The research reported in this document was made possible by support extended to the University of Illinois, Coordinated Science Laboratory, jointly by the Department of the Army (Signal Corps), Department of the Navy (Office of Naval Research), and the Department of the Air Force (Office of Scientific Research) under Signal Corps Contract DA-36-039-SC-85122.
ABSTRACT

The report considers optimization of landing order with respect to runway acceptance rate and total delay for planes desiring to land at the same one-runway airport. Two cases determined by the character of the minimum landing separation are recognized: (a) constant landing separation time and (b) variable landing separation time. For constant separation time, it is shown that the first-come-first-served landing order both maximizes the runway acceptance rate and minimizes the total delay. For variable separation time, a class of permutations allowing maximum runway acceptance rate is defined and rules for selecting a permutation yielding minimum total delay are given.
CONTENTS

<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1. The criteria</td>
<td>1</td>
</tr>
<tr>
<td>1.1. Total delay</td>
<td>2</td>
</tr>
<tr>
<td>1.2. Runway acceptance rate</td>
<td>5</td>
</tr>
<tr>
<td>2. Constant time separation</td>
<td>7</td>
</tr>
<tr>
<td>3. Variable time separation</td>
<td>10</td>
</tr>
<tr>
<td>3.1. Special case: constant distance separation</td>
<td>10</td>
</tr>
<tr>
<td>3.1.1. Two planes with no time restriction</td>
<td>11</td>
</tr>
<tr>
<td>3.1.2. Two planes with time restriction</td>
<td>13</td>
</tr>
<tr>
<td>3.1.3. More than two planes</td>
<td>14</td>
</tr>
<tr>
<td>3.1.4. An application</td>
<td>15</td>
</tr>
<tr>
<td>3.2. General case</td>
<td>19</td>
</tr>
<tr>
<td>Appendix</td>
<td>21</td>
</tr>
</tbody>
</table>
In developing a logic for automatic air traffic control there arises the problem of establishing a general procedure for setting the order of landing. In principle, an order optimizing any arbitrarily selected criterion can be determined by examining all possible permutations and selecting one that yields the optimal criterion value. In practice however, when the number of permutations is large, such a method takes too much time. This note gives a simplified, time-saving procedure for optimizing the landing order with respect to one or both of the selected criteria.

1. THE CRITERIA

The two criteria arbitrarily selected for consideration are total delay (the sum of the amounts of time each aircraft is kept waiting), and runway acceptance rate (number of planes landing on a given runway per unit time). If planes $P_1, P_2, P_3, \ldots$ desire to land at the same one-runway airport, it is assumed that each plane needs a minimum time $t_1, t_2, t_3, \ldots$ to land in absence of any other planes, and that the planes are required to land so as to maintain a minimum time separation $t_{i+1}$ between planes $P_{i-1}$ and $P_i$ landing in succession.

---

1 Preliminary work on automatic air traffic control carried out at the Coordinated Science Laboratory will be described comprehensively in reports R-145 and R-146.

2 Duane Cooper and Linton Kypta of the Coordinated Science Laboratory read a draft of the report and offered helpful suggestions.

3 Some other possible criteria are maximum delay of a single plane, maximum fuel consumption of a single plane while waiting, total fuel consumption of all planes while waiting, etc.
1.1. Total delay

For some landing order, say $P_1 P_2 P_3 \ldots$, the minimum time required to land the first $i$ planes is

$$s_i = t_i$$

(1) $s_i = s_{i-1} + \max (\alpha_i, t_1 - s_{i-1}), \text{ for } i = 2, 3, 4, \ldots$. 

For example, in the situation shown in Figures 1a and 1b, $s_3 = t_1 + t_2 + t_3$ for the $P_1 P_2 P_3 \ldots$ sequence and $s_3 = t_1 + (t_3 - t_1) + t_2$ for $P_1 P_3 P_2 \ldots$.

