# 科技部補助專題研究計畫成果報告
# 期中進度報告

## 一些生產與配送整合排程問題之研究(第1年)

計 畫 類 別 ： 個別型計畫
計 畫 編 號 ： MOST 104-2221-E-252-002-MY2
執 行 期 間 ： 104年08月01日至105年07月31日
執 行 單 位 ： 南開科技大學工業管理系

計 畫 主 持 人： 許洲榮

報 告 附 件 ： 出席國際會議研究心得報告及發表論文

處 理 方 式 ：
公開資訊：期中報告不提供公開查詢

中 華 民 國 105 年 05 月 30 日

中文摘要：整合排程問題是一類具有廣泛實際應用背景的新問題。在這一模型中，製造商首先完成一批工作的加工，然後把加工完的工作單獨或分批配送到客戶。每個工作有個重量，每批總重量不能超過配送批的限制。每批配送產生固定的運輸成本。整合排程問題的解包含生產排程和配送排程。生產排程確定工作的加工順序，配送排程確定使用多少配送批，每批的配送時間和路徑，每個工作所在批和配送時間。總成本函數包含兩部份，一部份與工作加工順序有關，另一部份與工作配送有關。目標是確定工作加工順序，配送批及配送時間使總成本最小。文獻中大部份考慮分批工作加工排程問題的研究不涉及配送成本。本計劃主要研究生產與配送整合的排程問題。分析問題各種情況的計算複雜度，對於P問題，提供多項式演算法。對於NP-hard問題，則依其複雜度提供動態規劃演算法、分枝界限演算法、啟發式演算法或近似演算法。這類問題的研究一方面可以豐富排程問題理論成果，另一方面可以為解決實際問題提供切實可行的方法。

中文關鍵詞：整合排程；供應鏈管理；啟發式演算法；最壞情況性能比；近似演算法

英文摘要：The integrated scheduling problem is a new class problem with extensive practical background. In this model, the manufacturer would complete the process first and then deliver the complete jobs individually or in batches to customers. Each job has a weight, and the total weight in a batch should not exceed the capacity of the delivery batch. Each delivery batch incurs a fixed distribution cost. The solution to a given integrated scheduling problem consists of a production schedule and a delivery schedule. Production schedule specifies processing sequence. Delivery schedule specifies how many shipments are used, the departure time and traveling route of each shipment, which jobs are in each shipment, and when each job is delivered. The total cost function is usually composed of two parts, one part with the job processing sequence, the other part of the job distribution. The goal is to determine the order of processing of jobs, delivery batch and delivery time of the minimum total cost. Many models that integrate job processing scheduling with batching have been studied in the literature. Almost all of those models do not involve distribution cost. This project considers detailed scheduling level integration of production and distribution. We analyze the computational complexity of various cases of the problems by either providing a polynomial algorithm for P problems or providing a dynamic programming algorithm, branch and bound algorithm, heuristic algorithm or approximation algorithm for NP-hard problems. On one hand, our research could contribute to the enrichment of theoretical accomplishment; on the other hand, it could provide useful methods to solve practical problems.

英文關鍵詞： Integrated scheduling; Supply chain management; Heuristic algorithm; Worst-case performance ratio; Approximation algorithm

# 科技部補助專題研究計畫成果報告

（■期中進度報告/□期末報告）

## （一些生產與配送整合排程問題之研究）

計畫類別：■個別型計畫　　□整合型計畫
計畫編號：MOST 104-2221-E-252-002-MY2
執行期間：104 年 08 月 01 日至 105 年 07 月 31 日

執行機構及系所：南開科技大學工業管理系

計畫主持人：許洲榮
共同主持人：
計畫參與人員：張育豪、李浚榕

本計畫除繳交成果報告外，另含下列出國報告，共 _1_ 份：
□執行國際合作與移地研究心得報告
■出席國際學術會議心得報告

期末報告處理方式：

1. 公開方式：
   ■非列管計畫亦不具下列情形，立即公開查詢
   □涉及專利或其他智慧財產權，□一年□二年後可公開查詢
2. 「本研究」是否已有嚴重損及公共利益之發現：■否 □是
3. 「本報告」是否建議提供政府單位施政參考 ■否 □是，____（請列舉提供之單位；本部不經審議，依勾選逕予轉送）

中　華　民　國　105　年　05　月　31　日

# 行政院國科技部專題研究計畫成果報告

## 一些生產與配送整合排程問題之研究

計畫編號:MOST 104-2221-E-252-002-MY2
執行期限:2014 年 08 月 01 日至 2015 年 05 月 31 日
主持人:許洲榮
計畫參與人員:張育豪、李浚榕

## 一、中文摘要

整合排程問題是一類具有廣泛實際應用背景的新問題。在這一模型中，製造商首先完成一批工作的加工，然後把加工完的工作單獨或分批配送到客戶。每個工作有個重量，每批總重量不能超過配送批的限制。每批配送產生固定的運輸成本。整合排程問題的解包含生產排程和配送排程。生產排程確定工作的加工順序，配送排程確定使用多少配送批，每批的配送時間和路徑，每個工作所在批和配送時間。總成本函數包含兩部份，一部份與工作加工順序有關，另一部份與工作配送有關。目標是確定工作加工順序，配送批及配送時間使總成本最小。文獻中大部份考慮分批工作加工排程問題的研究不涉及配送成本。本計劃主要研究生產與配送整合的排程問題。分析問題各種情況的計算複雜度，對於 P 問題，提供多項式演算法。對於 NP-hard 問題，則依其複雜度提供動態規劃演算法、分枝界限演算法、啟發式演算法或近似演算法。這類問題的研究一方面可以豐富排程問題理論成果，另一方面可以為解決實際問題提供切實可行的方法。

**關鍵字**：整合排程；供應鏈管理；啟發式演算法；最壞情況性能比；近似演算法

## Abstract

The integrated scheduling problem is a new class problem with extensive practical background. In this model, the manufacturer would complete the process first and then deliver the complete jobs individually or in batches to customers. Each job has a weight, and the total weight in a batch should not exceed the capacity of the delivery batch. Each delivery batch incurs a fixed distribution cost. The solution to a given integrated scheduling problem consists of a production schedule and a delivery schedule. Production schedule specifies processing sequence. Delivery schedule specifies how many shipments are used, the departure time and traveling route of each shipment, which jobs are in each shipment, and when each job is delivered. The total cost function is usually composed of two parts, one part with the job processing sequence, the other part of the job distribution. The goal is to determine the order of processing of jobs, delivery batch and delivery time of the minimum total cost. Many models that integrate job processing scheduling with batching have been studied in the literature. Almost all of those models do not involve distribution cost. This project considers detailed scheduling level integration of production and distribution. We analyze the computational complexity of various cases of the problems by either providing a polynomial algorithm for P problems or providing a dynamic programming algorithm, branch and bound algorithm, heuristic algorithm or approximation algorithm for NP-hard problems. On one hand, our research could contribute to the enrichment of theoretical accomplishment; on the other hand, it could provide useful methods to solve practical problems.

**Keywords:** Integrated scheduling; Supply chain management; Heuristic algorithm; Worst-case performance ratio; Approximation algorithm

## 二、緣由與目的

Production and distribution operations are two key operational functions in a supply chain. To achieve optimal operational performance in a supply chain, it is critical to integrate production and outbound delivery schedules. Some studies have tried to integrate production and distribution operations. Lee and Chen [10] study various problems with the constraint that there are a limited number of transporters available for job delivery. Their models do not consider delivery costs, and the objective is to optimize a delivery time related performance. Cheng et al. [5] consider a single machine problem, the objective function takes into account the delivery costs and the sum of earliness of the orders. Hall and Potts [8] consider various problems with an objective function combining a regular order delivery performance measure and delivery costs subject to the constraint that that each delivery batch has an unlimited capacity. Pundoor and Chen [13] studied the model with one manufacturer and one or more customers. The objective function considered both the maximum tardiness and total delivery cost. For various cases of the problem studied, they either provided an efficient algorithm or proved the intractability of the problem. Chen and Vairaktarakis [2] investigate models integrating production scheduling with delivery routing, the objective is to minimize a weighted sum of a lead time performance and total distribution cost. All of those models assume that each order has a same weight and the capacity of a batch is simply represent it by the maximum number of orders that can be included. There are some papers considered models where each order has a generally different weight. Chang and Lee [1] considered a single machine problem assume that the completed orders delivered by a single vehicle to one customer. The objective is to minimize the time when all jobs are completed and delivered to the customer area and the vehicle returns to the machine, an approximation algorithm with a worst-case ratio of 5/3 is given. Zhong at el. [17] considered the same problem, present an improved approximation algorithm with a worst-case ratio of 3/2. Chen and Pundoor [3] considered a single machine model in which each order has a generally different weight and the capacity of a batch is represented by the total weight it can carry. The objective is to find jointly a schedule for order processing and a way of packing completed orders to form delivery batches such that the total distribution cost is minimized subject to the constraint that a given customer service level is guaranteed. For two customer service constraints studied, they clarified the complexity of each problem and develop fast heuristics for the NP-hard problems and analyze their worst-case performance bounds. Recently, Chen [4] provided a survey of integrated production and outbound distribution scheduling. He present a unified model representation scheme, classify existing models into several different classes, and for each class of the models give an overview of the optimality properties, computational tractability, and solution algorithms for the various problems studied in the literature. All above models assume that the delivery dates are decision variables, however in practical environment, the distribution of orders is often not done by the manufacturer but by the third-party logistics to complete. There few paper studied problems where vehicles have fixed delivery departure dates. Li et al. [11] considered a problem involving a computer assembly manufacturer trying to synchronize the schedule of assembly operations with order delivery by air transportation. This problem is strongly NP-hard. They proposed a decomposition-based heuristic. Li et al. [12] studied the same problem but with a rescheduling consideration. Wang et al. [16] considered a problem of coordinating mail processing and distribution scheduling at a mail processing and distribution center. The authors show that this problem is strongly NP-hard and propose dispatching rules and heuristics. Stecke and Zhao [15] considered a problem in the context of a make-to-order computer manufacturer with a commit-to-delivery business mode in which each order is promised a delivery deadline. They shown that the problem with splittable delivery is polynomially solvable and the problem with nonsplittable delivery is strongly NP-hard and give a heuristic.

In this paper, we consider the integrated scheduling with fixed delivery departure dates. The manufacturer first needs to process a set of orders and then deliver the completed orders in several batches to the customer. The distribution is done by a third party logistics company. The problem is to find a schedule of processing of order, delivery batches and delivery times such that the total cost is minimized. The problem can be described as follows.

## 三、問題與相關解決方法

The manufacturer receives a set of *n* orders from the customer at time 0, $N = \{1, 2, ..., n\}$, which are to be processed on a single machine. Each order $j$ is associated with a set of integer parameters: processing time $p_j$, deadline $d_j$, and weight $w_j$. The processing speed of order $j$ is $w_j / p_j$, which is the amount of weight corresponding to every unit of the processing time of order $j$. Completed orders need to be delivered to the customer in batches. We assume that there are $m$ pre-specified fixed delivery departure dates : $t_1, ..., t_m$ $(t_1 < ... < t_m)$ and there are enough homogeneous vehicles available at each delivery departure date. The capacity of each delivery batch (vehicle) is $b$ units $(w_j \le b, j \in N)$. There is a fixed delivery cost per batch regardless of the total weight it carries. We use $c_0$ denote the delivery cost. Since all the orders are delivered to the same customer and the shipping time of any batch is the same, we assume without loss of generality that the shipping time is 0. In a given schedule, we define:

$s_j$: the starting time of order $j$.

