	87	203	3
208	台北市信	台北市中山區中山北路二段48巷7號	377	70	335	661	36	96	339	791	5
2097	台北市信	台北市大同區重慶北路二段171號	377	7	336	662	40	100	129	301	4
2180	台北市信	台北市中山區林森北路410號	377	93	337	663	40	100	93	217	3
2189	台北市信	台北市文山區羅斯福路六段39巷	377	334	338	664	94	154	185	431	6
238	台北市大	台北市松山區民生東路三段135號	469	140	339	665	36	96	66	154	2
179	台北市大	台北市中山區撫順街16號	469	109	340	666	36	96	45	105	1
2292	台北市大	台北市士林區克強路3號	469	259	341	667	94	154	45	105	3
235	台北市信	台北市士林區中山北路七段140巷1號	373	267	342	668	36	96	225	525	6
210	台北市信	台北市士林區文昌路95號	373	282	343	669	36	96	63	147	3
264	台北市信	台北市大安區通安街139號	373	48	344	670	36	96	90	210	3
2044	台北市信	台北市文山區景福街225號	373	343	345	671	40	100	80	186	4
2082	台北市信	台北市中山區撫順街16號	373	109	346	672	40	100	102	238	6
2193	台北市信	台北市信義區忠孝東路五段71巷17-22	373	170	347	673	94	154	327	763	18
168	台北市土	台北市信義區於廠路88號	405	192	348	674	36	96	168	392	3
134	台北市土	台北市中正區思源街1號	405	335	349	675	36	96	147	343	3

距離矩陣 (由於資料龐大距離矩陣僅節錄前 30 個節點組成之矩陣-30*30)

999	11	6	5	5	4	4	5	3	12	6	12	13	6	6	2	6	6	3	11	7	5	8	5	8	3	14	12	4
11	999	7	10	9	8	7	6	11	4	7	21	15	9	8	11	9	7	11	3	17	6	14	6	18	11	16	14	7
6	7	999	4	8	2	2	1	7	9	1	16	10	3	1	5	3	0	7	9	11	6	8	1	12	7	11	9	1
5	10	4	999	9	3	3	4	7	13	5	12	9	3	4	3	3	4	7	12	7	8	4	4	8	7	9	8	3
5	9	8	9	999	7	7	7	3	9	8	15	17	9	8	6	9	8	3	9	11	3	13	7	12	3	18	16	7
4	8	2	3	7	999	1	2	7	10	2	14	10	3	2	4	3	2	7	10	10	6	6	2	11	7	11	9	1
4	7	2	3	7	1	999	1	6	10	2	14	10	3	2	4	3	2	6	10	10	6	6	1	11	6	12	9	1
5	6	1	4	7	2	1	999	7	9	1	16	11	4	2	5	4	1	7	9	11	5	8	0	12	7	12	10	1
3	11	7	7	3	7	6	7	999	11	8	13	16	9	8	5	9	7	0	11	9	5	11	7	10	0	17	15	6
12	4	9	13	9	10	10	9	11	999	10	22	18	12	10	12	12	9	11	1	17	6	16	9	19	11	19	17	10
6	7	1	5	8	2	2	1	8	10	999	16	9	2	1	6	2	1	8	9	12	6	8	1	13	8	11	8	2
12	21	16	12	15	14	14	16	13	22	16	999	17	15	16	11	15	16	13	22	5	17	10	16	4	13	16	13	15
13	15	10	9	17	10	10	11	16	18	9	17	999	8	9	11	8	10	16	17	14	15	9	11	14	16	3	4	11
6	9	3	3	9	3	3	4	9	12	2	15	8	999	2	5	0	3	9	11	10	8	6	4	11	9	9	6	4
6	8	1	4	8	2	2	2	8	10	1	16	9	2	999	5	2	1	8	10	11	7	7	2	12	8	10	8	2
2	11	5	3	6	4	4	5	5	12	6	11	11	5	5	999	5	5	5	11	6	6	6	5	7	5	12	10	4
6	9	3	3	9	3	3	4	9	12	2	15	8	0	2	5	999	3	9	11	10	8	6	4	11	9	9	6	4
6	7	0	4	8	2	2	1	7	9	1	16	10	3	1	5	3	999	7	9	11	6	8	1	12	7	11	9	1
3	11	7	7	3	7	6	7	0	11	8	13	16	9	8	5	9	7	999	11	9	5	11	7	10	0	17	15	6
11	3	9	12	9	10	10	9	11	1	9	22	17	11	10	11	11	9	11	999	17	6	16	9	18	11	19	16	9
7	17	11	7	11	10	10	11	9	17	12	5	14	10	11	6	10	11	9	17	999	12	6	11	1	9	13	10	10
5	6	6	8	3	6	6	5	5	6	6	17	15	8	7	6	8	6	5	6	12	999	11	5	13	5	16	14	5
8	14	8	4	13	6	6	8	11	16	8	10	9	6	7	6	6	8	11	16	6	11	999	8	7	11	8	5	7
5	6	1	4	7	2	1	0	7	9	1	16	11	4	2	5	4	1	7	9	11	5	8	999	12	7	12	10	1
8	18	12	8	12	11	11	12	10	19	13	4	14	11	12	7	11	12	10	18	1	13	7	12	999	10	12	10	11
3	11	7	7	3	7	6	7	0	11	8	13	16	9	8	5	9	7	0	11	9	5	11	7	10	999	17	15	6
14	16	11	9	18	11	12	12	17	19	11	16	3	9	10	12	9	11	17	19	13	16	8	12	12	17	999	3	12
12	14	9	8	16	9	9	10	15	17	8	13	4	6	8	10	6	9	15	16	10	14	5	10	10	15	3	999	10
4	7	1	3	7	1	1	1	6	10	2	15	11	4	2	4	4	1	6	9	10	5	7	1	11	6	12	10	999

單位：分鐘

■ 實務例題 (題組 3)

餐廳餐點資訊 (由於資料龐大餐廳餐點資訊僅節錄前 5 家餐廳)

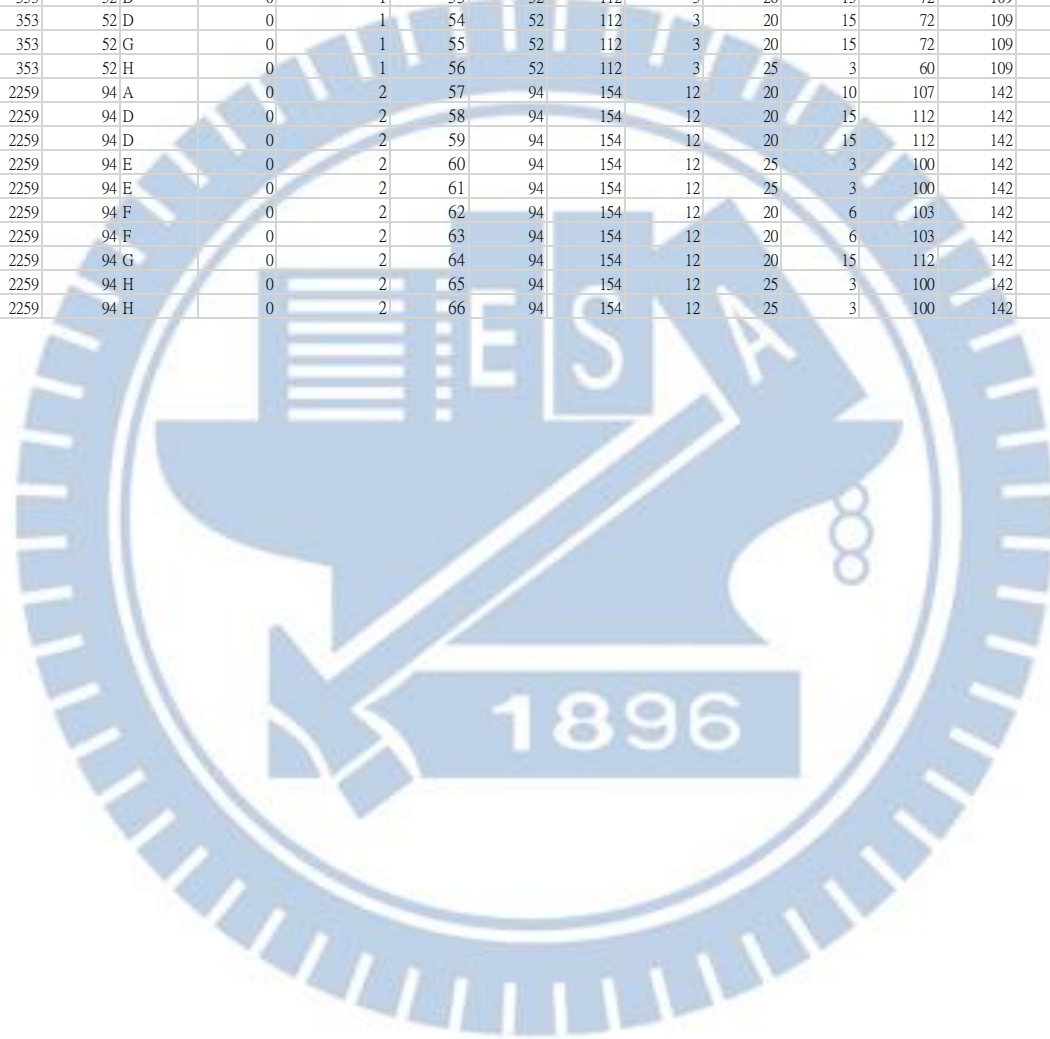
餐廳 4

外送訂單編號	餐點種類	訂單種類	餐廳訂單編號	餐點編號	時窗上界	時窗下界	與顧客距	賞味期限	製作時間	更新後時窗	更新後時窗	價格	餐廳編號
	31 F	1	5	1	31	71	0	20	6	37	71	90	4
	52 C	1	6	2	52	92	0	20	15	67	92	220	4
	54 A	1	7	3	54	94	0	20	10	64	94	190	4
	55 D	1	8	4	55	95	0	20	15	70	95	200	4
	58 E	1	9	5	58	98	0	25	3	61	98	170	4
	76 A	1	10	6	76	116	0	20	10	86	116	190	4
	76 F	1	11	7	76	116	0	20	6	82	116	90	4
	96 A	1	12	8	96	136	0	20	10	106	136	190	4
	101 E	1	13	9	101	141	0	25	3	104	141	170	4
	112 F	1	14	10	112	152	0	20	6	118	152	90	4
	113 C	1	15	11	113	153	0	20	15	128	153	220	4
	114 B	1	16	12	114	154	0	20	15	129	154	230	4
373	52 A	0	1	13	52	112	7	20	10	67	105	190	4
373	52 C	0	1	14	52	112	7	20	15	72	105	220	4
373	52 E	0	1	15	52	112	7	25	3	60	105	170	4
385	52 B	0	2	16	52	112	11	20	15	72	101	230	4
385	52 C	0	2	17	52	112	11	20	15	72	101	220	4
385	52 C	0	2	18	52	112	11	20	15	72	101	220	4
385	52 C	0	2	19	52	112	11	20	15	72	101	220	4
385	52 E	0	2	20	52	112	11	25	3	60	101	170	4
385	52 F	0	2	21	52	112	11	20	6	63	101	90	4
2253	94 D	0	3	22	94	154	6	20	15	112	148	200	4
2253	94 E	0	3	23	94	154	6	25	3	100	148	170	4
2255	94 B	0	4	24	94	154	9	20	15	112	145	230	4
2255	94 C	0	4	25	94	154	9	20	15	112	145	220	4
2255	94 D	0	4	26	94	154	9	20	15	112	145	200	4
2255	94 D	0	4	27	94	154	9	20	15	112	145	200	4
2255	94 E	0	4	28	94	154	9	25	3	100	145	170	4
2255	94 F	0	4	29	94	154	9	20	6	103	145	90	4

餐廳 6

外送訂單	餐點時間	餐點種類	訂單種類	餐廳訂單編號	餐點編號	時窗上界	時窗下界	與顧客距離	賞味期限	製作時間	更新後時窗	更新後時窗	價格	餐廳編號
	13 F		1	3	1	13	53	0	20	6	19	53	140	6
	13 H		1	4	2	13	53	0	25	3	16	53	100	6
	13 G		1	5	3	13	53	0	20	15	28	53	450	6
	14 B		1	6	4	14	54	0	20	15	29	54	240	6
	14 E		1	7	5	14	54	0	25	3	17	54	200	6
	19 B		1	8	6	19	59	0	20	15	34	59	240	6
	19 F		1	9	7	19	59	0	20	6	25	59	140	6
	19 H		1	10	8	19	59	0	25	3	22	59	100	6
	44 E		1	11	9	44	84	0	25	3	47	84	200	6
	44 H		1	12	10	44	84	0	25	3	47	84	100	6
	44 B		1	13	11	44	84	0	20	15	59	84	240	6
	47 G		1	14	12	47	87	0	20	15	62	87	450	6
	47 H		1	15	13	47	87	0	25	3	50	87	100	6
	47 G		1	16	14	47	87	0	20	15	62	87	450	6
	47 C		1	17	15	47	87	0	20	15	62	87	270	6
	52 D		1	18	16	52	92	0	20	15	67	92	260	6
	52 C		1	19	17	52	92	0	20	15	67	92	270	6
	52 D		1	20	18	52	92	0	20	15	67	92	260	6
	52 A		1	21	19	52	92	0	20	10	62	92	200	6
	52 D		1	22	20	52	92	0	20	15	67	92	260	6
	62 E		1	23	21	62	102	0	25	3	65	102	200	6
	62 F		1	24	22	62	102	0	20	6	68	102	140	6
	62 A		1	25	23	62	102	0	20	10	72	102	200	6
	62 B		1	26	24	62	102	0	20	15	77	102	240	6
	62 H		1	27	25	62	102	0	25	3	65	102	100	6
	68 F		1	28	26	68	108	0	20	6	74	108	140	6
	72 C		1	29	27	72	112	0	20	15	87	112	270	6
	79 C		1	30	28	79	119	0	20	15	94	119	270	6
	79 E		1	31	29	79	119	0	25	3	82	119	200	6
	79 A		1	32	30	79	119	0	20	10	89	119	200	6
	79 G		1	33	31	79	119	0	20	15	94	119	450	6
	84 G		1	34	32	84	124	0	20	15	99	124	450	6
	84 G		1	35	33	84	124	0	20	15	99	124	450	6
	84 E		1	36	34	84	124	0	25	3	87	124	200	6
	92 H		1	37	35	92	132	0	25	3	95	132	100	6
	92 A		1	38	36	92	132	0	20	10	102	132	200	6
	92 D		1	39	37	92	132	0	20	15	107	132	260	6
	92 C		1	40	38	92	132	0	20	15	107	132	270	6
	96 E		1	41	39	96	136	0	25	3	99	136	200	6
	96 D		1	42	40	96	136	0	20	15	111	136	260	6
	96 G		1	43	41	96	136	0	20	15	111	136	450	6
	96 C		1	44	42	96	136	0	20	15	111	136	270	6
	102 B		1	45	43	102	142	0	20	15	117	142	240	6
	102 H		1	46	44	102	142	0	25	3	105	142	100	6
	105 A		1	47	45	105	145	0	20	10	115	145	200	6
	105 E		1	48	46	105	145	0	25	3	108	145	200	6
	105 B		1	49	47	105	145	0	20	15	120	145	240	6
	109 B		1	50	48	109	149	0	20	15	124	149	240	6
	113 E		1	51	49	113	153	0	25	3	116	153	200	6

