跡見学園マネジメント学部紀要　第２号　（2004年3月15日）

An Analytical Approach utilizing Monge Structure and its Application to Dynamical Aircraft Sequencing Problem

Teruaki Hanaoka

Abstract

In this paper we give an analytical method to the aircraft sequencing problem on a single runway so as to minimize the sum of the time interval called the time separation between successive landing aircraft in a final approach. If cost matrix of the separation time holds Monge properties, the optimum sequence of this problem is given by a non-heavier order of the types classed on aircraft take-off weight. Although the proposed method is a solution method for solving the static aircraft sequencing problem fundamentally, this method is extended to the dynamic aircraft sequencing problem here. The validity of this method is shown by comparison by the numerical examples with other methods containing the technique of the First Come-First Served strategy.

Key words: Aircraft sequencing, Monge property, Optimization problem, Dynamic Aircraft sequencing

1. Introduction

A demand for air traffic in the world in recent years is on a way of an increase. However the present condition is insufficient in a total volume of equipments of the runway for landing and take-off sufficient in order to fulfill such demand of the airtraffic in the main cities in the world. In this paper the method of improving the use efficiency of the runway classified into a short-term policy is considered, which is considered as policies relieving the confusion based on the increase in demand for air traffic. Nowadays, the research from a viewpoint which performs efficient use of the runway towards effective realization of the air traffic flow, especially the theoretical research of the methodology such as reducing a burden in a work of airport controllers can't be seen so much though it is very important.

The problem which determines a desirable landing order of the aircraft which entries into the terminal area of an airport is called the aircraft sequencing problem (ASP). When there are many aircraft, the ASP is difficult in calculation execution to obtain the optimum solution since the ASP is the problem called the NP-hard similar to the traveling-salesman problem. The ASP is classified in two types of "dynamic" and "static". Dear *et al.*, [5] discussed the dynamic problem in which the mixed composition of the type of aircraft waiting a landing within the station called a stack is changed every time. They have avoided an increase in the amount of calculation of dynamic problems by restricting the scope of its problem between each period, and limiting a movable quantity from the landing order of the aircraft by the First-Come First-Served (FCFS) which is a simple strategy to land the aircraft at the order of an arrival. The method which limited the maximum movable quantity from the landing order of the aircraft based on the FCFS is known as the Constrained Position Shifting (CPS), which is the concept first introduced by Dear *et al.*. This prespecified movable number is called the Maximum Position Shift (MPS). In the static problem they emphasize that all the aircraft taken into the consideration for landing should exist in the same stack at the same time so that arbitrary aircraft can land at any time. Psaraftis [6] which pointed out the importance of including all the aircraft taken into consideration formulated the single machine-N job group scheduling problem with the CPS by the dynamic programming (DP), and was applied to the static ASP. He denotes that the running time of the DP algorithm is both polynomial functions of the maximum number per group of aircraft classified by weight-class, and exponential functions of the number of groups N. This shows that his approach is effective to problems in which N is small.

In this paper we propose an analytical method to solving the ASP on a single runway so as to minimize the sum of the time interval called the *time separation* between successive landing aircraft in a final approach. We have slightly suspected as whether the optimum solution of the ASP would be making it land at the small order of the type of aircraft, experientially. In this paper, it is shown clearly by proof that the method is the optimum. Moreover, some sufficient conditions for this method being realized were considered.

The proposed analytical method may have a possibility of reducing a burden in a work of airport controllers because this method is possible to calculate the optimum permutation of landing aircraft immediately. In the paper this method is compared with other methods containing the FCFS strategy.

2. Statement of Problem

2.1 Aircraft sequencing problem

Consider a single runway as shown in Fig. 1, which is used for landing of aircraft. Aircraft paths are merged over the gate to the final approach. To prevent collision between successive aircraft and wake turbulence to a trailing aircraft j by a leading one i, a minimum distance separation $s_{e(i), e(j)}$ between i and j is strictly required, while they are both airborne. The Japanese CAB's air traffic control system standard [4] subdivides aircraft into three classes to $s_{e(i), e(j)}$ (Table 1). Those are the types of 'heavy' (H), 'large/medium' (L/M) and 'small' (S).