$C_j$: completion time of order $j$.

$D_j$: delivery time of order $j$, which is the time when order $j$ is received by the customer.

$D_{\max} = \max\{D_j, j \in N\}$: maximum delivery time of orders.

$T_j = \max\{0, D_j - d_j\}$: the delivery tardiness of order $j$.

$T_{\max} = \max\{T_j, j \in N\}$: maximum delivery tardiness of orders.

Let $W_{sum} = \sum_{j=1}^{n} w_j$ and $P_{sum} = \sum_{j=1}^{n} p_j$. Clearly, if the latest departure date $t_m$ is less than $P_{sum}$ then not all the orders can be delivered and hence such a problem is infeasible. Since there are enough homogeneous vehicles available at each delivery departure date, we assume that $t_{m-1} < P_{sum} < t_m$.

We consider the following three cases of the way an order can be produced and delivered [3]:

Case (i) Non-splittable production and delivery (denoted as NSP-NSD, for ease of presentation): An order cannot be split in terms of production or delivery.

Case (ii) Non-splittable production, but splittable delivery (NSP-SD): An order cannot be split in terms of production, but can be split in terms of delivery.

Case (iii) Splittable production and delivery (SP-SD): An order can be split in terms of both production and delivery.

If order processing is non-splittable, the processing of an order $j$ cannot be preempted and must be carried out in $p_j$ consecutive time units. If order processing is splittable, it is allowed to split an order $j$ into any number of parts and each part is allowed to have a non-integer processing time, as long as the total processing time spent on all the parts together is equal to $p_j$. In this case, the completion time of an order is the completion time of its last part, and the departure time of an order is the departure time of the batch containing the last part of the order. We assume that the weight of each split part is linearly proportional to its processing time, i.e., if a part of order $j$ requires a processing time $t$, then its weight is $t w_j / p_j$.

The problem is to find a joint schedule of production and distribution such that an objective function that takes into account both customer service level and total distribution cost is optimized. We consider three problems. By the five-field notation in

Chen [4], the problem can be denoted as:

$1||V(\infty,b), fdep|1|f+TDC$,

$f\in\left\{D_{\max},\sum D_j,T_{\max}\right\}$.

For ease of presentation, we denote problem as (1), (2), and (3) for the three cases $f\in\left\{D_{\max},\sum D_j,T_{\max}\right\}$, respectively.

We first present some results. In the following, the term a partial order means a part of an order if it is split, and the term first (last) part of an order is a part of the order that is processed earliest (latest) among all split parts of the order. By a similar argument as in Chen and Pundoor [3], we can present some preliminary results about the structure of an optimal schedule.

**Lemma 1**. There exists an optimal schedule for all the three cases of three problems where

(1) There is no inserted idle time between orders and partial orders processed at the manufacturer.

(2) The departure time of each delivery batch is the earliest delivery departure date when all the orders and partial orders in it are completed processing.

**Lemma 2**. There exists an optimal schedule for cases NSP-NSD and SP-SD of each of three problems where all the orders and partial orders that are delivered in the same batch are processed consecutively at the manufacturer.

**Lemma 3**. There exists an optimal schedule for case SP-SD of each of three problems where

(1) If a delivery batch contains a partial order which is not the first part of an order, then this batch is full.

(2) In each delivery batch containing partial orders, the weight of each partial order is an integer.

## 1. Problem $1||V(\infty,b), fdep|1|D_{\max}+TDC$

In this section, we consider the problem (1): $1||V(\infty,b), fdep|1|D_{\max}+TDC$. Clearly, case NSP-NSD of each of three problems is strongly NP-hard because the classical bin-packing problem, which is strongly NP-hard [7], is a special case of it with zero processing times. We first present the two batching procedures.

When the order processing is non-splittable, to schedule a given subset of orders $Q$ starting from a given time $t$ in a particular step, the following *FFD* procedure is used.

Procedure *FFD*

Input: A subset of orders $Q$ and a starting time $t$ for the first order.

Step 1: Assign the orders of $Q$ to delivery batches using the *First-Fit-Decreasing* (*FFD*) rule [6] for the classical bin-packing problem. By the *FFD* rule, orders in $Q$ are first sorted in the non-increasing sequence of their weights. Orders are assigned to batches by this sequence. Initially there are no existing batches. The very first order is assigned to a new batch. After that, if an order can fit into one of the existing batches, it is then assigned to the existing batch with smallest index. If an order cannot fit into any existing batch, it is then assigned to a new batch. Repeat this until every order in $Q$ is assigned. Let $h$ be the number of batches formed by the *FFD* rule for the orders in $Q$. Let $P_k$ be the total processing time of the orders assigned to batch $B_k$, for $k=1,\dots,h$.

Step 2: Let $\tau=t$. For $k=1,\dots,h$, process the orders of batch $B_k$ in time interval $[\tau,\tau+P_k]$ and deliver this batch at delivery departure date $\tilde{t}_k=\min\left\{t_i \mid t_i\ge\tau+P_k\right\}$, and update $\tau=\tau+P_k$.

When the order processing is splittable, to schedule a given subset of orders $Q$ starting from a given time $t$ in a particular step, the following *FB* procedure is used.

Procedure *FB*

Input: A subset of orders $Q$ and a starting time $t$ for the first order.

Step 1: Specify a sequence of the orders in $Q$ and denote it by $([1],\dots,[u])$, where $u=|Q|$.

Step 2: Let $h=\left\lceil\left(\sum_{j=1}^{u}w_{[j]}\right)\big/b\right\rceil$. Assign the orders of $Q$ to $h$ delivery batches using the following *Full-Batch* (*FB*) rule. Take the whole orders $[1],\dots,[i_1]$ and a portion $\alpha$ $(0<\alpha<1)$ of $[i_1+1]$, such that $\sum_{j=1}^{u} w_{[j]}+\alpha w_{[i_1+1]}=b$, and assign them to the first delivery batch. Take the remaining part of order $[i_1+1]$ and a number of whole orders $[i_1+2],...,[i_2]$ and possibly a partial order $[i_2+1]$, such that their total weight is exactly $b$, and assign them to the second delivery batch. Repeat the above until all the orders are assigned. Clearly, all the batches except possibly the last one are full. Let $P_k$ be the total processing time of the orders and partial orders assigned to batch $B_k$, for $k=1,...,h$.

Step 3: Let $\tau=t$. For $k=1,...,h$, process the orders of batch $B_k$ in time interval $[\tau,\tau+P_k]$ and deliver this batch at delivery departure date $\tilde{t}_k=\min\{t_i \mid t_i\ge\tau+P_k\}$, and update $\tau=\tau+P_k$.

By *FFD* procedure, we propose an approximation algorithm for case NSP-NSD of the problem

$$1\|V(\infty,b),\ fdep\,|1|\,D_{\max}+TDC.$$

Algorithm 1

Step 0: Input: A set of orders $Q=N$ and a starting time $t=0$ for the first order.

Step 1: Applying the procedure *FFD*, get all orders processing sequence in time interval $[0,P_{sum}]$, batches and corresponding delivery departure dates.

Step 2: Let $h$ be the number of batches formed by the *FFD* rule for the orders in $Q$, calculate objective $D_{\max}+TC=t_m+hc_0$.

**Theorem 1**. The worst-case performance ratio of Algorithm 1 for case NSP-NSD of problem (1) is bounded by $3/2$.

**Proof** Let $\pi$ is the solution generated by Algorithm 1 (for case NSP-NSD), the number of batches in the solution is $h$, then the objective value is $Z=t_m+hc_0$. Let $\pi*$ is the optimal solution of case NSP-NSD of problem (1), the number of batches in the optimal solution is $h*$, the objective value is $Z*=D^*_{\max}+h*c_0$.

Since $t_{m-1}<P_{sum}\le t_m$, then $D^*_{\max}=t_m$. Note that the worst-case performance ratio of *FFD* for bin-packing problem is bounded by $3/2$ [14], we have $h\le(3/2)h*$. Therefore

$$Z=t_m+hc_0\le t_m+(3/2)h*\le(3/2)Z*.$$

This completes the proof of Theorem.

When the order processing is splittable, by the *FB* procedure, we propose the following polynomial optimal algorithm.

Algorithm 2

Step 0: Input: A set of orders $Q=N$ and a starting time $t=0$ for the first order.

Step 1: Applying the procedure *FB*, get all orders processing sequence in time interval $[0,P_{sum}]$, batches and corresponding delivery departure dates.

Step 2: The number of batches by procedure *FB* is $h=\lceil W_{sum}/b\rceil$, calculate objective $D_{\max}+TC=t_m+hc_0$.

Similar to case NSP-NSD, for any solution of problem (1), we have $D_{\max}=t_m$. Since the total weight of all orders is $W_{sum}=\sum_{j=1}^{n}w_j$ and the capacity of delivery batch is $b$, there are at least $\lceil W_{sum}/b\rceil$ batches in any feasible solution. Therefore we have following solution.

**Theorem 2**. Algorithm 2 finds an optimal solution for cases NSP-SD and SP-SD of problem (1).

## 2. Problem $1\|V(\infty,Q),\ fdep\,|1|\sum D_j+TDC$

In this section, we consider the problem (2): $1\|V(\infty,Q),\ fdep\,|1|\sum D_j+TDC$. We first analyze the complexity of the problem, then present the approximation algorithms.

### 2.1 Solvability of Cases NSP-SD and SP-SD

Clearly, case NSP-NSD of problems (2) is strongly NP-hard. We will show that both cases NSP-SD and SP-SD of the problem (2) are strongly NP-hard.

We give a result which will be used later in the NP-hardness proofs.

**Lemma 4**. Given any positive integer $k$ and any $k$ numbers $E_1,...,E_k$ with

$E_1 < ... < E_k$ , the following holds: $\sum_{j=1}^{k} E_j y_j \ge 3\sum_{j=1}^{h} E_j$ for any non-negative integers $y_1,...,y_k$ satisfying: $\sum_{j=1}^{k} y_j \ge 3(k-i+1)$ for all $i = 1,...,k$ [3].

We first consider case NSP-SD of the problem (2).

**Theorem 3**. The problem (2) with non-splittable production but splittable delivery is strongly NP-hard.

**Proof** The proof is by reduction from the following problem, which is known to be strongly NP-complete [7].

3-Partition Problem (3-PP): given $K \in Z_+$ and a set $A = \{a_1,...,a_{3h}\}$ of $3h$ integers such that $K/4 < a_j < K/2$ for $1 \le j \le 3h$ and $\sum_{j=1}^{3h} a_j = hK$ , can $A$ be partitioned into disjoint sets $A_1,...,A_h$ such that $\sum_{a_j \in A_i} = K$ for each $1 \le i \le h$ ?

For any given instance $I$ of the 3PP, we construct the corresponding instance $II$ of problem (2) as follows:

$n = 3h$ orders, $N = \{1,...,3h\}$ with $p_j = hK - a_j$ , $w_j = hK + a_j$ , $j = 1,...,3h$ .

$m = h$ fixed delivery departure dates: $t_i = (3h-1)Ki$ , $i = 1,...,h$ .

Batch capacity: $b = (3h+1)K$ .

Threshold cost:

$$Y = \frac{3}{2}(3h-1)h(h+1)K + hc_0 ,$$

where $c_0 > \frac{3}{2}(3h-1)h(h+1)K$ .

