東　　海　　大　　學

工業工程與經營資訊研究所

碩士論文

應用新的研究方法探討有提早/延遲懲罰的空間排程問題

研 究 生：歐陽敬倫
指導教授：彭泉　博士
賴奕銓 博士

中 華 民 國 九 十 七 年 六 月

Process Application of New Approaches to Space Scheduling Problems with Early and Tardy Penalty

By
Chin-Lun Ouyang

Advisor: Dr. Chyuang Perng
Dr. Yi-Chiuan Lai

A Thesis
Submitted to the Institute of Industrial Engineering and Enterprise Information at Tunghai University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in
Industrial Engineering and Enterprise Information

June 2008

Taichung, Taiwan , Republic of China

應用新的研究方法探討有提早/延遲懲罰的空間排程問題

學生：歐陽敬倫　　　　　　　　　　　　　　指導教授：彭泉　博士
賴奕銓　博士

東海大學工業工程與經營資訊研究所

摘要

在有限的空間限制下，空間排成問題對於高科技組裝廠而言是個複雜且重要的議題，因為空間排程會影響整個工作產能的效益。為了能夠有效的了提高工作產能，把用有限空間的資源最佳化利用是必要的。雖然有許多研究針對工作排程問題進行討論，但是對於空間排程問題這領域的研究卻是很缺乏的。

在本研究中，我們提出兩個主題來進行討論，這兩主題的共同目標就是尋找在不同績效標準下最適合的排程法則，例如總延遲成本、製程時間和空間利用率。在本研究的第一個主題，我們提出一個新的空間搜尋法則，也就是最大接觸法，來和西北角法，運用不同的派工法則比較各項績效指標。由於在先前的空間排程問題研究上，並沒有討論提早懲罰和延遲懲罰對空間排程問題的效果和影響，因此我們會再地二個主題探討此議題。

從 OR library 和台灣的企業蒐集的資料，經過我們用統計方法分析和電腦程式運算後得到的結果，發現運用最大接觸法取得的績效指標優於西北角法，而且在各項績效指標中表現較好的派工法則也在我們的研究中得到。

關鍵字詞：排程法則、提早懲罰和延遲懲罰、排程、空間限制、空間排程問題

Application of New Approaches to Space Scheduling Problems with Early and Tardy Penalty

Student: Chin-Lun Ouyang   Advisor: Dr. Chyuang Perng
Dr. Yi-Chiuan Lai

Department of Industrial Engineering and Enterprise Information
Tunghai University

ABSTRACT

Space scheduling problem with limited space constraint is a complicated and important issue in high-tech assembly factory, it will affect the production capacity of jobs. In order to effectively use the space capacity, we have to arrange the sequence of jobs so as to maximize space utilization. Although there are many studies in job scheduling and space allocation problems, there are not many researches regarding space scheduling problems.

Two topics were discussed in this study, both aimed to find a better dispatching rule for different performance measurements such as total tardiness, makespan and space utilization. In the First topic, we develop a new space allocation algorithm, namely, Longest Contact Edge Algorithm (LCEA) and compare the performance indicators with the Northwest Algorithm (NWA) by different dispatching rules. We take early and tardy penalty into consideration in space scheduling problem in the second topic, due to the effect of early and tardy penalty did not be considered in previous studies.

After statistic approach and computation results of data from OR library and companies in Taiwan, we found that LCEA is more efficient than NWA for obtaining better schedules, and the preferable dispatching rule for each performance measurements is developed.

Keywords: dispatching rules, early and tardy penalty, scheduling, space constraint, space scheduling problem

誌謝

本論文能夠順利的完成，感謝的人很多。其中最要感謝的莫過於指導老師彭泉老師和賴奕銓老師，除了在專業知識、論文寫作和研究方法的傳授外，更以許多個人的經驗和智慧，引導我在生活上或處事上建立更成熟的態度和開闊的人生觀，使我在各方面受益良多。同時也要感謝研究室的各個老師，蔡禎騰老師、林水順老師、邱文志老師和邱創鈞老師在口試期間和研究是會議期間給予論文上的意見和指導。

另外感謝何子平學長和眾多學長姐在這段期間的關心和照顧，對於論文的結構和文法上提供許多建議，以及程式撰寫方面的指導和協助本研究資料的取得和蒐集，有了這些我幫助，我才能完成龐大數據的演算和分析取得實驗結果。還有系上助理的素卿、玉玲和宏華感謝你們對我種種的協助和照顧。另依方面，研究室的諸位同學們，尚琳、嘉南、志宇、宗億和政憲以及與學弟妹們，鉉文、任志、美瑜、玠昀和珈綸對我的協助和支持讓我感受到在這裡的懷念和歡樂的氣氛，帶給我許多的回憶。

最後，謹以這份研究獻給我的父母和爺爺。在這段求學生涯中，給予不斷的支持和鼓勵，讓我無後顧之憂地完成學業，沒有你們就沒有我今日的成就。

歐陽敬倫　謹誌於

東海大學工業工程與經營資訊研究所

中華民國九十七年六月

Table of Contents

List of Tables

List of Figures

Chapter 1 General Introduction

In recent years, high-tech industries play important roles in Taiwan. These high-tech industries, such as TFT-LCD (Thin Film Transistor-Liquid Crystal Display) and semiconductor manufacturers need clean room and expensive equipment to produce products on the shop floor. The equipment for such industries is huge and expensive and required to be manufactured in a certain level of cleanness. Therefore, such machinery industry needs certain amount of space to assembly machines. However, limited and expensive land resource is a big constraint for such high-tech industries. On the other hand, it is an expensive cost to broaden the shop floor scale of the shop floor. In order to increase the productivity, it is necessary to make efficient use of the space on the shop floor.

A job scheduling problem with space resource constraints is one of resource constraint scheduling problems. Perng *et al*. (2007, 2008a, 2008b) defined it as a space scheduling problem. The machine assembly process requires a certain amount of complete space on the shop floor in the factory for a period of time. The sizes of the shop floor and machines will determine the number of machines which can be assembled at the same time. The sequence of jobs and allocation of jobs in a factory will affect the performance of the shop floor. If there are too many fragmentary free spaces in a factory, it will decrease the production capacity since the manufacturing process needs a complete space to assembly the machine. On the other hand, because each machine has its own space requirements based on its shape, the shop floor needs to be scanned before each new arrival order can be allocated to the shop floor. If the factory is at capacity (i.e. no more free space or the job is too big to fit in the available space), the new order must wait for the space taken by existing jobs to complete the assembly. As shown in Figure 1.1, job D2 can’t be assigned into the shop floor due to limited space and it will be on the queue to wait for other jobs on the shop floor to be done. Figure 1.2shows that job A3 is done and there is enough space to fit it job D2.

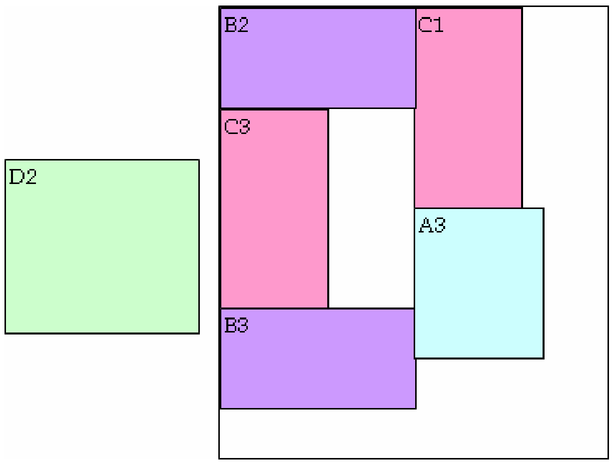


Figure 1.1 Space constraints

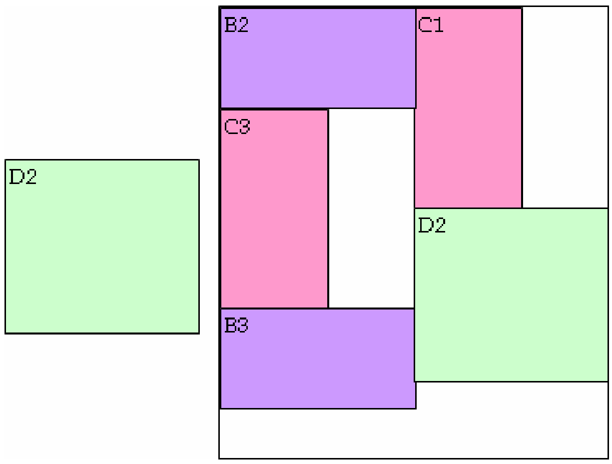


Figure 1.2 Allocation of a new job in the sequence

Obstacles such as pillars or resting area on the shop floor will also influence allocations of jobs. Figure 1.3 is an example of a shop floor with obstacles. This will result in different performances of utilization of shop floor space. Figure 1.4 demonstrates that obstacles are presented on the shop floor and the areas with obstacles can't be assigned to any job. On the other hand, the entire shop floor space can be used in Figure 1.5.

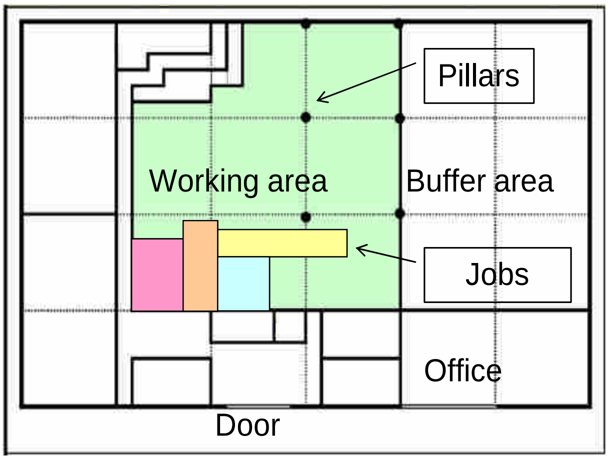


Figure 1.3 Obstacles and jobs within the shop floor

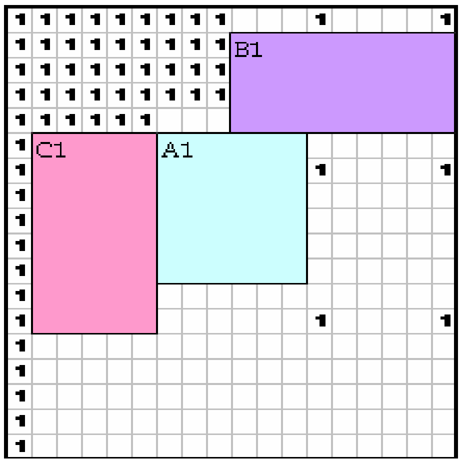


Figure 1.4 Space allocations with obstacles (1 stands for obstacles)

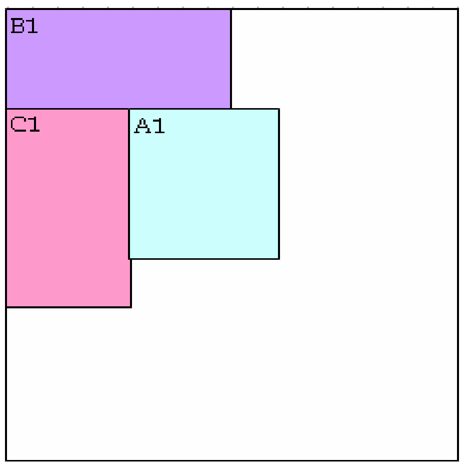


Figure 1.5 Space allocations without obstacles

Two topics were discussed in this study. Both topics aimed to find a better dispatching rule for different performance measurements such as total tardiness, makespan and space utilization. We develop a new space allocation algorithm, namely, Longest Contact Edge Algorithm (LCEA) to compare the performance indicators (Makespan, Total Tardiness and Space Utilization) with the Northwest Algorithm (NWA) in previous studies of space scheduling problem under the same conditions and restrictions, by different dispatching rules (Shortest Processing Time, Longest Processing Time, First Come First Serve, Earliest Due Date, Smallest Space Requirement and largest Space Requirement). We also take early and tardy penalty into consideration in space scheduling problem.