3135	60 B	2	52	50	150	160	0	20	15	150	160	240	6
173	36 G	2	53	51	36	96	1	20	15	84	94	450	6
353	52 A	0	1	52	52	112	3	20	10	67	109	200	6
353	52 D	0	1	53	52	112	3	20	15	72	109	260	6
353	52 D	0	1	54	52	112	3	20	15	72	109	260	6
353	52 G	0	1	55	52	112	3	20	15	72	109	450	6
353	52 H	0	1	56	52	112	3	25	3	60	109	100	6
2259	94 A	0	2	57	94	154	12	20	10	107	142	200	6
2259	94 D	0	2	58	94	154	12	20	15	112	142	260	6
2259	94 D	0	2	59	94	154	12	20	15	112	142	260	6
2259	94 E	0	2	60	94	154	12	25	3	100	142	200	6
2259	94 E	0	2	61	94	154	12	25	3	100	142	200	6
2259	94 F	0	2	62	94	154	12	20	6	103	142	140	6
2259	94 F	0	2	63	94	154	12	20	6	103	142	140	6
2259	94 G	0	2	64	94	154	12	20	15	112	142	450	6
2259	94 H	0	2	65	94	154	12	25	3	100	142	100	6
2259	94 H	0	2	66	94	154	12	25	3	100	142	100	6



餐廳 7

外送訂單	點餐時間	餐點種類	訂單種類	餐廳訂單編號	餐點編號	時窗上界	時窗下界	與顧客距	賞味期限	製作時間	更新後時窗	更新後時窗	價格	餐廳編號
		41 F	1	6	1	41	81	0	20	6	47	81	140	7
		41 D	1	7	2	41	81	0	20	15	56	81	260	7
		41 C	1	8	3	41	81	0	20	15	56	81	270	7
		41 D	1	9	4	41	81	0	20	15	56	81	260	7
		46 H	1	10	5	46	86	0	25	3	49	86	100	7
		46 E	1	11	6	46	86	0	25	3	49	86	200	7
		46 B	1	12	7	46	86	0	20	15	61	86	240	7
		46 A	1	13	8	46	86	0	20	10	56	86	200	7
		63 B	1	14	9	63	103	0	20	15	78	103	240	7
		63 D	1	15	10	63	103	0	20	15	78	103	260	7
		63 E	1	16	11	63	103	0	25	3	66	103	200	7
		63 A	1	17	12	63	103	0	20	10	73	103	200	7
		67 G	1	18	13	67	107	0	20	15	82	107	450	7
		67 A	1	19	14	67	107	0	20	10	77	107	200	7
		67 F	1	20	15	67	107	0	20	6	73	107	140	7
		68 F	1	21	16	68	108	0	20	6	74	108	140	7
		68 H	1	22	17	68	108	0	25	3	71	108	100	7
		68 H	1	23	18	68	108	0	25	3	71	108	100	7
		68 C	1	24	19	68	108	0	20	15	83	108	270	7
		85 E	1	25	20	85	125	0	25	3	88	125	200	7
		85 F	1	26	21	85	125	0	20	6	91	125	140	7
		85 E	1	27	22	85	125	0	25	3	88	125	200	7
		85 D	1	28	23	85	125	0	20	15	100	125	260	7
		85 B	1	29	24	85	125	0	20	15	100	125	240	7
		104 F	1	30	25	104	144	0	20	6	110	144	140	7
		104 D	1	31	26	104	144	0	20	15	119	144	260	7
		104 H	1	32	27	104	144	0	25	3	107	144	100	7
		104 E	1	33	28	104	144	0	25	3	107	144	200	7
		110 H	1	34	29	110	150	0	25	3	113	150	100	7
		110 B	1	35	30	110	150	0	20	15	125	150	240	7
		110 A	1	36	31	110	150	0	20	10	120	150	200	7
		110 G	1	37	32	110	150	0	20	15	125	150	450	7
		111 F	1	38	33	111	151	0	20	6	117	151	140	7
		111 C	1	39	34	111	151	0	20	15	126	151	270	7
		114 E	1	40	35	114	154	0	25	3	117	154	200	7
		114 H	1	41	36	114	154	0	25	3	117	154	100	7
		114 C	1	42	37	114	154	0	20	15	129	154	270	7
		114 A	1	43	38	114	154	0	20	10	124	154	200	7
		114 D	1	44	39	114	154	0	20	15	129	154	260	7
		114 D	1	45	40	114	154	0	20	15	129	154	260	7

386	52 E	0	1	41	52	112	13	25	3	60	99	200	7
386	52 F	0	1	42	52	112	13	20	6	63	99	140	7
386	52 G	0	1	43	52	112	13	20	15	72	99	450	7
386	52 G	0	1	44	52	112	13	20	15	72	99	450	7
386	52 H	0	1	45	52	112	13	25	3	60	99	100	7
294	52 A	0	2	46	52	112	10	20	10	67	102	200	7
294	52 A	0	2	47	52	112	10	20	10	67	102	200	7
294	52 B	0	2	48	52	112	10	20	15	72	102	240	7
294	52 C	0	2	49	52	112	10	20	15	72	102	270	7
294	52 C	0	2	50	52	112	10	20	15	72	102	270	7
294	52 C	0	2	51	52	112	10	20	15	72	102	270	7
294	52 D	0	2	52	52	112	10	20	15	72	102	260	7
294	52 D	0	2	53	52	112	10	20	15	72	102	260	7
294	52 D	0	2	54	52	112	10	20	15	72	102	260	7
294	52 D	0	2	55	52	112	10	20	15	72	102	260	7
294	52 E	0	2	56	52	112	10	25	3	60	102	200	7
294	52 E	0	2	57	52	112	10	25	3	60	102	200	7
294	52 F	0	2	58	52	112	10	20	6	63	102	140	7
294	52 F	0	2	59	52	112	10	20	6	63	102	140	7
294	52 H	0	2	60	52	112	10	25	3	60	102	100	7
2218	94 B	0	3	61	94	154	3	20	15	112	151	240	7
2218	94 B	0	3	62	94	154	3	20	15	112	151	240	7
2319	94 A	0	4	63	94	154	15	20	10	107	139	200	7
2319	94 B	0	4	64	94	154	15	20	15	112	139	240	7
2319	94 B	0	4	65	94	154	15	20	15	112	139	240	7
2319	94 D	0	4	66	94	154	15	20	15	112	139	260	7
2319	94 D	0	4	67	94	154	15	20	15	112	139	260	7
2319	94 E	0	4	68	94	154	15	25	3	100	139	200	7
2319	94 E	0	4	69	94	154	15	25	3	100	139	200	7
2319	94 E	0	4	70	94	154	15	25	3	100	139	200	7
2319	94 E	0	4	71	94	154	15	25	3	100	139	200	7
2319	94 H	0	4	72	94	154	15	25	3	100	139	100	7
2319	94 H	0	4	73	94	154	15	25	3	100	139	100	7
2319	94 H	0	4	74	94	154	15	25	3	100	139	100	7
2330	94 E	0	5	75	94	154	17	25	3	100	137	200	7

餐廳 8

外送訂單	點餐時間	餐點種類	訂單種類	餐廳訂單編號	餐點編號	時窗上界	時窗下界	與顧客距	賞味期限	製作時間	更新後時窗	更新後時窗	價格	餐廳編號
	59	H	1	2	1	59	99	0	25	3	62	99	100	8
	65	A	1	3	2	65	105	0	20	10	75	105	200	8
	67	H	1	4	3	67	107	0	25	3	70	107	100	8
	67	D	1	5	4	67	107	0	20	15	82	107	260	8
	68	A	1	6	5	68	108	0	20	10	78	108	200	8
	68	B	1	7	6	68	108	0	20	15	83	108	240	8
	68	E	1	8	7	68	108	0	25	3	71	108	200	8
	70	F	1	9	8	70	110	0	20	6	76	110	140	8
	70	H	1	10	9	70	110	0	25	3	73	110	100	8
	75	C	1	11	10	75	115	0	20	15	90	115	270	8
	75	G	1	12	11	75	115	0	20	15	90	115	450	8
	75	E	1	13	12	75	115	0	25	3	78	115	200	8
	76	C	1	14	13	76	116	0	20	15	91	116	270	8
	76	D	1	15	14	76	116	0	20	15	91	116	260	8
	76	A	1	16	15	76	116	0	20	10	86	116	200	8
	76	E	1	17	16	76	116	0	25	3	79	116	200	8
	83	A	1	18	17	83	123	0	20	10	93	123	200	8
	88	E	1	19	18	88	128	0	25	3	91	128	200	8
	88	D	1	20	19	88	128	0	20	15	103	128	260	8
	88	F	1	21	20	88	128	0	20	6	94	128	140	8
	91	D	1	22	21	91	131	0	20	15	106	131	260	8
	91	E	1	23	22	91	131	0	25	3	94	131	200	8
	91	D	1	24	23	91	131	0	20	15	106	131	260	8
	91	D	1	25	24	91	131	0	20	15	106	131	260	8
	91	E	1	26	25	91	131	0	25	3	94	131	200	8
	93	B	1	27	26	93	133	0	20	15	108	133	240	8
	93	E	1	28	27	93	133	0	25	3	96	133	200	8
	97	H	1	29	28	97	137	0	25	3	100	137	100	8
	98	D	1	30	29	98	138	0	20	15	113	138	260	8
	98	F	1	31	30	98	138	0	20	6	104	138	140	8
	98	E	1	32	31	98	138	0	25	3	101	138	200	8
	98	A	1	33	32	98	138	0	20	10	108	138	200	8
	98	D	1	34	33	98	138	0	20	15	113	138	260	8
	99	C	1	35	34	99	139	0	20	15	114	139	270	8
	101	G	1	36	35	101	141	0	20	15	116	141	450	8
	101	H	1	37	36	101	141	0	25	3	104	141	100	8
	101	G	1	38	37	101	141	0	20	15	116	141	450	8
	114	D	1	39	38	114	154	0	20	15	129	154	260	8
390	52	D	0	1	39	52	112	3	20	15	72	109	260	8
390	52	G	0	1	40	52	112	3	20	15	72	109	450	8

餐廳 11

外送訂單編號	餐點時間	餐點種類	訂單種類	餐廳訂單編號	餐點編號	時窗上界	時窗下界	與顧客距	賞味期限	製作時間	更新後時窗	更新後時窗	價格	餐廳編號
	54	B	1	3	1	54	94	0	25	3	57	94	150	11
	54	C	1	4	2	54	94	0	20	6	60	94	70	11
	59	A	1	5	3	59	99	0	20	10	69	99	160	11
	68	A	1	6	4	68	108	0	20	10	78	108	160	11
	77	A	1	7	5	77	117	0	20	10	87	117	160	11
	77	B	1	8	6	77	117	0	25	3	80	117	150	11
	85	B	1	9	7	85	125	0	25	3	88	125	150	11
	100	A	1	10	8	100	140	0	20	10	110	140	160	11
	113	B	1	11	9	113	153	0	25	3	116	153	150	11
	113	A	1	12	10	113	153	0	20	10	123	153	160	11
357	52	A	0	1	11	52	112	4	20	10	67	108	160	11
357	52	C	0	1	12	52	112	4	20	6	63	108	70	11
2313	94	B	0	2	13	94	154	11	25	3	100	143	150	11
2313	94	B	0	2	14	94	154	11	25	3	100	143	150	11
2313	94	C	0	2	15	94	154	11	10	6	103	143	70	11
2313	94	C	0	2	16	94	154	11	10	6	103	143	70	11

外送訂單資訊 (由於資料龐大外送訂單資訊僅節錄前 50 張訂單)

