CE 191: Civil and Environmental Engineering Systems Analysis

LEC 09 : Mixed Integer Programming

Professor Scott Moura
Civil & Environmental Engineering
University of California, Berkeley

Fall 2014

Intro to Mixed Integer Programming

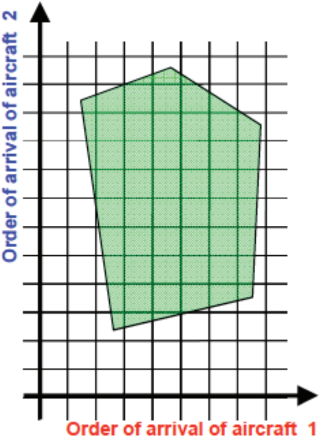
Problem Statement

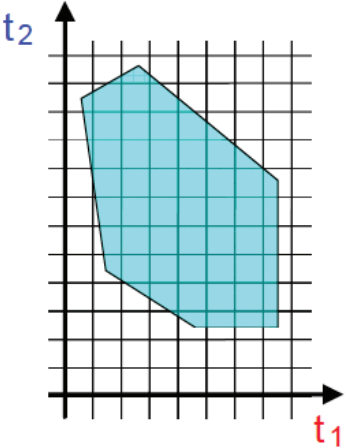
Some decision variables are integers, some are not. Suppose $x = [x_1, x_2, x_3, x_4]^T$.

$$\begin{aligned} \min \quad & c^T x \\ \text{s. to} \quad & Ax \leq b \\ & x_1, x_2 \in \mathbb{R} \\ & x_3, x_4 \in \mathbb{Z} \end{aligned}$$

Graphical Interpretation

Some variables are integer, some are not.



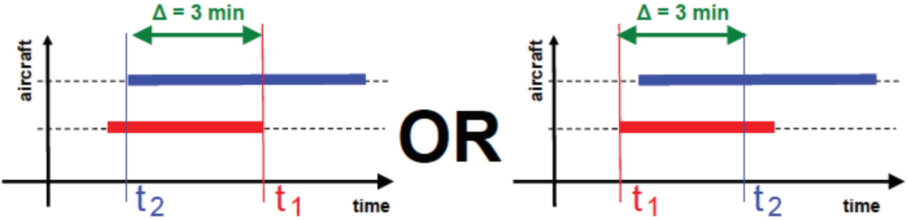


The solution for these two variables has to be on the grid.

It does not matter where the solution for these two variables is

Queueing airplane landings - Decide the order

Decision variable: order of arrival



$$t_1 - t_2 \geq \Delta \qquad\qquad t_2 - t_1 \geq \Delta$$

Another way to express this:

$$|t_1 - t_2| \geq \Delta$$

Absolute values

Absolute values of this form are not linear, not affine. That is, they're difficult.

$$|t_1 - t_2| \geq \Delta$$

Absolute values can be expressed as a **logical disjuntion**.
This is a mathematical way of saying "OR".

$$t_2 - t_1 \geq \Delta \qquad \text{OR} \qquad t_1 - t_2 \geq \Delta$$
$$\text{if} \quad t_2 \geq t_1 \qquad\qquad \text{otherwise}$$

Or put more simply

$$|t_1 - t_2| \geq \Delta$$

AND is easy, **OR** is difficult

Reminder: you have already used **AND** many times

$$
\begin{aligned}
\text{min:} \quad f(x_1, x_2) &= c_1 x_1 + a_2 x_2 \\
\text{s. to:} \quad a_1 x_1 &\le c_{max} \\
a_2 x_2 &\ge a_{min} \\
a_1 x_1 + a_2 x_2 &\ge a_{min} \\
a_2 x_2 &\ge 2 a_1 x_1 \\
a_2 x_2 &\le 2 a_1 x_1
\end{aligned}
$$

All of the constraints are logical **AND**.

Transformation of **OR** into an **AND**

Pick a very large number M.
Also consider a decision variable $d \in \{0, 1\}$.