Chapter 2 A New Space Allocation Algorithm for Space Scheduling Problem

2.1 Introduction

Space resource constraint is an important issue in space scheduling problems. In order to effectively use the space capacity, we have to arrange the sequence of jobs so as to maximize space utilization. In this study, we focus on the effect of space allocation algorithm to space scheduling problem. Due to the NWA is the only space allocation algorithm used in previous studies of space scheduling problems, we develop a new space allocation algorithm, namely, Longest Contact Edge Algorithm (LCEA) and aim to find better space allocation under different performance measurements.

2.2 Literature review

2.2.1 Scheduling problems

Production scheduling has been widely studied in many researches. It is a decision making process that plays an important rule in many industries or services. Pinedo (2002) defines the goal of production scheduling is to maximize the efficiency of the operation and reduce costs. It is also a great tool to optimize use of available resources such as machines, labor, material and time. Johnson (1954) was the first to study one machine scheduling problems. He allocated limited machine resources to process the tasks, and first to find optimal solutions for a two-machine flow shop problem where each job has no more than two operations. Backer (1990) defines scheduling as a management of resources to complete jobs over time.

The main purpose of production scheduling is to minimize the cost from production activity. By solving scheduling problems, dispatching rules play an important role in determining the sequence of jobs. Rajendran (1997) stated that the decision to which job is to be loaded on a machine, when the machine becomes free, is normally made with the help of dispatching rules. He also categorized dispatching rules into following classifications: 1. Process-time based rules, such as SPT (shortest process-time rule) and LPT (longest process-time rule); 2. Due-date based rules, such as EDD (earliest due date); 3. Combination rules, such as least slack rule and critical ratio; 4. Rules that are neither process-time based nor due-date based, such as WINQ rule (total work-content of jobs in the queue of next operation of a job). Sule (1996) mentioned that EDD and LPT can reduce maximum lateness when dealing with single-machine scheduling and multi-machine scheduling problems, respectively. Panwalker (1997) classified dispatching rules as priority rules, heuristic scheduling rules and others. As for priority rules, it can be divided into simple priority rules, combination of simple priority rules and weighted priority indexes. Chryssolou and Subramaniam (2001) used genetic algorithm (GA) for job shop scheduling problems. Rajendran and Ziegler (2001) investigated the performance of dispatching rules and a heuristic for scheduling in static flow shops with missing operations. Pugazhendhi (2004) stated that the performance of a dispatching rule will be influenced by various parameters such as the utilization level of shop floor and allowance factor, and that no single rule has been found to be the best under all conditions. Muzrak and Bayhan (2006) classified the dispatching rules for job shop scheduling problems. The list of rules were FCFS (first come first serve), SPT (shortest processing time), WSPT (weighted SPT), WLWKR (weighted least work remaining), EDD (earliest due date), MDD (modified due date), SLACK (least slack), CR (critical ratio), S/OPN (slack per remaining operation), MDSPRO (modified slack per remaining operation), S/RPT (slack per remaining processing time), ODD (operation due date), OSLACK (operation slack), OCR (operation critical ratio), ATC (apparent tardiness cost), COVERT (cost over time), SB (shifting bottleneck) and WINQ (work in next queue). Hung and Chang (2002) applied the dispatching rule, the random rule, in a semiconductor industry. Rhee *et al.* (2004) used PERT to workflow management system. They discovered PERT was better than SPT or FCFS dispatching rules.

2.2.2 Space allocation problem

In this study, we focused on scheduling problems with limited space capacity. The size of the job and the space available on the shop flow will determine how the job will be allocated. It is similar to the DLP (dynamic layout problem) and to other many researches, namely, circuit board design; layout design of hospitals, schools, and airports; warehouses; storages. Erel *et al*. (2003) used enumerative heuristic to reduce the total flow cost when arranging equipments in the factory. Balakrishnan *et al.* (2003) applied the technique of the dynamic plant layout problem (DPLP) to deal with the design of MBRLP (multi-period layout plans). Dunker *et al.* (2005) presented an algorithm combining dynamic programming and genetic search for solving a dynamic facility layout problem. Mckendall *et al.* (2006) proposed that manufacturing plants must be able to operate efficiently and respond quickly to changes in product mix and demand. They considered the problem of arranging and rearranging (when there were changes in the flows of materials between departments) manufacturing facilities such that the sum of the material handling and rearrangement costs were minimized. In their paper, simulated annealing (SA) were employed for the DLP. Erel *et al*. (2003) used the dynamic layout problem to address the situation where the traffic among the various units within a facility changes over time. Its objective was to determine a layout for each period in a planning horizon such that the total flow and the relocation costs were minimized. Perng *et al.* (2007) developed two space related dispatching rules, namely, small space requirement (SSR) first and large space requirement (LSR) first, to the space resource constrained problem. Perng *et al.* (2008a) proposed a space scheduling problem with obstacles. In a space scheduling problem, certain amounts of obstacles such as pillars and office areas were presented on the shop floor. Perng *et al.* (2008b) applied container loading problem (CLP) heuristics into a space scheduling problem.

As mentioned previously, in order to increase the production capacity, it is necessary to manage the limited space resource of the shop floor. To effectively manage the shop floor, the space on the shop floor could be divided into several chunks and some jobs could be allocated for simultaneous assembly. Hegazy (1999) purposed that space planning should divide into equal chunks; each chunk is a unit square of the space. In this study, the shop floor and jobs are divided into equal chunks; each chunk is a unit of grid. Kelley (1960) set the division points of sheets as a previous cutting plane problem. Each grid represented a unit of working space requirement, and an order was completed on working space which composes of these grids. This method was also called a set of grid square (Egeblad *et al.*, 2007).

After dividing the shop floor and jobs to several chunks, a sequence of orders are then allocated to the shop floor. The sequence of new orders has to be decided. The factory has to offer a complete space to allocate the new order based on its shape. After some orders were assigned into the shop flow, the remaining space on the shop floor will become irregular shapes. To efficiently use the space on the shop floor, we will try to fit in as many orders as possible. This is similar to bin packing, nesting and knapsack problems. Dantzig (1957) considered a hitch-hiker has to fill up his knapsack by selecting from among various possible objects those which will give him maximum comfort. Johnson (1973) determined how to put the many objects in the least number of fixed space bins. Chow (1979) dealing with nesting problem, a big piece must be divided into smaller irregular shaped pieces, minimizing waste. Gomes and Oliveira (2002) solved nesting problems based on a 2-exchange neighborhood search. Tay *et al.* (2002) presented a new method of solving the pattern nesting problem on irregular-shaped stock using genetic algorithms. Bischoff (2006) proposed a new heuristic to a 3-D bin packing problem. The results demonstrated that it outperformed previous literatures. Sciomachen and Tanfani (2007) surveyed the problem of determining stowage plans for containers in a ship. Egeblad *et al*. (2007) used fast neighborhood search for 2D and 3D dimensional nesting problems. Yang and Lin (2007) applied genetic algorithms to shoe making nesting. They tried to make shoes with the minimum waste of raw materials. Lee *et al*. (2008) transformed irregular shapes to polygons. Their results showed that a quick location and movement algorithm (QLMA) took less time to calculate a layout and the space utilization outperformed previous literature’s methods.

The goal of bin packing is to pack a collection of objects into the minimum number of fixed-size bins. Nesting is the process of efficiently manufacturing parts from flat raw material while the knapsack problem aims to maximize the best choice of essentials that can fit into one bag to be carried on a trip. However, there are still some differences between our study of space scheduling and bin packing, nesting and knapsack problem. We list two major differences: One of the differences is that space scheduling problem considers duration of time condition. Under this condition, the space location of orders on the shop floor is related to the operation time of each job.

The other difference is that the purpose of the space scheduling problem is to determine the sequence of jobs in order to optimize performance measurements instead of only choosing objects to optimize utilization.

2.3 Problem Formulation

Let N denote a set of n jobs. Let $s_1, s_2, ..., s_n$ denote the start time for each job. Let $t_1, t_2, ..., t_n$ denote the processing time requirement for each job. Let $d_1, d_2, ..., d_n$ denote the due date specified by customers for each job. Let σ denote an arbitrary sequence. All possible permutations are [1-2-…-(n-1)-n], [1-2-…-n-(n-1)],…, [n-(n-1)-…-1]. There have n! possibilities. The traditional dispatching rule, such as SPT, FCFS, or EDD rule, is one of the permutation results. For a job *i*, a tardiness will be denoted by T_i. The summary of notations is shown as follows:

N : a set of n jobs

s_i : start time for each job *i*

t_i : processing time requirement for each job *i*

d_i : due date specified by customers for each job *i*

σ : an arbitrary sequence

T_i : tardiness for job *i*

P : unit tardiness penalty

$f(\sigma)$: tardy cost of σ sequence

$$T_i = \text{Max}\{s_i + t_i - d_i, 0\}$$

The objective is to find a sequence to minimize the tardy cost.

$$\textbf{Min} \quad f(\sigma) = P\sum_{i=1}^{n} T_i \tag{1}$$

In the cost function $f(\sigma)$, P denotes the unit tardiness penalty. This research assumes that if a job *i* has been completed early, the job has to stay in the factory until due date which is specified by customers arrives. Thus, earliness will not occur.

Let C_j denote completion time for each job *j*. Let $C_{\max}$ denote the makespan. Thus,

$$C_j = s_j + p_j$$

$$C_{\max} = \text{Max}\left\{C_j\right\}$$

$f_2(\sigma)$: makespan of σ sequence

The objective (2) is to find a sequence to minimize the makespan.

Min $f_2(\sigma) = C_{\max}$ (2)

Further, let S_t denote the space utilization of date *t* and S denote the average space utilization.

l : length of the working place

w : width of the working place

T : total time of the operation

s_i : start time for each job *i*

sp_i : shipping date for each job *i*

a_i : length of each job *i*

b_i : width of each job *i*

α_i : is a Boolean variable of each job *i*

$$\alpha_i = \begin{cases} 1 & \text{if } s_i \le t \le sp_i \\ 0 & \text{other} \end{cases}$$

$$S_t = \frac{\sum_{i=1}^{n} \alpha_i a_i b_i}{(l \times w - obstacle)}$$

$$S = \frac{\sum_{t=1}^{T} S_t}{T}$$

$f_3(\sigma)$: space utilization of σ sequence

The objective (3) is to find a sequence to maximize the space utilization of all jobs.

Max $f_3(\sigma) = S$ (3)

2.4 Longest Contact Edge Algorithm

The Northwest algorithm is a basic space allocation approach using in Perng *et al*. (2007). It was used to allocate jobs into a factory. In this study, we attend to propose a new space allocation algorithm, which differ from the northwest approach. The flow chart of longest contact edge algorithm is shown in Figure 2.1.

The space allocation procedure of the longest contact edge algorithm is almost the same as Perng *et al.* (2007). This study extended northwest algorithm (NWA) into Longest Contact Edge algorithm (LCEA). The difference between LCEA and NWA is that LCEA includes the steps of evaluation of every candidate working area. The procedure of LECA is as follows:

Step1: Search free of reference point

In a factory, the program search grid (1,1), grid (1,2),…, grid (1,j), grid (2,1), grid (2,2), …, grid (i,j), further, if it is a feasible reference point, then search the job's working space.

Step 2: Search a job's working space

It will both search the area which can fit in the job. If the area is free, then it will be defined as a candidate *c*. Further, let S represent all possible candidates *c*. $c \in S$.