外送訂單#	餐廳地址	顧客地址	餐廳編號	顧客編號	取餐節點	顧客節點	送達時間	送達時間	CF	PF	容量
3135	台北市大安區永康街6巷6號號	台北市萬華區成都路88號	365	302	300	575	170	180	0	0	1
3139	台北市松山區民生東路三段134號A棟1樓	台北市中正區濟南路二段37號	402	121	301	576	170	180	0	0	1
3140	台北市中正區仁愛路二段108號	台北市中山區復興南路一段28號	439	103	302	577	170	180	0	0	2
3133	台北市大安區忠孝東路四段126-1號	台北市萬華區中華路二段484巷6號	451	314	303	578	170	180	0	0	3
3141	台北市中山區中山北路一段121巷9號	台北市中山區新生北路二段82之5號	459	106	304	579	170	180	0	0	1
3142	台北市中山區中山北路一段105巷22號	台北市中山區新生北路二段38號	467	104	305	580	170	180	0	0	15
3128	台北市中山區吉林路226號1樓	台北市中山區林森北路410號	483	93	306	581	170	180	0	0	2
3134	台北市中正區重慶南路一段5巷1號	台北市中正區汀州路三段230巷14弄2號	522	331	307	582	170	180	0	0	3
3127	台北市士林區天玉街38巷15號	台北市大安區大安路二段63號	524	18	308	583	170	180	0	0	7
3136	台北市大安區忠孝東路四段126之13號	台北市中山區新生北路二段82之5號	526	106	309	584	170	180	0	0	5
3132	台北市大同區承德路一段1號	台北市大安區和平東路三段63號	393	34	310	585	170	180	0	0	1
3137	台北市大同區承德路一段1號	台北市萬華區萬大路346號	393	312	311	586	170	180	0	0	4
3129	台北市中山區市民大道三段2號	台北市信義區忠孝東路五段236巷15號	593	169	312	587	170	180	0	0	18
373	台北市大同區赤峰街59號	台北市松山區八德路四段138號	480	131	313	588	52	112	174	406	3
385	台北市大同區赤峰街59號	台北市內湖區瑞光路583巷24號	480	208	314	589	52	112	345	805	6
2253	台北市大同區赤峰街59號	台北市松山區饒河街	480	161	315	590	94	154	56	130	2
2255	台北市大同區赤峰街59號	台北市中山區農安街30號	480	108	316	591	94	154	167	389	6
353	台北市大同區承德路一段1號	台北市中山區長安東路二段68號	393	321	317	592	52	112	381	889	5
2259	台北市大同區承德路一段1號	台北市士林區天母西路9號	393	280	318	593	94	154	308	718	10
386	台北市大安區忠孝東路四段98號	台北市士林區中山北路六段77號	378	261	319	594	52	112	402	938	5
294	台北市大安區忠孝東路四段98號	台北市內湖區環山路一段25號	378	205	320	595	52	112	981	2289	15
2218	台北市大安區忠孝東路四段98號	台北市文山區一壽街44巷1號	378	359	321	596	94	154	72	168	2
2319	台北市大安區忠孝東路四段98號	台北市信義區松隆路5-1	378	181	322	597	94	154	345	805	12
2330	台北市大安區忠孝東路四段98號	台北市士林區中山北路六段419巷37-1號	378	281	323	598	94	154	30	70	1
390	台北市信義區松壽路20號	台北市南港區中坡南路55號	409	195	324	599	52	112	213	497	2
357	台北市大安區敦化南路二段164號	台北市中正區羅斯福路三段62號	562	125	325	600	52	112	69	161	2
2313	台北市大安區敦化南路二段164號	台北市士林區中山北路七段152-1號	562	266	326	601	94	154	66	154	4
370	台北市士林區忠誠路二段200號2樓	台北市中山區民生東路三段67號	386	76	327	602	52	112	474	1106	7
388	台北市士林區忠誠路二段200號2樓	台北市大安區羅斯福路二段77巷	386	326	328	603	52	112	132	308	2
2281	台北市士林區忠誠路二段200號2樓	台北市大安區基隆路四段43號	386	329	329	604	94	154	87	203	3
2398	台北市士林區忠誠路二段200號2樓	台北市文山區萬和街1號	386	348	330	605	115	175	164	382	4
2407	台北市士林區忠誠路二段200號2樓	台北市內湖區內湖路一段737巷71號	386	220	331	606	115	175	71	165	3
339	台北市信義區松壽路22號2樓	台北市信義區基隆路二段111號	377	190	332	607	52	112	468	1092	6
2189	台北市信義區松壽路22號2樓	台北市文山區羅斯福路六段39巷	377	334	333	608	94	154	185	431	6
2356	台北市信義區松壽路22號2樓	台北市松山區八德路四段138號	377	131	334	609	115	175	225	525	5
2448	台北市信義區松壽路22號2樓	台北市中正區羅斯福路一段8之1號	377	325	335	610	115	175	378	882	12
317	台北市大安區安東街57-1號	台北市內湖區文德路2號	469	214	336	611	52	112	90	210	3
2292	台北市大安區安東街57-1號	台北市士林區克強路3號	469	259	337	612	94	154	45	105	3
2401	台北市大安區安東街57-1號	台北市內湖區內湖路一段360巷2號	469	209	338	613	115	175	57	133	3
367	台北市萬華區南寧路50號	台北市萬華區昆明街214-1號	501	303	339	614	52	112	183	427	3
2299	台北市萬華區南寧路50號	台北市信義區忠孝東路五段236巷15號	501	169	340	615	94	154	98	228	3
389	台北市內湖區瑞光路478巷24號	台北市中山區長安東路一段16號	491	95	341	616	52	112	117	273	2
2320	台北市內湖區瑞光路478巷24號	台北市士林區中山北路七段141巷25號	491	268	342	617	94	154	107	249	5
381	台北市信義區信義路四段298號1F	台北市大安區忠孝東路三段217巷1弄6號	490	37	343	618	52	112	57	133	1
335	台北市大安區四維路208巷14號	台北市內湖區東湖路106巷7弄3號	749	237	344	619	52	112	1068	2492	15
321	台北市中山區中山北路一段140巷23號	台北市松山區延壽街356巷4弄2號	429	148	345	620	52	112	408	952	8
2276	台北市中山區中山北路一段140巷23號	台北市萬華區華西街	429	306	346	621	94	154	101	235	3
2451	台北市中山區中山北路一段140巷23號	台北市大同區雙連街43號	429	14	347	622	115	175	447	1043	15
358	台北市大安區師大路114巷5號	台北市文山區木柵路一段17巷1號	463	345	348	623	52	112	117	273	2
2335	台北市大安區師大路114巷5號	台北市信義區松廬路15號	463	182	349	624	94	154	51	119	2

距離矩陣 (由於資料龐大距離矩陣僅節錄前 30 個節點組成之矩陣-30*30)

999	9	15	8	5	13	9	7	4	4	3	1	5	3	4	3	9	5	6	0	13	7	4	3	4	12	2	3	17	2
9	999	12	3	4	12	9	11	11	12	8	11	13	6	9	7	0	14	14	9	11	11	10	6	8	18	8	12	14	8
15	12	999	15	14	2	6	9	11	14	17	15	12	14	11	15	12	16	16	15	2	10	11	14	18	27	16	17	2	16
8	3	15	999	5	14	12	11	11	12	7	10	12	6	9	6	3	13	13	9	14	11	10	6	7	16	7	11	16	7
5	4	14	5	999	13	9	7	7	8	5	7	9	2	5	3	5	10	10	5	13	7	6	2	5	15	4	8	16	4
13	12	2	14	13	999	4	7	10	12	16	13	11	12	10	13	11	14	15	13	1	8	9	13	17	25	15	15	4	15
9	9	6	12	9	4	999	3	6	8	12	9	6	8	6	9	9	11	11	9	4	4	6	9	13	21	11	11	8	11
7	11	9	11	7	7	3	999	3	6	9	7	3	7	4	7	10	8	8	7	8	1	3	7	11	19	8	9	11	8
4	11	11	11	7	10	6	3	999	3	7	4	2	5	3	5	11	5	5	4	10	2	1	5	8	16	6	6	13	6
4	12	14	12	8	12	8	6	3	999	5	3	3	6	4	6	13	2	2	4	12	6	3	6	6	14	5	4	16	5
3	8	17	7	5	16	12	9	7	5	999	3	7	4	6	3	8	6	6	3	16	9	6	3	2	10	1	3	19	1
1	11	15	10	7	13	9	7	4	3	3	999	5	5	4	4	11	4	4	1	13	7	4	5	4	12	3	2	17	3
5	13	12	12	9	11	6	3	2	3	7	5	999	7	4	7	13	5	5	5	11	3	3	7	9	17	6	7	14	6
3	6	14	6	2	12	8	7	5	6	4	5	7	999	3	1	6	8	9	3	12	6	5	0	4	13	2	7	16	2
4	9	11	9	5	10	6	4	3	4	6	4	4	3	999	4	9	6	6	3	10	3	2	4	7	15	5	6	13	5
3	7	15	6	3	13	9	7	5	6	3	4	7	1	4	999	7	8	8	3	13	7	5	0	4	13	2	6	16	2
9	0	12	3	5	11	9	10	11	13	8	11	13	6	9	7	999	14	14	9	11	11	11	6	7	18	8	12	13	8
5	14	16	13	10	14	11	8	5	2	6	4	5	8	6	8	14	999	0	5	15	7	5	8	7	12	7	3	18	7
6	14	16	13	10	15	11	8	5	2	6	4	5	9	6	8	14	0	999	6	15	7	5	9	8	12	7	3	18	7
0	9	15	9	5	13	9	7	4	4	3	1	5	3	3	3	9	5	6	999	13	6	4	3	4	12	2	3	16	2
13	11	2	14	13	1	4	8	10	12	16	13	11	12	10	13	11	15	15	13	999	8	10	13	17	25	15	15	4	15
7	11	10	11	7	8	4	1	2	6	9	7	3	6	3	7	11	7	7	6	8	999	3	7	10	18	8	9	12	8
4	10	11	10	6	9	6	3	1	3	6	4	3	5	2	5	11	5	5	4	10	3	999	4	8	16	5	6	13	5
3	6	14	6	2	13	9	7	5	6	3	5	7	0	4	0	6	8	9	3	13	7	4	999	4	13	2	6	16	2
4	8	18	7	5	17	13	11	8	6	2	4	9	4	7	4	7	7	8	4	17	10	8	4	999	10	2	5	20	2
12	18	27	16	15	25	21	19	16	14	10	12	17	13	15	13	18	12	12	12	25	18	16	13	10	999	11	10	28	11
2	8	16	7	4	15	11	8	6	5	1	3	6	2	5	2	8	7	7	2	15	8	5	2	2	11	999	4	18	0
3	12	17	11	8	15	11	9	6	4	3	2	7	7	6	6	12	3	3	3	15	9	6	6	5	10	4	999	19	4
17	14	2	16	16	4	8	11	13	16	19	17	14	16	13	16	13	18	18	16	4	12	13	16	20	28	18	19	999	18
2	8	16	7	4	15	11	8	6	5	1	3	6	2	5	2	8	7	7	2	15	8	5	2	2	11	0	4	18	999

單位：分鐘

■ 實務例題結果 (題組 1)

餐廳生產排程 (由於資料龐大餐廳生產排程結果僅節錄前 5 家餐廳)

餐廳 1

time_1: 0.004					
seq_					
machine1:	8	1	6	5	7
machine2:	9	2	4	7	
machine3:	10	3	7		
machine4:	11	7			
machine5:	12	7			
machine6:	13	7			
machine7:	14	7			
machine8:	15	7			
s_time					
machine1:	0	62	68	74	0
machine2:	0	65	75	0	
machine3:	0	71	0		
machine4:	0	0			
machine5:	0	0			
machine6:	0	0			
machine7:	0	0			
machine8:	0	0			

餐廳 6

[illegible]

餐廳 7

time 1: 0.428										
seq_										
machine1:	42	1	35	33	34	19	39	38	37	41
machine2:	43	2	20	40	36	41				
machine3:	44	3	21	41						
machine4:	45	4	22	41						
machine5:	46	5	23	41						
machine6:	47	6	24	41						
machine7:	48	7	25	41						
machine8:	49	8	26	41						
machine9:	50	9	27	41						
machine10:	51	10	28	41						
machine11:	52	11	29	41						
machine12:	53	12	30	41						
machine13:	54	13	31	41						
machine14:	55	14	32	41						
machine15:	56	15	41							
machine16:	57	16	41							
machine17:	58	17	41							
machine18:	59	18	41							
s time										
machine1:	0	13	28	31	37	58	73	83	89	0
machine2:	0	13	58	73	88	0				
machine3:	0	25	60	0						
machine4:	0	26	78	0						
machine5:	0	29	78	0						
machine6:	0	29	78	0						
machine7:	0	29	78	0						
machine8:	0	36	82	0						
machine9:	0	36	82	0						
machine10:	0	43	82	0						
machine11:	0	43	82	0						
machine12:	0	43	82	0						
machine13:	0	47	82	0						
machine14:	0	52	83	0						
machine15:	0	52	0							
machine16:	0	52	0							
machine17:	0	52	0							
machine18:	0	58	0							

餐廳 8

time_1: 1.125									
seq_									
machine1:	92	1	55	51	60	19	84	91	
machine2:	93	2	59	88	61	20	91		
machine3:	94	3	58	54	53	63	21	91	
machine4:	95	4	56	90	87	62	64	22	91
machine5:	96	5	57	85	23	91			
machine6:	97	6	86	83	24	91			
machine7:	98	7	89	25	91				
machine8:	99	52	8	26	91				
machine9:	100	9	27	91					
machine10:	101	10	28	91					
machine11:	102	11	29	91					
machine12:	103	12	30	91					
machine13:	104	13	31	91					
machine14:	105	14	32	91					
machine15:	106	15	91						
machine16:	107	16	91						
machine17:	108	17	91						
machine18:	109	18	91						
s_time									
machine1:	0	39	45	55	65	82	85	0	
machine2:	0	39	54	57	72	82	0		
machine3:	0	39	45	48	51	66	83	0	
machine4:	0	49	55	58	64	70	73	83	0
machine5:	0	49	55	65	83	0			
machine6:	0	49	64	79	83	0			
machine7:	0	52	67	90	0				
machine8:	0	48	52	90	0				
machine9:	0	61	90	0					
machine10:	0	61	91	0					
machine11:	0	61	91	0					
machine12:	0	61	91	0					
machine13:	0	61	91	0					
machine14:	0	61	93	0					
machine15:	0	62	0						
machine16:	0	62	0						
machine17:	0	75	0						
machine18:	0	75	0						

餐廳 9

time_1: 1.15									
seq_									
machine1:	18	1	11	7	10	12	14	16	15
machine2:	19	2	9	8	13	17			17
machine3:	20	3	17						
machine4:	21	4	17						
machine5:	22	5	17						
machine6:	23	6	17						
machine7:	24	17							
machine8:	25	17							
s_time									
machine1:	0	38	44	47	53	63	69	75	81
machine2:	0	45	51	54	64	0			
machine3:	0	60	0						
machine4:	0	82	0						
machine5:	0	82	0						
machine6:	0	84	0						
machine7:	0	0							
machine8:	0	0							

外送員配送路線 (由於資料龐大外送員配送路線結果僅節錄前 5 位外送員)