Table 1. Minimum distance separation (in nautical miles)

Leading aircraft, trailing aircraft:

$$(s_{e(i),e(j)}) = \begin{array}{c} \\ H \\ L/M \\ S \end{array} \begin{array}{c} H \quad L/M \quad S \\ \left[\begin{array}{ccc} 4 & 5 & 6 \\ 3 & 3 & 5 \\ 3 & 3 & 3 \end{array}\right] \end{array}$$

Table 2. Ground speed and runway occupancy time

Type	$v_{e(i)}$	$o_{e(i)}$
	(knot)	(sec)
H	v_H	70
	$\vee$	
L	v_L	60
	$\vee$	
M	v_M	55
	$\vee$	
S	v_S	50

Table 3. Minimum time separation (in seconds)

Leading aircraft, trailing aircraft:

$$(T_{e(i),e(j)}) = \begin{array}{c} \\ H \\ L \\ M \\ S \end{array} \begin{array}{c} H \quad L \quad M \quad S \\ \left[\begin{array}{cccc} 96 & 157 & 207 & 320 \\ 72 & 83 & 123 & 262 \\ 72 & 83 & 98 & 236 \\ 72 & 83 & 98 & 120 \end{array}\right] \end{array}$$

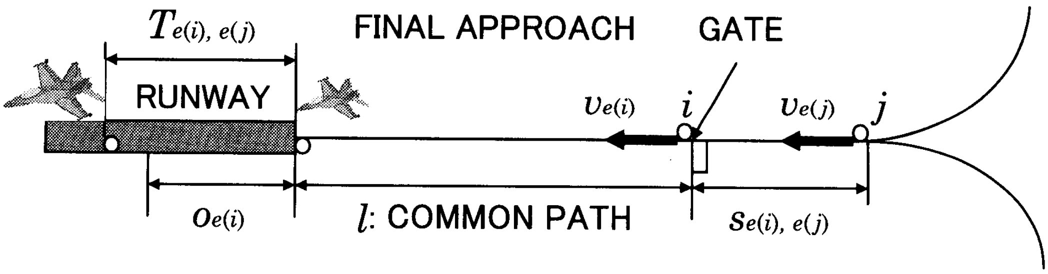


Figure 1. Final approach and runway

On the other hand a leading aircraft i must be safely out of runway before the next trailing aircraft j begins to touch down onto the runway. Using Blumstein's formula [2], the minimum distance separation $s_{e(i), e(j)}$ based such safety rules can be changed to the interval called the minimum time separation $T_{e(i), e(j)}$ between successive arrivals at the runway as that

$$T_{e(i), e(j)} = \max\left[\frac{l + s_{e(i), e(j)}}{v_{e(j)}} - \frac{l}{v_{e(i)}},\ o_{e(i)}\right] \quad \text{for } v_{e(i)} > v_{e(j)} \tag{1}$$

$$T_{e(i), e(j)} = \max\left[\frac{s_{e(i), e(j)}}{v_{e(j)}},\ o_{e(i)}\right] \quad \text{for } v_{e(i)} \leq v_{e(j)} \tag{2}$$

where l is the length of the common path in final approach, $v_{e(i)}$ and $o_{e(i)}$ are ground speed on the approach and runway occupancy time which are functions of the type of aircraft $e(i)$, respectively. Assuming $l=5$ nautical miles, $v_{e(i)}$ and $o_{e(i)}$ in Table 2, a matrix of the minimum time separation $T_{e(i), e(j)}$ is obtained (Table 3). Then, the ASP considered in this paper is the sequencing problem to the set of n given aircraft in a single runway, so as to minimize the total of the processing cost defined by the minimum time separation $T_{e(i), e(j)}$.

2.2 Formulation of the ASP

The ASP is defined as

$P0$: Minimize $\quad J=\sum_{i}\sum_{j} T_{e(i),e(j)} x_{ij} \quad (3)$

Subject to $\quad \sum_{j=1}^{n} x_{ij} \leq 1, \quad i=1, \ldots, n \quad (4)$

$$\sum_{i=1}^{n} x_{ij} \leq 1, \quad j=1, \ldots, n \quad (5)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}=n-1 \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad (7)$$

$x_{ij}\ (1 \leq i \neq j \leq n)$ doesn't generate any subtours more than one (8)

where

$$x_{ij}=\begin{cases} 1, & \text{if leading } i \text{ is followed by trailing } j \\ 0, & \text{otherwise.} \end{cases}$$

The objective function (3) minimizes the total minimum time separation. The assignment constraints (4) and (5) together with the integrality constraints (7) guarantee that at most one trailing aircraft is assigned to each leading aircraft and guarantee that at most one leading aircraft is assigned to each trailing aircraft, respectively. (6) denotes that exactly $n-1$ trailing aircraft ($n-1$ leading aircraft) exist in the permutation of n aircraft. The 'subtour' in the (8) is defined as a partial permutation except the complete permutation ordered by each of a set of n aircraft which makes a landing exactly once onto the runway. The subtour elimination constraints (8) guarantee that all subtours are removed, but the complete permutation isn't removed. However, the problem P0 has a structure similar to the Traveling Salesman Problem to be proven NP-hard by R. M. Karp in 1972.