For sufficiently large M, the following two statements are equivalent:

Statement 1:

OR $$\begin{cases} t_1 - t_2 \geq \Delta & \text{if } t_1 \geq t_2 \\ t_2 - t_1 \geq \Delta & \text{o.w.} \end{cases}$$

Statement 2:

AND $$\begin{cases} t_1 - t_2 \geq \Delta - Md \\ t_1 - t_2 \leq -\Delta + M(1 - d) \end{cases}$$

Logical Explanation

Suppose $M = 10^9$. Two cases to investigate:

Case 1 : $d = 0$ (Order: t_2, t_1, i.e. $t_2 < t_1$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta$$

Logical Explanation

Suppose $M = 10^9$. Two cases to investigate:

Case 1 : $d = 0$ (Order: t_2, t_1, i.e. $t_2 < t_1$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta$$

$$t_1 - t_2 \leq -\Delta + M(1-d) \quad \rightarrow \quad t_1 - t_2 \leq -\Delta + 10^9 \leq 10^9$$

Logical Explanation

Suppose $M = 10^9$. Two cases to investigate:

Case 1 : $d = 0$ (Order: t_2, t_1, i.e. $t_2 < t_1$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta$$

$$t_1 - t_2 \leq -\Delta + M(1-d) \quad \rightarrow \quad t_1 - t_2 \leq -\Delta + 10^9 \leq 10^9$$

The second condition is always true for sufficiently large M.

Combine conditions: $\Delta \leq t_1 - t_2 \leq 10^9$

Logical Explanation

Suppose $M = 10^9$. Two cases to investigate:

Case 1 : $d = 0$ (Order: t_2, t_1, i.e. $t_2 < t_1$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta$$

$$t_1 - t_2 \leq -\Delta + M(1-d) \quad \rightarrow \quad t_1 - t_2 \leq -\Delta + 10^9 \leq 10^9$$

The second condition is always true for sufficiently large M.

Combine conditions: $\Delta \leq t_1 - t_2 \leq 10^9$

Case 2 : $d = 1$ (Order: t_1, t_2, i.e. $t_1 < t_2$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta - 10^9 \geq -10^9$$

Logical Explanation

Suppose $M = 10^9$. Two cases to investigate:

Case 1 : $d = 0$ (Order: t_2, t_1, i.e. $t_2 < t_1$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta$$

$$t_1 - t_2 \leq -\Delta + M(1-d) \quad \rightarrow \quad t_1 - t_2 \leq -\Delta + 10^9 \leq 10^9$$

The second condition is always true for sufficiently large M.

Combine conditions: $\Delta \leq t_1 - t_2 \leq 10^9$

Case 2 : $d = 1$ (Order: t_1, t_2, i.e. $t_1 < t_2$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta - 10^9 \geq -10^9$$

$$t_1 - t_2 \leq -\Delta + M(1-d) \quad \rightarrow \quad t_1 - t_2 \leq -\Delta$$

Logical Explanation

Suppose $M = 10^9$. Two cases to investigate:

Case 1 : $d = 0$ (Order: t_2, t_1, i.e. $t_2 < t_1$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta$$

$$t_1 - t_2 \leq -\Delta + M(1-d) \quad \rightarrow \quad t_1 - t_2 \leq -\Delta + 10^9 \leq 10^9$$

The second condition is always true for sufficiently large M.

Combine conditions: $\Delta \leq t_1 - t_2 \leq 10^9$

Case 2 : $d = 1$ (Order: t_1, t_2, i.e. $t_1 < t_2$)

$$t_1 - t_2 \geq \Delta - Md \quad \rightarrow \quad t_1 - t_2 \geq \Delta - 10^9 \geq -10^9$$

$$t_1 - t_2 \leq -\Delta + M(1-d) \quad \rightarrow \quad t_1 - t_2 \leq -\Delta$$

The first condition is always true for sufficiently large M.

Combine conditions: $\Delta \leq t_2 - t_1 \leq 10^9$

Transformation of **OR** into an **AND**

Pick a very large number M.
Also consider a decision variable $d \in \{0, 1\}$.