In general, the delay of the $i$-th plane in the sequence is

$$\text{delay of } P_i = \begin{cases} 
    s_i - t_i, & \text{if } s_i > t_i, \\
    0, & \text{if } s_i \leq t_i,
\end{cases}$$

which is

(2) $$\frac{1}{2}(s_i - t_1) + \frac{1}{2}|s_i - t_1|.$$ 

If we let "delay ($P_1 P_2 \ldots P_n$)" denote the total delay of the first $n$ planes resulting from the landing sequence $P_1 P_2 \ldots P_n P_{n+1} \ldots$, then this total delay is

(3) $$\text{delay (} P_1 P_2 \ldots P_n) = \sum_{i=1}^{n} \left[ \frac{1}{2}(s_i - t_1) + \frac{1}{2}|s_i - t_1| \right].$$
Minimum time

Actual time

\[ t_i = \text{minimum time to land needed by plane } P_i \]

\[ \hat{t}_i = \text{minimum separation time between planes } P_{i-1} \text{ and } P_i \text{ at landing} \]

\[ s_i = \text{actual time used to land the first } i \text{ planes} \]

*In this figure and in the ones that follow, the upper axis is used to plot, for each plane, the minimum time required to land the plane; the lower axis is used to plot the actual landing time of the planes which are landing in a prescribed sequence. For each plane, plots of minimum time and actual time are connected with a line to facilitate interpretation.*
Minimum time

Actual time

Figure 1b
1.2. Runway acceptance rate

Order the planes $P_{\alpha}, P_{\beta}, P_{\gamma}, \ldots$ by $t_1$ so that $t_1 \leq t_{i+1}$ and let this order be $P_1P_2P_3\ldots$. Then, the smallest $k$ such that

\[
t_1 + \sum_{i=2}^{k+1} \hat{t}_i < t_{k+1},
\]

defines the smallest group of planes $P_1, P_2, \ldots, P_k$ which can be landed without affecting the landing rate of the remaining planes. In Figure 1c, for example, $k = 6$.

Call the class of permutations of $P_1, P_2, \ldots, P_k$ which satisfy (4) class $C_1$. For example, permutations $P_1P_2P_3P_4P_5P_6$ (Figure 1c) and $P_1P_3P_2P_4P_5P_6$ (Figure 1d) belong to $C_1$, while $P_1P_2P_3P_5P_2P_4P_6$ (Figure 1e) does not. If we agree to calculate the runway acceptance rate for periods

---

**Figure 1c**
Landing order P, P3 P2 P4 P5 P6 P7

Figure 1d

Landing order P1 P3 P5 P2 P4 P6 P7

Figure 1e
greater than $t_{k+1}$, all the permutations in $C_1$ can be considered equivalent with respect to their effect on the acceptance rate, for none of these permutations affects the landing rate of the remaining planes. The overall runway acceptance rate, therefore, can be maximized simply by taking only permutations in $C_1$.

Permutations in $C_1$, in turn, can be optimized with respect to another criterion. Below, minimization of total delay for permutations in $C_1$ is considered for two cases: the case in which the minimum separation time is the same (constant) for all aircraft, and the case in which it depends in some way upon which plane is being considered (variable). Instances in which it is known that a strictly minimum delay permutation belongs to $C_1$ are indicated.

2. CONSTANT TIME SEPARATION

If we let $\hat{t}_1 = \hat{t}_2 = \ldots = \hat{t}_k = \hat{t}$ and, as before, assume $P_1P_2P_3\ldots$ ordered so that $t_1 \leq t_{i+1}$, the condition defining $k$ for $C_1$ becomes

$$(4a) \quad t_1 + k\hat{t} < t_{k+1}.$$  

There may exist in $C_1$ permutations which produce landing separations greater than the required minimum, that is, permutations for which

$$(5) \quad t_1 + q\hat{t} < t_i, \quad i < k,$$

where $q$ is the number of planes ahead of $P_1$ in a given sequence. Let such permutations constitute class $C_2$, a subclass in $C_1$. Permutations in $C_2$, however, produce more total delay than those that are not in $C_2$, since
every permutation in $C_2$ must be of the form $P_1P_2 \ldots P_i \ldots P_j \ldots P_k$, with $t_j < t_i$. For example, permutation $P_1P_3P_2P_4P_5P_6$ shown in Figure 1d belongs to $C_2$ and has $P_2$ as the $P_j$ plane.