3. Approach from Partial Assignment Problems

3.1 Relaxation problem of the ASP

Consider the following relaxation problem which has been removed the equation (8) from the P0, and replaced $T_{e(i),e(j)}$ by cost matrix c_{ij} which is equivalent to it.

$$P1: \text{Minimize} \quad J=\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{9}$$

$$\text{Subject to} \quad \sum_{j=1}^{n} x_{ij} \leq 1, \quad i=1, \ldots, n \tag{10}$$

$$\sum_{i=1}^{n} x_{ij} \leq 1, \quad j=1, \ldots, n \tag{11}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} = n-1 \tag{12}$$

$$x_{ij} \in \{0, 1\}, \quad i, j=1, \ldots, n \tag{13}$$

where x_{ij} is assignment variables which assign j to i. $x_{ij}=1$ if it assigns j to i and $x_{ij}=0$ otherwise. Each of elements x_{ij} of a matrix $X=(x_{ij})$ corresponds to each of elements c_{ij} of a cost matrix $C=(c_{ij})$. (9) is the total cost based on the whole assignments. The problem P1 is called the partial assignment problem (PAP) which assigns $n-1$ trailing aircraft to $n-1$ leading ones among n aircraft. It is obvious that any feasible solutions of P0 are always included in the feasible region of P1 because P1 is the relaxation problem of P0, therefore the feasible region of P1 is lager than that of P0. Thus, if the optimum solution of P1 has the complete permutation on n aircraft, this solution is the optimum solution of the original problem P0.

3.2 Property of cost matrix

In order to investigate the cost structure and properties in the cost matrix $C=(c_{ij})$, we give some definitions and prove some auxiliary theorems.

[Definition 3.1]: *totally monotone*

An $n \times n$ matrix $C=(c_{ij})$ is called a *totally monotone* if any $j<k$ satisfies $c_{ij} \leq c_{ik}$ and $c_{ji} \geq c_{ki}$ such that

$$C = \begin{bmatrix} c_{11} & \leq & c_{12} & \leq & \cdots & \leq & c_{1n} \\ \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \cdots & & \rotatebox{90}{$\leq$} \\ c_{21} & \leq & c_{22} & \leq & \cdots & \leq & c_{2n} \\ \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \cdots & & \rotatebox{90}{$\leq$} \\ \vdots & & \vdots & & \ddots & \vdots & \vdots \\ \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \cdots & & \rotatebox{90}{$\leq$} \\ c_{n1} & \leq & c_{n2} & \leq & \cdots & \leq & c_{nn} \end{bmatrix}$$

Then, the value of elements of C decreases left-downward.

[Definition 3.2]: *Monge matrix*

An $n \times n$ matrix $C=(c_{ij})$ is called a *Monge matrix* if C satisfies the so-called *Monge property* such that

$$c_{ij}+c_{i+k,\,j+l} \leq c_{i,\,j+l}+c_{i+k,\,j} \tag{14}$$

$$1 \leq i < i+k \leq n,\ 1 \leq j < j+l \leq n,\ k,\ l \geq 1$$

for the four elements $c_{ij},\ c_{i,\,j+l},\ c_{i+k,\,j},\ c_{i,+k,\,j+l}$ [3]. Here, let call an $(n-1)\times(n-1)$ matrix $C'=(c_{ij})(2\leq i\leq n,\ 1\leq j\leq n-1)$ the *left-lower* matrix of C. Moreover, let call an $1\times n$ matrix $C^{up}=(c_{ij})(i=1,\ 1\leq j\leq n)$, an $(n-1)\times n$ matrix $C^{low}=(c_{ij})(2\leq i\leq n,\ 1\leq j\leq n)$ and an $(n-1)\times 1$ matrix $C''=(c_{ij})(2\leq i\leq n,\ j=n)$ the *upper* matrix, the *lower* matrix and the *right−down* matrix of C, respectively.

[Definition 3.3]: *locally monotone*

An $n \times n$ matrix $C=(c_{ij})$ is called a *locally monotone* if any j and i in the $C=(c_{ij})$ satisfies the following two properties:

$$\text{(i) } c_{1j} \geq \max\{c_{ij} | 2 \leq i \leq n\},\ 1 \leq j \leq n \tag{15}$$

$$\text{(ii) } \max\{c_{ij} | 1 \leq j \leq n-1\} \leq c_{in},\ 1 \leq i \leq n.$$

Using those definitions, the following lemma is obtained.