Step 3: Evaluation of each candidate solution.

Based on Longest Contact Edge evaluation function (4), the program will count the value of each candidate solution from S. The reference point coordinate is (rpx_c, rpy_c), which is the same to the northwest point in Perng *et al.* (2007).

Step 4: To choose the allocation reference point

The program chooses the largest value of evaluation function. If there is a tie, it will choose the first one encountered.

Step 5: To allocate the job

Allocate the job (which is based on step 4) into the factory.

Notations are showed below:

c : a candidate solution number, $c \in S$

k : job number ($k = 1, 2, ..., n$)

a_k : the length of job k

b_k : the width of job k

$grid(i, j)$: is a Boolean variable at coordinate (i, j)

$$grid(i, j) = \begin{cases} 1 & \text{occupied by any job or obstacle} \\ 0 & \text{other} \end{cases}$$

(rpx_c, rpy_c) : reference point coordinate, $c \in S$

The objective is to find a maximum z value by c from S .

$$z = \operatorname*{Max}_{c \in S} \left\{ \sum_{i=rpx_c}^{rpx+a_k-1} grid(i, rpy_c - 1) + \sum_{i=rpx_c}^{rpx+a_k-1} grid(i, rpy_c + b_k) \right.$$

$$\left. + \sum_{j=rpy_c}^{rpy+b_k-1} grid(rpx_c - 1, j) + \sum_{j=rpy_c}^{rpy+b_k-1} grid(rpx_c + a_k, j) \right\} \quad (4)$$

In LCEA, we increase one unit of $grid(i, j)$ for both the length and width of the working space; however, the jobs will adjoin the original edges of the working space. Let $grid(i,j)=1$, if $grid(i,j)$ is occupied by any job or obstacles. Let $grid(i,j)=0$, if $grid(i,j)$ is an available space. The factory's working area concept is shown as Figure 2.2.

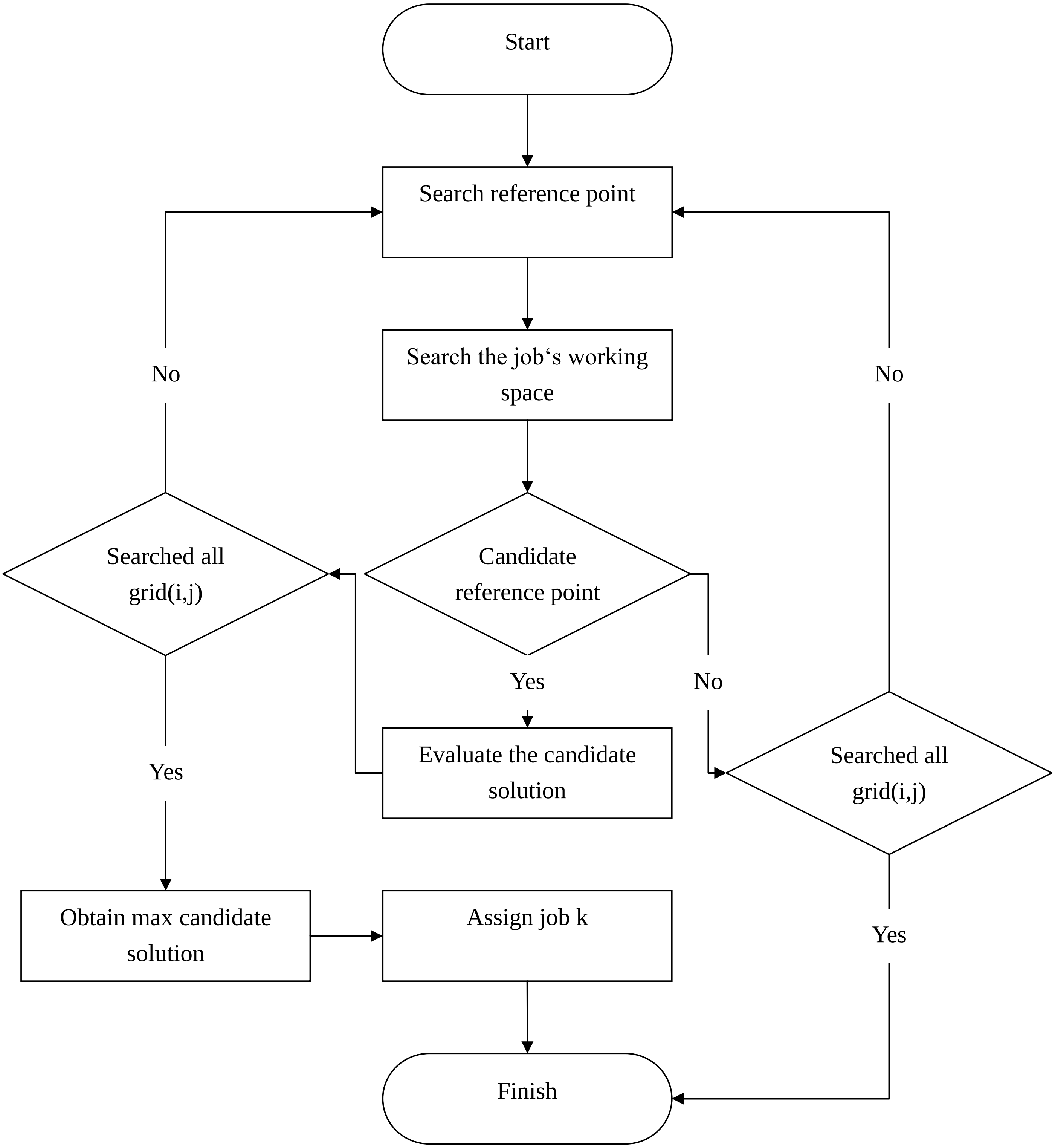


Figure 2.1 Flow chart of the Longest Contact Edge Algorithm

1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋰	0	1
1	0	0	0	0	0	⋯	0	1
1	0	0	0	0	0	⋯	0	1
1	0	0	0	0	0	⋯	0	1
1	0	0	0	0	0	⋯	0	1
1	0	0	0	0	0	⋯	0	1
1	1	1	1	1	1	⋯	1	1

Figure 2.2 Concept of the Longest Contact Edge Algorithm

2.5 Experimental Design

We use t test for two matched samples of dispatching rules to compare the performances between the NWA and LCEA, the hypothesis is shown as Table 2.1. We also use one-way ANOVA for performances between different dispatching rules using LCEA, the hypothesis is shown in Table 2.2. Both experimental designs analyze within and without obstacles in the shop floor (Figure 2.3 and Figure 2.4), the level of significance in both experimental designs is 0.05.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.3 The shop floor without obstacles

1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.4 The shop floor with obstacles

Table 2.1 Hypothesis of t test for two matched samples

Performance	Hypothesis	
Makespan	H_0: $\mu_{NWSPT} = \mu_{LCEASPT}$	H_0: $\mu_{NWEDD} = \mu_{LCEAEDD}$
	H_1: $\mu_{NWSPT} \neq \mu_{LCEASPT}$	H_1: $\mu_{NWEDD} \neq \mu_{LCEAEDD}$
	H_0: $\mu_{NWLPT} = \mu_{LCEALPT}$	H_0: $\mu_{NWSSR} = \mu_{LCEASSR}$
	H_1: $\mu_{NWLPT} \neq \mu_{LCEALPT}$	H_1: $\mu_{NWSSR} \neq \mu_{LCEASSR}$
	H_0: $\mu_{NWFCFS} = \mu_{LCEFCFS}$	H_0: $\mu_{NWLSR} = \mu_{LCEALSR}$
	H_1: $\mu_{NWFCFS} \neq \mu_{LCEAFCFS}$	H_1: $\mu_{NWSPT} \neq \mu_{LCEASPT}$
Space Utilization	H_0: $\mu_{NWSPT} = \mu_{LCEASPT}$	H_0: $\mu_{NWEDD} = \mu_{LCEAEDD}$
	H_1: $\mu_{NWSPT} \neq \mu_{LCEASPT}$	H_1: $\mu_{NWEDD} \neq \mu_{LCEAEDD}$
	H_0: $\mu_{NWLPT} = \mu_{LCEALPT}$	H_0: $\mu_{NWSSR} = \mu_{LCEASSR}$
	H_1: $\mu_{NWLPT} \neq \mu_{LCEALPT}$	H_1: $\mu_{NWSSR} \neq \mu_{LCEASSR}$
	H_0: $\mu_{NWFCFS} = \mu_{LCEFCFS}$	H_0: $\mu_{NWLSR} = \mu_{LCEALSR}$
	H_1: $\mu_{NWFCFS} \neq \mu_{LCEAFCFS}$	H_1: $\mu_{NWSPT} \neq \mu_{LCEASPT}$
Tardiness	H_0: $\mu_{NWSPT} = \mu_{LCEASPT}$	H_0: $\mu_{NWEDD} = \mu_{LCEAEDD}$
	H_1: $\mu_{NWSPT} \neq \mu_{LCEASPT}$	H_1: $\mu_{NWEDD} \neq \mu_{LCEAEDD}$
	H_0: $\mu_{NWLPT} = \mu_{LCEALPT}$	H_0: $\mu_{NWSSR} = \mu_{LCEASSR}$
	H_1: $\mu_{NWLPT} \neq \mu_{LCEALPT}$	H_1: $\mu_{NWSSR} \neq \mu_{LCEASSR}$
	H_0: $\mu_{NWFCFS} = \mu_{LCEFCFS}$	H_0: $\mu_{NWLSR} = \mu_{LCEALSR}$
	H_1: $\mu_{NWFCFS} \neq \mu_{LCEAFCFS}$	H_1: $\mu_{NWSPT} \neq \mu_{LCEASPT}$

Table 2.2 Hypothesis of one-way ANOVA

Performance	Hypothesis
Makespan	H_0: $\mu_{SPT} = \mu_{LPT} = \mu_{FCFS} = \mu_{EDD} = \mu_{SSR} = \mu_{LSR}$
	H_1: $\mu_{SPT} \neq \mu_{LPT} \neq \mu_{FCFS} \neq \mu_{EDD} \neq \mu_{SSR} \neq \mu_{LSR}$
Space Utilization	H_0: $\mu_{SPT} = \mu_{LPT} = \mu_{FCFS} = \mu_{EDD} = \mu_{SSR} = \mu_{LSR}$
	H_1: $\mu_{SPT} \neq \mu_{LPT} \neq \mu_{FCFS} \neq \mu_{EDD} \neq \mu_{SSR} \neq \mu_{LSR}$
Tardiness	H_0: $\mu_{SPT} = \mu_{LPT} = \mu_{FCFS} = \mu_{EDD} = \mu_{SSR} = \mu_{LSR}$
	H_1: $\mu_{SPT} \neq \mu_{LPT} \neq \mu_{FCFS} \neq \mu_{EDD} \neq \mu_{SSR} \neq \mu_{LSR}$

Chapter 3 Early and Tardy Penalties in Space Scheduling Problems

3.1 Introduction

As we stated before in previous chapter, in order to optimize the use of space capacity, we have to arrange the sequence of jobs so as to maximize space utilization. However, there are many causes of different conditions which will affect the job's sequence; penalty cost is a given example. Early and tardy penalty is broadly used in many scheduling problems. The purpose is to produce the goods at the best timing to minimize the cost of storage or lost of good will. However, the effect of early and tardy penalty did not be considered in previous studies. In this study, we take early and tardy penalty into consideration in space scheduling problem. We aim to find better space allocation under early and tardy penalty consideration.