To observe the delay characteristics of such permutations, consider permutation

$$c_1 = P_1P_2 \ldots P_{i^*} \ldots P_j \ldots P_{n} \ldots P_k,$$

where $i^*$ is the smallest $i$ for which $c_1$ satisfies (5) and $t_j < t_{i^*}$. Transposing the $P_j$ for which $t_{i^*-1} < t_j < t_{i^*}$ to obtain

$$c_2 = P_1P_2 \ldots P_{i^*-1}P_jP_{i^*} \ldots P_{n} \ldots P_k,$$

has the consequences that (a) the total delay of $P_1, P_2, \ldots, P_{i^*-1}$ remains unchanged, (b) the delay of $P_j$ is decreased by more than $r$, where $r$ is the number of planes between $P_j$ and $P_{i^*-1}$ before the transposition, (c) the total delay of $P_{i^*}, P_{i^*+1}, \ldots, P_{j-1}$ was increased by less than $r$, and (d) the total delay of $P_{j+1}, P_{j+2}, \ldots, P_k$ was at most decreased. Therefore,

$$\text{delay (c_1)} > \text{delay (c_2)}.$$

If $c_2$ belongs to $C_2$, the process is repeated until a permutation $c_n$ which does not belong to $C_2$ is obtained, so that the permutations are ordered by delay:

$$\text{delay (c_1)} > \text{delay (c_2)} > \ldots > \text{delay (c_n)}.$$

---

1 As an example, consider transposing $P_2$ in Figure 1d to obtain permutation $P_1P_3P_2P_4P_5P_6$. 

---
Thus, if we let \( C_1 - C_2 = C_3 \), for every permutation in \( C_2 \) we can find a permutation \( c_n \) in \( C_3 \) which produces less total delay. Similarly, it can be shown that permutations not in \( C_1 \) result in more delay than those in \( C_3 \).

The expression for total delay is

\[
\text{delay}(P_1, P_2, \ldots, P_k) = \sum_{i=1}^{k} \left[ \frac{1}{2}(s_i - t_i) + \frac{1}{2}|s_i - t_i| \right],
\]

but since \( s_i \geq t_{i-1} \) for \( i \leq k \) for permutations in \( C_3 \) this reduces to

\[
\sum_{i=1}^{k} (s_i - t_i),
\]

which is

\[
\sum_{i=1}^{k} \left[ t_1 + (i-1)t^n - t_i \right],
\]

with the result upon summation:

\[
(6) \quad \text{delay}(P_1, P_2, \ldots, P_k) = kt_1 + \frac{1}{2}k(k-1)t^n - \sum_{i=1}^{k} t_i.
\]

Since the value of (6) does not depend on the order of summation, the total delay for all permutations in \( C_3 \) is constant. Furthermore, since permutations not in \( C_3 \) result in greater delay, the expression in (6) gives the minimum total delay.

In particular, the intuitively desirable first-to-land-first-served permutation with \( t_1 \leq t_2 \leq \ldots \leq t_n \) belongs to \( C_3 \) and consequently yields minimum total delay. Thus, with constant time landing separation, the first-to-land-first-served sequencing yields minimum total delay.
3. VARIABLE TIME SEPARATION

The optimization of landing order for variable time separation is somewhat less straightforward and, except for two instances, the rules given are complex. First, the special case of the frequently employed constant distance separation is discussed and an application described; then, the rules for the general case are given.

3.1. Special case: constant distance separation

When constant distance is used to determine the minimum separation, a point M on the line of final approach is selected and, at any given instant, only one plane is allowed between the point M and the turn-off point on the runway. In such a case, let \( m_i \) be the minimum arrival time of plane \( P_i \) at the point M and \( t_i \), as before, the minimum time to land. Then, \( t_i - m_i > 0 \) and the minimum landing time separation between the \( P_{i-1} \) and \( P_i \) planes is \( t_i - m_i \). Substituting \( t_i - m_i \) for \( t_i \) in (4), we find the condition defining \( k \) for class \( C_1 \) to be

\[
 t_1 + \sum_{i=2}^{k} (t_i - m_i) + (t_{k+1} - m_{k+1}) < t_{k+1},
\]

which is

\[
(4b) \quad t_1 + \sum_{i=2}^{k} (t_i - m_i) < m_{k+1}.
\]

---

1 Work is now in progress to determine whether, given a wide variation in the terminal area speeds, a reasonably small constant time landing separation can be arranged for an automatic system.
The procedure for selecting the delay minimizing permutation from $C_1$ is introduced by deriving the rules for the simple case of two planes with no time restrictions. Then, the rules for two planes with time restriction are stated and are extended to the general case of $n$ planes.