[Lemma 3.1]

If an $n \times n$ matrix $C=(c_{ij})$ in the partial assignment problem P1 is *totally monotone*, then the problem P1 is reduced to an assignment subproblem P2 which is composed for an $(n-1)\times(n-1)$ left-lower matrix C', as shown below.

$P2$: Minimize $\quad J=\sum_{i=2}^{n}\sum_{j=1}^{n-1}c_{ij}x_{ij}$ (16)

Subject to $\quad \sum_{j=1}^{n-1}x_{ij}=1, \quad i=2, ..., n$ (17)

$\sum_{i=2}^{n}x_{ij}=1, \quad j=1, ..., n-1$ (18)

$x_{ij}\in\{0, 1\}, \quad i=2, ..., n$ (19)

$j=1, ..., n-1$

Proof: It is obvious because any assignments in C^{up} and C'' can replace the assignments having smaller cost in C' on the same columns and rows, respectively by the definition 3.1, (10) and (11).

[Lemma 3.2]

If an $(n-1)\times(n-1)$ *left−down* matrix C' is *Monge matrix* and has exactly one assignment $x_{ij}=1$ for any rows i and any column j in the $X'=(x_{ij})$ corresponding to c_{ij}, then the sum of the cost of $n-1$ diagonal elements $c_{ii}=(1\leq i\leq n-1)$ in the $C'=(c_{ij})(1\leq i, j\leq n-1$ for renumbering of i and j) is less or equal than the sum of the cost of $n-1$ elements c_{ij} corresponding to any assignments in the $X'=(x_{ij})$. Furthermore, this diagonal assignment serves as the optimum permutation of the cost minimum which consists of $n-1$ elements simultaneously.

Proof: Consider a set of two elements $x_{im}=1$ $(1\leq m\leq n-1)$ and $x_{li}=1$ $(1\leq l\leq n-1)$ on the row i and column i, respectively. If either $x_{im}=1$ or $x_{li}=1$ is a diagonal element, such two elements hold the same position, but otherwise by *Monge property*, we can obtain the following relation

$$c_{ii}+c_{lm}\leq c_{im}+c_{li} \tag{20}$$

Therefore, a more smaller sum of the cost $c_{ii}+c_{lm}$ is obtained by assigning 0 to x_{im} and x_{li} on those low i and column i and assigning 1 to x_{ii} and x_{lm}. By means of repeating those operations from $i=1$ to $i=n-1$ to the C' and X', finally the assignment by $n-1$ diagonal elements is obtained which is that of the minimum cost. This assignment constitutes the permutation which surely consists of $n-1$ elements.

[Lemma 3.3]

If an $n\times n$ matrix $C=(c_{ij})$ in the partial assignment problem P1 is *locally monotone*, then the problem P1 is reduced to an assignment subproblem P2 which is composed for an $(n-1)\times(n-1)$ left-lower matrix C'.

Proof: It is the same as the proof of Lemma 3.1 and proven by the definition 3.3, (10) and (11).

[Theorem 3.1]

If an $n\times n$ matrix $C=(c_{ij})$ is the *totally monotone* and *Monge matrix*, then the optimum assignment and its cost for the problem P1 are given by the $n-1$ diagonal elements x_{ii} and $c_{ii}(1\leq i\leq n-1)$ in the left-down matrices X' and C' respectively. This assignment serves as the optimum permutation of the Problem P1 simultaneously.

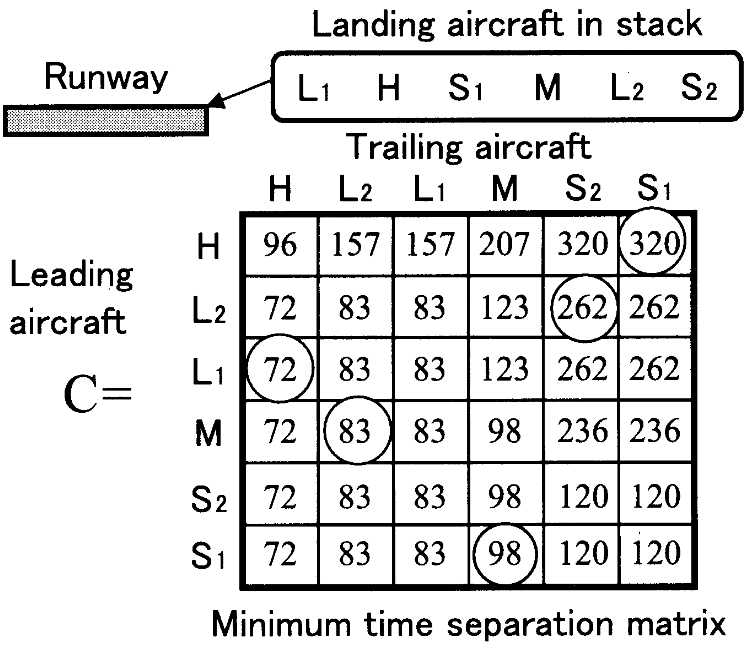
[Corollary 3.1]