For sufficiently large M, the following two statements are equivalent:

Statement 1:

$$\textbf{OR} \qquad \begin{cases} t_1 - t_2 \geq \Delta & \text{if } t_1 \geq t_2 \\ t_2 - t_1 \geq \Delta & \text{o.w.} \end{cases}$$

Statement 2:

$$\textbf{AND} \qquad \begin{cases} t_1 - t_2 \geq \Delta - Md \\ t_1 - t_2 \leq -\Delta + M(1 - d) \end{cases}$$

Transform an **OR** condition to an **AND** condition,
at the expense of an added binary variable d.
Variable d encodes the **order**.

$d = 0 \rightarrow \text{Order} : t_2, t_1.$

$d = 1 \rightarrow \text{Order} : t_1, t_2.$

Why is this useful?

Now you can pose the problem of earliest arrival time for the last aircraft with a decision variable for the order of arrival. That is, you can deal with continuous and discrete variables.

$$\text{min}: \quad dt_1 + (1-d)t_2$$

$$\begin{aligned}
\text{s. to:} \quad t_1 - t_2 &\geq \Delta - Md \\
t_1 - t_2 &\leq -\Delta + M(1-d) \\
t_1 &\leq b_1 \\
t_1 &\geq a_1 \\
t_2 &\leq b_2 \\
t_2 &\geq a_2
\end{aligned}$$

Why is this useful?

Now you can pose the problem of earliest arrival time for the last aircraft with a decision variable for the order of arrival. That is, you can deal with continuous and discrete variables.

min : $\quad dt_1 + (1-d)t_2 \qquad$ Aircraft 1 or 2 is last, if d = 1 or 0 respectively.

$$
\text{s. to:} \quad
\begin{aligned}
t_1 - t_2 &\geq \Delta - Md \\
t_1 - t_2 &\leq -\Delta + M(1-d) \\
t_1 &\leq b_1 \\
t_1 &\geq a_1 \\
t_2 &\leq b_2 \\
t_2 &\geq a_2
\end{aligned}
$$

Why is this useful?

Now you can pose the problem of earliest arrival time for the last aircraft with a decision variable for the order of arrival. That is, you can deal with continuous and discrete variables.

min : $\quad d t_1 + (1 - d) t_2 \qquad$ Aircraft 1 or 2 is last, if d = 1 or 0 respectively.

s. to:

$$
\begin{aligned}
t_1 - t_2 &\geq \Delta - Md && \text{Aircrafts 1 \& 2 are separated by at least } \Delta. \\
t_1 - t_2 &\leq -\Delta + M(1 - d) \\
t_1 &\leq b_1 \\
t_1 &\geq a_1 \\
t_2 &\leq b_2 \\
t_2 &\geq a_2
\end{aligned}
$$

Why is this useful?

Now you can pose the problem of earliest arrival time for the last aircraft with a decision variable for the order of arrival. That is, you can deal with continuous and discrete variables.

$$\text{min}: \quad dt_1 + (1-d)t_2 \qquad \text{Aircraft 1 or 2 is last, if d = 1 or 0 respectively.}$$

$$\begin{aligned}
\text{s. to:} \quad t_1 - t_2 &\geq \Delta - Md && \text{Aircrafts 1 \& 2 are separated by at least } \Delta. \\
t_1 - t_2 &\leq -\Delta + M(1-d) && \\
t_1 &\leq b_1 && \text{Aircraft 1 arrives in } [a_1, b_1]. \\
t_1 &\geq a_1 && \\
t_2 &\leq b_2 && \\
t_2 &\geq a_2 &&
\end{aligned}$$

Why is this useful?

Now you can pose the problem of earliest arrival time for the last aircraft with a decision variable for the order of arrival. That is, you can deal with continuous and discrete variables.

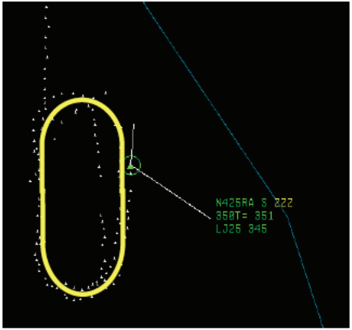
min : $\quad dt_1 + (1-d)t_2$ $\quad$ Aircraft 1 or 2 is last, if d = 1 or 0 respectively.

s. to:

$$
\begin{aligned}
t_1 - t_2 &\geq \Delta - Md && \text{Aircrafts 1 \& 2 are separated by at least } \Delta. \\
t_1 - t_2 &\leq -\Delta + M(1-d) && \\
t_1 &\leq b_1 && \text{Aircraft 1 arrives in } [a_1, b_1]. \\
t_1 &\geq a_1 && \\
t_2 &\leq b_2 && \text{Aircraft 2 arrives in } [a_2, b_2]. \\
t_2 &\geq a_2 &&
\end{aligned}
$$

How many holding patterns should an aircraft fly?