3.2 Literature review

3.2.1 Space scheduling problem

Space scheduling problem is a newly risen research, which is developed from several combination studies of space constraint, scheduling and allocation problem. It is necessary to stretch in each of the study's methods, knowledge, and applications which will aid the efforts to our purpose in solving space scheduling problems. Perng *et al*. (2007, 2008a, 2008b) defined a job scheduling problem with space resource constraints as a space scheduling problem. The machine assembly process requires a certain amount of complete space on the shop floor in the factory for a period of time. The sizes of the shop floor and machines will determine the number of machines which can be assembled at the same time. Space scheduling problem considers duration of time condition, due to the space location of orders on the shop floor is related to the operation time of each job. Space scheduling problem also determine the sequence of jobs in order to optimize performance measurements instead of only choosing objects to optimize utilization. It is noticeable to consider space allocation algorithms from nesting, bin packing or knapsack problem; and consider dispatching rules such as traditional dispatching rules (SPT, LPT, FCFS, EDD) to space scheduling problem, as to increase the production capacity of the limited space resource in the shop floor.

Space scheduling problem is similar to the DLP (dynamic layout problem) and to other many researches, namely, circuit board design; layout design of hospitals, schools, and airports; warehouses; storages. Erel *et al*. (2003) used enumerative heuristic to reduce the total flow cost when arranging equipments in the factory. Balakrishnan *et al.* (2003) applied the technique of the dynamic plant layout problem (DPLP) to deal with the design of MBRLP (multi-period layout plans). Dunker *et al.* (2005) presented an algorithm combining dynamic programming and genetic search for solving a dynamic facility layout problem. Mckendall *et al.* (2006) proposed that manufacturing plants must be able to operate efficiently and respond quickly to changes in product mix and demand. They considered the problem of arranging and rearranging (when there were changes in the flows of materials between departments) manufacturing facilities such that the sum of the material handling and rearrangement costs were minimized. In their paper, simulated annealing (SA) were employed for the DLP. Erel *et al*. (2003) used the dynamic layout problem to address the situation where the traffic among the various units within a facility changes over time. Its objective was to determine a layout for each period in a planning horizon such that the total flow and the relocation costs were minimized. Perng *et al.* (2007) developed two space related dispatching rules, namely, small space requirement (SSR) first and large space requirement (LSR) first, to the space resource constrained problem. Perng *et al.* (2008a) proposed a space scheduling problem with obstacles. In a space scheduling problem, certain amounts of obstacles such as pillars and office areas were presented on the shop floor. Perng *et al.* (2008b) applied container loading problem (CLP) heuristics into a space scheduling problem.

3.2.2 Early and Tardy Penalty

Early and tardy penalties are often compared to just in time system (JIT). The purpose is to emphasize producing goods only when they are needed and to force jobs to be completed as close to their due dates as possible. Early cost may represent the cost of completing a project early, deterioration in the production of perishable goods or a holding cost for finished goods. The tardy cost can represent rush shipping costs, lost sales and loss of goodwill. Idle time must also be avoided for machines with high operating costs since the cost of keeping the machine running is then higher than the earliness cost incurred by completing a job before its due date. Thus, an ideal schedule is one in which all jobs are completed exactly on their due dates.

The assumption that no machine idle time is allowed reflects a production setting where the cost of machine idleness is higher than the early cost incurred by completing any job before its due date, or the capacity of the machine is limited, so that the machine must indeed be kept running. Studies showed that the best results are provided by the heuristics that explicitly consider both early and tardy costs. Heuristics have been used for the problem with a linear objective function. Kim (1994) considered the non-weighted linear problem with inserted idle time. Liaw (1999) developed an efficient lower and upper bounds for scheduling problem by a given set of independent jobs on a single machine to minimize the sum of weighted earliness and weighted tardiness without considering machine idle time. Feldmann (2003) used three meta-heuristics, evolutionary search (ES), simulated annealing (SA) and threshold accepting (TA), to find a schedule which minimizes the sum of earliness and tardiness costs on single-machine problems. Valente (2005) found a dispatch rule and a greedy procedure that minimizes the sum of the weighted quadratic earliness and tardiness costs for a single machine scheduling problem with quadratic earliness and tardiness costs, and machine idle time was not considered.

3.3 Problem Formulation

Let *N* denote a set of *n* jobs. Let $s_1, s_2, ..., s_n$ denote the start time for each job. Let $t_1, t_2, ..., t_n$ denote the processing time requirement for each job. Let $d_1, d_2, ..., d_n$ denote the due date specified by customers for each job. Let σ denote an arbitrary sequence. All possible permutations are [1-2-…-(n-1)-n], [1-2-…-n-(n-1)],…, [n-(n-1)-…-1]. There have *n*! possibilities. The traditional dispatching rule, such as SPT, FCFS, or EDD rule, is one of the permutation results. For a job *i*, a earliness will be denoted by E_i, a tardiness will be denoted by T_i. The summary of notations is shown as follows:

N: a set of n jobs

s_i : start time for each job i

t_i : processing time requirement for each job i

d_i : due date specified by customers for each job i

σ : an arbitrary sequence

E_i : a earliness for job i

T_i : a tardiness for job i

P_1 : unit earliness penalty

P_2 : unit tardiness penalty

$E_i = \text{Max}\{d_i - s_i - t_i, 0\}$

$T_i = \text{Max}\{s_i + t_i - d_i, 0\}$

$f_5(\sigma)$: total cost by σ processing sequence

The objective is to find a sequence to minimize the total penalty cost.

$$\textbf{Min} \quad f_5(\sigma) = P_1 \sum_{i=1}^{n} E_i + P_2 \sum_{i=1}^{n} T_i \tag{5}$$

In the cost function (5), P_1 denotes the unit early penalty and P_2 denotes the unit tardy penalty. We assume that if a job i have been completed early, the job will be moved to the buffer or storage. Thus, earliness will occur.

Chapter 4 Results and Discussion

In this study, we compare the performance between northwest algorithm and the longest contact edge algorithm under different number of jobs (25, 50 and 75). Thirty different data sets, with and without obstacles, were included in this study. We also employ LCEA for space scheduling problem considering early and tardy penalties.

For the computations of the performance of the proposed system under traditional dispatching rules, the programs were developed by using PHP and Microsoft Visual Basic languages. We used a Pentium IV (Celeron CPU 2.40GHz) computer for the computations. The test data were obtained from the OR-Library (Beasley 1990, 2008) and based on Taillard (1993) due to real data were too few for overall testing. However, job size requirement for the scheduling problem of the library were not available. Therefore, the job size requirement and order information were obtained from a company located in central Taiwan. The data were also obtained from Perng *et al*. (2007). All calculations are round up to second decimal place.

The objective of chapter 2 was to compare northwest algorithm with longest contact edge algorithm in space scheduling problems and to find a better dispatching rule for each performance measurement. There were some assumptions, namely, all of the orders are rectangles, a job will not be moved until it is done and due, there is no constraint on job's height. The following Figures and Tables are results by applying NWA and LCEA for space scheduling problems with and without obstacles: Figure 4.1 and Tables 4.1-4.4 showed the makespan in 25-job data, Figure 4.2 and Tables 4.5-4.8 showed the tardiness, Figure 4.3 and Tables 4.9-4.12 showed the space utilization; Figure 4.4 and Tables 4.13-4.16 showed the makespan in 50-job data, Figure 4.5 and Tables 4.17-4.20 showed the tardiness, Figure 4.6 and Tables 4.21-4.24 showed the space utilization; Figure 4.7 and Tables 4.25-4.28 showed the makespan in 75-job data, Figure 4.8 and Tables 4.29-4.32 showed the tardiness, Figure 4.9 and Tables 4.33-4.36 showed the space utilization.. Sipser (2006) represents T(n) as the mean time complexity.

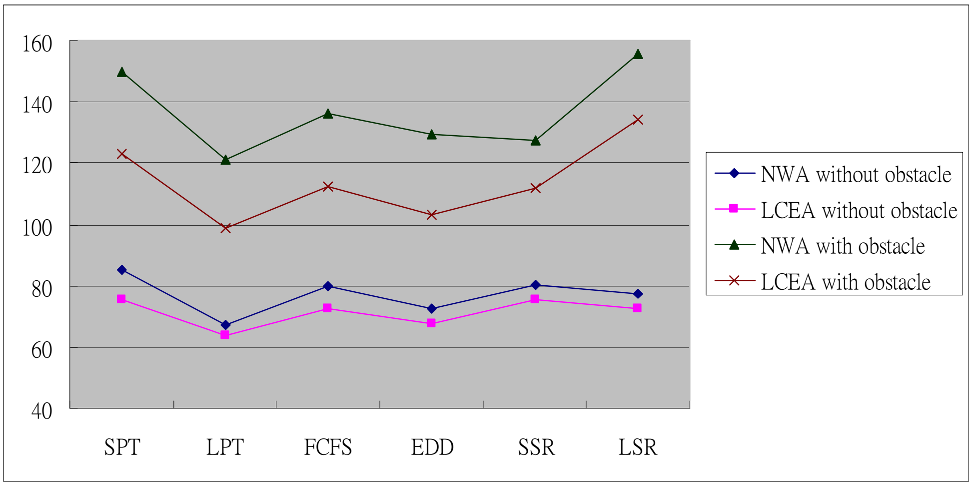


Figure 4.1 Makespan in 25-job data

Table 4.1 Makespan using NWA without obstacles in 25-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	133	101	125	118	122	127
Min	59	56	59	57	59	58
Mean	85.2	67.43	79.93	72.53	80.37	77.3
S.D.	20.13	12.72	19.12	15.76	17.66	17.29
T(n)	142455	140084.4	143633.4	124962.9	141535.8	171410.2

Table 4.2 Makespan using LCEA without obstacles in 25-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	128	95	112	108	115	111
Min	56	56	57	56	56	58
Mean	75.67	63.9	72.4	67.9	75.47	72.6
S.D.	19.05	9.85	16.01	13.38	17.4	14.84
T(n)	1221082	1232975	1238529	1293922	1679210	921304.1

Table 4.3 Makespan using NWA with obstacles in 25-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	256	228	271	237	234	366
Min	64	59	62	60	61	59
Mean	149.57	121.2	136.23	129.6	127.57	155.73
S.D.	58.5	55.43	61.49	55.77	51.3	83.67
T(n)	157701.1	152270.1	154527.3	142678.7	182989.7	162014.8

Table 4.4 Makespan using LCEA with obstacles in 25-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	261	195	239	235	219	299
Min	59	56	57	57	57	57
Mean	123.23	98.63	112.6	102.93	111.7	134.43
S.D.	58.73	44.2	51.78	47.72	46.25	77
T(n)	1129084	1160516	1148896	1169171	1401045	1057499

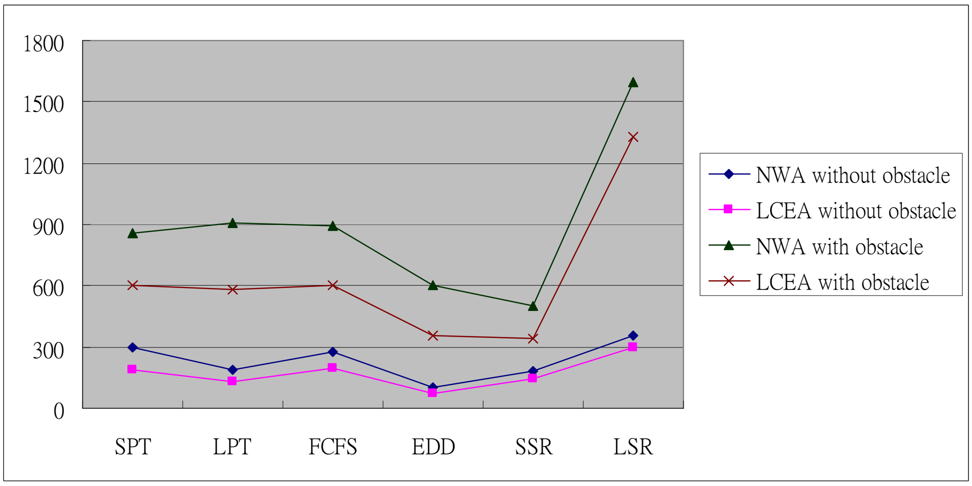


Figure 4.2 Tardiness in 25-job data

Table 4.5 Tardiness using NWA without obstacles in 25-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	856	697	819	522	769	1028
Min	0	0	0	0	0	0
Mean	297.3	186.5	278.2	98.7	184.6	356
S.D.	217.24	208.83	226.74	132.89	184.84	246.82
T(n)	142455	140084.4	143633.4	124962.9	141535.8	171410.2