外送員 0

deliver 0						
Take 0 go vertex 0 and wait for 0(TW:25 in 25/999, mshift:2, CP:0/50)						
Take 7 go vertex 512 and wait for 0(TW:32 in 25/999, mshift:2, CP:9/50)						
Take 3 go vertex 536 and wait for 0(TW:35 in 25/999, mshift:2, CP:22/50)						
Take 1 go vertex 415 and wait for 0(TW:36 in 25/999, mshift:2, CP:37/50)						
Take 2 go vertex 809 and wait for 2(TW:40 in 40/100, mshift:0, CP:28/50)						
Take 0 go vertex 712 and wait for 0(TW:40 in 40/100, mshift:0, CP:13/50)						
Take 8 go vertex 357 and wait for 0(TW:48 in 25/999, mshift:0, CP:25/50)						
Take 5 go vertex 575 and wait for 0(TW:53 in 53/53, mshift:0, CP:32/50)						
Take 3 go vertex 654 and wait for 0(TW:56 in 21/81, mshift:0, CP:20/50)						
Take 3 go vertex 833 and wait for 0(TW:59 in 40/100, mshift:0, CP:7/50)						
Take 1 go vertex 872 and wait for 0(TW:60 in 53/63, mshift:0, CP:0/50)						
Take 5 go vertex 366 and wait for 0(TW:65 in 60/70, mshift:0, CP:1/50)						
Take 0 go vertex 371 and wait for 0(TW:65 in 65/65, mshift:0, CP:19/50)						
Take 2 go vertex 663 and wait for 0(TW:67 in 60/80, mshift:1, CP:18/50)						
Take 2 go vertex 668 and wait for 0(TW:69 in 65/75, mshift:1, CP:0/50)						
Take 5 go vertex 338 and wait for 1(TW:75 in 75/76, mshift:0, CP:15/50)						
Take 6 go vertex 635 and wait for 0(TW:81 in 75/81, mshift:0, CP:0/50)						
Take 11 go vertex 320 and wait for 0(TW:92 in 92/92, mshift:0, CP:9/50)						
Take 6 go vertex 617 and wait for 0(TW:98 in 92/100, mshift:2, CP:0/50)						
Take 5 go vertex 304 and wait for 17(TW:120 in 120/130, mshift:0, CP:6/50)						
Take 5 go vertex 303 and wait for 0(TW:125 in 120/125, mshift:0, CP:13/50)						
Take 3 go vertex 601 and wait for 12(TW:140 in 140/150, mshift:0, CP:7/50)						
Take 6 go vertex 600 and wait for 0(TW:146 in 140/150, mshift:0, CP:0/50)						
Take 8 go vertex 301 and wait for 0(TW:154 in 150/155, mshift:0, CP:5/50)						
Take 4 go vertex 300 and wait for 0(TW:158 in 150/160, mshift:0, CP:6/50)						
Take 12 go vertex 598 and wait for 0(TW:170 in 170/180, mshift:0, CP:1/50)						
Take 10 go vertex 597 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 1

deliver 1						
Take 0 go vertex 1 and wait for 0(TW:25 in 25/999, mshift:11, CP:0/50)						
Take 1 go vertex 562 and wait for 11(TW:37 in 37/38, mshift:0, CP:5/50)						
Take 5 go vertex 859 and wait for 0(TW:42 in 37/53, mshift:0, CP:0/50)						
Take 6 go vertex 478 and wait for 0(TW:48 in 45/55, mshift:0, CP:1/50)						
Take 2 go vertex 534 and wait for 0(TW:50 in 50/50, mshift:0, CP:4/50)						
Take 4 go vertex 775 and wait for 0(TW:54 in 45/70, mshift:0, CP:3/50)						
Take 1 go vertex 442 and wait for 0(TW:55 in 55/58, mshift:0, CP:12/50)						
Take 4 go vertex 831 and wait for 0(TW:59 in 50/60, mshift:0, CP:9/50)						
Take 3 go vertex 739 and wait for 0(TW:62 in 55/68, mshift:0, CP:0/50)						
Take 10 go vertex 592 and wait for 0(TW:72 in 70/80, mshift:0, CP:1/50)						
Take 6 go vertex 457 and wait for 0(TW:78 in 78/78, mshift:0, CP:8/50)						
Take 3 go vertex 754 and wait for 0(TW:81 in 78/81, mshift:0, CP:1/50)						
Take 6 go vertex 889 and wait for 0(TW:87 in 70/89, mshift:0, CP:0/50)						
Take 2 go vertex 429 and wait for 0(TW:89 in 89/89, mshift:0, CP:7/50)						
Take 2 go vertex 726 and wait for 0(TW:91 in 89/99, mshift:0, CP:0/50)						
Take 4 go vertex 423 and wait for 0(TW:95 in 94/95, mshift:0, CP:5/50)						
Take 5 go vertex 720 and wait for 0(TW:100 in 94/100, mshift:0, CP:0/50)						
Take 10 go vertex 307 and wait for 10(TW:120 in 120/125, mshift:5, CP:6/50)						
Take 5 go vertex 604 and wait for 15(TW:140 in 140/150, mshift:9, CP:0/50)						
Take 3 go vertex 306 and wait for 7(TW:150 in 150/155, mshift:2, CP:2/50)						
Take 3 go vertex 309 and wait for 0(TW:153 in 150/155, mshift:2, CP:17/50)						
Take 1 go vertex 305 and wait for 0(TW:154 in 150/160, mshift:2, CP:21/50)						
Take 4 go vertex 302 and wait for 0(TW:158 in 150/160, mshift:2, CP:22/50)						
Take 3 go vertex 606 and wait for 9(TW:170 in 170/180, mshift:0, CP:7/50)						
Take 4 go vertex 603 and wait for 0(TW:174 in 170/180, mshift:0, CP:5/50)						
Take 2 go vertex 602 and wait for 0(TW:176 in 170/180, mshift:0, CP:1/50)						
Take 4 go vertex 599 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 2

deliver 2						
Take 0 go vertex 2 and wait for 0(TW:25 in 25/999, mshift:10, CP:0/50)						
Take 3 go vertex 322 and wait for 9(TW:37 in 37/38, mshift:1, CP:3/50)						
Take 1 go vertex 416 and wait for 3(TW:41 in 41/41, mshift:0, CP:8/50)						
Take 2 go vertex 619 and wait for 0(TW:43 in 37/53, mshift:0, CP:5/50)						
Take 1 go vertex 713 and wait for 0(TW:44 in 41/51, mshift:0, CP:0/50)						
Take 1 go vertex 341 and wait for 0(TW:45 in 45/45, mshift:0, CP:6/50)						
Take 8 go vertex 638 and wait for 0(TW:53 in 45/55, mshift:1, CP:0/50)						
Take 10 go vertex 433 and wait for 0(TW:63 in 59/66, mshift:1, CP:2/50)						
Take 4 go vertex 730 and wait for 0(TW:67 in 59/76, mshift:1, CP:0/50)						
Take 5 go vertex 547 and wait for 1(TW:73 in 73/73, mshift:0, CP:10/50)						
Take 7 go vertex 844 and wait for 0(TW:80 in 73/81, mshift:0, CP:0/50)						
Take 2 go vertex 473 and wait for 0(TW:82 in 80/87, mshift:0, CP:2/50)						
Take 4 go vertex 388 and wait for 0(TW:86 in 84/88, mshift:0, CP:8/50)						
Take 4 go vertex 770 and wait for 0(TW:90 in 80/100, mshift:0, CP:6/50)						
Take 9 go vertex 685 and wait for 0(TW:99 in 84/99, mshift:0, CP:0/50)						
Take 8 go vertex 310 and wait for 13(TW:120 in 120/125, mshift:5, CP:8/50)						
Take 6 go vertex 607 and wait for 14(TW:140 in 140/150, mshift:3, CP:0/50)						
Take 7 go vertex 314 and wait for 3(TW:150 in 150/155, mshift:0, CP:4/50)						
Take 5 go vertex 312 and wait for 0(TW:155 in 150/155, mshift:0, CP:9/50)						
Take 5 go vertex 308 and wait for 0(TW:160 in 150/160, mshift:0, CP:10/50)						
Take 7 go vertex 611 and wait for 3(TW:170 in 170/180, mshift:0, CP:6/50)						
Take 10 go vertex 609 and wait for 0(TW:180 in 170/180, mshift:0, CP:1/50)						
Take 0 go vertex 605 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 3

deliver 3						
Take 0 go vertex 3 and wait for 0(TW:25 in 25/999, mshift:5, CP:0/50)						
Take 7 go vertex 509 and wait for 2(TW:34 in 34/41, mshift:3, CP:2/50)						
Take 1 go vertex 513 and wait for 2(TW:37 in 37/38, mshift:1, CP:6/50)						
Take 2 go vertex 452 and wait for 1(TW:40 in 40/40, mshift:0, CP:11/50)						
Take 3 go vertex 403 and wait for 1(TW:44 in 44/45, mshift:1, CP:15/50)						
Take 1 go vertex 700 and wait for 0(TW:45 in 44/55, mshift:1, CP:11/50)						
Take 3 go vertex 806 and wait for 0(TW:48 in 34/51, mshift:1, CP:9/50)						
Take 0 go vertex 810 and wait for 0(TW:48 in 37/51, mshift:1, CP:5/50)						
Take 2 go vertex 749 and wait for 0(TW:50 in 40/51, mshift:1, CP:0/50)						
Take 1 go vertex 362 and wait for 2(TW:53 in 53/53, mshift:0, CP:5/50)						
Take 6 go vertex 659 and wait for 0(TW:59 in 53/63, mshift:2, CP:0/50)						
Take 6 go vertex 364 and wait for 1(TW:66 in 66/70, mshift:1, CP:2/50)						
Take 3 go vertex 431 and wait for 0(TW:69 in 66/70, mshift:1, CP:6/50)						
Take 2 go vertex 661 and wait for 0(TW:71 in 66/80, mshift:1, CP:4/50)						
Take 3 go vertex 728 and wait for 0(TW:74 in 66/80, mshift:1, CP:0/50)						
Take 5 go vertex 321 and wait for 1(TW:80 in 80/80, mshift:0, CP:8/50)						
Take 1 go vertex 378 and wait for 1(TW:82 in 82/92, mshift:0, CP:10/50)						
Take 8 go vertex 618 and wait for 0(TW:90 in 80/90, mshift:0, CP:2/50)						
Take 7 go vertex 675 and wait for 0(TW:97 in 82/100, mshift:3, CP:0/50)						
Take 3 go vertex 315 and wait for 20(TW:120 in 120/125, mshift:5, CP:7/50)						
Take 6 go vertex 612 and wait for 14(TW:140 in 140/150, mshift:6, CP:0/50)						
Take 6 go vertex 316 and wait for 4(TW:150 in 150/160, mshift:2, CP:3/50)						
Take 3 go vertex 311 and wait for 0(TW:153 in 150/155, mshift:2, CP:6/50)						
Take 5 go vertex 613 and wait for 12(TW:170 in 170/180, mshift:2, CP:3/50)						
Take 8 go vertex 608 and wait for 0(TW:178 in 170/180, mshift:2, CP:0/50)						

外送員 4

deliver 4						
Take 0 go vertex 4 and wait for 0(TW:25 in 25/999, mshift:7, CP:0/50)						
Take 4 go vertex 504 and wait for 6(TW:35 in 35/38, mshift:1, CP:3/50)						
Take 3 go vertex 801 and wait for 0(TW:38 in 35/51, mshift:1, CP:0/50)						
Take 2 go vertex 374 and wait for 0(TW:40 in 40/41, mshift:1, CP:3/50)						
Take 5 go vertex 671 and wait for 0(TW:45 in 40/51, mshift:1, CP:0/50)						
Take 6 go vertex 463 and wait for 1(TW:52 in 52/53, mshift:0, CP:4/50)						
Take 1 go vertex 453 and wait for 0(TW:53 in 53/53, mshift:0, CP:12/50)						
Take 2 go vertex 760 and wait for 0(TW:55 in 52/63, mshift:0, CP:8/50)						
Take 1 go vertex 750 and wait for 0(TW:56 in 53/63, mshift:0, CP:0/50)						
Take 1 go vertex 377 and wait for 0(TW:57 in 57/58, mshift:0, CP:9/50)						
Take 2 go vertex 456 and wait for 0(TW:59 in 59/69, mshift:0, CP:10/50)						
Take 3 go vertex 674 and wait for 0(TW:62 in 57/68, mshift:0, CP:1/50)						
Take 4 go vertex 441 and wait for 0(TW:66 in 63/67, mshift:0, CP:4/50)						
Take 5 go vertex 738 and wait for 0(TW:71 in 63/77, mshift:0, CP:1/50)						
Take 1 go vertex 427 and wait for 0(TW:72 in 72/73, mshift:0, CP:8/50)						
Take 6 go vertex 753 and wait for 0(TW:78 in 59/79, mshift:0, CP:7/50)						
Take 3 go vertex 724 and wait for 0(TW:81 in 72/81, mshift:0, CP:0/50)						
Take 5 go vertex 555 and wait for 0(TW:86 in 86/91, mshift:1, CP:2/50)						
Take 6 go vertex 523 and wait for 0(TW:92 in 92/97, mshift:1, CP:6/50)						
Take 3 go vertex 852 and wait for 0(TW:95 in 86/100, mshift:1, CP:4/50)						
Take 4 go vertex 820 and wait for 0(TW:99 in 92/100, mshift:1, CP:0/50)						
Take 8 go vertex 313 and wait for 43(TW:150 in 150/160, mshift:10, CP:1/50)						
Take 10 go vertex 610 and wait for 10(TW:170 in 170/180, mshift:10, CP:0/50)						

已被接受但尚未取餐或尚未送達之訂單 (即必須納入下一規劃時段考慮)

	舊編號		餐廳	顧客	對應車輛	容量	新編號		取餐上界	取餐下界	送達上界	送達下界
3115	301	598	420	282	0	5	952	1001	154	155	160	170
3110	303	600	402	10	0	7	953	1002	125	125	136	146
3117	305	602	382	35	1	4	955	1004	154	160	166	176
3114	315	612	577	22	3	7	959	1008	120	125	130	140
29	335	632	448	20	15	3	961	1010	64	65	56	66
82	337	634	600	272	28	2	962	1011	46	53	52	62
36	350	647	405	329	24	2	964	1013	57	64	64	74
51	359	656	631	212	12	2	965	1014	71	71	65	75
70	361	658	508	217	21	3	967	1016	50	50	48	58
120	389	686	472	300	20	3	969	1018	72	73	71	81
24	412	709	584	212	12	4	973	1022	64	65	65	75
121	418	715	400	258	18	3	974	1023	70	74	70	80
6	432	729	482	188	14	2	976	1025	78	84	71	81
45	435	732	619	286	31	1	977	1026	60	65	58	68
84	441	738	433	17	4	3	979	1028	66	67	61	71
14	444	741	453	309	15	3	980	1029	73	74	67	77
63	457	754	382	81	1	7	982	1031	78	78	71	81
13	477	774	374	341	22	3	984	1033	61	65	59	69
110	485	782	418	291	8	6	986	1035	71	71	70	80
46	491	788	478	170	7	1	987	1036	63	67	62	72
62	493	790	632	97	13	5	989	1038	60	60	59	69
27	581	878	459	39	28	2	997	1046	74	74	70	80
7	346	643		265	28	1		1051		52	50	60
116	386	683		113	6	4		1055		41	37	47
18	452	749		133	3	5		1060		40	40	50
21	513	810		39	3	4		1066		38	38	48
86	533	830		206	17	2		1067		45	45	55
73	548	845		188	21	2		1068		45	38	48

■ 實務例題結果 (題組 2)

餐廳生產排程 (由於資料龐大餐廳生產排程結果僅節錄前 5 家餐廳)

餐廳 4

time 1: 0.112												
seq_												
machine1:	25	1	8	14	13	11	19	22	23	16	17	24
machine2:	26	2	12	10	21	24						
machine3:	27	3	9	20	24							
machine4:	28	4	15	18	24							
machine5:	29	5	24									
machine6:	30	6	24									
machine7:	31	7	24									
machine8:	32	24										
machine9:	33	24										
machine10:	34	24										
machine11:	35	24										
machine12:	36	24										
s_time												
machine1:	0	37	47	62	72	78	112	115	121	136	139	0
machine2:	0	53	63	78	112	0						
machine3:	0	53	68	112	0							
machine4:	0	61	64	112	0							
machine5:	0	86	0									
machine6:	0	86	0									
machine7:	0	106	0									
machine8:	0	0										
machine9:	0	0										
machine10:	0	0										
machine11:	0	0										
machine12:	0	0										