### 3.1.1 Two planes and no time restriction

In the case of two planes $P_1$ and $P_2$ with $t_1, m_1, t_2, m_2$ and a free runway, the total delay can be minimized by forming $t_i + m_i$ for $i = 1, 2$ and landing the plane with the smaller sum first. To show it, we obtain from (2)

$$
\text{delay } (P_1 P_2) = \begin{cases} 
  t_1 - m_2, & \text{if } m_2 \leq t_1, \\
  0, & \text{if } m_2 > t_1,
\end{cases}
$$

$$
\text{delay } (P_2 P_1) = \begin{cases} 
  t_2 - m_1, & \text{if } m_1 \leq t_2, \\
  0, & \text{if } m_1 > t_2.
\end{cases}
$$

The relationship between delay $(P_1 P_2)$ and delay $(P_2 P_1)$ can be expressed as a function of relationships between $m_i$ and $t_j$, $i \neq j$, as follows:

(a) delay $(P_1 P_2) \leq$ delay $(P_2 P_1)$ when either

(i) $m_2 > t_1$ and $m_1 \leq t_2$, or

(ii) $m_2 \leq t_1$ and $m_1 \leq t_2$ and $t_1 - m_2 \leq t_2 - m_1$, 

(b) delay \(P_1P_2\) > delay \(P_2P_1\) when either

(iii) \(m_2 < t_1\) and \(m_1 > t_2\), or

(iv) \(m_2 < t_1\) and \(m_1 < t_2\) and \(t_1 - m_2 > t_2 - m_1\).

Conditions (i), (ii), (iii), and (iv) are mutually exclusive; also, they are exhaustive since condition \(m_2 > t_1\) and \(m_1 > t_2\) cannot occur because of the constraint that \(m_1 < t_1\).

When either (i) or (ii) is true, \(t_1 + m_1 < t_2 + m_2\) is true, or in symbols

\[(i) \text{ or } (ii) \implies t_1 + m_1 < t_2 + m_2,\]

for

\[m_2 > t_1\] and \(m_1 < t_2 \implies t_1 + m_1 < t_2 + m_2,\]

and

\[t_1 - m_2 < t_2 - m_1 \implies t_1 + m_1 < t_2 + m_2.\]

Also, when \(t_1 + m_1 < t_2 + m_2\) is true, either (i) or (ii) is true, or in symbols

\[t_1 + m_1 < t_2 + m_2 \implies (i) \text{ or } (ii),\]

because when \(t_1 + m_1 < t_2 + m_2\), (iii) is false, since

\[m_2 < t_1\] and \(m_1 > t_2 \implies t_1 + m_1 > t_2 + m_2;\]

and so is (iv), since

\[t_1 + m_1 < t_2 + m_2 \implies t_1 - m_1 < t_2 - m_2.\]
Therefore, we have
\[ t_1 + m_1 \leq t_2 + m_2 \iff (i) \text{ or } (ii), \]
or
\[ t_1 + m_1 \leq t_2 + m_2 \iff \text{delay } (P_1P_2) \leq \text{delay } (P_2P_1). \]

3.1.2 Two planes with time restriction

For two planes \( P_1 \) and \( P_2 \) with \( t_1, m_1, t_2, \) and \( m_2 \) as before, but with the added restriction that the immediately preceding plane lands at time \( s_p \), the rules can be derived from a straightforward examination of the list of the 64 possible combinations and an application of results of the preceding section. The rules are

I: For \( m_i < s_p, i = 1, 2 \)
\[ t_1 - m_1 \leq t_2 - m_2 \iff \text{delay } (P_1P_2) \leq \text{delay } (P_2P_1). \]

II: For \( m_i \leq s_p \) and \( m_j > s_p \)
\[ 2(s_p - m_i) + (t_1 + m_1) \leq (t_j + m_j) \iff \text{delay } (P_1P_j) \leq \text{delay } (P_jP_1), \]
or
\[ 2(s_p - m_j) + (t_1 - m_1) \leq (t_j - m_j) \iff \text{delay } (P_iP_j) \leq \text{delay } (P_jP_i). \]