(If Part) Assume that $I$ has a solution $A_1,...,A_h$ . For convenience, denote by $A_i$ $(i = 1,...,h)$, the set of orders corresponding to the elements of set C, i.e., $A_i = \{ j \mid a_j \in A_i \}$ $(i = 1,...,h)$ . Process the orders in the sequence $(A_1,...,A_h)$ , the sequence of orders within each $A_i$ is immaterial. For each $i = 1,...,h$ , the three orders of $A_i$ complete processing at time $(3h-1)K$ , and total weight of these three orders is $(3h+1)K = b$ . Deliver the three orders of $A_i$ in one batch at delivery departure date $t_i$ , for $i = 1,...,h$ . Clearly, in this solution, all batch are full batch, the total delivery time is $\frac{3}{2}(3h-1)h(h+1)K$ , the total delivery cost is $hc_0$ , and the objective value is $\frac{3}{2}(3h-1)h(h+1)K + hc_0$ .

(Only If Part) Assume that $II$ has a solution $\pi$ with total cost no more than $Y$ . Since the total weight of all orders is $h(3h+1)K$ and the batch capacity is $b = (3h+1)K$ , then we need at least $h$ delivery batches. Note that the threshold cost is $\frac{3}{2}(3h-1)h(h+1)K + hc_0$ and $c_0 > \frac{3}{2}(3h-1)h(h+1)K$ , we need at most $h$ delivery batches. Therefore, we can conclude that in schedule $\pi$ there are exactly $h$ delivery batches and each batch is full, the total delivery time smaller than or equal to $\frac{3}{2}(3h-1)h(h+1)K$ .

Suppose that there are $h$ batches in schedule $\pi$: $B_1(\pi),...,B_h(\pi)$ . Let $n_i$ be the number of orders and partial orders in $B_i(\pi)$, and $t_i^\pi$ be the departure time of delivery batch $B_i(\pi)$ in schedule $\pi$ . We have following results.

(1). $n_1 + ... + n_i \le 3i$ , for $i = 1,...,h$ .

We prove this by contradiction. Suppose for some $i$ , $n_1 + ... + n_i \ge 3i$ , then the total weight of the orders in $B_1(\pi),...,B_i(\pi)$ is more than $(3i+1)hK \ge i(3h+1)K = ib$ . This means that more than $i$ batches are needed to deliver those orders, which is a contradiction with the fact that those orders are delivered in $i$ batches in $\pi$ .

(2). $t_i^\pi \ge t_i$ , for $i = 1,...,h$ .

Since $B_i(\pi)$ is delivered at $t_i^\pi$ $(i = 1,...,h)$, then the orders in $B_1(\pi),...,B_i(\pi)$ have completed processing at $t_i^\pi$ . Note that each batch is full batch, the total weight of the orders in $B_1(\pi),...,B_i(\pi)$ is $ib = i(3h+1)K$ .

Recall that the processing speed of an order $j$ is defined as the ratio of its weight to its processing time $w_j/p_j$, to fill one unit of weight, we need at least $\min\{p_j/w_j \mid j \in N\}$ units of processing time. Let $a_{\max} = \max\{a_j \mid j \in A\}$, then

$$\min\{p_j/w_j \mid j \in N\} = \frac{hK - a_{\max}}{hK + a_{\max}}$$
$$\geq \frac{hK - K/2}{hK + K/2} = \frac{2h-1}{2h+1}. \quad (1)$$

From (1), the time when $i$ batches $B_1(\pi),...,B_i(\pi)$ are fully filled, we need at least $\frac{2h-1}{2h+1}(3h+1)Ki > (3h-1)K(i-1) = t_{i-1}$ units of processing time. Thus $t_i^\pi \geq t_i$.

Since $n_1 + ... + n_i \leq 3i$ and $n_1 + ... + n_h = 3h$, we have $n_i + ... + n_h \geq 3(h-i+1) \quad (i = 1,...,h)$.

Now, we show that $n_1 = ... = n_h = 3$ by contradiction. Suppose that there exists some $k$ ,such that $n_{k+1} = ... = n_h = 3$ and $n_k > 3$. Define $y_k = n_k \quad (k = 1,...,k-2)$, $y_{k-1} = n_{k-1} + n_k - 3$. It can be easily verified that $\sum_{j=1}^{k-1} y_j \geq 3[(k-1)-i+1]$ for all $i = 1,...,k-1$. From Lemma 4, we have

$$\sum_{j=1}^{k-1} t_j y_j \geq 3\sum_{j=1}^{k-1} t_j. \quad (2)$$

Therefore, the total delivery time in schedule $\pi$ is

$$\sum_{j=1}^{n} D_j = \sum_{i=1}^{h} n_i t_i^\pi \geq \sum_{i=1}^{h} n_i t_i$$
$$= \left(\sum_{i=1}^{k-1} n_i t_i\right) + 3t_k + (n_k - 3)t_k + \sum_{i=k+1}^{h} 3t_i$$
$$= \left(\sum_{i=1}^{k-1} n_i t_i\right) + (n_k - 3)t_{k-1} + \left(\sum_{i=k}^{h} 3t_i\right) + (n_k - 3)(3h-1)K$$
$$= \left(\sum_{i=1}^{k-2} n_i t_i\right) + (n_{k-1} + n_k - 3)t_{k-1} + \left(\sum_{i=k}^{h} 3t_i\right) + (n_k - 3)(3h-1)K$$
$$= \left(\sum_{j=1}^{k-1} y_j t_j\right) + \left(\sum_{i=k}^{h} 3t_i\right) + (n_k - 3)(3h-1)K$$

From (2), we have

$$\sum_{j=1}^{n} D_j \geq 3\left(\sum_{j=1}^{k-1} t_j\right) + \left(\sum_{j=k}^{h} 3t_j\right) + (n_k - 3)(3h-1)K$$
$$= 3\left(\sum_{j=1}^{h} t_j\right) + (n_k - 3)(3h-1)K$$
$$= (3/2)(3h-1)h(h+1)K + (n_k - 3)(3h-1)K$$
$$> (3/2)(3h-1)h(h+1)K. \quad (3)$$

This contradicts the cost in $\pi$ is no more than $(3/2)(3h-1)h(h+1)K + hc_0$. Therefore, $n_1 = ... = n_h = 3$. This means that in schedule $\pi$, for every $i = 1,...,h$, the first $i$ delivery batches $B_1(\pi),...,B_i(\pi)$ together deliver the first $3i$ orders and possibly a part of the $(3i+1)st$ order.

Let the processing time and weight of the part of the $(3i+1)st$ order covered in the $i$th batch $B_i(\pi)$ be denoted as $\alpha_i$ and $\beta_i$. If batch $B_i(\pi)$ does not cover a part of the $(3i+1)st$ order, then $\alpha_i = \beta_i = 0$. Let the processing sequence of orders under schedule $\pi$ be denoted as $([1],...,[3h])$. Let the total processing time and the total weight of the orders in $B_i(\pi)$ be denoted as $P_{B_i}$ and $W_{B_i}$ $(i = 1,...,h)$. Thus, the total weight of the orders covered in the first $i$ batches $B_1(\pi),...,B_i(\pi)$ is

$$\sum_{k=1}^{i} W_{B_k} = 3ihK + \left(\sum_{j=1}^{3i} a_{[j]}\right) + \beta_i.$$

Since each batch is full, then $3ihK + \left(\sum_{j=1}^{3i} a_{[j]}\right) + \beta_i = ib$, which implies that

$$\sum_{j=1}^{3i} a_{[j]} = iK - \beta_i. \quad (4)$$

Note that the total processing time of the orders covered in the first $i$ batches $B_1(\pi),...,B_i(\pi)$ is

$$\sum_{k=1}^{i} P_{B_k} = 3ihK - \left(\sum_{j=1}^{3i} a_{[j]}\right) + \alpha_i.$$ From (4), we have

$$\sum_{j=1}^{n} D_j \geq \sum_{i=1}^{h}\left(3ihK - iK + \alpha_i + \beta_i\right)$$
$$= (3/2)(3h-1)h(h+1)K + \sum_{i=1}^{h}\left(\alpha_k + \beta_i\right).$$

Since the total delivery time in schedule $\pi$ must be no more than $(3/2)(3h-1)h(h+1)K$, we have $\alpha_i = \beta_i = 0$ $(i = 1,...,h)$. Therefore, $\sum_{j=1}^{3i} a_{[j]} = iK \quad (i = 1,...,h)$ and $I$ has a solution$([1],[2],[3]),([4],[5],[6]),...,$

$([3h-2],[3h-1],[3h])$. This completes the proof of Theorem.

In case SP-SD, both processing preemption and delivery split of an order are allowed.

It can be easily verified that all the results proved in the proof of Theorem 3 up to the result $n_1 = ... = n_h = 3$ in the "Only If" part apply to the problem with case SP-SD as well. The arguments given there after the result $n_1 = ... = n_h = 3$ are applicable to case NSP-SD only, but can be slightly modified as follows to work for the problem with case SP-SD: The result "$n_1 = ... = n_h = 3$" means that in schedule $\pi$, for every $i = 1,...,h$, the first $i$ delivery batches $B_1(\pi),...,B_i(\pi)$ together deliver $3i$ orders and possibly parts of some other orders. Let the total processing time and total weight of the parts of the other orders covered in these batches be denoted as $\alpha_i$ and $\beta_i$. Let the completion sequence of orders under schedule $\pi$ be denoted as $([1],...,[3h])$. Then the rest of the proof Theorem 3 apply to the problem with case SP-SD as well. Therefore we have following solution.

**Theorem 4**. The problem (2) with splittable production and delivery is strongly NP-hard.

## 2.2 Approximation algorithms

We first propose an approximation algorithm for case NSP-NSD of problem (2), which is a dynamic programming based approach. The dynamic program builds up a schedule step by step from time 0 to $P_{sum}$, and in each step a subset of orders is scheduled for processing and delivery. Let $m = \lceil W_{sum}/b \rceil$. Clearly, at least(most) $m(n)$ delivery batches are necessary to deliver the $n$ orders of $N$.

**Algorithm 3**

Step 0: Re-index the orders of $N$ in *SPT* rule, i.e., $p_1 \le ... \le p_n$.

Step 1: Run the following dynamic programming algorithm.

Define value function $F(i,j)$ to be the minimum total cost(sum of total delivery time and total delivery cost) of the first $j$ orders $1,...,j$ given that they are processed from time 0 without idle time and that they are delivered in $i$ batches.

*Initial conditions*: $F(0,0) = 0$ and $F(i,j) = \infty$ for any $(i,j)$ satisfying: $i < 0$, or $i = 0$ and $j > 0$.

*Recursive relations*: For $i = m,...,n$, and $j = 1,...,n$,

$F(i,j) = \min\{F(i - g(k+1,j),k) +$
$D_{sum}(k+1,j) + g(k+1,j)c_0 \mid k = 0,..., j-1\}$,

where $g(k+1,j)$ is the number of delivery batches formed by applying the procedure *FFD* to the subset of orders $Q = \{k+1,...,j\}$ with the starting time $t = \sum_{j=1}^{k} p_j$, and $D_{sum}(k+1,j)$ is the corresponding total delivery time of the orders of $Q$.

*Solutions*:

$F(q,n) = \min\{F(i,n) \mid i = m,...,n\}$.

Step 2: Let the corresponding schedule be denoted as $\pi(q,n)$, and the number of delivery batches used is $q$.