餐廳 6

time 1: 0.662													
seq													
machine1:	78	1	63	19	66	56	55	37	67	46	47	48	77
machine2:	79	2	20	53	38	75	76	72	77				
machine3:	80	3	21	51	39	73	71	77					
machine4:	81	4	22	52	40	70	77						
machine5:	82	5	23	49	41	77							
machine6:	83	6	65	24	42	77							
machine7:	84	7	62	25	68	43	77						
machine8:	85	8	64	26	50	54	57	74	44	77			
machine9:	86	9	61	27	58	69	45	77					
machine10:	87	10	28	77									
machine11:	88	11	59	29	77								
machine12:	89	12	60	30	77								
machine13:	90	13	31	77									
machine14:	91	14	32	77									
machine15:	92	15	33	77									
machine16:	93	16	34	77									
machine17:	94	17	35	77									
machine18:	95	18	36	77									
s time													
machine1:	0	19	60	66	69	84	90	99	109	124	130	150	0
machine2:	0	19	67	82	99	102	105	111	0				
machine3:	0	28	67	82	102	108	111	0					
machine4:	0	29	69	84	107	110	0						
machine5:	0	29	69	84	107	0							
machine6:	0	34	60	69	107	0							
machine7:	0	34	60	69	112	127	0						
machine8:	0	34	60	72	82	85	88	112	127	0			
machine9:	0	44	60	77	80	112	127	0					
machine10:	0	47	77	0									
machine11:	0	60	70	77	0								
machine12:	0	60	70	87	0								
machine13:	0	60	94	0									
machine14:	0	60	94	0									
machine15:	0	60	94	0									
machine16:	0	60	94	0									
machine17:	0	65	99	0									
machine18:	0	65	99	0									

餐廳 8

time_1: 0.771							
seq_							
machine1:	59	1	56	57	19	37	58
machine2:	60	2	20	38	58		
machine3:	61	3	21	39	58		
machine4:	62	4	22	40	58		
machine5:	63	5	23	41	58		
machine6:	64	6	24	42	58		
machine7:	65	7	25	58			
machine8:	66	8	26	58			
machine9:	67	9	27	58			
machine10:	68	10	28	58			
machine11:	69	11	29	58			
machine12:	70	12	30	58			
machine13:	71	13	31	58			
machine14:	72	14	32	58			
machine15:	73	15	33	58			
machine16:	74	16	34	58			
machine17:	75	17	35	58			
machine18:	76	18	36	58			
s_time							
machine1:	0	49	64	67	90	113	0
machine2:	0	61	90	113	0		
machine3:	0	61	91	113	0		
machine4:	0	61	91	113	0		
machine5:	0	61	91	113	0		
machine6:	0	61	91	114	0		
machine7:	0	61	93	0			
machine8:	0	62	93	0			
machine9:	0	62	103	0			
machine10:	0	75	103	0			
machine11:	0	75	106	0			
machine12:	0	82	106	0			
machine13:	0	82	106	0			
machine14:	0	83	106	0			
machine15:	0	83	106	0			
machine16:	0	83	108	0			
machine17:	0	83	108	0			
machine18:	0	90	108	0			

餐廳 15

time 1: 1.301													
seq_													
machine1:	83	1	57	77	78	59	19	37	55	79	81	82	
machine2:	84	2	58	20	38	52	56	80	82				
machine3:	85	3	75	21	39	82							
machine4:	86	4	72	22	40	82							
machine5:	87	5	23	41	82								
machine6:	88	6	24	42	82								
machine7:	89	7	25	31	43	82							
machine8:	90	8	26	44	82								
machine9:	91	73	9	27	45	82							
machine10:	92	74	10	28	46	82							
machine11:	93	11	29	47	82								
machine12:	94	12	30	48	82								
machine13:	95	76	13	49	82								
machine14:	96	14	32	50	82								
machine15:	97	15	33	51	82								
machine16:	98	16	34	82									
machine17:	99	17	35	53	82								
machine18:	100	18	36	54	82								
s time													
machine1:	0	36	56	59	62	65	74	100	115	125	131	0	
machine2:	0	41	56	74	100	104	119	134	0				
machine3:	0	41	60	74	100	0							
machine4:	0	41	60	74	100	0							
machine5:	0	48	74	103	0								
machine6:	0	48	74	103	0								
machine7:	0	53	74	89	103	0							
machine8:	0	74	77	103	0								
machine9:	0	60	74	84	103	0							
machine10:	0	60	74	89	103	0							
machine11:	0	74	89	104	0								
machine12:	0	74	89	104	0								
machine13:	0	60	74	104	0								
machine14:	0	74	89	104	0								
machine15:	0	74	89	104	0								
machine16:	0	74	89	0									
machine17:	0	74	89	104	0								
machine18:	0	74	89	105	0								

餐廳 17

time_1: 1.778											
seq_											
machine1:	89	1	75	19	37	55	83	87	85	84	88
machine2:	90	2	20	38	56	82	88				
machine3:	91	3	21	39	57	88					
machine4:	92	4	22	40	58	88					
machine5:	93	5	78	23	41	59	88				
machine6:	94	6	77	24	42	60	88				
machine7:	95	7	76	25	43	61	88				
machine8:	96	8	26	44	62	88					
machine9:	97	9	27	45	63	88					
machine10:	98	10	28	46	64	88					
machine11:	99	11	29	47	65	88					
machine12:	100	12	30	48	66	88					
machine13:	101	13	31	49	67	86	88				
machine14:	102	14	32	50	68	88					
machine15:	103	15	33	51	69	88					
machine16:	104	16	34	52	88						
machine17:	105	17	35	53	88						
machine18:	106	18	36	54	88						
s_time											
machine1:	0	26	55	64	79	94	112	115	118	121	0
machine2:	0	29	64	79	94	112	0				
machine3:	0	38	64	79	94	0					
machine4:	0	38	64	79	94	0					
machine5:	0	47	53	64	79	94	0				
machine6:	0	47	62	65	79	94	0				
machine7:	0	47	62	65	79	94	0				
machine8:	0	47	66	79	95	0					
machine9:	0	47	66	81	95	0					
machine10:	0	47	66	81	96	0					
machine11:	0	48	66	81	96	0					
machine12:	0	48	66	81	96	0					
machine13:	0	48	66	81	108	111	0				
machine14:	0	49	66	81	108	0					
machine15:	0	49	66	81	108	0					
machine16:	0	64	67	81	0						
machine17:	0	64	79	94	0						
machine18:	0	64	79	94	0						

外送員配送路線 (由於資料龐大外送員配送路線結果僅節錄前 5 位外送員)

外送員 0

deliver 0						
Take 0 go vertex 0 and wait for 0(TW:48 in 48/999, mshift:0, CP:12/50)						
Take 1 go vertex 306 and wait for 0(TW:49 in 0/999, mshift:0, CP:13/50)						
Take 4 go vertex 301 and wait for 0(TW:53 in 0/999, mshift:0, CP:14/50)						
Take 0 go vertex 300 and wait for 0(TW:53 in 0/999, mshift:0, CP:16/50)						
Take 3 go vertex 1052 and wait for 0(TW:56 in 46/56, mshift:0, CP:4/50)						
Take 6 go vertex 311 and wait for 0(TW:62 in 0/999, mshift:0, CP:9/50)						
Take 1 go vertex 541 and wait for 0(TW:63 in 0/999, mshift:0, CP:14/50)						
Take 1 go vertex 961 and wait for 0(TW:64 in 64/65, mshift:0, CP:17/50)						
Take 2 go vertex 1010 and wait for 0(TW:66 in 56/66, mshift:0, CP:14/50)						
Take 2 go vertex 867 and wait for 0(TW:68 in 40/100, mshift:1, CP:9/50)						
Take 4 go vertex 625 and wait for 0(TW:72 in 67/77, mshift:1, CP:10/50)						
Take 2 go vertex 960 and wait for 1(TW:75 in 75/78, mshift:0, CP:13/50)						
Take 6 go vertex 1009 and wait for 0(TW:81 in 71/81, mshift:0, CP:10/50)						
Take 2 go vertex 520 and wait for 0(TW:83 in 82/83, mshift:0, CP:16/50)						
Take 0 go vertex 400 and wait for 1(TW:84 in 84/86, mshift:0, CP:22/50)						
Take 3 go vertex 951 and wait for 0(TW:87 in 67/92, mshift:0, CP:21/50)						
Take 1 go vertex 846 and wait for 0(TW:88 in 82/93, mshift:0, CP:15/50)						
Take 1 go vertex 380 and wait for 0(TW:89 in 89/89, mshift:0, CP:25/50)						
Take 2 go vertex 726 and wait for 0(TW:91 in 84/96, mshift:2, CP:19/50)						
Take 3 go vertex 706 and wait for 0(TW:94 in 89/96, mshift:2, CP:9/50)						
Take 5 go vertex 347 and wait for 0(TW:99 in 0/999, mshift:2, CP:27/50)						
Take 2 go vertex 673 and wait for 0(TW:101 in 94/154, mshift:2, CP:9/50)						
Take 10 go vertex 309 and wait for 0(TW:111 in 0/999, mshift:2, CP:13/50)						
Take 7 go vertex 954 and wait for 2(TW:120 in 120/130, mshift:0, CP:19/50)						
Take 5 go vertex 953 and wait for 0(TW:125 in 125/125, mshift:0, CP:26/50)						
Take 4 go vertex 565 and wait for 0(TW:129 in 126/130, mshift:0, CP:28/50)						
Take 2 go vertex 891 and wait for 0(TW:131 in 126/140, mshift:0, CP:26/50)						
Take 1 go vertex 1003 and wait for 0(TW:132 in 130/140, mshift:0, CP:20/50)						
Take 9 go vertex 626 and wait for 0(TW:141 in 140/150, mshift:0, CP:18/50)						
Take 5 go vertex 1002 and wait for 0(TW:146 in 136/146, mshift:0, CP:11/50)						
Take 8 go vertex 952 and wait for 0(TW:154 in 154/155, mshift:0, CP:16/50)						
Take 14 go vertex 1001 and wait for 0(TW:168 in 160/170, mshift:0, CP:11/50)						
Take 6 go vertex 637 and wait for 0(TW:174 in 170/180, mshift:0, CP:6/50)						
Take 0 go vertex 632 and wait for 0(TW:174 in 170/180, mshift:0, CP:5/50)						
Take 3 go vertex 635 and wait for 0(TW:177 in 170/180, mshift:0, CP:1/50)						
Take 3 go vertex 627 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 1

deliver 1						
Take 0 go vertex 1 and wait for 0(TW:42 in 42/999, mshift:1, CP:0/50)						
Take 7 go vertex 967 and wait for 1(TW:50 in 50/50, mshift:0, CP:3/50)						
Take 1 go vertex 415 and wait for 0(TW:51 in 0/999, mshift:0, CP:8/50)						
Take 7 go vertex 1016 and wait for 0(TW:58 in 48/58, mshift:0, CP:5/50)						
Take 1 go vertex 741 and wait for 0(TW:59 in 36/96, mshift:4, CP:0/50)						
Take 0 go vertex 987 and wait for 4(TW:63 in 63/67, mshift:0, CP:1/50)						
Take 7 go vertex 1036 and wait for 0(TW:70 in 62/72, mshift:0, CP:0/50)						
Take 5 go vertex 303 and wait for 0(TW:75 in 0/999, mshift:0, CP:1/50)						
Take 0 go vertex 963 and wait for 0(TW:75 in 75/76, mshift:0, CP:16/50)						
Take 6 go vertex 1012 and wait for 0(TW:81 in 71/81, mshift:0, CP:1/50)						
Take 5 go vertex 314 and wait for 0(TW:86 in 0/999, mshift:1, CP:4/50)						
Take 6 go vertex 383 and wait for 0(TW:92 in 87/94, mshift:1, CP:9/50)						
Take 1 go vertex 431 and wait for 0(TW:93 in 93/97, mshift:1, CP:11/50)						
Take 2 go vertex 709 and wait for 0(TW:95 in 87/96, mshift:1, CP:6/50)						
Take 2 go vertex 757 and wait for 0(TW:97 in 93/100, mshift:2, CP:4/50)						
Take 4 go vertex 307 and wait for 1(TW:102 in 0/999, mshift:1, CP:19/50)						
Take 3 go vertex 506 and wait for 1(TW:106 in 106/113, mshift:0, CP:21/50)						
Take 2 go vertex 832 and wait for 0(TW:108 in 106/123, mshift:0, CP:19/50)						
Take 1 go vertex 305 and wait for 0(TW:109 in 0/999, mshift:0, CP:21/50)						
Take 6 go vertex 304 and wait for 0(TW:115 in 0/999, mshift:0, CP:23/50)						
Take 5 go vertex 957 and wait for 0(TW:120 in 120/125, mshift:0, CP:31/50)						
Take 2 go vertex 956 and wait for 0(TW:122 in 120/125, mshift:0, CP:37/50)						
Take 8 go vertex 1006 and wait for 0(TW:130 in 130/140, mshift:0, CP:29/50)						
Take 10 go vertex 1005 and wait for 0(TW:140 in 130/140, mshift:0, CP:23/50)						
Take 5 go vertex 630 and wait for 0(TW:145 in 140/150, mshift:5, CP:21/50)						
Take 9 go vertex 955 and wait for 0(TW:154 in 154/160, mshift:6, CP:25/50)						
Take 3 go vertex 629 and wait for 13(TW:170 in 170/180, mshift:0, CP:24/50)						
Take 3 go vertex 631 and wait for 0(TW:173 in 170/180, mshift:0, CP:22/50)						
Take 2 go vertex 1004 and wait for 0(TW:175 in 166/176, mshift:0, CP:18/50)						
Take 5 go vertex 640 and wait for 0(TW:180 in 170/180, mshift:0, CP:15/50)						
Take 0 go vertex 633 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 2