If an $n\times n$ matrix $C=(c_{ij})$ is the *locally monotone* and *Monge matrix*, then the optimum assignment and its cost for the problem P1 is given by the $n-1$ diagonal elements x_{ii} and $c_{ii}(1\leq i\leq n-1)$ in the left-down matrices X' and C' respectively. This assignment serves as the optimum permutation of the Problem P1 simultaneously.

4. Structure of the ASP

4.1. Extension of cost matrix

Consider the case where six aircraft land onto the runway as shown in Fig. 2. Since the



	H	L2	L1	M	S2	S1
H	96	157	157	207	320	320
L2	72	83	83	123	262	262
L1	72	83	83	123	262	262
M	72	83	83	98	236	236
S2	72	83	83	98	120	120
S1	72	83	83	98	120	120

Figure 2. Assignment of landing order by matrix C

minimum time separation matrix ($T_{e(i),\, e(j)}$) generated by the Blumstein's formula is the 4×4 matrix, first, 6×6 cost matrix C for the six aircraft (n=6) is created using matrix ($T_{e(i),\, e(j)}$). In the matrix C, it assigns a row index one after another by the large order of the type of aircraft. It assigns similarly about columns of C. The aircraft of the same type assigns a following aircraft first. For example, the assignment of the case where the aircraft land in the order of $L_1 \rightarrow H \rightarrow S_1 \rightarrow M \rightarrow L_2 \rightarrow S_2$ is shown by round mark ○ in Fig. 2. The indices such as L_1, L_2 express the index of the aircraft number of the same type. The total of the number in the round marks is the total use time of the runway.

4.2. Structure of cost matrix

Since the matrix T shown in Table 3 has the properties of both *totally monotone* and *locally monotone*, it is clear that C generated by the completely same Blumstein's formula also has the properties of the both. As a special case, the portion with which it is not satisfied of *Monge property* in the matrix T exists. The place without *Monge property* (non-*Monge*) is a part of the small matrix which consists of four elements T_{HM}, T_{HS}, T_{LM} and T_{LS} of the rectangle containing the element T_{HS}. The cause which produces the portion of *non-Monge* mainly relates to the value of T_{HS}. That is, it is the case where the leading aircraft is the type of H and the trailing the type of S. This is equivalent to the case where the trailing aircraft is influenced most with vortex generated by the leading aircraft. In all the places except the element T_{HS} in the matrix T, the *Monge property* is satisfied.

4.3. Examinations on non-Monge matrix

However, as the proof of Remma 3.2, the influence of non-*Monge* is completely extinguished using the properties of *Monge* (14) and the *monotone* in the process which changes the cost sum of two elements into the much smaller cost sum of other two ones. If the following conditions are realized theoretically, the optimum solution of the ASP isn't influenced by the portion of non-*Monge* of T_{HS}.

$$(T_{HS} - T_{HH}) - (T_{LS} - T_{LH})$$

$$+ (T_{HS} - T_{HM}) - (T_{LS} - T_{LM}) \geq 0 \qquad (21)$$

The 3rd term and the 4th one of the left side of (21) are the portion without structure of *Monge.* Since the difference of the 1st term and the 2nd one are usually larger than that of the 4th and the 3rd, the (21) is satisfied. Therefore, the following theorem is obtained.

[Theorem 4.1]

For the aircraft sequencing problem P0 with $n \times n$ matrix $C=(c_{ij})$ made by the Blumstein's formula, the optimum assignment to the problem P0 is given by the $n-1$ diagonal elements $c_{ii}(1 \leq i \leq n-1)$ in the left-down matrix C'. Therefore the optimum sequence of this problem is given by a non-heavier order of the types classed on the aircraft take-off weight.

As shown in Fig. 3, this assignment of $S_1 \rightarrow S_2 \rightarrow M \rightarrow L_1 \rightarrow L_2 \rightarrow H$ shown by the round mark ○ makes only one sequence having the minimum cost. The theorem 4.1 shows that the permutation arranged in small order about the type of the aircraft is the optimum permutation.