Table 4.6 Tardiness using LCEA without obstacles in 25-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	691	609	639	451	721	898
Min	0	0	0	0	0	0
Mean	189.57	130.13	195.1	71.6	147.5	297.27
S.D.	187.33	173.84	189.57	112.83	179.24	241.67
T(n)	1221082	1232975	1238529	1293922	1679210	921304.1

Table 4.7 Tardiness using NWA with obstacles in 25-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	1772	2435	2964	1973	1877	4385
Min	45	8	63	1	29	5
Mean	859.6	908.23	889.63	600.97	497.53	1599
S.D.	479.66	757.07	698.14	557.75	455.34	1256.18
T(n)	157701.1	152270.1	154527.3	142678.7	182989.7	162014.8

Table 4.8 Tardiness using LCEA with obstacles in 25-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	1994	2014	1808	1772	1381	3965
Min	0	1	0	0	0	0
Mean	603.1	580.97	599.6	354.2	338.27	1328.6
S.D.	505.45	628.44	534.87	421.68	355.84	1282.53
T(n)	1129084	1160516	1148896	1169171	1401045	1057499

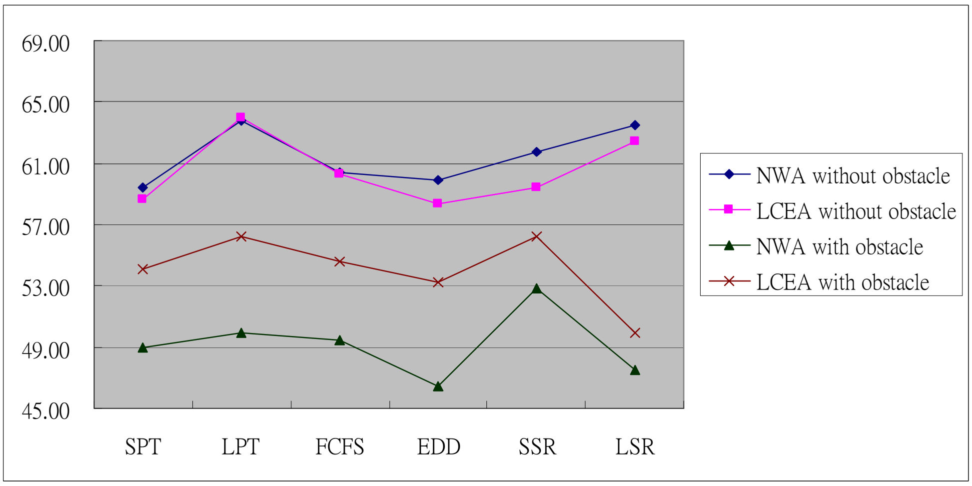


Figure 4.3 Space utilization in 25-job data

Table 4.9 Space utilization using NWA without obstacles in 25-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	71	79	70	72	75	77
Min	47	47	47	47	46	47
Mean	59.4	63.73	60.43	59.9	61.7	63.47
S.D.	6.32	7.29	5.61	6.81	6.9	7.83
T(n)	142455	140084.4	143633.4	124962.9	141535.8	171410.2

Table 4.10 Space utilization using LCEA without obstacles in 25-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	72	85	73	74	75	79
Min	39	40	38	42	41	44
Mean	58.67	63.93	60.3	58.37	59.47	62.43
S.D.	7.97	9.95	10.21	7.9	8.72	7.92
T(n)	1221082	1232975	1238529	1293922	1679210	921304.1

Table 4.11 Space utilization using NWA with obstacles in 25-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	62	66	64	62	65	67
Min	31	35	37	31	43	31
Mean	48.99	49.96	49.43	46.46	52.79	47.53
S.D.	9	9.77	8.86	8.46	6.11	10.87
T(n)	157701.1	152270.1	154527.3	142678.7	182989.7	162014.8

Table 4.12 Space utilization using LCEA with obstacles in 25-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	68	76	72	70	74	67
Min	33	35	35	29	44	31
Mean	54.06	56.23	54.63	53.23	56.26	49.93
S.D.	9.58	11.2	10.17	10.42	7.49	11.48
T(n)	1129084	1160516	1148896	1169171	1401045	1057499

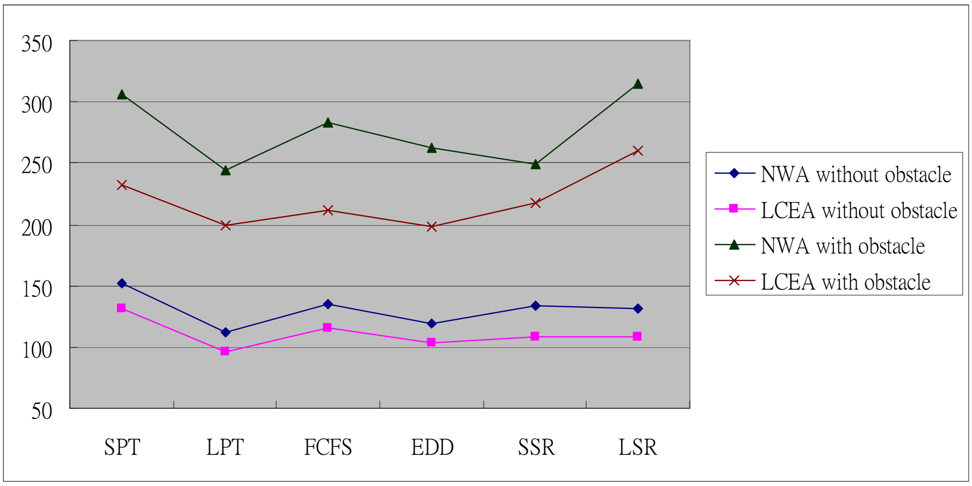


Figure 4.4 Makespan in 50-job data

Table 4.13 Makespan using NWA without obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	198	158	200	180	181	178
Min	107	71	89	79	93	86
Mean	151.5	111.8	134.57	119.37	134.3	130.8
S.D.	28.92	25.9	29.62	28.31	27.3	26.13
T(n)	345015.2	304846.5	327115.7	315741.6	400728	369289.9

Table 4.14 Makespan using LCEA without obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	201	153	173	160	168	168
Min	89	60	79	70	60	60
Mean	130.83	95.63	115.57	103.77	108.9	108.9
S.D.	29	27.07	26.72	22.37	27.04	27.04
T(n)	1873230	1923400	1888276	1919576	2655211	2655211

Table 4.15 Makespan using NWA with obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	486	384	474	457	425	623
Min	143	109	144	120	132	128
Mean	306.33	244.63	282.63	262.53	249.77	314.6
S.D.	100.39	88.03	100.54	101.09	82.89	126.27
T(n)	324420.2	298566.8	308262	296033.1	435340.9	304970.1

Table 4.16 Makespan using LCEA with obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	467	360	407	367	362	541
Min	109	85	95	88	102	95
Mean	232.73	199.53	211.43	198.6	218.1	259.6
S.D.	83.7	75.1	84.02	77.54	78.4	111.46
T(n)	2012006	2150939	2081974	2056968	2347393	2039592

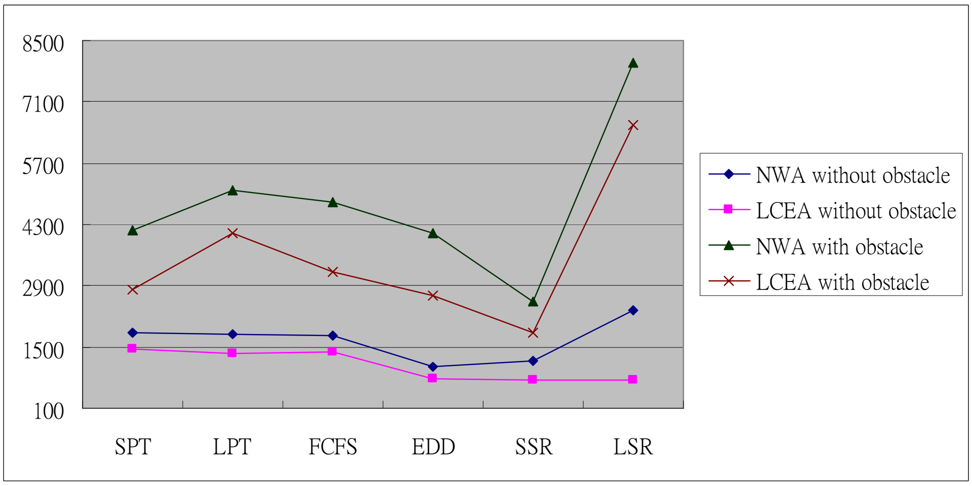


Figure 4.5 Tardiness in 50-job data

Table 4.17 Tardiness using NWA without obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	2893	3236	3177	2188	2305	3970
Min	911	531	803	313	468	1173
Mean	1835.87	1796.23	1767.63	1044.63	1179.47	2340.97
S.D.	527.83	758.95	692.99	531.45	498.5	718.67
T(n)	345015.2	304846.5	327115.7	315741.6	400728	369289.9

Table 4.18 Tardiness using LCETA without obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	2557	3346	2735	2299	1945	1945
Min	563	229	335	145	153	153
Mean	1456.47	1359	1393.9	771.57	740.43	740.43
S.D.	547.02	803.76	632.95	516.57	477.57	477.57
T(n)	1873230	1923400	1888276	1919576	2655211	2655211

Table 4.19 Tardiness using NWA with obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	7060	10192	8793	8876	4996	17092
Min	1719	1697	1901	1134	970	2792
Mean	4157.73	5068.37	4812.3	4091.1	2524	7979.33
S.D.	1459.87	2251.19	2031.03	2101.96	1131.68	3458.66
T(n)	324420.2	298566.8	308262	296033.1	435340.9	304970.1

Table 4.20 Tardiness using LCEA with obstacles in 50-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	5014	9957	7421	6825	4094	16288
Min	981	994	879	387	531	1547
Mean	2793.43	4099.33	3224.07	2685.27	1834.93	6569.23
S.D.	945.29	2268.91	1686.87	1678.94	999.95	3489.58
T(n)	2012006	2150939	2081974	2056968	2347393	2039592

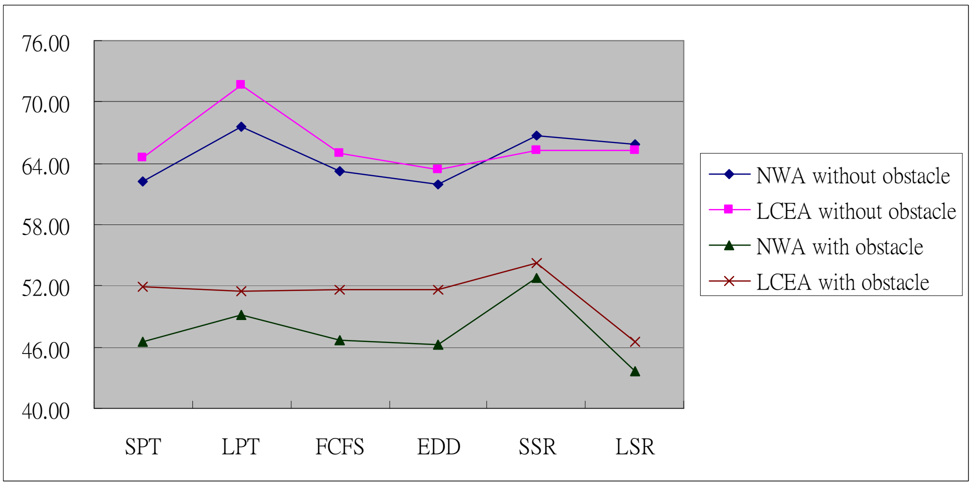


Figure 4.6 Space utilization in 50-job data

Table 4.21 Space utilization using NWA without obstacles in 50-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	69	77	70	69	73	77
Min	53	54	56	53	59	56
Mean	62.23	67.53	63.27	61.93	66.7	65.77
S.D.	4.11	5.31	3.99	3.97	3.3	4.72
T(n)	345015.2	304846.5	327115.7	315741.6	400728	369289.9