deliver 2						
Take 0 go vertex 2 and wait for 0(TW:41 in 41/999, mshift:0, CP:8/50)						
Take 0 go vertex 302 and wait for 0(TW:41 in 0/999, mshift:0, CP:13/50)						
Take 2 go vertex 1050 and wait for 0(TW:43 in 33/43, mshift:0, CP:10/50)						
Take 1 go vertex 1059 and wait for 0(TW:44 in 34/44, mshift:0, CP:5/50)						
Take 3 go vertex 308 and wait for 0(TW:47 in 0/999, mshift:0, CP:11/50)						
Take 3 go vertex 962 and wait for 0(TW:50 in 46/53, mshift:0, CP:13/50)						
Take 12 go vertex 1011 and wait for 0(TW:62 in 52/62, mshift:0, CP:11/50)						
Take 8 go vertex 965 and wait for 1(TW:71 in 71/71, mshift:0, CP:13/50)						
Take 4 go vertex 1014 and wait for 0(TW:75 in 65/75, mshift:0, CP:11/50)						
Take 0 go vertex 985 and wait for 5(TW:80 in 80/86, mshift:0, CP:14/50)						
Take 1 go vertex 1034 and wait for 0(TW:81 in 71/81, mshift:0, CP:11/50)						
Take 8 go vertex 445 and wait for 0(TW:89 in 89/89, mshift:0, CP:21/50)						
Take 2 go vertex 449 and wait for 0(TW:91 in 89/92, mshift:0, CP:28/50)						
Take 2 go vertex 775 and wait for 0(TW:93 in 89/100, mshift:0, CP:21/50)						
Take 3 go vertex 771 and wait for 0(TW:96 in 89/96, mshift:0, CP:11/50)						
Take 3 go vertex 386 and wait for 1(TW:100 in 100/110, mshift:3, CP:12/50)						
Take 3 go vertex 712 and wait for 0(TW:103 in 100/125, mshift:3, CP:11/50)						
Take 4 go vertex 566 and wait for 3(TW:110 in 110/110, mshift:0, CP:16/50)						
Take 2 go vertex 892 and wait for 0(TW:112 in 110/125, mshift:2, CP:11/50)						
Take 3 go vertex 378 and wait for 0(TW:115 in 112/122, mshift:2, CP:14/50)						
Take 2 go vertex 313 and wait for 0(TW:117 in 0/999, mshift:2, CP:16/50)						
Take 4 go vertex 959 and wait for 0(TW:121 in 120/125, mshift:2, CP:23/50)						
Take 6 go vertex 704 and wait for 0(TW:127 in 112/132, mshift:2, CP:20/50)						
Take 1 go vertex 1008 and wait for 2(TW:130 in 130/140, mshift:0, CP:13/50)						
Take 13 go vertex 628 and wait for 0(TW:143 in 140/150, mshift:0, CP:8/50)						
Take 12 go vertex 958 and wait for 0(TW:155 in 150/160, mshift:0, CP:9/50)						
Take 10 go vertex 1007 and wait for 0(TW:165 in 160/170, mshift:0, CP:8/50)						
Take 7 go vertex 639 and wait for 0(TW:172 in 170/180, mshift:0, CP:6/50)						
Take 8 go vertex 634 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 3

deliver 3						
Take 0 go vertex 3 and wait for 0(TW:44 in 44/999, mshift:0, CP:15/50)						
Take 1 go vertex 1057 and wait for 0(TW:45 in 35/45, mshift:0, CP:11/50)						
Take 3 go vertex 1066 and wait for 0(TW:48 in 38/48, mshift:0, CP:7/50)						
Take 0 go vertex 1065 and wait for 0(TW:48 in 38/48, mshift:0, CP:5/50)						
Take 2 go vertex 1060 and wait for 0(TW:50 in 40/50, mshift:0, CP:0/50)						
Take 1 go vertex 968 and wait for 2(TW:53 in 53/53, mshift:0, CP:5/50)						
Take 6 go vertex 1017 and wait for 0(TW:59 in 49/59, mshift:0, CP:0/50)						
Take 6 go vertex 423 and wait for 1(TW:66 in 66/66, mshift:0, CP:4/50)						
Take 1 go vertex 749 and wait for 0(TW:67 in 66/76, mshift:2, CP:0/50)						
Take 3 go vertex 969 and wait for 2(TW:72 in 72/73, mshift:0, CP:3/50)						
Take 9 go vertex 1018 and wait for 0(TW:81 in 71/81, mshift:0, CP:0/50)						
Take 2 go vertex 322 and wait for 1(TW:84 in 84/94, mshift:0, CP:1/50)						
Take 1 go vertex 454 and wait for 0(TW:85 in 85/85, mshift:0, CP:4/50)						
Take 1 go vertex 648 and wait for 0(TW:86 in 84/96, mshift:1, CP:3/50)						
Take 1 go vertex 780 and wait for 0(TW:87 in 85/95, mshift:1, CP:0/50)						
Take 1 go vertex 602 and wait for 0(TW:88 in 88/90, mshift:1, CP:6/50)						
Take 1 go vertex 323 and wait for 1(TW:90 in 90/90, mshift:0, CP:15/50)						
Take 6 go vertex 649 and wait for 0(TW:96 in 90/100, mshift:0, CP:6/50)						
Take 0 go vertex 928 and wait for 0(TW:96 in 88/96, mshift:0, CP:0/50)						
Take 0 go vertex 365 and wait for 7(TW:103 in 103/110, mshift:4, CP:2/50)						
Take 7 go vertex 691 and wait for 0(TW:110 in 103/125, mshift:4, CP:0/50)						
Take 3 go vertex 432 and wait for 3(TW:116 in 116/117, mshift:1, CP:5/50)						
Take 5 go vertex 599 and wait for 1(TW:122 in 122/122, mshift:0, CP:20/50)						
Take 3 go vertex 462 and wait for 0(TW:125 in 125/126, mshift:0, CP:22/50)						
Take 2 go vertex 758 and wait for 0(TW:127 in 116/127, mshift:0, CP:17/50)						
Take 2 go vertex 925 and wait for 0(TW:129 in 122/132, mshift:2, CP:2/50)						
Take 4 go vertex 788 and wait for 0(TW:133 in 125/141, mshift:2, CP:0/50)						
Take 6 go vertex 612 and wait for 1(TW:140 in 140/141, mshift:1, CP:6/50)						
Take 5 go vertex 938 and wait for 0(TW:145 in 140/151, mshift:6, CP:0/50)						
Take 12 go vertex 312 and wait for 0(TW:157 in 0/999, mshift:9, CP:4/50)						
Take 1 go vertex 310 and wait for 0(TW:158 in 0/999, mshift:9, CP:7/50)						
Take 5 go vertex 638 and wait for 7(TW:170 in 170/180, mshift:2, CP:3/50)						
Take 8 go vertex 636 and wait for 0(TW:178 in 170/180, mshift:2, CP:0/50)						

外送員 4

deliver 4						
Take 0 go vertex 4 and wait for 0(TW:45 in 45/999, mshift:2, CP:0/50)						
Take 3 go vertex 978 and wait for 2(TW:50 in 50/50, mshift:0, CP:4/50)						
Take 9 go vertex 1027 and wait for 0(TW:59 in 49/59, mshift:0, CP:0/50)						
Take 3 go vertex 401 and wait for 0(TW:62 in 60/70, mshift:3, CP:2/50)						
Take 1 go vertex 971 and wait for 3(TW:66 in 66/66, mshift:0, CP:5/50)						
Take 4 go vertex 727 and wait for 0(TW:70 in 60/80, mshift:0, CP:3/50)						
Take 2 go vertex 1020 and wait for 0(TW:72 in 65/75, mshift:0, CP:0/50)						
Take 3 go vertex 512 and wait for 0(TW:75 in 72/79, mshift:0, CP:2/50)						
Take 3 go vertex 970 and wait for 0(TW:78 in 78/83, mshift:0, CP:4/50)						
Take 2 go vertex 1019 and wait for 0(TW:80 in 70/80, mshift:0, CP:2/50)						
Take 1 go vertex 598 and wait for 2(TW:83 in 83/83, mshift:0, CP:6/50)						
Take 1 go vertex 586 and wait for 0(TW:84 in 84/94, mshift:0, CP:7/50)						
Take 1 go vertex 436 and wait for 0(TW:85 in 83/87, mshift:0, CP:13/50)						
Take 3 go vertex 838 and wait for 0(TW:88 in 72/89, mshift:0, CP:11/50)						
Take 4 go vertex 924 and wait for 0(TW:92 in 83/98, mshift:0, CP:7/50)						
Take 3 go vertex 912 and wait for 0(TW:95 in 84/100, mshift:0, CP:6/50)						
Take 1 go vertex 762 and wait for 0(TW:96 in 83/96, mshift:0, CP:0/50)						
Take 11 go vertex 377 and wait for 2(TW:109 in 109/110, mshift:0, CP:3/50)						
Take 4 go vertex 461 and wait for 0(TW:113 in 113/113, mshift:0, CP:16/50)						
Take 1 go vertex 703 and wait for 0(TW:114 in 109/123, mshift:1, CP:13/50)						
Take 8 go vertex 787 and wait for 0(TW:122 in 113/123, mshift:1, CP:0/50)						
Take 6 go vertex 369 and wait for 0(TW:128 in 127/137, mshift:4, CP:2/50)						
Take 3 go vertex 695 and wait for 0(TW:131 in 127/147, mshift:4, CP:0/50)						
Take 8 go vertex 403 and wait for 3(TW:142 in 142/143, mshift:1, CP:4/50)						
Take 7 go vertex 729 and wait for 0(TW:149 in 142/153, mshift:4, CP:0/50)						
Take 11 go vertex 316 and wait for 0(TW:160 in 0/999, mshift:8, CP:3/50)						
Take 6 go vertex 642 and wait for 4(TW:170 in 170/180, mshift:4, CP:0/50)						
Take 3 go vertex 315 and wait for 0(TW:173 in 0/999, mshift:4, CP:3/50)						
Take 3 go vertex 641 and wait for 0(TW:176 in 170/180, mshift:4, CP:0/50)						

已被接受但尚未取餐或尚未送達之訂單 (即必須納入下一規劃時段考慮)

	舊編號	餐廳	顧客	對應車輛	容量	新編號	取餐上界	取餐下界	送達上界	送達下界		
3124	304	630	436	182	1	2	850	962	115	999	135	145
3122	309	635	608	6	0	4	851	963	111	999	167	177
3119	313	639	548	164	2	2	852	964	117	999	162	172
3121	314	640	564	104	1	3	853	965	86	999	170	180
3125	315	641	517	139	4	3	854	966	173	999	166	176
173	322	648	393	294	3	1	855	967	84	94	76	86
226	330	656	386	1	28	3	856	968	65	66	65	75
238	339	665	469	140	39	2	857	969	91	98	84	94
264	344	670	373	48	36	3	858	970	81	81	73	83
181	350	676	405	210	27	3	859	971	74	77	74	84
153	356	682	491	103	16	1	860	972	63	73	67	77
265	362	688	393	180	34	1	861	973	81	89	82	92
147	366	692	372	162	6	4	862	974	86	87	85	95
182	372	698	364	69	31	3	863	975	78	85	74	84
256	380	706	511	36	0	10	864	976	89	89	84	94
195	383	709	545	174	1	5	865	977	92	94	85	95
189	388	714	604	348	32	2	866	978	81	81	78	88
267	390	716	434	207	20	4	867	979	73	77	77	87
178	392	718	540	23	30	3	868	980	89	93	85	95
247	400	726	403	27	0	6	869	981	84	86	81	91
219	406	732	395	52	22	2	870	982	86	91	80	90
215	410	736	396	222	41	3	871	983	63	65	64	74
186	423	749	704	183	3	4	872	984	66	66	57	67
185	427	753	394	170	22	3	873	985	92	94	86	96
258	429	755	391	121	29	2	874	986	86	88	84	94
137	434	760	528	6	24	6	875	987	66	67	64	74
150	435	761	400	92	17	5	876	988	82	82	74	84
158	436	762	479	341	4	6	877	989	85	87	86	96
145	440	766	423	86	26	3	878	990	88	88	85	95
236	441	767	423	78	8	3	879	991	95	95	86	96
242	445	771	387	163	2	10	880	992	89	89	86	96
163	453	779	437	63	14	5	881	993	75	76	75	85
132	454	780	433	95	3	3	882	994	85	85	77	87
197	456	782	433	32	13	3	883	995	86	87	85	95
152	459	785	420	39	5	2	884	996	82	86	75	85
131	463	789	471	39	34	4	885	997	90	90	86	96
206	477	803	369	17	34	4	886	998	78	78	76	86
188	480	806	369	74	11	2	887	999	90	91	83	93
200	487	813	443	330	12	6	888	1000	90	94	86	96
211	490	816	457	295	36	3	889	1001	90	91	86	96
170	496	822	374	77	33	2	890	1002	78	80	71	81
149	501	827	692	220	19	6	891	1003	84	85	83	93
175	502	828	370	16	11	10	892	1004	82	82	76	86
180	503	829	370	11	15	4	893	1005	90	90	86	96
164	504	830	370	251	37	2	894	1006	85	94	86	96
144	512	838	470	41	4	2	895	1007	75	79	78	88
142	515	841	532	146	5	9	896	1008	88	88	82	92
218	519	845	524	285	18	2	897	1009	87	87	82	92
223	520	846	484	143	0	6	898	1010	83	83	78	88
187	521	847	627	59	15	3	899	1011	62	62	61	71
266	522	848	503	212	6	4	900	1012	87	88	78	88
148	525	851	430	292	17	3	901	1013	90	91	84	94
165	528	854	390	144	33	2	902	1014	87	89	86	96
245	529	855	390	27	47	2	903	1015	71	74	68	78
129	534	860	419	293	29	5	904	1016	85	85	86	96
214	536	862	439	328	25	6	905	1017	74	74	68	78
249	538	864	381	116	14	4	906	1018	89	91	86	96
202	555	881	559	316	9	9	907	1019	87	87	84	94
169	558	884	360	184	33	5	908	1020	76	76	69	79
167	559	885	360	226	43	2	909	1021	80	80	77	87
198	568	894	367	24	22	3	910	1022	87	92	84	94
270	573	899	367	356	35	3	911	1023	84	85	86	96
222	583	909	407	346	16	4	912	1024	83	84	79	89
220	585	911	392	258	31	2	913	1025	81	85	85	95
254	589	915	525	350	35	2	914	1026	79	83	80	90
233	595	921	375	154	38	2	915	1027	92	96	85	95
229	596	922	375	228	21	1	916	1028	84	94	83	93
177	600	926	368	297	23	2	917	1029	93	94	84	94
155	602	928	368	122	3	6	918	1030	88	90	86	96
257	604	930	368	217	24	3	919	1031	79	84	84	94
190	608	934	362	112	14	8	920	1032	67	67	62	72
191	615	941	646	115	25	3	921	1033	83	83	83	93
184	618	944	361	180	21	3	922	1034	85	91	79	89
166	619	945	361	136	5	3	923	1035	95	95	86	96
244	620	946	361	221	28	3	924	1036	83	84	83	93
194	625	951	517	129	0	1	925	1037	72	77	77	87