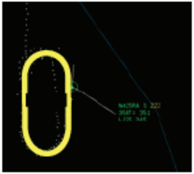
A holding pattern delays an aircraft by a fixed amount of time, e.g. T = 3 min.

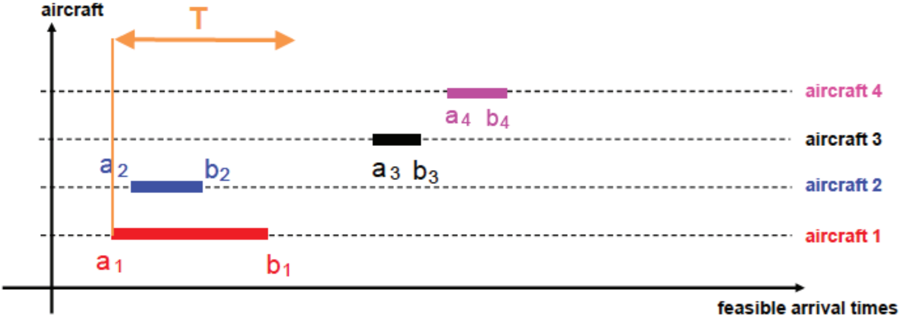


CTAS tracks courtesy of NASA Ames

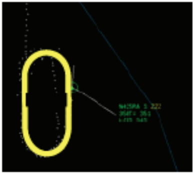
Question for ATC:
How many holding patterns should one aircraft do before it is allowed to land?

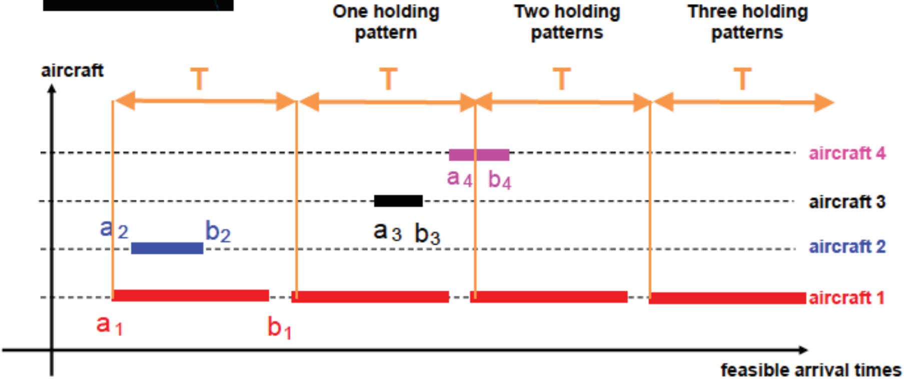
A holding pattern shifts arrival times by T



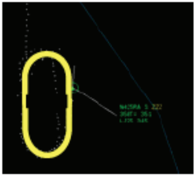


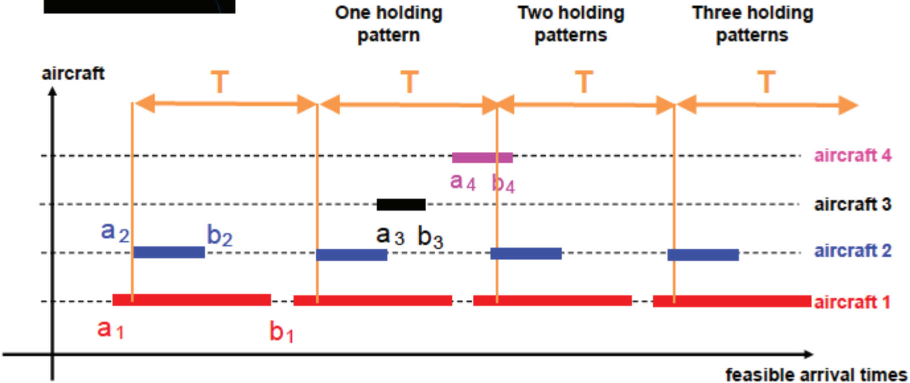
A holding pattern shifts arrival times by T