We note that the dynamic program considers all the schedules with the following easy-to-prove structure:

The order processing sequence can be divided into a number of blocks $G_1,...,G_u$ such that

(i). orders across different blocks are scheduled in *SPT* sequence, i.e., $p_{[i]} \le p_{[j]}$, for $[i] \in G_{i_1}$, $[j] \in G_{i_2}$ and $i_1 \le i_2$ $(1 \le i_1, i_2 \le u)$.

(ii). the orders within a block are scheduled by the procedure *FFD* and consequently they are divided into one or more subsets by the *FFD* rule, each delivered by a separate batch. Note that the orders within each block are not necessarily scheduled in *SPT* order.

For ease of presentation, we call a schedule with the above structure a *B-SPT-FFD* schedule. $\pi(q,n)$ is the optimal of all *B-SPT-FFD* schedules.

**Theorem 5**. The worst-case performance ratio of Algorithm 3 for case NSP-NSD of the problem (2) is bounded by 3.

**Proof** Given an optimal schedule $\pi^*$ for case NSP-NSD of the problem (2), let $h^*$

be the number of delivery batches used in $\pi*$. Let $n_i^*$ denote the number of orders and $t_i^*$ delivery departure date of the $i$th $(i=1,\dots,h*)$ batch of $\pi*$, then

$$h* \geq W_{sum}/b. \qquad (5)$$

We construct a *B-SPT-FFD* schedule $\pi$ based on $\pi*$ using the following procedures:

(i). Process the $n$ orders in *SPT* sequence. Denote this sequence as $([1],\dots,[n])$.

(ii). Divide the sequence $([1],\dots,[n])$ into $h*$ blocks of consecutive orders, denoted as $G_1,\dots,G_{h^*}$, such that the $i$th block $G_i$ consists of the $n_i^*$ orders:

$$G_i = \left(\left[\sum_{u=1}^{i-1} n_u^* + 1\right],\dots,\sum_{u=1}^{i} n_u^*\right)\ (i=1,\dots,h*).$$

Let the completion time of the last order of $G_i$ be $E_i$. It can be easily shown that

$$E_i \leq t_i^*,\ \ i=1,\dots,h*. \qquad (6)$$

This means that the departure time of $G_i$ is no more than $t_i^*$.

(iii). Denote the total weight of the orders in $G_i$ as $W_i$ $(i=1,\dots,h*)$. If $W_i \leq b$, then deliver all the orders of $G_i$ in a single batch at time $t_i^*$. Otherwise, apply the procedure *FFD* to the orders of $G_i$ and deliver all the orders of $G_i$ in a $k_i$ batches $G_i^1,\dots,G_i^{k_i}$ at time $t_i^*$. Denote the total weight of the orders in $G_i^l$ as $W_i^l$ $(l=1,\dots,k_i)$, $W_i=\sum_{l=1}^{k_i} W_i^l$. From procedure *FFD*, we have $W_i^l + W_i^{l+1} > b$ $(l=1,\dots,k_i-1)$, $W_i^{k_i}+W_i^1 > b$. Thus, $2\sum_{l=1}^{k_i} W_i^l > k_i b$, i.e., $2W_i > k_i b$, we have $k_i < 2W_i/b$.

Denote set $H_1=\{i \mid k_i=1, 1\leq i\leq h^*\}$ and $H_2=\{i \mid k_i \geq 2, 1\leq i\leq h^*\}$. Denote the total number of delivery batches in schedule $\pi$ by $h(\pi)$. Then

$$h(\pi) = |H_1| + \sum_{i\in H_2} k_i$$
$$\leq |H_1| + 2\sum_{i\in H_2} W_i/b$$
$$\leq |H_1| + 2W_{sum}/b$$
$$\leq h^* + 2h^* = 3h^*.$$

Therefore, the number of delivery batches used in $\pi$ is at most 3 times that of the optimal number of delivery batches. Clearly, the total delivery time of $\pi$ is no more than that of $\pi^*$. Since the dynamic program in the algorithm 3 considers all *B-SPT-FFD* schedules including $\pi$, we complete the proof of Theorem.

Now, we propose an approximation algorithm, call Algorithm 4, for cases NSP-SD and SP-SD of problem (2). The general idea and structure of Algorithm 4 are similar to that of the Algorithm 3. It is also dynamic programming based and the DP tries to find an optimal schedule among a subset of feasible schedules. However, since partial delivery of an order is allowed in cases NSP-SD and SP-SD of the problem, the procedure used to schedule a given subset of orders in each step of the DP is different.

**Algorithm 4**

Step 0: Re-index the orders of $N$ in *SPT* rule, i.e., $p_1 \leq \dots \leq p_n$.

Step 1: Run the following dynamic programming algorithm.

Define value function $F(i,j)$ to be the minimum total cost (sum of total delivery time and total delivery cost) of the first $j$ orders $1,\dots,j$ given that they are processed from time 0 without idle time and that they are delivered in $i$ batches.

*Initial conditions*: $F(0,0)=0$ and $F(i,j)=\infty$ for any $(i,j)$ satisfying: $i<0$, or $i=0$ and $j>0$.

*Recursive relations*: For $i=m,\dots,n$ and $j=1,\dots,n$,

$F(i,j)=\min\{F(i-g(k+1,j),k)+ D_{sum}(k+1,j)+g(k+1,j)c_0 \mid k=0,\dots,j-1\}$,

where $g(k+1,j)$ is the number of delivery batches formed by applying the procedure *FFD* to the subset of orders $Q=\{k+1,\dots,j\}$ with the starting time $t=\sum_{j=1}^{k} p_j$, and

$D_{sum}(k+1, j)$ is the corresponding total delivery time of the orders of $Q$.

*Solutions*:

$F(q,n) = \min\{F(i,n) \mid i = m,...,n\}$.

Step 2: Let the corresponding schedule be denoted as $\pi(q,n)$, and the number of delivery batches used is $q$.

The dynamic program considers all the schedules with the following structure:

The order processing sequence can be divided into a number of blocks $G_1,...,G_u$ such that

(i). orders across different blocks are scheduled in *SPT* sequence.

(ii). the orders within a block are scheduled by the procedure *FB* and consequently they are divided into one or more subsets by the *FB* rule, each delivered by a separate batch.

We call a schedule with the above structure a *B-SPT-FB* schedule. $\pi(q,n)$ is the optimal of all *B-SPT-FB* schedules.

**Theorem 6**. The worst-case performance ratio of Algorithm 4 for cases NSP-SD and SP-SD of the problem (2) is bounded by 2.

**Proof** Given an optimal schedule $\pi*$ for cases NSP-SD and SP-SD of the problem (2), let $h*$ be the number of delivery batches used in $\pi*$. Let $n_i^*$ denote the number of orders and $t_i^*$ delivery departure date of the $i$th $(i=1,...,h*)$ batch of $\pi*$.

We construct a *B-SPT-FB* schedule $\pi$ based on $\pi*$ using the following procedures:

(i). Process the *n* orders in *SPT* sequence. Denote this sequence as $([1],...,[n])$.

(ii). Divide the sequence $([1],...,[n])$ into $h*$ blocks of consecutive orders, denoted as $G_1,...,G_{h^*}$, such that the $i$th block $G_i$ consists of the $n_i^*$ orders:

$G_i = \left(\left[\sum_{u=1}^{i-1} n_u^* + 1\right],...,\sum_{u=1}^{i} n_u^*\right)$ $(i=1,...,h*)$.

The departure time of $G_i$ is no more than $t_i^*$.

(iii). Denote the total weight of the orders in $G_i$ as $W_i$ $(i=1,...,h*)$. If $W_i \le b$, then deliver all the orders of $G_i$ in a single batch at time $t_i^*$. Otherwise, apply the procedure *FFD* to the orders of $G_i$ and deliver all the orders of $G_i$ in a $k_i$ batches $G_i^1,...,G_i^{k_i}$ at time $t_i^*$.

Since the procedure *FB* generates for each block at most one batch which is less than full, then

$$W_i > (k_i - 1)b. \qquad (7)$$

Denote the total number of delivery batches in schedule $\pi$ by $h(\pi)$, we have

$$\begin{aligned} h(\pi) &= \sum_{i=1}^{h^*} k_i \\ &= h^* + \sum_{i=1}^{h^*} (k_i - 1) \\ &\le h^* + \sum_{i=1}^{h^*} W_i / b \\ &= h^* + W_{sum}/b \\ &\le h^* + h^* = 2h^*. \end{aligned}$$

Therefore, the number of delivery batches used in $\pi$ is at most 2 times that of the optimal number of delivery batches. Clearly, the total delivery time of $\pi$ is no more than that of $\pi^*$. Since the dynamic program in the algorithm 3 considers all *B-SPT-FB* schedules including $\pi$, we complete the proof of Theorem.

## Conclusions

In many situation, the companies worldwide now rely on third-party logistics providers for their daily distribution and third-party logistics have fixed delivery departure dates. In this paper, we have analyzed an integrated production distribution scheduling model where the orders generally have different sizes while the delivery batch capacity is finite and delivery batches have fixed delivery departure date. The total cost function is usually composed of two parts, one part with the job processing sequence, the other part of the job distribution. The goal is to determine a schedule of processing of order, delivery batches and delivery times such that the total cost is minimized. We first study the solvability of various cases of the problem by either providing an efficient algorithm or proving the intractability of the problem. We then develop an approximation algorithm with worst-case performance analysis for NP-hard problem.

There are several interesting problems for future research. First, we have assumed that there is one customer. A more realistic problem may be that there are more customers. Second, we have assumed that there are homogeneous vehicles available at each delivery departure date and there is a fixed delivery cost per batch regardless of the total weight it carries. Another extension to our model may be that at each delivery departure date, there are heterogeneous vehicles available and the delivery cost of a batch dependent on the weight carried by the batch.

## 四. 計劃成果自評

本計劃主要研究生產與配送整合的排程問題，是兩年期的研究計劃，探討三個不同的生產與配送模式對應三個研究目標函數，總共預定解決九個生產排程問題。第一年已經解決六個生產與配送整合的排程問題，撰寫成研究論文，詳細呈現在本報告中的第二和第三節，我們分析各問題的計算複雜度，對於 P 問題，提供多項式演算法，對於 NP-hard 問題，則依其複雜度提供動態規劃演算法、分枝界限演算法、啟發式演算法或近似演算法。這類問題的研究一方面可以豐富排程問題理論成果，另一方面可以為解決實際問題提供切實可行的方法。

第一年度計劃執行期間，已經發表兩篇註記感謝科技部補助的編號之期刊論文，以及一篇已修改(revised)論文，詳列如下：

1. Note on a unified approach to the single-machine scheduling problem with a deterioration effect and convex resource allocation, *Journal of Manufacturing Systems* [SCI; IF: 1.682], 38, pp.134-140. MOST 104-2221-E-252-002-MY2.

2. Unrelated Parallel-Machine Scheduling Problems with General Truncated Job-Dependent Learning Effect, *Journal of Applied Mathematics and Physics*. [Google-based Impact Factor: 0.37], 4, pp. 21-27. MOST 104-2221-E-252-002-MY2.

3. Scheduling deteriorating jobs with machine availability constraints to minimize the total completion time. *Journal of Industrial and Production Engineering*-Manuscript ID TJCI-2015-0009.R1

計劃執行期間，我們也培育、訓練了一位碩士班研究生和一位大學部學生，碩士班研究助理經過兩年的訓練，畢業後可以直接投入職場從事生產管理工作，亦可繼續就讀博士班，從事排程理論探討或應用研究。另一位大學部研究助理已經推甄錄取雲林科技大學碩士班。

整體而言，第一年研究計劃大致有達到預定目標，主持人由衷的感謝科技部的補助，讓我們可以充實實驗室設備和資源，以及參加國際研討會，拓展國際視野。

## 五、參考文獻


[1] Chang, Y.-C., C.-Y. Lee. 2004. Machine scheduling with job delivery coordination. European Journal of Operational Research, 158, 470-487.