5. Examples

5.1. Case of ***totally monotone***

The first example is related with the case in which the $n \times n$ matrix $C=(c_{ij})$ has the structure of the *totally monotone* and *Monge matrix.*

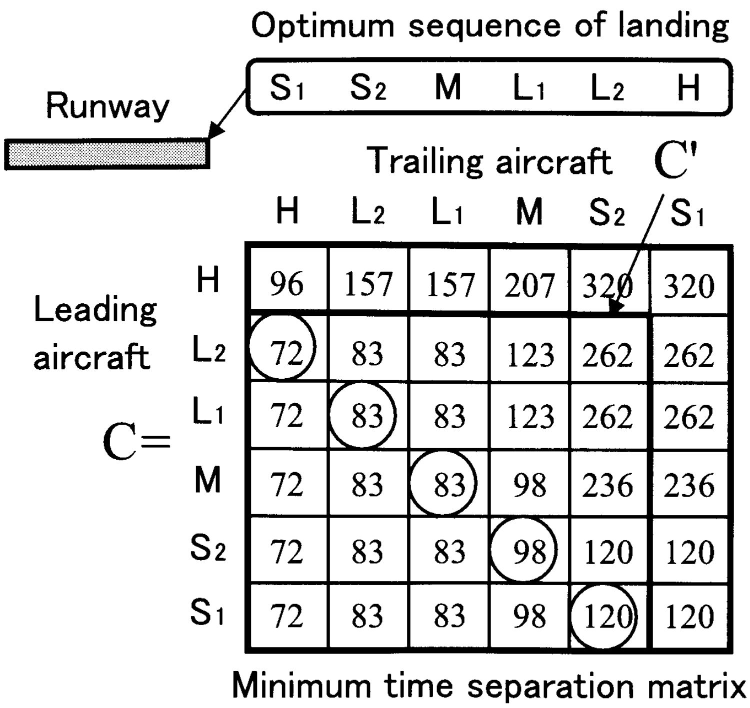


Figure 3. Assignment of optimum permutation of the ASP

we consider the example shown by [6]. Aircraft of $n=15$ classified into three types of H, L and M are waiting for landing. The minimum time separation matrix is given by Table 4 (in seconds). The problem is to find the optimum landing order to minimize the total sum of the minimum time separation between successive aircraft under the given cost matrix which has the structure of the *totally monotone* and the *Monge*. From the extended 15×15 cost matrix, we can obtain the optimum permutation which is given by the 14 diagonal elements in the *left-down* matrix C', ie. the optimum permutation of the landing aircraft is given by the non-decleasing order of the size of the aircraft.

Table 4. Minimum time separation (in seconds)

Leading aircraft, Trailing aircraft

$$(T_{e(i),\,e(j)}) = \begin{array}{c} \\ H \\ L \\ M \end{array} \begin{array}{c} \begin{array}{ccc} H & L & M \end{array} \\ \begin{bmatrix} 96 & 181 & 228 \\ 72 & 80 & 117 \\ 72 & 80 & 90 \end{bmatrix} \end{array}$$

The landing order and the total use time (in seconds) of the runway obtained by this method is shown with other methods in Table 5. In this table the MPS (5) is the approximation method with

Table 5. Example of the *totally monotone* ASP

Order	0	1	2	3	4	5	6	7	8
Initial	*L*	*H*	*H*	*M*	*L*	*L*	*M*	*L*	*H*
FCFS	*L*	*H*	*H*	*M*	*L*	*L*	*M*	*L*	*H*
MPS (5)	*L*	*H*	*H*	*H*	*L*	*L*	*L*	*L*	*M*
Proposed	*L*	*M*	*M*	*M*	*M*	*L*	*L*	*L*	*L*

9	10	11	12	13	14	15	use time	error (%)
L	*H*	*M*	*M*	*L*	*H*	*L*	—	—
L	*H*	*M*	*M*	*L*	*H*	*L*	1729	30.6
M	*M*	*M*	*L*	*L*	*H*	*H*	1400	5.8
L	*L*	*H*	*H*	*H*	*H*	*H*	1323	0

MPS=5 by [6]. The MPS is the DP-calculating method the CPS was incorporated. The proposed method does not include an error at all since this method constitutes the true optimum order. FCFS and MPS (5) methods include 5.8% and 30.6% of the error, respectively. The result shows that the optimum solution allows saving of about 6% to 30% on runway utilization.