III: For \( m_i > s_p, i = 1, 2 \)
\[ t_1 + m_1 \leq t_2 + m_2 \iff \text{delay } (P_1P_2) \leq \text{delay } (P_2P_1). \]

Note that under III, neither plane is able to land before time \( s_p + t_1 \) and the case is equivalent to that of the preceding section, i.e. to two planes with no time restriction.
3.1.3 More than two planes

The rules for two planes may be applied, with some modifications, to obtain minimal delay for \( n \) planes, with \( n > 2 \). The procedure consists essentially of three steps, the first two of which yield successive approximations and the third, by direct examination of the remaining possibilities, the final order. These steps are

(a) First, order the planes by \( t_i \), with \( t_i < t_{i+1} \) to reduce, in general, the number of transpositions required in the next step.

(b) Second, examine pairs of adjacent planes \( \ldots P_i P_h \ldots \) and proceed as follows:

(i) If Rule I applies, arrange the order which minimizes delay as indicated by the rule.

(ii) If Rule II applies, use order \( \ldots P_i P_h \ldots \) if \( m < s \); otherwise, use \( \ldots P_i P_h \ldots \).

(iii) If Rule III applies, use \( \ldots P_i P_h \ldots \) if \( t_i < t_h \); otherwise, use \( \ldots P_i P_h \ldots \).

Continue step (b) until no more transpositions are required. The resulting order (which must be in \( C_1 \) because in each instance minimum landing separation was used) is optimal for the special cases in which strict use of Rules II and III would yield the same order. Otherwise, it is approximately optimal and step (c) must be invoked:

(c) Third, starting with \( P_1 P_2 \), re-examine all pairs of adjacent planes \( \ldots P_i P_h \ldots \) in succession and, whenever Rule II or III applies and disagrees with the present order, compare delay \( (P_i P_h P_{i+1} \ldots P_k) \) with delay \( (P_h P_i P_{h+1} \ldots P_k) \) and select the order giving the smaller delay,
provided the order is in \(C_1\). After a necessary transposition, changing \(\ldots P_i P_j P_k \ldots\) to \(\ldots P_i P_j P_k \ldots\), backtrack one pair and resume examination with the pair \(\ldots P_i P_j \ldots\). The resulting order is optimal, for any additional transposition which gives a permutation in \(C_1\) either increases or leaves unchanged the total delay. In general, the order strictly minimizing total delay can be obtained by disregarding the restriction that the selected order be in \(C_1\).  

Neglecting the time required by step (a), steps (b) and (c) can be carried out by a computer in \(k\) passes over the list of \(k\) planes. It is estimated that each pass, except the last one, would require approximately time \(T\), where \(T\) is the time needed for calculation of total delay for one permutation of \(P_1, P_2, \ldots, P_k\). The last pass, it is estimated, would require not more than \((\frac{1}{4}k^2 + \frac{1}{2}k)T\). Therefore, the whole procedure could be executed approximately in time \((\frac{1}{4}k^2 + \frac{3}{2}k)T\), which is less than \(k!T\) for \(k > 3\). \(^1\) (\(k!T\) is the time required to calculate total delay for all the permutations of \(k\) planes.) In some applications, one of which is described below, it is possible to introduce further simplification of the procedure.

### 3.1.4. An application

In an experimental air-traffic-control logic programmed for a digital computer, the determination of landing order was carried out in the terminal area. The abstracted terminal area used contained the runway, the

---

1. At the time of writing, the estimates have not been verified by programming.

2. The program and the experimental framework are described in the previously cited Coordinated Science Laboratory Report R-146.
outer marker (OM), the gate, and four entry points deployed as shown in Figure 2. The planes were required to land so that at any time there would be only one plane between the outer marker and the turn-off point on the runway. A typical traffic pattern is shown in Figure 3.

The planes, upon reaching an entry point, were put on "hold" and at the same time became eligible to receive instructions to commence the approach. The problem was to issue these instructions in an order which minimized total delay for the planes waiting at the entry points. The solution turned out to be extremely simple and consisted of ordering the planes by the terminal area speed.  

To verify the solution, observe that (a) all the entry points are equidistant from the outer marker if the distance is measured along the approach paths, and (b) that for any plane $P_i$ with terminal area speed $v_i$, $t_i - m_i$ is inversely proportional to $v_i$.