[2] Chen, Z.-L., G. L. Vairaktarakis. 2005. Integrated scheduling of production and distribution operations. Management Science, 51, 614-628.

[3] Chen, Z.-L., G. Pundoor. 2009. Integrated order scheduling and packing. Production and Operations Management, 18, 672-692.

[4] Chen, Z.-L., 2010. Integrated production and outbound distribution scheduling: Review and extensions. Operations Research, 58, 130-148.

[5] Cheng, T. C. E., V. S. Gordon, M. Y. Kovalyov. 1996. Single machine scheduling with batch deliveries. European Journal of Operational Research , 94, 277-283.

[6] Coffman Jr., E. G., M. R. Garey, D. S. Johnson. 1997. Approximation algorithms for bin packing: A survey. In: D. S. Hochbaum, ed. Approximation Algorithms for NP-Hard Problems. PWS Publishing, Boston, 46-93.

[7] Garey, M. R., D. S. Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York.

[8] Hall, N. G., C. N. Potts. 2003. Supply chain scheduling: Batching and delivery. Operations Research, 51, 566-584.

[9] He, Y., W. Zhong, H. Gu. 2006. Improved algorithms for two single

machine scheduling problems. Theoretical Computer Science, 363, 257-265.
[10] Lee, C.-Y., Z.-L. Chen. 2001. Machine scheduling with transportation considerations. Journal of scheduling, 4, 3-24.
[11] Li, K. P., V. K. Ganesan, A. I. Sivakumar. 2005b. Synchronized scheduling of assembly and multi-destination air-transportation in a consumer electronics supply chain. International Journal of Production Research, 43, 2671-2685.
[12] Li, K. P., V. K. Ganesan, A. I. Sivakumar. 2006. Scheduling of single stage assembly with air transportation in a consumer electronic supply chain. Computers & Industrial Engineering, 51, 264-278.
[13] Pundoor, G., Z.-L. Chen. 2005. Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and total distribution cost. Naval Research Logistics, 52, 571-589.
[14] Simchi-Levi, D., 1994. New worst case results for the bin-packing problem. Naval Research Logistics, 41, 579-585.
[15] Stecke, K. E., X. Zhao. 2007. Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. Manufacturing & Service Operations Management, 9, 2, 206-224.
[16] Wang, Q., R. Batta, R. J. Szczerba. 2005. Sequencing the processing of incoming mail to match an outbound truck delivery schedule. Computers & operations research, 32, 1777-1791.
[17] Zhong, W., G. Dosa, Z. Tan. 2007. On the machine scheduling problem with job delivery coordination. European Journal of Operational Research, 182, 1057-1072.

# 科技部補助專題研究計畫出席國際學術會議心得報告

日期：105 年 01 月 17 日

| | | | |
|---|---|---|---|
| 計畫編號 | MOST 104-2221-E-252 -002 -MY2 | | |
| 計畫名稱 | 一些生產與配送整合排程問題之研究 | | |
| 出國人員姓名 | 許　洲　榮 | 服務機構及職稱 | 南開科技大學工業工程與管理系<br>教授 |
| 會議時間 | 105 年 1 月 14 日<br>至<br>105 年 1 月 16 日 | 會議地點 | 泰國曼谷 |
| 會議名稱 | (中文) 2016 年第二屆作業研究與模糊學國際會議<br>(英文) 2016 2nd Conference on Operations Research and Fuzziology (ORF 2016) | | |
| 發表題目 | (中文) 非等效平行機排程問題考慮工作處理時間具有一般截形的工作相依學習效應<br>(英文) Unrelated Parallel-Machine Scheduling Problems with General Truncated Job-Dependent Learning Effect | | |

一、參加會議經過

個人於 1 月 13 日(星期三) 9:30 出發，搭乘高鐵和統聯客運到達桃園國際機場，轉搭中華航空班機(CI-835)13:55(延誤至 14:20)出發，於 18:35(台北時間)到達曼谷蘇汪納蓬國際機場，1 月 14~16 日(星期四至星期六) 報到、參加會議，1 月 17 日(星期日) 返回台灣。

2016 年第二屆作業研究與模糊學國際會議由 Engineering Information Institute、1000 Thinktank 和 Scientific Research Publishing 合辦。我的論文投稿在技術會議(Technical

Sessions)第八主題：數學系列 II，論文編號(80006）。大會也邀請印度卡利亞尼大學(University of Kalyani, India)數學系的比斯瓦斯(Biswas)助理教授( Assistant Professor)進行專題報告：Optimization Under Hybrid Uncertainties。在現實生活中的決策情況，就算決策者有豐富經驗，在決策過程中他往往無法明確表達決策目標的目的，正是由於不僅是隨機的不確定性問題的描述，和不精確參與的目標制定這些問題的模型的過程，其次，參與該模型的參數往往非常的模糊性。許多學科的研究領域它正在成為利益的結果在於在隨機優化，模糊規劃等各類不精確的邊界是熟悉的在混合不確定性的優化。數學模型，包括模糊隨機變數的方法的發展，模糊數與模糊的目標顯示通過考慮不同類型的模糊環境下的概率分佈，研究中也探討一些衍生模型在真實生活中的應用。另一位主講人也是印度卡利亞尼大學(University of Kalyani, India)計算機科學與工程系的 Mukhopadhyay 助理教授：Multiobjective Genetic Algorithms for Clustering: Method and Applications。聚類是一個重要的資料採擷技術，其中一組模式，通常是在多維空間向量，被分組為基於某些相似性或不相似性的標準簇。聚類技術旨在找到輸入資料集的合適的組合，這樣一些標準，例如緊湊、分離和連接了優化。構成集群作為一個優化問題的一個簡單的方法是優化某些集群有效性指標，它反映了群集解決方案的好壞。該資料集的所有可能的分區和所述妥當性指標的相應值定義完整的搜索空間。傳統的劃分聚類技術，如 K-均值和模糊 C-手段，採用了搜索空間貪婪搜索技術來優化簇的緊湊性。這些演算法往往會卡在某個局部最優解依賴於初始聚類中心的選擇。此外，它們優化單個聚類有效性索引（緊湊在這種情況下），並且因此不包括資料集的不同的特性。為了克服局部最的問題，一些進化全域優化的工具，如基因演算法已被廣泛使用，以達到所選擇的有效性度量的全域最佳值。傳統的基於基因演算法的聚類技術使用一定的合理措

施，因為適應值。但是，沒有一個有效性的措施同樣適用於不同類型的資料集。因此，它是天然的同時優化多個這樣的措施，用於捕獲所述資料的不同的特性。多個目標同時優化提供更好的穩健性，以不同的資料屬性。因此，它是利用多目標基因演算法（MOGAS）的集群非常有用。在演講中，首先描述的多目標優化和柏拉圖最優一些準備工作。隨後，將描述一個多目標基於基因演算法的聚類演算法。最後演示多目標基因演算法聚類技術在遠端遙感和生物資訊學的一些應用。

個人此次與會發表之論文主題為：Unrelated Parallel-Machine Scheduling Problems with General Truncated Job-Dependent Learning Effect。在會議中，我針對非等效平行機生產環境、考慮工作處理時間(訂單)具有一般截形的工作相依學習效應排程問題，目標函數為機台總負荷、個別工作總處理時間和個別工作處理時間總差異最小化，證明問題的複雜度可以在多項式時間下求得最佳解，並且提供例子，利用指派問題求解。

二、與會心得

個人此次參加 2016 年第二屆作業研究與模糊學國際會議，茲將個人覺得印象深刻的主題摘錄如後報告：來自美國老道明大學（Old Dominion University）的 Gelareh Bakhtyar 發表：基於所指定的到達時間的時間相依真實網絡，這是一個廣泛應用的運輸問題，一般是利用最短路徑(Shortest path; SP)演算法求解動態網路系統，作者測試先進先出(First in First out; FIFO) 和非先進先出(Non-FIFO) 演算法，測試過程中利用 Backward Dijkstra SP 演算法排除一些不想要的解，以縮短求解時間。另一位來自美國老道明大學（Old Dominion University）的 Duc Nguyen 發表：Efficient generalized inverse for solving simultaneous linear equation。在工程和科學真實世界的應用上，常會遇到同時求解大規模線性方程組的案例，一般都用奇異係數矩陣

法求解小規模問題，作者提出一個效率數值程序求解同時大規模線性方程組，模擬結果顯示這個方法很也效率。中場休息時間，我與來自韓國成均館大學(Sungkyunkwan university)的 Seyed Ahmad Mojallal、以色列阿里埃勒大學(Ariel university) 的 Meir Lewkowicz 和中國中南大學(Central south university)的 Xuli Han 進行學術交流，互留名片，邀約互訪。

三、發表論文全文或摘要

**Parallel machines scheduling problems with general truncated job-dependent learning effect**


**Abstract**

In this paper, we consider scheduling problems with general truncated job-dependent learning effect on unrelated parallel machines. The objective functions are to minimize total machine load, total completion (waiting) time, total absolute differences in completion (waiting) times respectively. If the number of machines is unknown, these problems can be solved in $O\left(n^{m+2}\right)$ time respectively, where $m$ is the number of machines and $n$ is the number of jobs.



**Keywords:** Scheduling; Unrelated parallel machines; Truncated job-dependent learning


<論文全文煩請參看附錄>

**四、建議**

希望國科會能夠多補助學者出席國際學術會議，參加一次會議，開一次眼界，多多少少接觸各國學者，可以促成跨國合作研究的機會。

**五、攜回資料名稱及內容**

個人此次首次參加國際研討會吸取專家學者之新觀念，作為日後研究方向之參考。由於所有投稿論文都是由三位專家學者評審，接受的論文不發行論文集，全部論文由國際應用數學和物理期刊(Journal of Applied Mathematics and Physics)出版[Google-based

Impact Factor: 0.37]，研討會期間都能直接從網路搜尋，因此只帶回一本研討會議程指引（含所有投稿論文的摘要）。

**六、其他**

個人此次順利參加國際研討會，本人再次感謝國科會提供研討會之各項經費上的補助。

**Journal of Applied Mathematics and Physics, 2016, 4, 21-27**
Published Online January 2016 in SciRes. http://www.scirp.org/journal/jamp
http://dx.doi.org/10.4236/jamp.2016.41004

# Unrelated Parallel-Machine Scheduling Problems with General Truncated Job-Dependent Learning Effect

**Jibo Wang[1], Chou-Jung Hsu[2*]**

[1]School of Science, Shenyang Aerospace University, Shenyang, China
[2]Department of Industrial Management, Nan Kai University of Technology, Taiwan
Email: wangjibo75@163.com, *jrsheu@nkut.edu.tw

Received 22 November 2015; accepted 5 January 2016; published 12 January 2016

## Abstract

**In this paper, we consider scheduling problems with general truncated job-dependent learning effect on unrelated parallel-machine. The objective functions are to minimize total machine load, total completion (waiting) time, total absolute differences in completion (waiting) times respectively. If the number of machines is fixed, these problems can be solved in $O(n^{m+2})$ time respectively, where $m$ is the number of machines and $n$ is the number of jobs.**

## Keywords

**Scheduling, Unrelated Parallel Machines, Truncated Job-Dependent Learning**

## 1. Introduction

In modern planning and scheduling problems, there are many real situations where the processing time of jobs may be subject to change due to learning effect. An extensive survey of different scheduling models and problems with learning effects could be found in Biskup [1]. More recently, Janiak *et al.* [2] studied a single processor problem with a S-shaped learning model. They proved that the makespan minimization problem is strongly NP-hard. Lee [3] considered scheduling jobs with general position-based learning curves. For some single machine and a two-machine flowshop scheduling problems, they presented the optimal solution respectively. Lee [4] considered single-machine scheduling jobs with general learning effect and past-sequence-dependent setup time. For some single machine scheduling problems, they presented the optimal solution respectively. Lee and Wu [5], and Wu and Lee [6] considered scheduling jobs with learning effects. They proved that some single machine and flowshop scheduling problems can be solved in polynomial time respectively. Lee *et al.* [7] considered a single-machine scheduling problem with release times and learning effect. Lee *et al.* [8] considered a makespan minimization uniform parallel-machine scheduling problem with position-based learning curves. Lee and Chung [9], Sun *et al.* [10] [11], and Wang *et al.* [12] considered flow shop scheduling with learning effects. Wu *et al.* [13], Wu *et al.* [14], Wu *et al.* [15] and Wang *et al.* [16] considered scheduling problems with the

*Corresponding author.