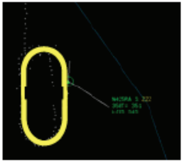


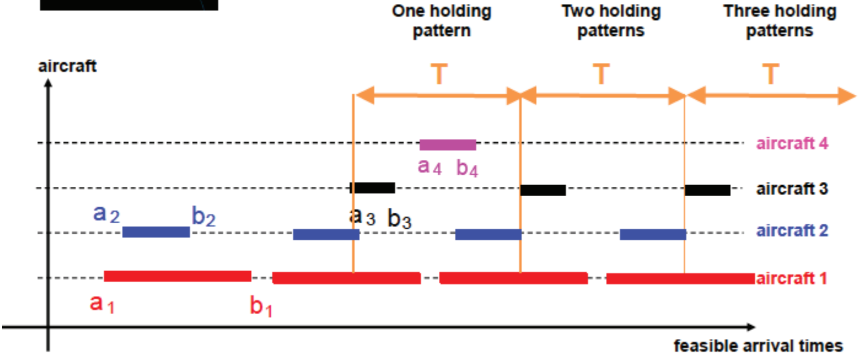
A holding pattern shifts arrival times by T



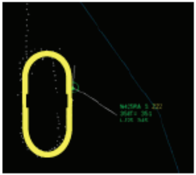


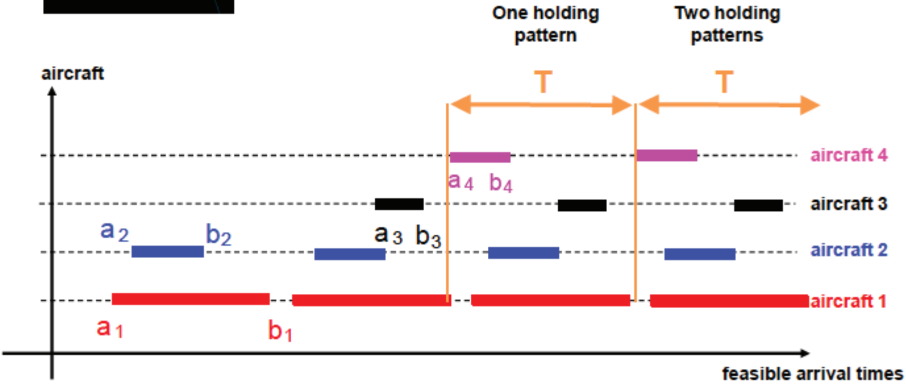
A holding pattern shifts arrival times by T