Table 4.22 Space utilization using LCEA without obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	72	87	73	74	73	73
Min	57	60	56	56	50	50
Mean	64.6	71.7	65	63.33	65.27	65.27
S.D.	3.91	5.81	4.19	4.33	6.16	6.16
T(n)	1873230	1923400	1888276	1919576	2655211	2655211

Table 4.23 Space utilization using NWA with obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	66	68	65	60	63	67
Min	38	38	39	38	44	34
Mean	46.46	49.16	46.69	46.23	52.73	43.69
S.D.	7.01	7.38	6	5.73	5.05	8.17
T(n)	324420.2	298566.8	308262	296033.1	435340.9	304970.1

Table 4.24 Space utilization using LCEA with obstacles in 50-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	63	69	67	65	69	70
Min	40	40	40	41	44	35
Mean	51.89	51.53	51.66	51.59	54.26	46.46
S.D.	6.48	7.87	7.66	6.66	5.93	9
T(n)	2012006	2150939	2081974	2056968	2347393	2039592

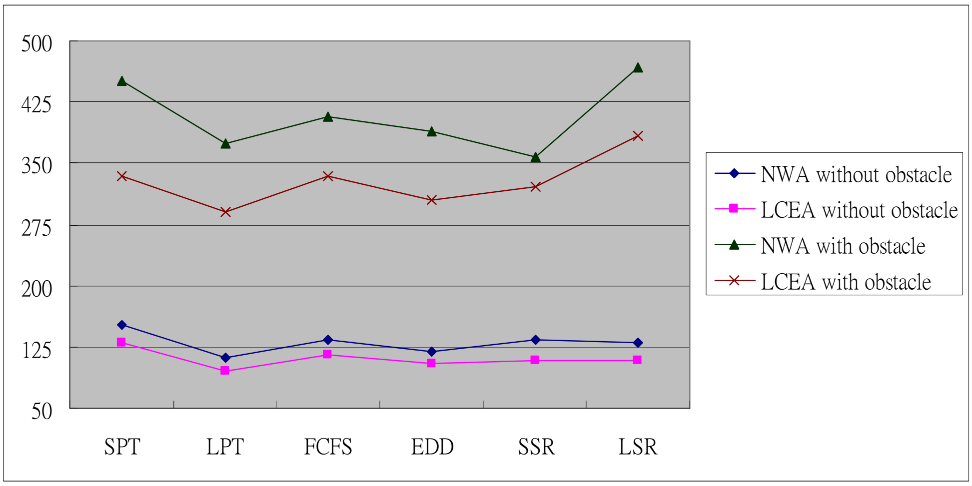


Figure 4.7 Makespan in 75-job data

Table 4.25 Makespan using NWA without obstacles in 75-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	198	158	200	180	181	178
Min	107	71	89	79	93	86
Mean	151.5	111.8	134.57	119.37	134.3	130.8
S.D.	28.92	25.9	29.62	28.31	27.3	26.13
T(n)	345015.2	304846.5	327115.7	315741.6	400728	369289.9

Table 4.26 Makespan using LCEA without obstacles in 75-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	201	153	173	160	168	168
Min	89	60	79	70	60	60
Mean	130.83	95.63	115.57	103.77	108.9	108.9
S.D.	29	27.07	26.72	22.37	27.04	27.04
T(n)	1873230	1923400	1888276	1919576	2655211	2655211

Table 4.27 Makespan using NWA with obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	733	771	681	694	618	792
Min	215	168	204	156	158	189
Mean	450.8	373.67	407.07	388.73	357.03	467.63
S.D.	144.34	146.37	138.96	147.22	131.7	157.1
T(n)	428377.7	410950.8	430524.3	416826.8	685861.6	444989

Table 4.28 Makespan using LCEA with obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	644	569	603	607	611	680
Min	168	93	150	113	144	146
Mean	333.87	289.87	334.63	305.43	322.33	382.9
S.D.	132.98	137.34	137.09	138.6	136.27	152.43
T(n)	3306373	3278287	3335410	3276354	3410630	2974632

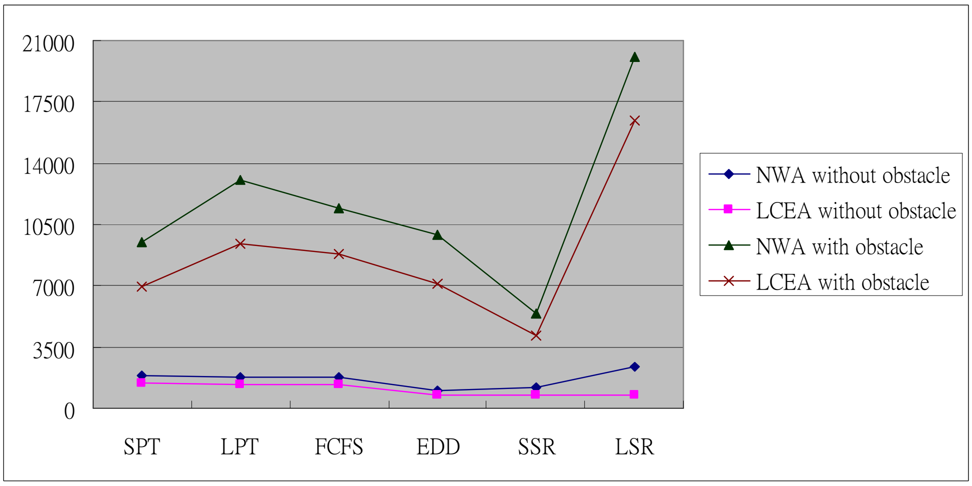


Figure 4.8 Tardiness in 75-job data

Table 4.29 Tardiness using NWA without obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	2893	3236	3177	2188	2305	3970
Min	911	531	803	313	468	1173
Mean	1835.87	1796.23	1767.63	1044.63	1179.47	2340.97
S.D.	527.83	758.95	692.99	531.45	498.5	718.67
T(n)	345015.2	304846.5	327115.7	315741.6	400728	369289.9

Table 4.30 Tardiness using LCEA without obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	2557	3346	2735	2299	1945	1945
Min	563	229	335	145	153	153
Mean	1456.47	1359	1393.9	771.57	740.43	740.43
S.D.	547.02	803.76	632.95	516.57	477.57	477.57
T(n)	1873230	1923400	1888276	1919576	2655211	2655211

Table 4.31 Tardiness using NWA with obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	15671	31051	17541	16475	11822	31487
Min	4649	5052	5198	2602	1514	7332
Mean	9513.63	13054.83	11400.37	9934.6	5424.5	20028.57
S.D.	3033.12	5613.84	3991.07	3824.93	2616.09	6497.76
T(n)	428377.7	410950.8	430524.3	416826.8	685861.6	444989

Table 4.32 Tardiness using LCEA with obstacles in 75-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	12737	23448	16497	13240	11066	30765
Min	3304	2071	2534	1289	987	4844
Mean	6952.4	9437.97	8821.93	7079.03	4184.33	16386.33
S.D.	2640.38	5295.58	4203.09	3442.91	2559.61	6841.92
T(n)	3306373	3278287	3335410	3276354	3410630	2974632

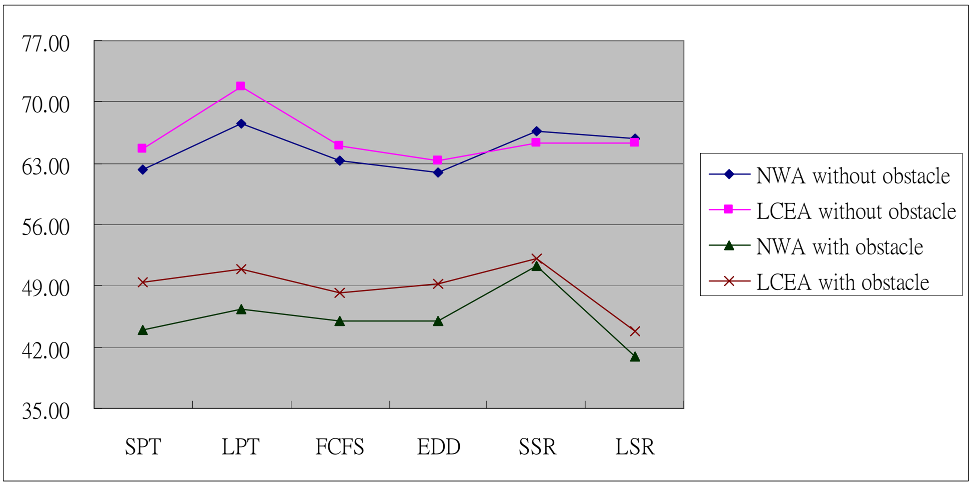


Figure 4.9 Space utilization in 75-job data

Table 4.33 Space utilization using NWA without obstacles in 75-jobs data

Dispatching Rules	**SPT**	**LPT**	**FCFS**	**EDD**	**SSR**	**LSR**
Max	69	77	70	69	73	77
Min	53	54	56	53	59	56
Mean	62.23	67.53	63.27	61.93	66.7	65.77
S.D.	4.11	5.31	3.99	3.97	3.3	4.72
T(n)	345015.2	304846.5	327115.7	315741.6	400728	369289.9

Table 4.34 Space utilization using LCEA without obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	72	87	73	74	73	73
Min	57	60	56	56	50	50
Mean	64.6	71.7	65	63.33	65.27	65.27
S.D.	3.91	5.81	4.19	4.33	6.16	6.16
T(n)	1873230	1923400	1888276	1919576	2655211	2655211

Table 4.35 Space utilization using NWA with obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	56	58	57	54	60	58
Min	38	36	39	39	41	32
Mean	43.99	46.29	45.06	44.93	51.26	40.86
S.D.	5.01	4.88	4.86	4.77	4.88	5.39
T(n)	428377.7	410950.8	430524.3	416826.8	685861.6	444989

Table 4.36 Space utilization using LCEA with obstacles in 75-jobs data

Dispatching Rules	SPT	LPT	FCFS	EDD	SSR	LSR
Max	61	68	65	64	64	61
Min	40	41	39	40	43	32
Mean	49.33	50.86	48.23	49.26	52.16	43.79
S.D.	6.1	7.68	6.72	6.53	5.68	6.74
T(n)	3306373	3278287	3335410	3276354	3410630	2974632

The statistical results of different number of jobs were shown in Tables 4.37-4.44. In these tables, ‘Yes’ represent for reject H_0 and ‘No’ if it dose not reject H_0 in the hypothesis of Chapter 2. Tables 4.37-4.42 showed the results of t test for two matched samples of dispatching rule using NWA and LCEA in 25-job data, 50-job data and 75-job data with and without obstacles Table 4.43 and Table 4.44 showed the results of one-way ANOVA between different dispatching rules using LCEA with and without space obstacles (Note: * represent NWA out perform LCEA).