**How to cite this paper:** Wang, J.B. and Hsu, C.-J. (2016) Unrelated Parallel-Machine Scheduling Problems with General Truncated Job-Dependent Learning Effect. *Journal of Applied Mathematics and Physics*, **4**, 21-27.
http://dx.doi.org/10.4236/jamp.2016.41004

truncated learning effect.

Recently, Wang *et al.* [17] considered several scheduling problems on a single machine with truncated job-dependent learning effect, *i.e.*, the actual processing time of job $J_j$ is $p_{jr}^A = p_j \max\{r^{a_j}, b\}$ if it is scheduled in the *r*th position of a sequence, where $a_j \le 0$ is the job-dependent learning index of job $J_j$, and *b* is a truncation parameter with $0 < b < 1$. In this paper, we study scheduling problems with general truncated job-dependent learning effect on unrelated parallel-machine. The objective is to minimize total machine load, total completion (waiting) time, total absolute differences in completion (waiting) times respectively.

## 2. Problems Description

There are *n* independent jobs $N = \{J_1, J_2, \cdots J_n\}$ to be processed on *m* unrelated paralle-machine $M = \{M_1, M_2, \cdots M_m\}$. Let $(n_1, n_2, \cdots n_m)$ denote a job-allocation vector, where $n_i$ denotes the number of jobs assigned to machine $M_i$, and $\sum_{i=1}^m n_i = n$. In this paper, we assume that the actual processing time of job $J_j$ scheduled on machine $M_i$ is

$$p_{ijr}^A = p_{ij} \max\{f_{ij}(r), b\}, \quad i = 1, 2, \ldots, m;\ r, j = 1, 2, \ldots, n, \tag{1}$$

where $p_{ij} \ge 0$ denotes the normal (basic) processing time of job $J_j$ $(j = 1, 2, \ldots, n)$ on machine $M_i$, $r$ is the position of a sequence, $b$ is a truncation parameter with $0 < b < 1$, $f_{ij}(r)$ is the general case of positional learning for job $J_j$ on machine $M_i$, special $f_{ij}(r) = r^{a_{ij}}$ is the polynomial learning index for job $J_j$ on machine $M_i$ $(a_{ij} < 0)$, $f_{ij}(r) = b_{ij}^{r-1}$ is the exponential learning index for job $J_j$ on machine $M_i$ $(0 < b_{ij} < 1)$.

Let $C_{ij}$ and $W_{ij} = C_{ij} - p_{ij}$ be the completion and waiting time for job $J_j$ on machine $M_i$ respectively. The goal is to determine the jobs assigned to corresponding each machine and the corresponding optimal schedule so that the following objective functions is to be minimized: the total machine load $\sum_{i=1}^m C_{\max}^i$, the total completion (waiting) times $\sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij} \left(\sum_{i=1}^m \sum_{j=1}^{n_i} W_{ij}\right)$, the total absolute differences in completion (waiting) times $\sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|C_{ik} - C_{ij}\right| \left(\sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|W_{ik} - W_{ij}\right|\right)$, where $C_{\max}^i$ denotes the makespan of machine $M_i$. Using the three-field notation [18] the problems can be denoted as $Rm\left|Y\right|Z$, where $Y$ denote the model (1), $Z \in \left\{\sum_{i=1}^m C_{\max}^i, \sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij}, \sum_{i=1}^m \sum_{j=1}^{n_i} W_{ij}, \sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|C_{ik} - C_{ij}\right|, \sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|W_{ik} - W_{ij}\right|\right\}$.

## 3. Main Results

Let $p_{ij}$ denote the actual processing time of a job when it is scheduled in position $j$ on machine $M_i$, then $f_{i[j]}(j)$, $J_{i[j]}$, $C_{i[j]}$, $W_{i[j]}$ are defined similarly.

**Lemma 1.** For a given permutation $\pi_i = \left(J_{i[1]}, J_{i[2]}, \ldots, J_{i[n_i]}\right)$ on machine $M_i$,

$$\sum_{i=1}^m C_{\max}^i = \sum_{i=1}^m \sum_{j=1}^{n_i} p_{i[j]} \max\left\{f_{i[j]}(j), b\right\}$$

$$\sum_{i=1}^m \sum_{j=1}^{n_i} C_{ij} = \sum_{i=1}^m \sum_{j=1}^{n_i} (n_i - j + 1)\, p_{i[j]} \max\left\{f_{i[j]}(j), b\right\}$$

$$\sum_{i=1}^m \sum_{j=1}^{n_i} W_{ij} = \sum_{i=1}^m \sum_{j=1}^{n_i} (n_i - j)\, p_{i[j]} \max\left\{f_{i[j]}(j), b\right\}$$

$$\sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|C_{ik} - C_{ij}\right| = \sum_{i=1}^m \sum_{j=1}^{n_i} (j-1)(n_i - j + 1) p_{i[j]} \max\left\{f_{i[j]}(j), b\right\} \quad \text{(Kanet [19])}$$

$$\sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|W_{ik} - W_{ij}\right| = \sum_{i=1}^m \sum_{j=1}^{n_i} j(n_i - j)\, p_{i[j]} \max\left\{f_{i[j]}(j), b\right\} \quad \text{(Bagchi [20]).}$$

If the vector $(n_1, n_2, \ldots, n_m)$ is given, let $X_{jir}$ be a $0/1$ variable such that $X_{jir} = 1$ if job $J_j\,(j = 1, 2, \ldots, n)$ is assigned at position $r\,(r = 1, 2, \ldots, n_i)$ on machine $M_i\,(i = 1, 2, \ldots, m)$, and $X_{jir} = 0$, otherwise. Then, the problem $Rm\left|Y\right|Z$ (where $\sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|C_{ik} - C_{ij}\right|, \sum_{i=1}^m \sum_{k=1}^{n_i} \sum_{j=k}^{n_i} \left|W_{ik} - W_{ij}\right|$) can be solved by the following assignment problem:

$$\min\ Z=\sum_{i=1}^{m}\sum_{r=1}^{n_i}\sum_{j=1}^{n}\lambda_{ir}p_{ij}\max\left\{f_{ij}(r),b\right\}X_{jir} \tag{2}$$

*s.t.*

$$\sum_{i=1}^{m}\sum_{r=1}^{n_i}X_{jir}=1,\ j=1,2,\ldots,n, \tag{3}$$

$$\sum_{j=1}^{n}X_{jir}=1,\ i=1,2,\ldots,m,\ r=1,2,\ldots,n_i, \tag{4}$$

$$X_{jir}=0\ \text{ or } 1,\quad j=1,2,\ldots,n,\ i=1,2,\ldots,m,\ r=1,2,\ldots,n_i, \tag{5}$$

where $\lambda_{ir}=1$ for $\sum_{i=1}^{m}C_{\max}^{i}$, $\lambda_{ir}=(n_i-r+1)$ for $\sum_{i=1}^{m}\sum_{k=1}^{n_i}C_{ik}$, $\lambda_{ir}=(n_i-r)$ for $\sum_{i=1}^{m}\sum_{k=1}^{n_i}W_{ik}$, $\lambda_{ir}=(r-1)(n_i-r+1)$ for $\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|C_{ik}-C_{ij}\right|$, $\lambda_{ir}=r(n_i-r)$ for $\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|W_{ik}-W_{ij}\right|$.

Now, the question is how many vectors $(n_1,n_2,\ldots,n_m)$ exist. Obviously $n_i$ may be 0, 1, 2, $\cdots$, $n$ $(i=1,2,\ldots,m)$. So if the numbers of jobs assigned to the first $m-1$ machines is given, the number of jobs assigned to the last machine is then determined uniquely ( $\sum_{i=1}^{m}n_i=n$ ). Therefore, the upper bound of $(n_1,n_2,\ldots,n_m)$ is $(n+1)^{m-1}$. Based on the above analysis, we have the following result.

**Theorem 1.** For a given constant $m$, $Rm|Y|Z$ can be solved in $O\left(n^{m+2}\right)$ time, where

$$Z\in\left\{\sum_{i=1}^{m}C_{\max}^{i},\sum_{i=1}^{m}\sum_{j=1}^{n_i}C_{ij},\sum_{i=1}^{m}\sum_{j=1}^{n_i}W_{ij},\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|C_{ik}-C_{ij}\right|,\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|W_{ik}-W_{ij}\right|\right\}.$$

**Proof.** As discussed above, to solve the problem $Rm|Y|Z$, polynomial number (*i.e.*, $(n+1)^{m-1}$) of assignment problems need to be solved. Each assignment problem is solved in $O\left(n^3\right)$ time (by using the Hungarian method). Hence, the time complexity of the problem $Rm|Y|Z$ can be solved in $O\left(n^{m+2}\right)$ time, where

$$Z\in\left\{\sum_{i=1}^{m}C_{\max}^{i},\sum_{i=1}^{m}\sum_{j=1}^{n_i}C_{ij},\sum_{i=1}^{m}\sum_{j=1}^{n_i}W_{ij},\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|C_{ik}-C_{ij}\right|,\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|W_{ik}-W_{ij}\right|\right\}.$$

Note that if the number of machines $m$ is fixed, then the problem $Rm|Y|Z$ can be solved in polynomial time. Based on the above analysis, we can determine the optimal solution for the problem $Rm|Y|Z$ via the following algorithm:

**Algorithm 1**

Step 1. For each possible vector $(n_1,n_2,\ldots,n_m)$, solve the assignment problem (2)-(5). Then, obtain the optimal schedule and the corresponding objective function $Z$.

Step 2. The optimal solution for the problem is the one with the minimum value of the objective function $Z$, where $Z\in\left\{\sum_{i=1}^{m}C_{\max}^{i},\sum_{i=1}^{m}\sum_{j=1}^{n_i}C_{ij},\sum_{i=1}^{m}\sum_{j=1}^{n_i}W_{ij},\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|C_{ik}-C_{ij}\right|,\sum_{i=1}^{m}\sum_{k=1}^{n_i}\sum_{j=k}^{n_i}\left|W_{ik}-W_{ij}\right|\right\}$.

The following example illustrates the working of Algorithm 1 to find the optimal solution for the problem $Rm|Y|\sum_{i=1}^{m}\sum_{j=1}^{n_i}C_{ij}$.