Table 6. Minimum time separation (in seconds)

Leading aircraft $(T_{e(i),e(j)})=$	Trailing aircraft: H	L	M	S
H	96	181	200	228
L	72	70	100	130
M	72	70	80	110
S	72	70	80	90

5.2. Case of *locally monotone*

The second example is related with the case in which the $n \times n$ matrix $C=(c_{ij})$ has't the *totally monotone*. we consider the example from real-world problems shown by Bianco and Bielli [1]. Aircraft of $n=30$ classified into four types of H, L, M and S are waiting for landing. The minimum time separation matrix $(T_{e(i),e(j)})$ is given by Table 6 (in seconds). Therefore, matrix $C=(c_{ij})$ hasn't the structure of *totally monotone* since for example the $C=(c_{ij})$ includes the elements which is in the relation of $c_{21}>c_{22}$, $c_{31}>c_{32}$ and $c_{41}>c_{42}$. However the $C=(c_{ij})$ has the structure of *locally monotone* and *Monge matrix.* The problem is to find the optimum landing order to minimize the total sum of the minimum time separation between successive aircraft under the given cost matrix which is the *locally monotone* and the *Monge.*

Table 7 illustrate the results of this realistic large scale problem. The total landing time is 3266 seconds for FCFS, 3083 seconds for Bianco's method and 2578 seconds for the proposed method. Therefore, the FCFS, Bianco' method and our method include 26.7%, 19.6% and 0% of the error, respectively. These results show that the proposed method can get about 20-26% of the savings of time on runway utilization.

6. Dynamic Aircraft Sequencing Problem

In this chapter, we consider applying the proposed technique to the dynamic aircraft sequencing problem. In practice, since the aircraft arrives at an airport every moment, the

Table 7. Example of the *locally monotone* ASP

Aircraft No.	Type	FCFS (sec)	Proposed Method (optimum)			Bianco's Method		
			Landing seq.	Type	Landing time (sec)	Landing seq.	Type	Landing time (sec)
1	*H*	0	1	*H*	0	1	*H*	0
2	*H*	96	14	*S*	228	2	*H*	96
3	*H*	192	23	*S*	318	3	*H*	192
4	*L*	392	13	*M*	398	5	*H*	288
5	*H*	464	20	*M*	478	6	*H*	384
6	*H*	560	27	*M*	558	4	*L*	584
7	*L*	760	4	*L*	628	8	*H*	656
8	*H*	832	7	*L*	698	11	*H*	752
9	*L*	1032	9	*L*	768	7	*L*	952
10	*L*	1112	10	*L*	838	9	*L*	1032
11	*H*	1184	15	*L*	908	12	*H*	1104
12	*H*	1280	16	*L*	978	14	*S*	1332
13	*M*	1461	17	*L*	1048	10	*L*	1412
14	*S*	1591	24	*L*	1118	15	*L*	1492
15	*L*	1671	26	*L*	1188	16	*L*	1572
16	*L*	1751	28	*L*	1258	13	*M*	1642
17	*L*	1831	2	*H*	1330	18	*H*	1714
18	*H*	1903	3	*H*	1426	19	*H*	1810
19	*H*	1999	5	*H*	1522	20	*M*	1991
20	*M*	2180	6	*H*	1618	17	*L*	2091
21	*H*	2252	8	*H*	1714	23	*S*	2201
22	*H*	2348	11	*H*	1810	21	*H*	2273
23	*S*	2656	12	*H*	1906	22	*H*	2369
24	*L*	2656	18	*H*	2002	25	*H*	2465
25	*H*	2728	19	*H*	2098	24	*L*	2665
26	*L*	2928	21	*H*	2194	26	*L*	2745
27	*M*	2998	22	*H*	2290	27	*M*	2815
28	*L*	3098	25	*H*	2386	28	*L*	2915
29	*H*	3170	29	*H*	2482	29	*H*	2987
30	*H*	3266	30	*H*	2578	30	*H*	3083

solution method for the static aircraft sequencing problem described above needs to correct to the solution method of a dynamic sequencing problem. Since the proposed method can calculate an optimum landing order in slight time, it does not need to restrict the number of aircraft in a stack. However, the solution by this method has the problem from which the landing order of a large-sized aircraft becomes behind more.

Here, in order to avoid such a problem and to apply this method to the dynamic problem, how to adjust the size of a stack is considered. There are two in the method. One is the method of making the number of aircraft per stack a variable under fixation of the number of stacks. Another is the method of making the number of stacks a variable under fixation of the number of the aircraft per stack. Although the proposed method is applicable to the both, the latter case is considered here.