Table 4.37 T test for two matched samples in 25-job data without space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
SPT	Yes	Yes	No
LPT	Yes	Yes	No
FCFS	Yes	Yes	No
EDD	Yes	Yes	Yes*
SSR	Yes	Yes	Yes*
LSR	Yes	Yes	Yes*

Table 4.38 T test for two matched samples in 25-job data with space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
SPT	Yes	Yes	Yes
LPT	Yes	Yes	Yes
FCFS	Yes	Yes	Yes
EDD	Yes	Yes	Yes
SSR	Yes	Yes	Yes
LSR	Yes	Yes	Yes

Table 4.39 T test for two matched samples in 50-job data without space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
SPT	Yes	Yes	Yes
LPT	Yes	Yes	Yes
FCFS	Yes	Yes	Yes
EDD	Yes	Yes	Yes
SSR	Yes	Yes	Yes*
LSR	Yes	Yes	No

Table 4.40 T test for two matched samples in 50-job data with space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
SPT	Yes	Yes	Yes
LPT	Yes	Yes	Yes
FCFS	Yes	Yes	Yes
EDD	Yes	Yes	Yes
SSR	Yes	Yes	Yes
LSR	Yes	Yes	Yes

Table 4.41 T test for two matched samples in 75-job data without space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
SPT	Yes	Yes	Yes
LPT	Yes	Yes	Yes
FCFS	Yes	Yes	Yes
EDD	Yes	Yes	Yes
SSR	Yes	Yes	Yes*
LSR	Yes	Yes	Yes*

Table 4.42 T test for two matched samples in 75-job data with space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
SPT	Yes	Yes	Yes
LPT	Yes	Yes	Yes
FCFS	Yes	Yes	Yes
EDD	Yes	Yes	Yes
SSR	Yes	Yes	Yes
LSR	Yes	Yes	Yes

Table 4.43 One-way ANOVA without space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
25 job	Yes	Yes	Yes
50 job	Yes	Yes	Yes
75 job	Yes	Yes	Yes

Table 4.44 One-way ANOVA with space obstacles

Performance	**Makespan**	**Tardiness**	**Space utilization**
25 job	No	Yes	No
50 job	No	Yes	Yes
75 job	No	Yes	Yes

Tables 4.45-4.46 listed the better dispatching rule for each performance measurement with and without obstacles in 25-job, 50-job and 75-job data by applying LCEA for allocation. The comparisons of dispatching rules of previous literature in Perng *et al*. (2007) were also cited in Table 4.47. Under NWA, EDD outperformed SPT, LPT, FCFS, SSR and LSR in total tardiness. LPT and LSR outperformed SPT, EDD, FCFS and SSR in space utilization and LPT, EDD and LSR outperformed SPT, FCFS and SSR in makespan. Perng *et al*. (2007) did not show the

experiment using number of 25 or 75 jobs option nor dose the result with obstacles. To sum up, except the space utilization has no significant difference with NWA and LCEA in 25-job data without obstacles, LCEA could provide a better schedule for space scheduling problem. However, longest lontact edge algorithm can not outperform northwest algorithm in time complexity due to more calculations are needed in LCEA.

Table 4.45 Better dispatching rule for each performance indicators without obstacles

Performance	**Makespan**	**Total Tardiness**	**Space utilization**
25 job	LPT	EDD	LPT*
50 job	LPT	SSR	LPT
75 job	LPT	SSR	LPT

(Note: * represent no significant difference between NWA and LCEA)

Table 4.46 Better dispatching rule for each performance indicators with obstacles

Performance	**Makespan**	**Total Tardiness**	**Space utilization**
25 job	LPT	SSR	SSR
50 job	EDD	SSR	SSR
75 job	LPT	SSR	SSR

Table 4.47 Scheduling performance in 50-job data from Perng *et al*. (2007)

Dispatching rule	**Total Tardiness**	**Space Utilization**	**Makespan**
SPT	Worse	Worse	Worse
LPT	Worse	Better	Better
EDD	Better	Worse	Better
FCFS	Worse	Worse	Worse
SSR	Worse	Worse	Worse
LSR	Worse	Better	Better

The objective of chapter 3 was to minimize the total penalty cost in our study. There is no obstacles consideration in chapter 3. We only focused in early and tardy penalties. Jobs will leave the working area when it is completed. A buffer or storage is considered. There were some assumptions, namely, the buffer or storage is available to fit in any number or any shape of jobs, the unit earliness penalty for all jobs is equal, the unit tardiness penalty for all jobs is also equal. The unit earliness penalty and unit tardiness penalty in this study was obtained from a machinery assemble company in Taiwan. Tables 4.48- 4.50 showed the earliness and tardiness in 25-job data, 50-job data and 75-job data using LCEA.

Table 4.48 Earliness and tardiness in 25-job data using LCEA

Dispatching Rule	**Earliness**				**Tardiness**			
	Max	**Min**	**Mean**	**S.D.**	**Max**	**Min**	**Mean**	**S.D.**
SPT	688	388	545.9	80.18	155	0	26.2	37.32
LPT	688	121	409.63	170.46	459	0	48.53	97.12
FCFS	688	246	495.47	113.42	198	0	33.87	51.83
EDD	689	128	476.93	150.54	111	0	10.73	26.84
SSR	704	294	529.77	97.8	179	0	32.67	48.59
LSR	688	167	428.8	141.25	215	0	46.03	57.55

Table 4.49 Earliness and tardiness in 50-job data using LCEA

Dispatching Rule	**Earliness**				**Tardiness**			
	Max	**Min**	**Mean**	**S.D.**	**Max**	**Min**	**Mean**	**S.D.**
SPT	1051	309	748.93	223.35	1085	0	264.9	285.3
LPT	798	52	277.9	182.24	2405	17	877.27	748.62
FCFS	1033	232	545.13	213.69	1416	32	599.17	432.64
EDD	802	53	324.07	189.05	1495	4	413.77	438.74
SSR	1146	523	866.63	181.58	1085	21	373.17	302.51
LSR	560	107	322	118.13	2016	141	828.93	552.73

Table 4.50 Earliness and tardiness in 75-job data using LCEA

Dispatching Rule	**Earliness**				**Tardiness**			
	Max	**Min**	**Mean**	**S.D.**	**Max**	**Min**	**Mean**	**S.D.**
SPT	1519	678	1025.67	214.37	2753	222	1327.17	727.23
LPT	605	36	235.3	133	7276	238	2826.13	1766.82
FCFS	1356	287	608.63	241.86	4310	133	1942.47	1080.02
EDD	1067	50	296.5	206.63	3661	12	1565.5	1020.19
SSR	1749	760	1199.13	252.6	1983	50	858.87	538.09
LSR	667	98	271.2	139.79	5070	351	2500.77	1378.86

Results of total penalty cost are shown in Tables 4.51-4.52. EDD outperformed SPT, LPT, FCFS, SSR and LSR in total penalty cost in 25-job data. SPT outperformed LPT, EDD, FCFS, SSR and LSR in total penalty cost in 50-job data. SSR outperformed SPT, LPT, EDD, FCFS and LSR in total penalty cost in 75-job data.

Table 4.51 Results of total penalty cost (unit: NT dollar)

Dispatching Rule	**25-job data**	**50-job data**	**75-job data**
SPT	2018372	9053876	38871798
LPT	2307023	24742433	78137257
FCFS	2108896	17744566	54769274
EDD	1429579	12129237	43678612
SSR	2157586	12306001	26429779
LSR	2284020	23520453	69291563

Table 4.52 Results of better dispatching rule of total penalty cost

Dispatching Rule	**25-job data**	**50-job data**	**75-job data**
SPT	Worse	Better	Worse
LPT	Worse	Worse	Worse
FCFS	Worse	Worse	Worse
EDD	Better	Worse	Worse
SSR	Worse	Worse	Better
LSR	Worse	Worse	Worse

Chapter 5 Conclusions and Recommendations

The space scheduling problem is an important issue in machinery assembly factory. In this study, two topics were considered. Chapter 2 compares northwest algorithm with a new space allocation algorithm we proposed in this study, namely, Longest Contact Edge Algorithm. Chapter 3 considered early and tardy penalty cost in space scheduling problems. Both topics aimed to find a better dispatching rule for each performance measurement. We employed the longest contact edge algorithm to allocate space for space scheduling problem and found that the longest contact edge algorithm is more efficient than northwest Algorithm for obtaining better schedules. However, longest contact edge algorithm results in more time complexity then northwest algorithm due to more calculation.

Some results show that NWA outperforms LCEA in space utilization. We suppose that without the influence of obstacles in the shop floor, the allocation of jobs are more freely decided and less restricted, which may lessen the effect of LCEA and result in better performances with NWA. These results only occur under the condition without obstacles. We also found that different dispatching rules have significant difference with longest contact edge algorithm in some performance measurements.

There are some assumptions in this study, namely, all of the orders are rectangles, a job will not be moved until it is done and due, there is no constraint on job's height, the buffer or storage is available to fit in any number or any shape of jobs, the unit earliness penalty for all jobs is equal and the unit tardiness penalty for all jobs is also equal. It may result in different conclusions if some of assumptions are relaxed.

References

[1] Backer,K.R. and G.D.Scudder (1990) Sequencing with earliness and tardiness penalties: A review, *Operations Research*, 38:22-36.

[2] Balakrishnan,J. , C.H.Cheng, D.G.Conway and C.M.Lau (2003) A hybrid genetic algorithm for the dynamic plant layout problem, *International Journal of Production Economics*, 86(2):107-120.

[3] Beasley,J.E. (1990) OR-Library: distributing test problems by electronic mail, *Journal of Operational Research Society*, 41(11):1069-1072.

[4] Beasley,J.E. (2008) OR-Library, *http://people.brunel.ac.uk/~mastjjb/jeb/info.htm*

[5] Bischoff,E.E. (2006) Three-dimensional packing of items with limited load bearing strength, *European Journal of Operational Research*, 168(3):952-966.

[6] Chow,W.W. (1979) Nesting of a single shape on a strip, *International Journal of Production Research*, 17(4):305-321.

[7] Chryssolouris,G. and V.Subramaniam (2001) Dynamic scheduling of manufacturing job shops using genetic algorithms, *Journal of Intelligent Manafacturing*, 12:281-293.

[8] Dantzig,G.B. (1957) Discrete-variable extremum problems, *Operations Research*, 5(2):266-277.

[9] Dunker,T., G.Radons and E.Westkamper (2005) Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem, *European Journal of Operational Research*, 165(1):55-69.

[10] Egeblad,J., B.K.Nielsen and A.Odgaard (2007) Fast neighborhood search for two-and three-dimensional nesting problems, *European Journal of Operational Research*, 183:1249-1266.

[11] Erel,E., J.B.Ghosh and J.T.Simon (2003) New heuristic for the dynamic layout problem, *Journal of the Operational Research Society*, 54(12):1275-1282.

[12] Gomes,A.M. and J.F.Oliveira (2002) A 2-exchange heuristic for nesting problems, *European Journal of Operational Research*, 141:359-370.

[13] Hegazy,T. and E.Emad (1999) EvoSite: Evolution-based model for site layout planning, *Journal of Computing in Civil Engineering*, 13:198-203.

[14] Hung,Y.F. and C.B.Chang (2002) Dispatching rules using flow time predictions for semiconductor wafer fabrications, *Journal of the Chinese Institute of Industrial Engineering*, 19(1):67-74.

[15] Johnson,S.M. (1954) Optimal two and three stage production schedules with setup times included, *Naval Research Logistics Quarterly*, 1:61-67.

[16] Johnson,D.S. (1973) Near-optimal bin packing algorithms, *Ph.D. Dissertation, Department of Mathematics, Massachusetts Institute of Technology*.