3115	952	1001	420	282	0	5	926	1038	154	155	158	168
3110	953	1002	402	10	0	7	927	1039	125	125	136	146
3111	954	1003	414	17	0	6	928	1040	120	130	122	132
3117	955	1004	382	35	1	4	929	1041	154	160	165	175
3112	956	1005	458	127	1	6	930	1042	122	125	130	140
3113	957	1006	498	290	1	8	931	1043	120	125	120	130
3109	958	1007	529	216	2	1	932	1044	155	160	155	165
3114	959	1008	577	22	2	7	933	1045	121	125	120	130
40	960	1009	538	141	0	3	934	1046	75	78	71	81
29	961	1010	448	20	0	3	935	1047	64	65	56	66
91	963	1012	402	229	1	15	936	1048	75	76	71	81
51	965	1014	631	212	2	2	937	1049	71	71	65	75
120	969	1018	472	300	3	3	938	1050	72	73	71	81
103	970	1019	411	151	4	2	939	1051	78	83	70	80
60	971	1020	411	187	4	3	940	1052	66	66	62	72
64	972	1021	456	200	6	3	941	1053	74	76	71	81
24	973	1022	584	212	7	4	942	1054	64	65	65	75
121	974	1023	400	258	8	3	943	1055	70	74	70	80
57	975	1024	385	339	9	7	944	1056	72	73	71	81
6	976	1025	482	188	10	2	945	1057	78	84	71	81
84	979	1028	433	17	9	3	946	1058	66	67	61	71
14	980	1029	453	309	11	3	947	1059	73	74	67	77
63	982	1031	382	81	12	7	948	1060	78	78	71	81
125	985	1034	418	210	2	3	949	1061	80	86	71	81
110	986	1035	418	291	13	6	950	1062	71	71	70	80
46	987	1036	478	170	1	1	951	1063	63	67	60	70
28	988	1037	660	172	14	3	952	1064	78	80	70	80
62	989	1038	632	97	8	5	953	1065	60	60	55	65
9	990	1039	370	332	15	3	954	1066	72	72	71	81
8	992	1041	591	57	16	4	955	1067	77	79	71	81
44	993	1042	360	46	5	2	956	1068	79	79	71	81
94	996	1045	375	246	19	2	957	1069	72	80	71	81
27	997	1046	459	39	20	2	958	1070	74	74	69	79
85	998	1047	361	102	21	3	959	1071	77	84	71	81
12	999	1048	361	114	22	2	960	1072	72	72	69	79
23	1000	1049	361	197	23	3	961	1073	65	69	62	72
3123	300	626		312	0	2		1074			131	141
3120	302	628		237	2	5		1075			133	143
3126	308	634		208	2	6		1076			170	180
239	361	687		341	26	6		1077			66	76
228	364	690		216	44	5		1078			61	71
243	379	705		205	51	2		1079			54	64
136	382	708		121	11	2		1080			60	70
141	387	713		123	48	3		1081			56	66
174	415	741		214	1	5		1082			49	59
250	416	742		344	29	3		1083			63	73
225	443	769		357	10	8		1084			82	92
160	452	778		330	15	6		1085			58	68
237	476	802		176	53	2		1086			64	74
204	479	805		292	45	6		1087			61	71
230	500	826		106	38	4		1088			67	77
133	508	834		180	13	2		1089			54	64
212	571	897		17	30	4		1090			54	64
262	582	908		32	22	2		1091			57	67
248	624	950		304	49	3		1092			52	62
36	964	1013		329	5	2		1093			64	74
45	977	1026		286	12	1		1094			58	68
31	983	1032		101	8	2		1095			57	67
13	984	1033		341	10	3		1096			59	69
26	991	1040		118	9	1		1097			53	63
59	994	1043		14	17	10		1098			70	80
11	995	1044		265	18	2		1099			71	81
3		1054		39	40	1		1100			68	78
58		1062		290	27	3		1101			60	70
39		1063		21	33	4		1102			63	73
87		1064		5	39	1		1103			68	78

■ 實務例題結果 (題組 3)

餐廳生產排程 (由於資料龐大餐廳生產排程結果僅節錄前 5 家餐廳)

餐廳 4

time_1: 0.155											
seq_											
machine1:	31	1	19	21	20	8	9	10	22	23	30
machine2:	32	2	14	5	15	25	29	28	30		
machine3:	33	26	30								
machine4:	34	4	13	24	30						
machine5:	35	27	30								
machine6:	36	6	30								
machine7:	37	7	30								
machine8:	38	17	30								
machine9:	39	18	30								
machine10:	40	3	16	30							
machine11:	41	11	30								
machine12:	42	12	30								
s_time											
machine1:	0	37	72	78	81	106	109	118	133	136	0
machine2:	0	67	82	85	88	112	118	121	0		
machine3:	0	112	0								
machine4:	0	70	80	112	0						
machine5:	0	112	0								
machine6:	0	86	0								
machine7:	0	86	0								
machine8:	0	72	0								
machine9:	0	72	0								
machine10:	0	67	82	0							
machine11:	0	128	0								
machine12:	0	129	0								

餐廳 6

time_1: 0.513											
seq_											
machine1:	68	1	19	52	51	37	60	62	66	50	67
machine2:	69	2	20	55	56	38	57	65	67		
machine3:	70	3	21	54	39	61	63	67			
machine4:	71	4	22	53	40	67					
machine5:	72	5	23	41	67						
machine6:	73	6	24	42	67						
machine7:	74	7	25	43	67						
machine8:	75	8	26	58	44	67					
machine9:	76	9	27	45	67						
machine10:	77	10	28	64	46	67					
machine11:	78	11	29	47	67						
machine12:	79	12	30	48	67						
machine13:	80	13	31	59	49	67					
machine14:	81	14	32	67							
machine15:	82	15	33	67							
machine16:	83	16	34	67							
machine17:	84	17	35	67							
machine18:	85	18	36	67							
s_time											
machine1:	0	19	67	77	92	107	110	116	119	150	0
machine2:	0	19	67	82	85	107	117	120	0		
machine3:	0	28	67	82	107	110	116	0			
machine4:	0	29	68	83	111	0					
machine5:	0	29	72	111	0						
machine6:	0	34	77	111	0						
machine7:	0	34	77	117	0						
machine8:	0	34	77	112	117	0					
machine9:	0	47	87	117	0						
machine10:	0	47	94	112	117	0					
machine11:	0	59	94	120	0						
machine12:	0	62	94	124	0						
machine13:	0	62	94	112	124	0					
machine14:	0	62	99	0							
machine15:	0	62	99	0							
machine16:	0	67	99	0							
machine17:	0	67	99	0							
machine18:	0	67	102	0							

餐廳 7

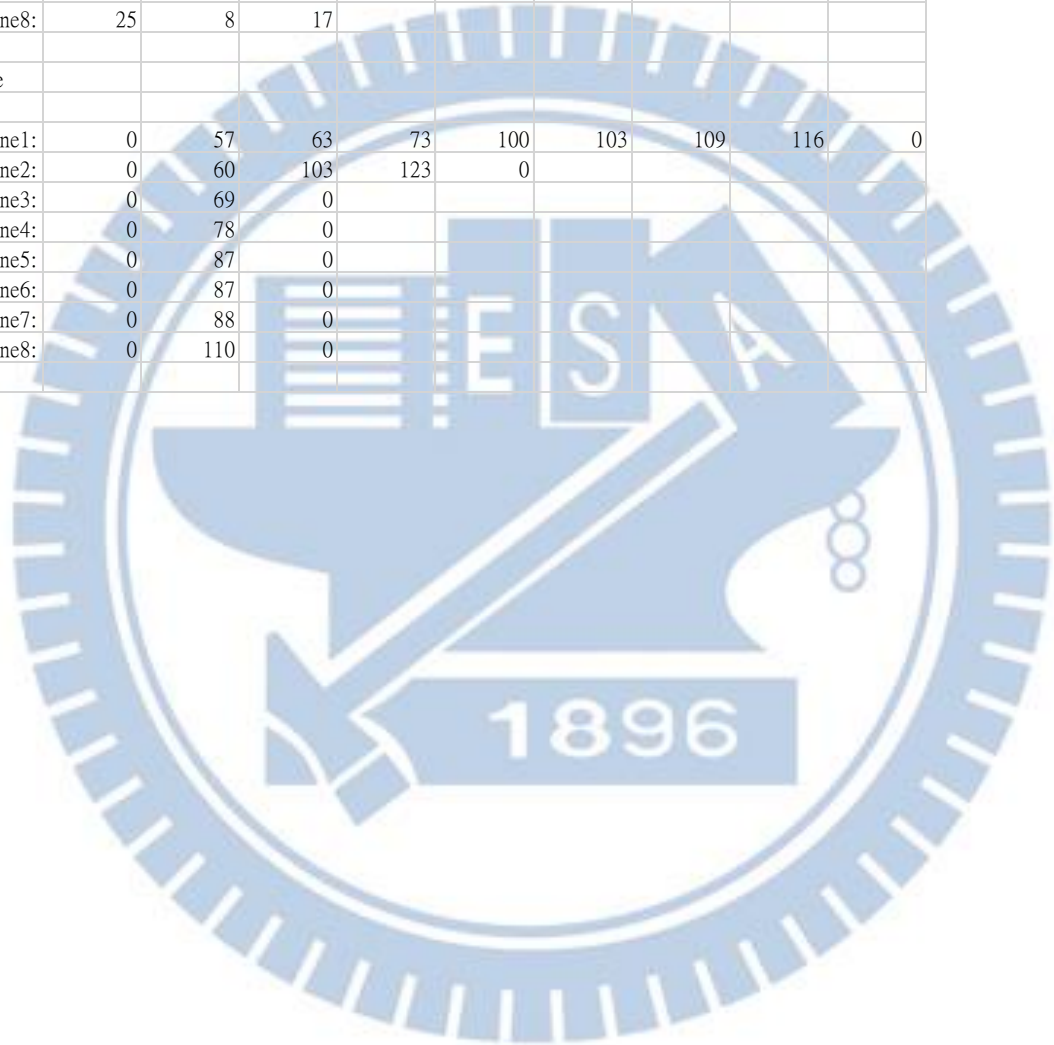
time_1: 0.994													
seq													
machine1:	77	1	43	19	62	37	69	75	73	76			
machine2:	78	2	44	41	42	20	61	38	68	72	74	76	
machine3:	79	3	21	39	71	70	76						
machine4:	80	4	22	25	40	76							
machine5:	81	5	23	76									
machine6:	82	6	12	24	76								
machine7:	83	7	76										
machine8:	84	8	26	67	76								
machine9:	85	9	45	27	65	76							
machine10:	86	10	28	66	76								
machine11:	87	11	29	64	76								
machine12:	88	30	63	76									
machine13:	89	13	31	76									
machine14:	90	14	32	76									
machine15:	91	15	33	76									
machine16:	92	16	34	76									
machine17:	93	17	35	76									
machine18:	94	18	36	76									
s_time													
machine1:	0	47	72	87	112	129	132	135	138	0			
machine2:	0	56	72	75	81	88	112	129	132	135	138	0	
machine3:	0	56	91	129	132	135	0						
machine4:	0	56	91	110	152	0							
machine5:	0	56	100	0									
machine6:	0	56	78	100	0								
machine7:	0	61	0										
machine8:	0	61	119	134	0								
machine9:	0	78	81	119	134	0							
machine10:	0	78	119	134	0								
machine11:	0	78	119	134	0								
machine12:	0	125	135	0									
machine13:	0	82	125	0									
machine14:	0	82	125	0									
machine15:	0	82	125	0									
machine16:	0	82	126	0									
machine17:	0	82	126	0									
machine18:	0	82	126	0									

餐廳 8

time_1: 1.035					
seq_					
machine1:	42	1	19	37	41
machine2:	43	2	20	38	41
machine3:	44	3	21	41	
machine4:	45	4	22	41	
machine5:	46	5	23	41	
machine6:	47	6	24	41	
machine7:	48	7	25	41	
machine8:	49	8	26	41	
machine9:	50	9	27	41	
machine10:	51	10	28	41	
machine11:	52	11	29	41	
machine12:	53	12	30	41	
machine13:	54	13	31	41	
machine14:	55	14	32	41	
machine15:	56	15	33	41	
machine16:	57	16	34	41	
machine17:	58	17	35	41	
machine18:	59	18	36	41	
s_time					
machine1:	0	62	103	118	0
machine2:	0	75	103	129	0
machine3:	0	75	106	0	
machine4:	0	82	106	0	
machine5:	0	82	106	0	
machine6:	0	83	106	0	
machine7:	0	83	106	0	
machine8:	0	83	108	0	
machine9:	0	83	108	0	
machine10:	0	90	108	0	
machine11:	0	90	113	0	
machine12:	0	90	113	0	
machine13:	0	91	113	0	
machine14:	0	91	113	0	
machine15:	0	91	113	0	
machine16:	0	91	114	0	
machine17:	0	93	116	0	
machine18:	0	93	116	0	

餐廳 11

time_1: 1.051									
seq_									
machine1:	18	1	12	11	14	13	16	9	17
machine2:	19	2	15	10	17				
machine3:	20	3	17						
machine4:	21	4	17						
machine5:	22	5	17						
machine6:	23	6	17						
machine7:	24	7	17						
machine8:	25	8	17						
s_time									
machine1:	0	57	63	73	100	103	109	116	0
machine2:	0	60	103	123	0				
machine3:	0	69	0						
machine4:	0	78	0						
machine5:	0	87	0						
machine6:	0	87	0						
machine7:	0	88	0						
machine8:	0	110	0						



外送員配送路線 (由於資料龐大外送員配送路線結果僅節錄前 5 位外送員)

外送員 0

deliver 0						
Take 0 go vertex 0 and wait for 0(TW:62 in 62/999, mshift:1, CP:2/50)						
Take 3 go vertex 872 and wait for 1(TW:66 in 66/66, mshift:0, CP:6/50)						
Take 1 go vertex 984 and wait for 0(TW:67 in 57/67, mshift:0, CP:2/50)						
Take 1 go vertex 473 and wait for 0(TW:68 in 0/999, mshift:1, CP:17/50)						
Take 1 go vertex 343 and wait for 0(TW:69 in 67/77, mshift:1, CP:18/50)						
Take 3 go vertex 618 and wait for 0(TW:72 in 67/87, mshift:1, CP:17/50)						
Take 3 go vertex 301 and wait for 0(TW:75 in 0/999, mshift:1, CP:18/50)						
Take 2 go vertex 863 and wait for 1(TW:78 in 78/85, mshift:0, CP:21/50)						
Take 5 go vertex 305 and wait for 0(TW:83 in 0/999, mshift:0, CP:36/50)						
Take 1 go vertex 975 and wait for 0(TW:84 in 74/84, mshift:0, CP:33/50)						
Take 1 go vertex 855 and wait for 0(TW:85 in 84/94, mshift:0, CP:34/50)						
Take 1 go vertex 967 and wait for 0(TW:86 in 76/86, mshift:0, CP:33/50)						
Take 3 go vertex 748 and wait for 0(TW:89 in 52/112, mshift:0, CP:18/50)						
Take 3 go vertex 873 and wait for 0(TW:92 in 92/94, mshift:0, CP:21/50)						
Take 1 go vertex 484 and wait for 0(TW:93 in 91/101, mshift:0, CP:22/50)						
Take 3 go vertex 985 and wait for 0(TW:96 in 86/96, mshift:0, CP:19/50)						
Take 7 go vertex 853 and wait for 0(TW:103 in 86/999, mshift:0, CP:22/50)						
Take 8 go vertex 759 and wait for 0(TW:111 in 91/111, mshift:0, CP:21/50)						
Take 9 go vertex 851 and wait for 0(TW:120 in 111/999, mshift:1, CP:25/50)						
Take 6 go vertex 427 and wait for 1(TW:127 in 127/137, mshift:0, CP:26/50)						
Take 5 go vertex 850 and wait for 0(TW:132 in 115/999, mshift:0, CP:28/50)						
Take 2 go vertex 1074 and wait for 0(TW:134 in 131/141, mshift:0, CP:26/50)						
Take 4 go vertex 702 and wait for 0(TW:138 in 127/147, mshift:0, CP:25/50)						
Take 1 go vertex 300 and wait for 0(TW:139 in 0/999, mshift:0, CP:26/50)						
Take 6 go vertex 962 and wait for 0(TW:145 in 135/145, mshift:0, CP:24/50)						
Take 9 go vertex 929 and wait for 0(TW:154 in 154/160, mshift:0, CP:28/50)						
Take 6 go vertex 852 and wait for 0(TW:160 in 117/999, mshift:0, CP:30/50)						
Take 4 go vertex 964 and wait for 0(TW:164 in 162/172, mshift:0, CP:28/50)						
Take 2 go vertex 1041 and wait for 0(TW:166 in 165/175, mshift:0, CP:24/50)						
Take 4 go vertex 576 and wait for 0(TW:170 in 170/180, mshift:0, CP:23/50)						
Take 3 go vertex 580 and wait for 0(TW:173 in 170/180, mshift:0, CP:8/50)						
Take 0 go vertex 965 and wait for 0(TW:173 in 170/180, mshift:0, CP:5/50)						
Take 4 go vertex 963 and wait for 0(TW:177 in 167/177, mshift:0, CP:1/50)						
Take 3 go vertex 575 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 1