**Example 1.** There are $n=5$ jobs and $f_{ij}(r)=r^{a_{ij}}$, The number of machines is $m=2$ and $p_{11}=15$, $p_{12}=11$, $p_{13}=14$, $p_{14}=3$, $p_{15}=9$, $p_{21}=12$, $p_{22}=10$, $p_{23}=9$, $p_{24}=16$, $p_{25}=8$, $a_{11}=-0.23$, $a_{12}=-0.32$, $a_{13}=-0.25$, $a_{14}=-0.35$, $a_{15}=-0.26$, $a_{21}=-0.32$, $a_{22}=-0.21$, $a_{23}=-0.31$, $a_{24}=-0.24$, $a_{25}=-0.29$, $b=0.7$ are given.

**Solution.** When $n_1=0,\ n_2=5$, the positional weights on machine $M_2$ are $\theta_{21}=5$, $\theta_{22}=4$, $\theta_{23}=3$, $\theta_{24}=2$, $\theta_{25}=1$. Then values $\theta_{ir}p_{ij}\max\left\{r^{a_{ij}},b\right\}$ are given in **Table 1** (the bold value is the optimal solution of the assignment problem (2)-(5)).We solve the assignment problem (2)-(5) to $\sum_{i=1}^{m}\sum_{j=1}^{n_i}C_{ij}=339.65119$.

When $n_1=1,\ n_2=4$, the positional weights on machine $M_1$ and $M_2$ are $\theta_{11}=1$, $\theta_{21}=4$, $\theta_{22}=3$, $\theta_{23}=2$, $\theta_{24}=1$. Then values $\theta_{ir}p_{ij}\max\left\{r^{a_{ij}},b\right\}$ are given in **Table 2**. We solve the assignment problem

**Table 1.** The $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ values of Example 1. for $n_1 = 0, n_2 = 5$.

| $ij \setminus ir$ | $\theta_{21}$ | $\theta_{22}$ | $\theta_{23}$ | $\theta_{24}$ | $\theta_{25}$ |
|---|---|---|---|---|---|
| $J_{21}$ | 60 | 38.45135 | 25.32933 | 16.80000 | **8.40000** |
| $J_{22}$ | 50 | 34.58149 | 23.81912 | **14.94849** | 7.13208 |
| $J_{23}$ | 45 | 29.03910 | **19.20685** | 12.60000 | 6.30000 |
| $J_{24}$ | **90** | 54.19170 | 36.87501 | 22.94328 | 11.20000 |
| $J_{25}$ | 40 | **27.09585** | 18.43750 | 11.20000 | 5.60000 |

**Table 2.** The $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ values of Example 1 for $n_1 = 1, n_2 = 4$.

| $ij \setminus ir$ | $\theta_{11}$ | $\theta_{21}$ | $\theta_{22}$ | $\theta_{23}$ | $\theta_{24}$ |
|---|---|---|---|---|---|
| $J_{i1}$ | 15 | 48 | 28.83852 | 16.88622 | **8.40000** |
| $J_{i2}$ | 11 | 40 | 25.93612 | **15.87942** | 7.13208 |
| $J_{i3}$ | 14 | 36 | **21.77933** | 12.80457 | 6.30000 |
| $J_{i4}$ | **3** | 64 | 40.64377 | 24.58334 | 11.20000 |
| $J_{i5}$ | 9 | **32** | 19.62965 | 11.63469 | 5.60000 |

(2)-(5) to obtain that the optimal schedule on machine $M_1$ is $[J_4]$, and on machine $M_2$ is $[J_5, J_3, J_2, J_1]$. The objective function is $\sum_{i=1}^{m}\sum_{j=1}^{n_i} C_{ij} = 81.05875$.

When $n_1 = 2, n_2 = 3$, the positional weights on machine $M_1$ and $M_2$ are $\theta_{11} = 2$, $\theta_{12} = 1$, $\theta_{21} = 3$, $\theta_{22} = 2$, $\theta_{23} = 1$. Then values $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ are given in **Table 3**. We solve the assignment problem (2)-(5) to obtain that the optimal schedule on machine $M_1$ is $[J_4, J_2]$, and on machine $M_2$ is $[J_5, J_3, J_1]$. The objective function is $\sum_{i=1}^{m}\sum_{j=1}^{n_i} C_{ij} = 61.77443$.

When $n_1 = 3, n_2 = 2$, the positional weights on machine $M_1$ and $M_2$ are $\theta_{11} = 3$, $\theta_{12} = 2$, $\theta_{13} = 1$, $\theta_{21} = 2$, $\theta_{22} = 1$. Then values $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ are given in **Table 4**. We solve the assignment problem (2)-(5) to obtain that the optimal schedule on machine $M_1$ is $[J_4, J_5, J_2]$, and on machine $M_2$ is $[J_3, J_1]$. The objective function is $\sum_{i=1}^{m}\sum_{j=1}^{n_i} C_{ij} = 59.38394$.

When $n_1 = 4, n_2 = 1$, the positional weights on machine $M_1$ and $M_2$ are $\theta_{11} = 4$, $\theta_{12} = 3$, $\theta_{13} = 2$, $\theta_{14} = 1$, $\theta_{21} = 1$. Then values $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ are given in **Table 5**. We solve the assignment problem (2)-(5) to obtain that the optimal schedule on machine $M_1$ is $[J_4, J_5, J_2, J_1]$, and on machine $M_2$ is $[J_3]$. The objective function is $\sum_{i=1}^{m}\sum_{j=1}^{n_i} C_{ij} = 69.93119$.

When $n_1 = 5, n_2 = 0$, the positional weights on machine $M_1$ and are $\theta_{11} = 5$, $\theta_{12} = 4$, $\theta_{13} = 3$, $\theta_{14} = 2$, $\theta_{15} = 1$. Then values $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ are given in **Table 6**. We solve the assignment problem (2)-(5) to obtain that the optimal schedule on machine $M_1$ is $[J_4, J_5, J_2, J_3, J_1]$. The objective function is
$\sum_{i=1}^{m}\sum_{j=1}^{n_i} C_{ij} = 98.58071$.

**Table 3.** The $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ values of Example 1 for $n_1 = 2, n_2 = 3$.

| $ij \setminus ir$ | $\theta_{11}$ | $\theta_{21}$ | $\theta_{22}$ | $\theta_{23}$ | $\theta_{24}$ |
|---|---|---|---|---|---|
| $J_{i1}$ | 30 | 12.78952 | 36 | 19.22568 | **8.44311** |
| $J_{i2}$ | 22 | **8.81177** | 30 | 17.29074 | 7.93971 |
| $J_{i3}$ | 28 | 11.77255 | 27 | **14.51955** | 6.40228 |
| $J_{i4}$ | **6** | 2.35375 | 48 | 27.09585 | 12.29167 |
| $J_{i5}$ | 18 | 7.51579 | **24** | 13.08643 | 5.81735 |

**Table 4.** The $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ values of Example 1 for $n_1 = 3, n_2 = 2$.

| $ij \setminus ir$ | $\theta_{11}$ | $\theta_{21}$ | $\theta_{22}$ | $\theta_{23}$ | $\theta_{24}$ |
|---|---|---|---|---|---|
| $J_{i1}$ | 45 | 25.57905 | 11.65074 | 24 | **9.61284** |
| $J_{i2}$ | 33 | 17.62354 | **7.739517** | 20 | 8.64537 |
| $J_{i3}$ | 42 | 23.54510 | 10.63770 | **18** | 7.25978 |
| $J_{i4}$ | **9** | 4.70751 | 2.10000 | 32 | 13.54792 |
| $J_{i5}$ | 27 | **15.03158** | 6.76380 | 16 | 6.54322 |

**Table 5.** The $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ values of Example 1 for $n_1 = 4, n_2 = 1$.

| $ij \setminus ir$ | $\theta_{11}$ | $\theta_{21}$ | $\theta_{22}$ | $\theta_{23}$ | $\theta_{24}$ |
|---|---|---|---|---|---|
| $J_{i1}$ | 60 | 38.36857 | 23.30147 | **10.90479** | 12 |
| $J_{i2}$ | 44 | 26.43531 | **15.47903** | 7.70000 | 10 |
| $J_{i3}$ | 56 | 35.31765 | 21.2754 | 9.89950 | **9** |
| $J_{i4}$ | **12** | 7.06126 | 4.20000 | 2.10000 | 16 |
| $J_{i5}$ | 36 | **22.54737** | 13.52761 | 6.30000 | 8 |

**Table 6.** The $\theta_{ir} p_{ij} \max\left\{r^{a_{ij}}, b\right\}$ values of Example 1 for $n_1 = 5, n_2 = 0$.

| $ij \setminus ir$ | $\theta_{21}$ | $\theta_{22}$ | $\theta_{23}$ | $\theta_{24}$ | $\theta_{25}$ |
|---|---|---|---|---|---|
| $J_{11}$ | 75 | 51.15809 | 34.95221 | 21.80959 | **10.50000** |
| $J_{12}$ | 55 | 35.24707 | **23.21855** | 15.40000 | 7.70000 |
| $J_{13}$ | 70 | 47.09020 | 31.91310 | **19.79899** | 9.80000 |
| $J_{14}$ | **15** | 9.41501 | 6.30000 | 4.20000 | 2.10000 |
| $J_{15}$ | 45 | **30.06317** | 20.29141 | 12.60000 | 6.30000 |

Hence, the optimal schedule on machine $M_1$ is $[J_4, J_5, J_2]$, and on machine $M_2$ is $[J_3, J_1]$. The optimal objective function is $\sum_{i=1}^{m}\sum_{j=1}^{n_i} C_{ij} = 59.38394$.

## Acknowledgements

Hsu was supported by the Ministry Science and Technology of Taiwan under Grant MOST 104-2221-E-252-002-MY2.

## References

[1] Biskup, D. (2008) A State-of-the-Art Review on Scheduling with Learning Effects. *European Journal of Operational Research*, **188**, 315-329. http://www.sciencedirect.com/science/article/pii/S0377221707005280 http://dx.doi.org/10.1016/j.ejor.2007.05.040

[2] Janiak, A., Janiak, W., Rudek, R. and Wielgus, A. (2009) Solution Algorithms for the Makespan Minimization Problem with the General Learning Model. *Computers and Industrial Engineering*, **56**, 1301-1308. http://www.sciencedirect.com/science/article/pii/S0360835208001654 http://dx.doi.org/10.1016/j.cie.2008.07.019

[3] Lee, W.-C. (2011) Scheduling with General Position-Based Learning Curves. *Information Sciences*, **181**, 5515-5522. http://www.sciencedirect.com/science/article/pii/S0020025511004130 http://dx.doi.org/10.1016/j.ins.2011.07.051

[4] Lee, W.-C. (2011) A note on Single-Machine Scheduling with General Learning Effect and Past-Sequence-Dependent Setup Time. *Computers and Mathematics with Applications*, **62**, 2095-2100. http://www.sciencedirect.com/science/article/pii/S0898122111005384 http://dx.doi.org/10.1016/j.camwa.2011.06.057

[5] Lee, W.-C. and Wu, C.-C. (2009) Some Single-Machine and m-Machine Flowshop Scheduling Problems with Learning Considerations. *Information Sciences*, **179**, 3885-3892. http://www.sciencedirect.com/science/article/pii/S0020025509003235 http://dx.doi.org/10.1016/j.ins.2009.07.011