If Rule I applies, we have

$$t_i - m_i \leq t_j - m_j \iff \text{delay } (P_i P_j) \leq \text{delay } (P_j P_i),$$

or

$$v_i \geq v_j \iff \text{delay } (P_i P_j) \leq \text{delay } (P_j P_i).$$

and if Rule II applies, we have

$$2(s_p - m_i) + (t_i + m_i) \leq (t_j + m_j) \iff \text{delay } (P_i P_j) \leq \text{delay } (P_j P_i),$$

or

$$2(s_p - m_j) + (t_i - m_i) \leq (t_j - m_j) \iff \text{delay } (P_i P_j) \leq \text{delay } (P_j P_i).$$

---

3 Ordering by speed alone in this case theoretically allows some planes to be held at an entry point indefinitely. In the experimental program the maximum waiting time was monitored and, whenever it exceeded a preset number, ordering by speed was replaced with ordering by waiting time.
But, for \( s_p > m_i \), we obtain

\[
2(s_p - m_i) + (t_i + m_i) \leq (t_j + m_j) \implies (t_i + m_i) \leq (t_j + m_j),
\]

or

\[
2(s_p - m_i) + (t_i + m_i) \leq (t_j + m_j) \implies v_i \geq v_j.
\]

Also, we have

\[
v_i \geq v_j \implies (t_i - m_i) \leq (t_j - m_j)
\]

and, for \( s_p \leq m_j \),

\[
(t_i - m_i) \leq (t_j - m_j) \implies 2(s_p - m_j) + (t_i - m_i) \leq (t_j - m_j).
\]

Therefore, for Rule II we obtain

\[
v_i \geq v_j \iff \text{delay } (P_iP_j) \leq \text{delay } (P_jP_i).
\]

If Rule III applies, we have

\[
t_i + m_i \leq t_j + m_j \iff \text{delay } (P_iP_j) \leq \text{delay } (P_jP_i),
\]

or

\[
v_i \geq v_j \iff \text{delay } (P_iP_j) \leq \text{delay } (P_jP_i).
\]

Thus, regardless of which rule applies, we obtain

\[
v_i \geq v_j \iff \text{delay } (P_iP_j) \leq \text{delay } (P_jP_i),
\]

and so ordering by speed minimizes total delay. As a partial check of the validity of these results an alternate derivation applicable to groups of planes with \( m_i \leq s_p \) for all planes is given in the Appendix.
3.2. The general case of variable time separation

The transition from the constant distance landing separation to the general case of variable time separation is easily accomplished by a straightforward substitution of

\[(7) \quad \hat{t}_i = t_i - m_i.\]

For two planes, using (7) in the rules of Section 3.1.2, we obtain

Ia: for \(t_i - \hat{t}_i \leq s_p, \ i = 1, 2,\)

\[\hat{t}_i \leq \hat{t}_2 \iff \text{delay (} P_1 P_2 \text{)} \leq \text{delay (} P_2 P_1 \text{);}\]

IIa: for \(t_i - \hat{t}_i \leq s_p \) and \(t_j - \hat{t}_j > s_p,\)

\[\frac{1}{2}(\hat{t}_1 + \hat{t}_2) \leq t_2 - s_p \iff \text{delay (} P_1 P_2 \text{)} \leq \text{delay (} P_2 P_1 \text{);}\]

IIIa: for \(t_i - \hat{t}_i > s_p, \ i = 1, 2,\)

\[t_1 - t_2 \leq \frac{1}{2}(\hat{t}_1 - \hat{t}_2) \iff \text{delay (} P_1 P_2 \text{)} \leq \text{delay (} P_2 P_1 \text{)}.\]

To obtain the optimal order for \(n\) planes, first order the planes by \(t_i\) so that \(t_i < t_{i+1}\). Then, examine pairs of adjacent planes \(\ldots P_i P_{i+1}\) and

(i) if Rule Ia applies, use the order indicated by the rule;

(ii) if Rule IIa applies, use order \(\ldots P_i P_{i+1}\) if \(t_i - \hat{t}_i < s_p,\) otherwise, use \(\ldots P_i P_{i+1}\);