deliver 1						
Take 0 go vertex 1 and wait for 0(TW:58 in 58/999, mshift:0, CP:5/50)						
Take 1 go vertex 1082 and wait for 0(TW:59 in 49/59, mshift:0, CP:0/50)						
Take 2 go vertex 860 and wait for 2(TW:63 in 63/73, mshift:0, CP:1/50)						
Take 9 go vertex 309 and wait for 0(TW:72 in 0/999, mshift:0, CP:6/50)						
Take 1 go vertex 972 and wait for 0(TW:73 in 67/77, mshift:0, CP:5/50)						
Take 2 go vertex 881 and wait for 0(TW:75 in 75/76, mshift:0, CP:10/50)						
Take 3 go vertex 895 and wait for 0(TW:78 in 75/79, mshift:0, CP:12/50)						
Take 3 go vertex 858 and wait for 0(TW:81 in 81/81, mshift:0, CP:15/50)						
Take 2 go vertex 970 and wait for 0(TW:83 in 73/83, mshift:0, CP:12/50)						
Take 2 go vertex 993 and wait for 0(TW:85 in 75/85, mshift:0, CP:7/50)						
Take 1 go vertex 1007 and wait for 0(TW:86 in 78/88, mshift:1, CP:5/50)						
Take 2 go vertex 302 and wait for 0(TW:88 in 0/999, mshift:1, CP:7/50)						
Take 2 go vertex 857 and wait for 1(TW:91 in 91/98, mshift:0, CP:9/50)						
Take 3 go vertex 969 and wait for 0(TW:94 in 84/94, mshift:0, CP:7/50)						
Take 4 go vertex 535 and wait for 0(TW:98 in 0/999, mshift:0, CP:20/50)						
Take 1 go vertex 306 and wait for 0(TW:99 in 0/999, mshift:0, CP:22/50)						
Take 8 go vertex 810 and wait for 0(TW:107 in 52/112, mshift:0, CP:9/50)						
Take 6 go vertex 304 and wait for 0(TW:113 in 0/999, mshift:0, CP:10/50)						
Take 7 go vertex 928 and wait for 0(TW:120 in 120/130, mshift:0, CP:16/50)						
Take 5 go vertex 927 and wait for 0(TW:125 in 125/125, mshift:0, CP:23/50)						
Take 3 go vertex 545 and wait for 0(TW:128 in 128/128, mshift:0, CP:28/50)						
Take 1 go vertex 493 and wait for 1(TW:130 in 130/130, mshift:0, CP:37/50)						
Take 0 go vertex 1040 and wait for 0(TW:130 in 122/132, mshift:0, CP:31/50)						
Take 4 go vertex 820 and wait for 0(TW:134 in 128/138, mshift:0, CP:26/50)						
Take 6 go vertex 768 and wait for 0(TW:140 in 130/143, mshift:0, CP:17/50)						
Take 2 go vertex 347 and wait for 0(TW:142 in 142/142, mshift:0, CP:32/50)						
Take 1 go vertex 372 and wait for 0(TW:143 in 139/143, mshift:0, CP:34/50)						
Take 1 go vertex 1039 and wait for 0(TW:144 in 136/146, mshift:1, CP:27/50)						
Take 3 go vertex 622 and wait for 0(TW:147 in 142/152, mshift:1, CP:12/50)						
Take 5 go vertex 647 and wait for 0(TW:152 in 139/153, mshift:1, CP:10/50)						
Take 3 go vertex 557 and wait for 1(TW:156 in 156/157, mshift:1, CP:12/50)						
Take 1 go vertex 832 and wait for 0(TW:157 in 156/167, mshift:5, CP:10/50)						
Take 5 go vertex 374 and wait for 5(TW:167 in 167/171, mshift:0, CP:12/50)						
Take 2 go vertex 649 and wait for 0(TW:169 in 167/175, mshift:0, CP:10/50)						
Take 1 go vertex 581 and wait for 0(TW:170 in 170/180, mshift:0, CP:8/50)						
Take 1 go vertex 584 and wait for 0(TW:171 in 170/180, mshift:0, CP:3/50)						
Take 0 go vertex 579 and wait for 0(TW:171 in 170/180, mshift:0, CP:2/50)						
Take 2 go vertex 854 and wait for 0(TW:173 in 173/999, mshift:0, CP:5/50)						
Take 3 go vertex 966 and wait for 0(TW:176 in 166/176, mshift:0, CP:2/50)						
Take 2 go vertex 577 and wait for 0(TW:178 in 170/180, mshift:2, CP:0/50)						

外送員 2

deliver 2						
Take 0 go vertex 2 and wait for 0(TW:62 in 62/999, mshift:1, CP:11/50)						
Take 2 go vertex 856 and wait for 1(TW:65 in 65/66, mshift:0, CP:14/50)						
Take 10 go vertex 968 and wait for 0(TW:75 in 65/75, mshift:0, CP:11/50)						
Take 3 go vertex 861 and wait for 3(TW:81 in 81/89, mshift:0, CP:12/50)						
Take 9 go vertex 973 and wait for 0(TW:90 in 82/92, mshift:0, CP:11/50)						
Take 2 go vertex 865 and wait for 0(TW:92 in 92/94, mshift:0, CP:16/50)						
Take 3 go vertex 977 and wait for 0(TW:95 in 85/95, mshift:0, CP:11/50)						
Take 2 go vertex 421 and wait for 7(TW:104 in 104/105, mshift:0, CP:16/50)						
Take 1 go vertex 696 and wait for 0(TW:105 in 104/112, mshift:0, CP:11/50)						
Take 2 go vertex 332 and wait for 0(TW:107 in 100/107, mshift:0, CP:17/50)						
Take 2 go vertex 607 and wait for 0(TW:109 in 100/112, mshift:0, CP:11/50)						
Take 1 go vertex 499 and wait for 0(TW:110 in 108/115, mshift:0, CP:14/50)						
Take 0 go vertex 774 and wait for 0(TW:110 in 108/112, mshift:0, CP:11/50)						
Take 3 go vertex 543 and wait for 0(TW:113 in 113/113, mshift:0, CP:20/50)						
Take 2 go vertex 354 and wait for 0(TW:115 in 112/122, mshift:2, CP:23/50)						
Take 3 go vertex 818 and wait for 0(TW:118 in 113/123, mshift:2, CP:14/50)						
Take 1 go vertex 933 and wait for 2(TW:121 in 121/125, mshift:0, CP:21/50)						
Take 6 go vertex 629 and wait for 0(TW:127 in 112/132, mshift:0, CP:18/50)						
Take 1 go vertex 1045 and wait for 0(TW:128 in 120/130, mshift:0, CP:11/50)						
Take 13 go vertex 1075 and wait for 0(TW:141 in 133/143, mshift:0, CP:6/50)						
Take 13 go vertex 926 and wait for 0(TW:154 in 154/155, mshift:0, CP:11/50)						
Take 14 go vertex 1038 and wait for 0(TW:168 in 158/168, mshift:0, CP:6/50)						
Take 9 go vertex 1076 and wait for 0(TW:177 in 170/180, mshift:3, CP:0/50)						

外送員 3

deliver 3						
Take 0 go vertex 3 and wait for 0(TW:59 in 59/999, mshift:1, CP:0/50)						
Take 14 go vertex 859 and wait for 1(TW:74 in 74/77, mshift:0, CP:3/50)						
Take 10 go vertex 971 and wait for 0(TW:84 in 74/84, mshift:0, CP:0/50)						
Take 2 go vertex 862 and wait for 0(TW:86 in 86/87, mshift:0, CP:4/50)						
Take 1 go vertex 900 and wait for 0(TW:87 in 87/88, mshift:0, CP:8/50)						
Take 1 go vertex 1012 and wait for 0(TW:88 in 78/88, mshift:0, CP:4/50)						
Take 7 go vertex 974 and wait for 0(TW:95 in 85/95, mshift:0, CP:0/50)						
Take 7 go vertex 345 and wait for 0(TW:102 in 101/102, mshift:0, CP:8/50)						
Take 6 go vertex 620 and wait for 0(TW:108 in 101/112, mshift:1, CP:0/50)						
Take 5 go vertex 449 and wait for 0(TW:113 in 107/117, mshift:1, CP:2/50)						
Take 2 go vertex 724 and wait for 0(TW:115 in 107/127, mshift:1, CP:0/50)						
Take 1 go vertex 310 and wait for 0(TW:116 in 0/999, mshift:1, CP:1/50)						
Take 1 go vertex 307 and wait for 0(TW:117 in 0/999, mshift:1, CP:4/50)						
Take 2 go vertex 931 and wait for 1(TW:120 in 120/125, mshift:0, CP:12/50)						
Take 2 go vertex 930 and wait for 0(TW:122 in 122/125, mshift:0, CP:18/50)						
Take 8 go vertex 1043 and wait for 0(TW:130 in 120/130, mshift:0, CP:10/50)						
Take 10 go vertex 1042 and wait for 0(TW:140 in 130/140, mshift:0, CP:4/50)						
Take 1 go vertex 322 and wait for 0(TW:141 in 138/142, mshift:1, CP:16/50)						
Take 3 go vertex 378 and wait for 0(TW:144 in 141/145, mshift:1, CP:23/50)						
Take 3 go vertex 653 and wait for 0(TW:147 in 141/154, mshift:1, CP:16/50)						
Take 4 go vertex 597 and wait for 0(TW:151 in 138/154, mshift:1, CP:4/50)						
Take 3 go vertex 932 and wait for 1(TW:155 in 155/160, mshift:0, CP:5/50)						
Take 10 go vertex 1044 and wait for 0(TW:165 in 155/165, mshift:0, CP:4/50)						
Take 11 go vertex 585 and wait for 0(TW:176 in 170/180, mshift:0, CP:3/50)						
Take 4 go vertex 582 and wait for 0(TW:180 in 170/180, mshift:0, CP:0/50)						

外送員 4

deliver 4						
Take 0 go vertex 4 and wait for 0(TW:59 in 59/999, mshift:2, CP:0/50)						
Take 5 go vertex 303 and wait for 0(TW:64 in 0/999, mshift:2, CP:3/50)						
Take 0 go vertex 875 and wait for 2(TW:66 in 66/67, mshift:0, CP:9/50)						
Take 5 go vertex 311 and wait for 0(TW:71 in 0/999, mshift:0, CP:13/50)						
Take 3 go vertex 987 and wait for 0(TW:74 in 64/74, mshift:0, CP:7/50)						
Take 5 go vertex 465 and wait for 0(TW:79 in 0/999, mshift:1, CP:13/50)						
Take 2 go vertex 876 and wait for 1(TW:82 in 82/82, mshift:0, CP:18/50)						
Take 1 go vertex 988 and wait for 0(TW:83 in 74/84, mshift:0, CP:13/50)						
Take 3 go vertex 869 and wait for 0(TW:86 in 84/86, mshift:0, CP:19/50)						
Take 3 go vertex 864 and wait for 0(TW:89 in 89/89, mshift:0, CP:29/50)						
Take 1 go vertex 539 and wait for 0(TW:90 in 89/99, mshift:0, CP:31/50)						
Take 1 go vertex 981 and wait for 0(TW:91 in 81/91, mshift:0, CP:25/50)						
Take 2 go vertex 740 and wait for 0(TW:93 in 52/112, mshift:0, CP:19/50)						
Take 1 go vertex 976 and wait for 0(TW:94 in 84/94, mshift:0, CP:9/50)						
Take 4 go vertex 476 and wait for 3(TW:101 in 101/101, mshift:0, CP:18/50)						
Take 5 go vertex 814 and wait for 0(TW:106 in 89/109, mshift:0, CP:16/50)						
Take 5 go vertex 751 and wait for 0(TW:111 in 101/111, mshift:0, CP:7/50)						
Take 7 go vertex 529 and wait for 0(TW:118 in 118/120, mshift:0, CP:12/50)						
Take 2 go vertex 560 and wait for 0(TW:120 in 118/120, mshift:0, CP:27/50)						
Take 4 go vertex 568 and wait for 0(TW:124 in 119/126, mshift:0, CP:29/50)						
Take 1 go vertex 804 and wait for 0(TW:125 in 118/130, mshift:0, CP:24/50)						
Take 1 go vertex 566 and wait for 0(TW:126 in 126/129, mshift:0, CP:30/50)						
Take 2 go vertex 835 and wait for 0(TW:128 in 118/130, mshift:0, CP:15/50)						
Take 6 go vertex 841 and wait for 0(TW:134 in 126/140, mshift:0, CP:9/50)						
Take 5 go vertex 843 and wait for 0(TW:139 in 119/139, mshift:0, CP:7/50)						
Take 13 go vertex 308 and wait for 0(TW:152 in 0/999, mshift:2, CP:14/50)						
Take 17 go vertex 583 and wait for 1(TW:170 in 170/180, mshift:1, CP:7/50)						
Take 8 go vertex 586 and wait for 0(TW:178 in 170/180, mshift:1, CP:3/50)						
Take 1 go vertex 578 and wait for 0(TW:179 in 170/180, mshift:1, CP:0/50)						

簡歷

中文姓名：鄭介東

英文姓名：Chieh-Tung Cheng

出生日期：民國82年7月11日

連絡地址：台中市北屯區東山路一段225-16巷8號

連絡電話：0983863505

電子信箱：danny40263@gmail.com

2019年1月國立交通大學運輸與物流管理學系碩士畢業

2015年6月國立交通大學運輸與物流管理學系學士畢業