[17] Kelley,J.E. (1960) The cutting-plane method for solving convex programs, *Journal of the Society for Industrial and Applied*, 8(4):703-712.

[18] Kim,Y.D. and C.A.Yano (1994) Minimizing mean tardiness and earliness in single-machine scheduling problems with unequal due dates, *Naval Research Logistics*, 41:913-933.

[19] Lee,W.C., H.Mab and B.W.Cheng, (2008) A heuristic for nesting problems of irregular shapes, *Computer-Aided Design*, 40(5):625-633.

[20] Liaw,C.F. (1999) A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem, *Computers & Operations Research*, 26(7):679-693.

[21] Martin,F. and D.B. (2003) Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches, *Computers & Industrial Engineering*, 44(2):307-323.

[22] Mckendall,A.R., J., J.Shang and S.Kuppusamy (2006) Simulated annealing heuristics for the dynamic facility layout problem, *Computers and Operations Research*, 33(8):2431-2444.

[23] Mizrak,P. and G.M.Bayhan (2006) Comparative study of dispatching rules in a real-life job shop environment, *Applied Artificial Intelligence*, 20:585-607.

[24] Panwalker,S.S. and W.Iskander (1997) A Survey of Scheduling Rules, *Operations Research*, 25(1):45-61.

[25] Perng,C. and Z.P.Ho (2004) Applying information technique to layout on Semi-conductor equipments factory, *The Third Conference on Innovation and Technology Management*, 114:1-10.

[26] Perng,C., Y.C.Lai, Z.Y.Zhuang and Z.P.Ho (2007) Job scheduling in machinery industry with space constraint, *System Analysis Section The Fourth Conference on Operations Research of Taiwan*, 5:1-10.

[27] Perng,C., Y.C.Lai and Z.P.Ho (2008) Jobs scheduling in an assembly factory with space obstacles, *The 18th International Conference on Flexible Automation and Intelligent Manufacturing*, Sweden, Skovde, June.30-July.2.

[28] Perng,C., S.S.Lin and Z.P.Ho (2008) On space resource constrained job scheduling problems - A container loading heuristic approach, *The 4th International Conference on Natural Computation*, 3122:1-5.

[29] Perng,C., Y.C.Lai, Z.Y.Zhuang and Z.P.Ho (2006) The layout study of applying scheduling technique to job based assembly factory, *Scheduling & VRP Section, The Third Conference on Operations Research of Taiwan*, 59:1-10.

[30] Pinedo,M. (2002) Scheduling theory, algorithms and systems, *second edition Prentice Hall press-New Jersey*.

[31] Pugazhendhi,S., S.Thiagarajan, C.Rajendran and N.Anantharaman (2004) Relative performance evaluation of permutation and non-permutation schedules in flowline-based manufacturing systems with flowtime objective, *International Journal of Advance Manufacturing Technology*, 23:820-830.

[32] Holthaus,O. and C.Rajendran (1997) Efficient dispatching rules for scheduling in a job shop, *International Journal of Production Economics*, 48(1):87-105.

[33] Rajendran.C and H.Ziegler (2001) A performance analysis of dispatching rules and a heuristic in static flowshops with missing operations of jobs, *European Journal of Operational Research*, 131(1):622-634.

[34] Rhee,S.H., H.Bae and Y.Kim (2004) A dispatching rule for efficient workflow, *Concurrent Engineering*, 12(4):305-318.

[35] Sciomachen,A. and E.Tanfani (2007) A 3D-BPP approach for optimizing

stowage plans and terminal productivity, *European Journal of Operational Research*, 183(3):1433-1446.

[36] Sipser,M. (2006) Time complexity: Introduction to the theory of computation, *2nd edition*, USA, Thomson Course Technology press.

[37] Sule,D.R. (1996) Industrial Scheduling, *PWS Publishing*.

[38] Valente,J.M.S. and R.A.F.S.A. (2005) Improved heuristics for the early/tardy scheduling problem with no idle time, *Computers & Operations Research*, 32(3):557-569.

[39] Taillard,E. (1993) Benchmarks for basic scheduling problems, *European Journal of Operations Research*, 64:278-285.

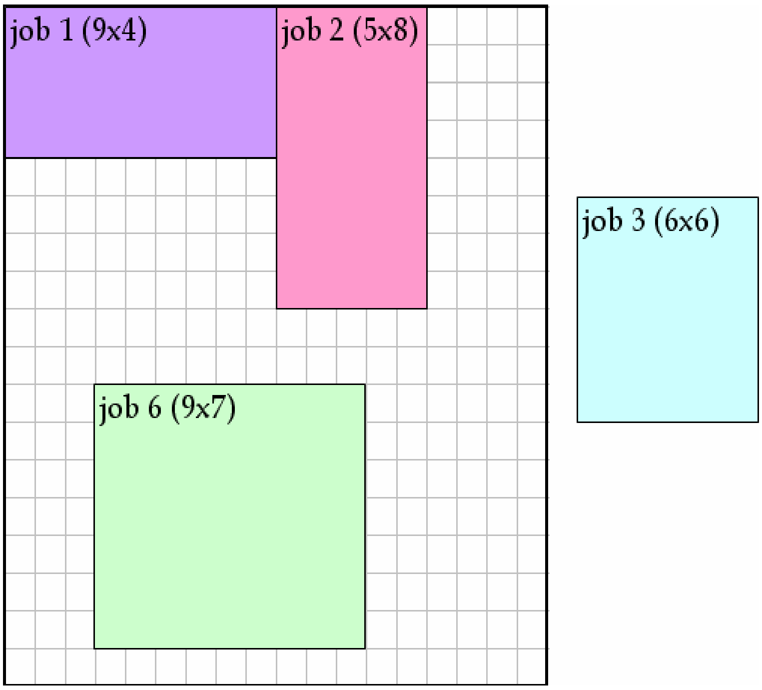
[40] Tay,F.E.H., T.Y.Chong and F.C.Lee (2002) Pattern nesting on irregular-shaped stock using genetic algorithms, *Engineering Applications of Artifical Intelligence*, 15:551-558.

[41] Yang,H.H. and C.L.Lin, (2007) On genetic algorithms for shoe making nesting-A Taiwan case, *Expert Systems with Applications*, Vol.doi:10.1016/j.eswa.2007.10.043, 2007, pp.-.

Appendix

Example of Longest Contact Edge Algorithm

After a sequence of job scheduling, we tend to assign job 3 into the working space (18×18 grids) with previous job, job 1, job 2 and job 6. The job shape of job1 is 9×4, job2 5×8, job6 9×7 and job3 6×6, the figures of the example are shown below:



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

Step1: Search free of reference point

In a factory, search grid (1,1), grid(1,2),…, grid(1,j), grid(2,1), grid (2,2), …, grid(18,18), further, if it is a feasible reference point, then search the job's working space.

Step 2: Search job3's working space

Search both the area of job3's length x job3's width and job3's width x job3's length. There are only nine reference points that are not occupied by previous jobs or obstacles, which were (1,14), (2,14), (3,14), (4,14), (13,10), (13,9), (13,8), (13,7) and (13,6). As a result, these reference points are candidate working area of job3.

$$S = \{(1,14), (2,14), (3,14), (4,14), (13,10), (13,9), (13,8), (13,7), (13,6)\}$$

Step 3: Evaluate every candidate working area

Based on Longest Contact Edge evaluation function (4), we compute the values from every candidate working area. All eight candidate values are listed below:

Candidate value of reference point (1,14)

$$\sum_{x=1}^{6} grid(x,8)+\sum_{x=1}^{6} grid(x,15)+\sum_{y=9}^{14} grid(0,y)+\sum_{y=9}^{14} grid(7,y)=3+6+6+0=15$$

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

Candidate value of reference point (2,14)

$$\sum_{x=2}^{7} grid(x,8)+\sum_{x=2}^{7} grid(x,15)+\sum_{y=9}^{14} grid(1,y)+\sum_{y=9}^{14} grid(8,y)=4+6+0+0=10$$

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

Candidate value of reference point (3,14)

$$\sum_{x=3}^{8} grid(x,8) + \sum_{x=3}^{8} grid(x,15) + \sum_{y=9}^{14} grid(2,y) + \sum_{y=9}^{14} grid(9,y) = 5+6+0+0 = 11$$

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

Candidate value of reference point (4,14)

$$\sum_{x=3}^{9} grid(x,8) + \sum_{x=3}^{9} grid(x,15) + \sum_{y=9}^{14} grid(3,y) + \sum_{y=9}^{14} grid(10,y) = 6+6+0+4 = 16$$

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

Candidate value of reference point (13,10)

$$\sum_{x=13}^{18} grid(x,4) + \sum_{x=13}^{18} grid(x,11) + \sum_{y=5}^{10} grid(12,y) + \sum_{y=5}^{10} grid(19,y) = 0+2+4+6 = 12$$

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

Candidate value of reference point (13,9)

$$\sum_{x=13}^{18} grid(x,3) + \sum_{x=13}^{18} grid(x,10) + \sum_{y=4}^{9} grid(12,y) + \sum_{y=4}^{9} grid(19,y) = 0+0+5+6 = 11$$

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

Candidate value of reference point (13,8)

$$\sum_{x=13}^{18} grid(x,2)+\sum_{x=13}^{18} grid(x,9)+\sum_{y=3}^{8} grid(12,y)+\sum_{y=3}^{8} grid(19,y)=0+0+6+6=12$$

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

Candidate value of reference point (13,7)

$$\sum_{x=13}^{18} grid(x,1)+\sum_{x=13}^{18} grid(x,8)+\sum_{y=2}^{7} grid(12,y)+\sum_{y=2}^{7} grid(19,y)=0+0+5+6=11$$

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

Candidate value of reference point (13,5)

$$\sum_{x=13}^{18} grid(x,0) + \sum_{x=13}^{18} grid(x,7) + \sum_{y=1}^{6} grid(12,y) + \sum_{y=1}^{6} grid(19,y) = 6 + 6 + +0 = 14$$

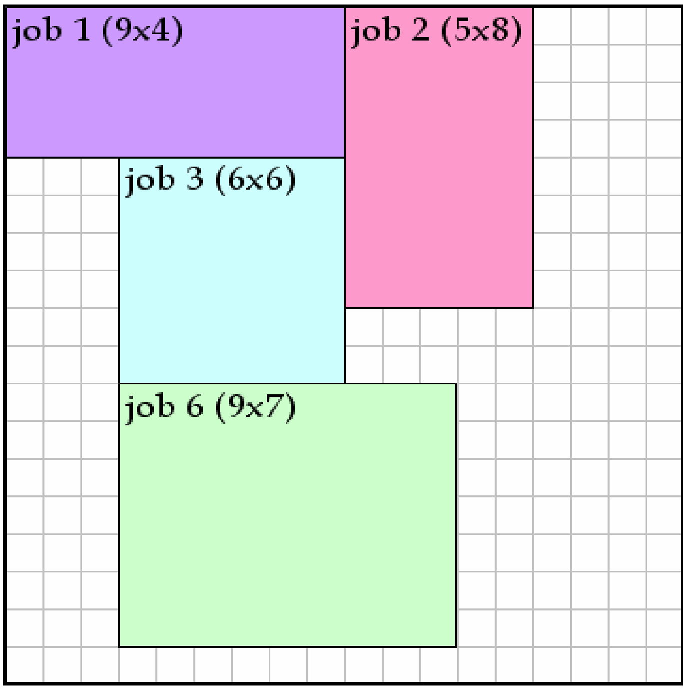
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
14	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1
13	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1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

Step 4: Choose the allocation reference point

Due to reference point (4,14) has the maximum value of evaluation function 16, we choose reference point (4,14) to allocate job3.

Step 5: Allocate job3

Allocate jo3 into the factory as figure below:


job 1 (9x4)
job 2 (5x8)
job 3 (6x6)
job 6 (9x7)