(iii) if Rule IIIa applies, use \(\ldots P_i P_{i+1}\) if \(t_i < t_{i+1}\), otherwise use \(\ldots P_i P_{i+1}\).
Continue this procedure until no more transpositions are required.
The resulting order, as in Section 3.1.3, belongs to $C_1$ and is optimal for a class of special cases and approximately optimal in general. To obtain the optimal order in the general case, complete the step (c) of the procedure described in Section 3.1.3.
APPENDIX

Special case: \( m^{*+j} \leq s^{*} \) for \( i^{*} + j \leq k \). If in some permutation in \( C_{i} \) there is a \( P^{*} \) such that \( m_{i} \leq s_{i}^{*} \) for \( i > i^{*} \), then the delay of \( P^{*+1}, P^{*+2}, \ldots, P_{k} \) can be minimized by ordering by \( t_{i} - m_{i} \) and landing first the plane corresponding to the smallest \( t_{i} - m_{i} \). Figure 4 shows an example of such a sequence with \( P_{i} \) as the \( P^{*} \) plane.

We obtain from (3)

\[
\text{delay} (P_{1}P_{2} \ldots P_{k}) = \sum_{i=1}^{k} \left[ \frac{1}{2}(s_{i} - t_{i}) + \frac{1}{2}|s_{i} - t_{1}| \right],
\]

or

\[
\sum_{i=1}^{i^{*}} \left[ \frac{1}{2}(s_{i} - t_{i}) + \frac{1}{2}|s_{i} - t_{1}| \right] + \sum_{i=i^{*}+1}^{k} \left[ \frac{1}{2}(s_{i} - t_{i}) + \frac{1}{2}|s_{i} - t_{1}| \right]
\]

which is, since \( s_{i} \geq t_{1} \) for \( i > i^{*} \),

\[
\sum_{i=1}^{i^{*}} \left[ \frac{1}{2}(s_{i} - t_{i}) + \frac{1}{2}|s_{i} - t_{1}| \right] + \sum_{i=i^{*}+1}^{k} (s_{i} - t_{i}).
\]

Thus, given \( s_{i}^{*} \) the delay of \( P^{*+1}, P^{*+2}, \ldots, P_{k} \) is determined by

\[
\sum_{i=i^{*}+1}^{k} (s_{i} - t_{i}). \quad \text{However, this may be expanded as}
\]

\[
\sum_{i=i^{*}+1}^{k} (s_{i} - t_{i}) = (s_{i^{*}+1} - t_{i^{*}+1}) + (s_{i^{*}+2} - t_{i^{*}+2}) + \ldots + (s_{k} - t_{k}),
\]

or, by substituting for \( s_{i^{*}+1}, s_{i^{*}+2}, \ldots, s_{i^{*}+k} \), as
Figure 4
Collecting like terms, we obtain
\[
(k-1) s_i + \left[ k-(i+1) \right] t_{i+1} + \left[ k-(i+2) \right] t_{i+2} + \cdots + \left[ k-(k-1) \right] t_{k-1} + \\
-(k-i) m_{i+1} - \left[ k-(i+1) \right] m_{i+2} - \cdots - \left[ k-(k-1) \right] m_k
\]
and, grouping by coefficients,
\[
(k-1) s_i + \left[ k-(i+1) \right] (t_{i+1} - m_{i+1}) + \left[ k-(i+2) \right] (t_{i+2} - m_{i+2}) + \\
+ \cdots + \left[ k-(k-1) \right] (t_{k-1} - m_{k-1}) - m_{i+1} - m_{i+2} - \cdots - m_{k-1} - m_k
\]
which can be written as
\[
(k-1) s_i + \sum_{i=i+1}^{k} (k-i) (t_i - m_i) - \sum_{i=i+1}^{k} m_i
\]
In (8), only the value of \( \sum_{i=i+1}^{k} (k-i) (t_i - m_i) \) depends on the landing order of \( P_{i+1}, P_{i+2}, \ldots, P_k \). Given \( s_i \), the total delay of \( P_{i+1}, P_{i+2}, \ldots, P_k \) can be minimized, therefore, by ordering the planes by \( t_i - m_i \), so that \( t_i - m_i \leq t_{i+1} - m_{i+1} \). In particular, if \( t_i - m_i \) is inversely proportional to the speed, the delay can be minimized by ordering by speed alone.