[6] Wu, C.-C. and Lee, W.-C. (2009) Single-Machine and Flowshop Scheduling with a General Learning Effect Model. *Computers & Industrial Engineering*, **56**, 1553-1558. http://www.sciencedirect.com/science/article/pii/S036083520800260X http://dx.doi.org/10.1016/j.cie.2008.10.002

[7] Lee, W.-C., Wu, C.-C. and Hsu, P.-H. (2010) A Single-Machine Learning Effect Scheduling Problem with Release Times. *Omega—The International Journal of Management Science*, **38**, 3-11. http://www.sciencedirect.com/science/article/pii/S0305048309000024 http://dx.doi.org/10.1016/j.omega.2009.01.001

[8] Lee, W.-C., Yeh, W.-C. and Chuang, M.C. (2012) Uniform Parallel-Machine Scheduling to Minimize Makespan with Position-Based Learning Curves. *Computers & Industrial Engineering*, **63**, 813-818. http://www.sciencedirect.com/science/article/pii/S0360835212001283 http://dx.doi.org/10.1016/j.cie.2012.05.003

[9] Lee, W.-C. and Chung, Y.-H. (2013) Permutation Flowshop Scheduling to Minimize the Total Tardiness with Learning Effects. *International Journal of Production Economics*, **141**, 327-334. http://www.sciencedirect.com/science/article/pii/S0925527312003623 http://dx.doi.org/10.1016/j.ijpe.2012.08.014

[10] Sun, L.-H., Cui, K., Chen, J.-H., Wang, J. and He, X.-C. (2013) Research on Permutation Flow Shop Scheduling Problems with General Position-Dependent Learning Effects. *Annals of Operations Research*, **211**, 473-480. http://link.springer.com/article/10.1007/s10479-013-1481-6 http://dx.doi.org/10.1007/s10479-013-1481-6

[11] Sun, L.-H., Cui, K., Chen, J.-H., Wang, J. and He, X.-C. (2013) Some Results of the Worst-Case Analysis for Flow Shop Scheduling with a Learning Effect. *Annals of Operations Research*, **211**, 481-490. http://link.springer.com/article/10.1007/s10479-013-1368-6 http://dx.doi.org/10.1007/s10479-013-1368-6

[12] Wang, X.-Y., Zhou, Z., Zhang, X., Ji, P. and Wang, J.-B. (2013) Several Flow Shop Scheduling Problems with Truncated Position-Based Learning Effect. *Computers & Operations Research*, **40**, 2906-2929. http://www.sciencedirect.com/science/article/pii/S0305054813001743 http://dx.doi.org/10.1016/j.cor.2013.07.001

[13] Wu, C.-C., Yin, Y. and Cheng, S.-R. (2011) Some Single-Machine Scheduling Problems with a Truncation Learning Effect. *Computers & Industrial Engineering*, **60**, 790-795.
http://www.sciencedirect.com/science/article/pii/S0360835211000362
http://dx.doi.org/10.1016/j.cie.2011.01.016

[14] Wu, C.-C., Yin, Y. and Cheng, S.-R. (2013) Single-Machine and Two-Machine Flowshop Scheduling Problems with Truncated Position-Based Learning Functions. *Journal of the Operation Research Society*, **64**, 147-156.
http://www.palgrave-journals.com/jors/journal/v64/n1/abs/jors201246a.html
http://dx.doi.org/10.1057/jors.2012.46

[15] Wu, C.-C., Yin, Y., Wu, W.-H. and Cheng, S.-R. (2012) Some Polynomial Solvable Single-Machine Scheduling Problems with a Truncation Sum-of-Processing-Times Based Learning Effect. *European Journal of Industrial Engineering*, **6**, 441-453. http://www.inderscienceonline.com/doi/abs/10.1504/EJIE.2012.047665
http://dx.doi.org/10.1504/ejie.2012.047665

[16] Wang, J.-B., Wang, X.-Y., Sun, L.-H. and Sun, L.-Y. (2013) Scheduling Jobs with Truncated Exponential Learning Functions. *Optimization Letters*, **7**, 1857-1873. http://link.springer.com/article/10.1007/s11590-011-0433-9
http://dx.doi.org/10.1007/s11590-011-0433-9

[17] Wang, X.-R., Wang, J.-B., Jin, J. and Ji, P. (2014) Single Machine Scheduling with Truncated Job-Dependent Learning Effect. *Optimization Letters*, **8**, 669-677. http://link.springer.com/article/10.1007/s11590-012-0579-0
http://dx.doi.org/10.1007/s11590-012-0579-0

[18] Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G. (1979) Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, **5**, 287-326.
http://www.sciencedirect.com/science/article/pii/S016750600870356X
http://dx.doi.org/10.1016/S0167-5060(08)70356-X

[19] Kanet, J.J. (1981) Minimizing Variation of Flow Time in Single Machine Systems. *Management Science*, **27**, 1453-1459. http://pubsonline.informs.org/doi/abs/10.1287/mnsc.27.12.1453
http://dx.doi.org/10.1287/mnsc.27.12.1453

[20] Bagchi, U.B. (1989) Simultaneous Minimization of Mean and Variation of Flow-Time and Waiting Time in Single Machine Systems. *Operations Research*, **37**, 118-125. http://pubsonline.informs.org/doi/abs/10.1287/opre.37.1.118
http://dx.doi.org/10.1287/opre.37.1.118

# 科技部補助計畫衍生研發成果推廣資料表

日期:2016/05/30

| | |
|---|---|
| 科技部補助計畫 | 計畫名稱: 一些生產與配送整合排程問題之研究 |
| | 計畫主持人: 許洲榮 |
| | 計畫編號: 104-2221-E-252-002-MY2　　學門領域: 作業研究 |

無研發成果推廣資料

# 104年度專題研究計畫研究成果彙整表

| **計畫主持人：**許洲榮 | | | **計畫編號：**104-2221-E-252-002-MY2 | | | | |
|---|---|---|---|---|---|---|---|
| **計畫名稱：**一些生產與配送整合排程問題之研究 | | | | | | | |
| 成果項目 | | | 量化 | | | 單位 | 備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等） |
| | | | 實際已達成數（被接受或已發表） | 預期總達成數（含實際已達成數） | 本計畫實際貢獻百分比 | | |
| 國內 | 論文著作 | 期刊論文 | 0 | 0 | 100% | 篇 | |
| | | 研究報告/技術報告 | 0 | 0 | 100% | | |
| | | 研討會論文 | 0 | 0 | 100% | | |
| | | 專書 | 0 | 0 | 100% | 章/本 | |
| | 專利 | 申請中件數 | 0 | 0 | 100% | 件 | |
| | | 已獲得件數 | 0 | 0 | 100% | | |
| | 技術移轉 | 件數 | 0 | 0 | 100% | 件 | |
| | | 權利金 | 0 | 0 | 100% | 千元 | |
| | 參與計畫人力（本國籍） | 碩士生 | 1 | 1 | 100% | 人次 | |
| | | 博士生 | 0 | 0 | 100% | | |
| | | 博士後研究員 | 0 | 0 | 100% | | |
| | | 專任助理 | 0 | 0 | 100% | | |
| 國外 | 論文著作 | 期刊論文 | 1 | 0 | 100% | 篇 | |
| | | 研究報告/技術報告 | 0 | 0 | 100% | | |
| | | 研討會論文 | 1 | 0 | 100% | | |
| | | 專書 | 0 | 0 | 100% | 章/本 | |
| | 專利 | 申請中件數 | 0 | 0 | 100% | 件 | |
| | | 已獲得件數 | 0 | 0 | 100% | | |
| | 技術移轉 | 件數 | 0 | 0 | 100% | 件 | |
| | | 權利金 | 0 | 0 | 100% | 千元 | |
| | 參與計畫人力（外國籍） | 碩士生 | 0 | 0 | 100% | 人次 | |
| | | 博士生 | 0 | 0 | 100% | | |
| | | 博士後研究員 | 0 | 0 | 100% | | |
| | | 專任助理 | 0 | 0 | 100% | | |
| 其他成果（無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。） | | 針對研究問題，我們分析各問題的計算複雜度，對於P問題，提供多項式演算法，對於NP-hard問題，則依其複雜度提供動態規劃演算法、分枝界限演算法、啟發式演算法或近似演算法。這類問題的研究一方面可以豐富排程問題理論以及組合最佳化的理論成果，另一方面可以為解決實際問題提供切實可行的方法。 | | | | | |

| | 成果項目 | 量化 | 名稱或內容性質簡述 |
|---|---|---|---|
| 科教處計畫加填項目 | 測驗工具(含質性與量性) | 0 | |
| | 課程/模組 | 0 | |
| | 電腦及網路系統或工具 | 0 | |
| | 教材 | 0 | |
| | 舉辦之活動/競賽 | 0 | |
| | 研討會/工作坊 | 0 | |
| | 電子報、網站 | 0 | |
| | 計畫成果推廣之參與（閱聽）人數 | 0 | |

# 科技部補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估
   ■達成目標
   □未達成目標（請說明，以100字為限）
   　　□實驗失敗
   　　□因故實驗中斷
   　　□其他原因
   說明：

2. 研究成果在學術期刊發表或申請專利等情形：
   論文：□已發表 □未發表之文稿 ■撰寫中 □無
   專利：□已獲得 □申請中 ■無
   技轉：□已技轉 □洽談中 ■無
   其他：（以100字為限）
   第一年度計劃執行期間預定解決之問題已經完全解決，並且撰寫成研究論文，詳細呈現在本報告中的第二和第三節中。
   第一年度計劃執行期間，已經發表兩篇註記感謝科技部補助的編號之期刊論文，以及一篇已修改(revised)論文，詳列於期中結案報告中。

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以500字為限）
   本計劃主要研究生產與配送整合的排程問題，是兩年期的研究計劃，探討三個不同的生產與配送模式對應三個研究目標函數，總共預定解決九個生產排程問題。第一年已經解決六個生產與配送整合的排程問題，撰寫成研究論文，詳細呈現在本報告中的第二和第三節，我們分析各問題的計算複雜度，對於P問題，提供多項式演算法，對於NP-hard問題，則依其複雜度提供動態規劃演算法、分枝界限演算法、啟發式演算法或近似演算法。這類問題的研究一方面可以豐富排程問題理論成果，另一方面可以為解決實際問題提供切實可行的方法。
   　　第一年度計劃執行期間，已經發表兩篇註記感謝科技部補助的編號之期刊論文，以及一篇已修改(revised)論文，詳列如下：
   1. Note on a unified approach to the single-machine scheduling problem with a deterioration effect and convex resource allocation, Journal of Manufacturing Systems [SCI; IF: 1.682], 38, pp.134-140. MOST 104-2221-E-252-002-MY2.
   2. Unrelated Parallel-Machine Scheduling Problems with General Truncated Job-Dependent Learning Effect, Journal of Applied Mathematics and Physics. [Google-based Impact Factor: 0.37], 4, pp.

21-27. MOST 104-2221-E-252-002-MY2.
3. Scheduling deteriorating jobs with machine availability constraints to minimize the total completion time. Journal of Industrial and Production Engineering- Manuscript ID TJCI-2015-0009.R1

計劃執行期間，我們也培育、訓練了一位碩士班研究生和一位大學部學生，碩士班研究助理經過兩年的訓練，畢業後可以直接投入職場從事生產管理工作，亦可繼續就讀博士班，從事排程理論探討或應用研究。另一位大學部研究助理已經推甄錄取雲林科技大學碩士班。

整體而言，第一年研究計劃大致有達到預定目標，主持人由衷的感謝科技部的補助，讓我們可以充實實驗室設備和資源，以及參加國際研討會，拓展國際視野。