On our dynamic problem, number n of aircraft per stack is specified first. When n aircraft arrive into the first stack, the dynamic ASP to n aircraft is solved. Next, the stack is cleared. We call this stack the second stack. Then, after n following aircraft reach the 2nd stack, the 2nd dynamic problem is solved. This processing is performed about all the stacks. Actually, one stack can be substituted for all the stacks. Fig. 4 shows the time until the last aircraft ends landing, when changing the number of aircraft in the stack for the same example of *locally monotone* ASP. That is, it is the completion time of arrival of the aircraft. As shown in this figure, the completion time of arrival increases toward the completion time of the FCFS strategy gradually as the number of aircraft per stack decreases. However, if the number of aircraft per stack is

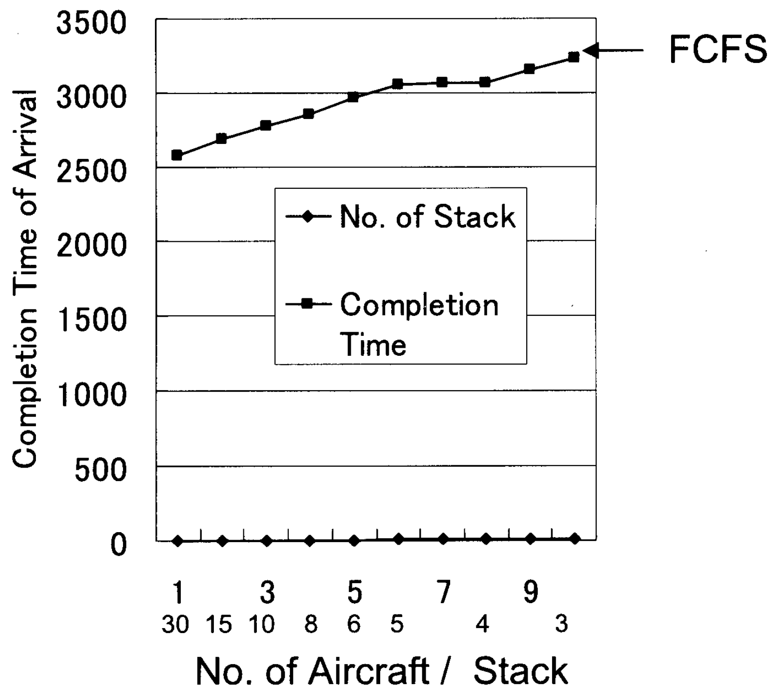


Figure 4. Completion Time of Arrival versus Number of Aircraft Contained in One Stack

reduced by increasing the number of stacks, although the total of landing time will increase, the problem that landing of a large-sized aircraft becomes behind is avoidable.

7. Conclusions

In this paper we explained an analytical method to obtain optimal solutions for the ASP on a single runway by directing our attention to the structure of the ASP and by considering the relaxation problem of the original ASP using the PAP.

It was shown that the optimal solution could be found easily in ASP which had *Monge structure.* On the other hand, the ASP so as to minimize the sum of the minimum time separation calculated by the Blumstein' formula has the structure of *monotone.* Therefore, The optimum sequence for the ASP can be obtained easily.

In this paper it is shown that the influence of non-*Monge* is completely extinguished by using the properties of the *Monge* and the *monotone.* This analysis shows that the optimum assignment to the ASP is given by the non-heavier order of the types classed on aircraft take-off weight. By this simple strategy, the optimum solution can be obtained immediately. However, in order to apply to realistic traffic, this method needs to extend or modify according to various situations.

The proposed method is extended to the dynamic aircraft sequencing problem. If the number of aircraft per stack is reduced by increasing the number of stacks, although the total of landing time will increase, the problem that landing of a large-sized aircraft becomes behind is avoidable.

Reference

[1] L. Bianco and M. Bielli. System aspects and optimization models in atc planning. 47-99, 1993.

[2] A. Blumstein. The landing capacity of a runway. *Operations Research,* 7, 752-763, 1959.

[3] R. E. Burkard, B. Klinz, and R. Rudolf. Perspectives of monge properties in optimization. Technical report, Karl-Franzens-Universität Graz & Technische Universität Graz., 1994.

[4] CAB. *Air traffic control system standard.* Obunshorin, Tokyo, 1999.

[5] R. Dear and Y. S. Sherif. Algorithm for computer assisted sequencing and scheduling of terminal area operations. *Transportation Research. Part A*, 25, 129-139, 1991.

[6] H. N. Psaraftis. A dynamic programming approach for sequencing identical groups of jobs. *Operations Research*, 28, 1347-1359, 1980.