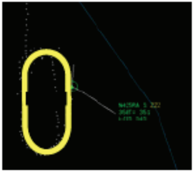


A holding pattern shifts arrival times by T

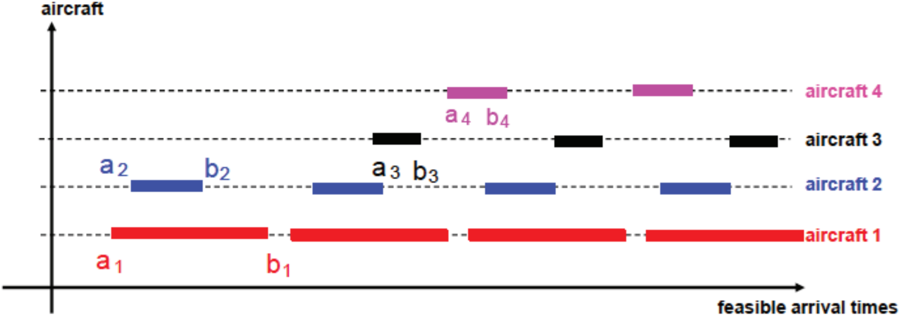




A holding pattern can be expressed as a MILP



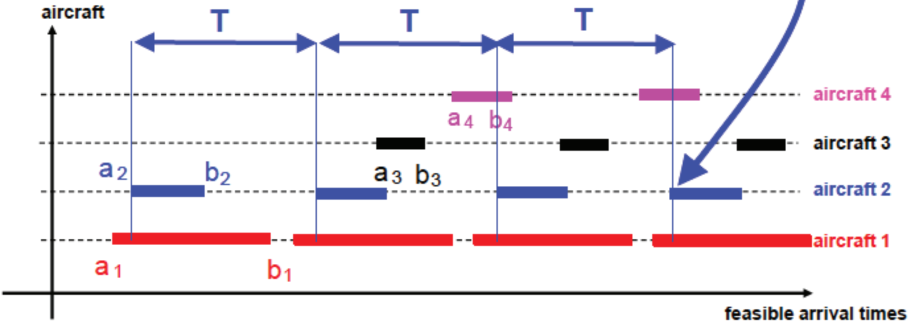
This is the set in which we want to schedule aircraft. We seek one arrival time for each aircraft in each of the colored sets.



A holding pattern can be expressed as a MILP

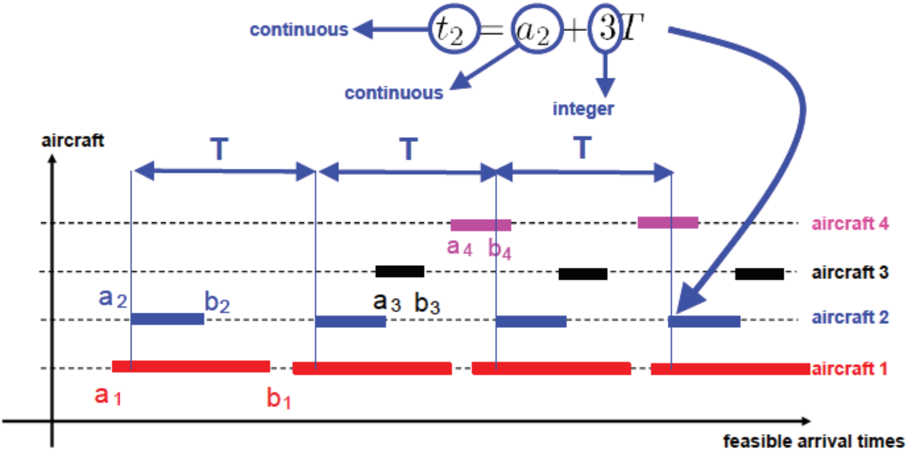
The number of holding patterns is a decision variable (the decision is actually made by the human Air Traffic Controller).

$$t_2 = a_2 + 3T$$



A holding pattern can be expressed as a MILP

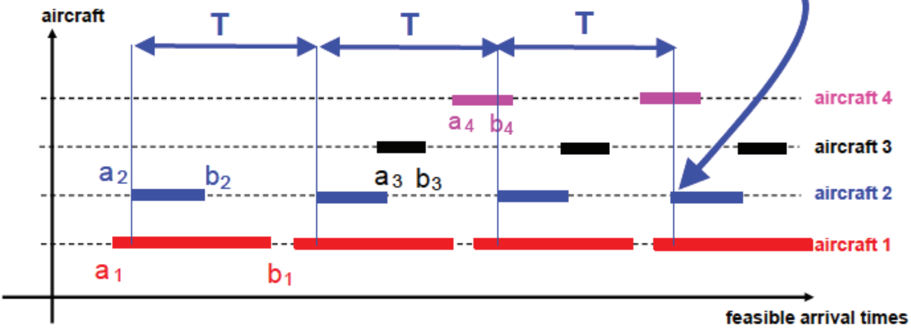
The number of holding patterns is a decision variable (the decision is actually made by the human Air Traffic Controller).



A holding pattern can be expressed as a MILP

Actually, the human air traffic controller has the possibility to schedule aircraft 2 anywhere in the fourth time interval (i.e. with three holding patterns).

$$t_2 \in [a_2 + 3T, b_2 + 3T]$$

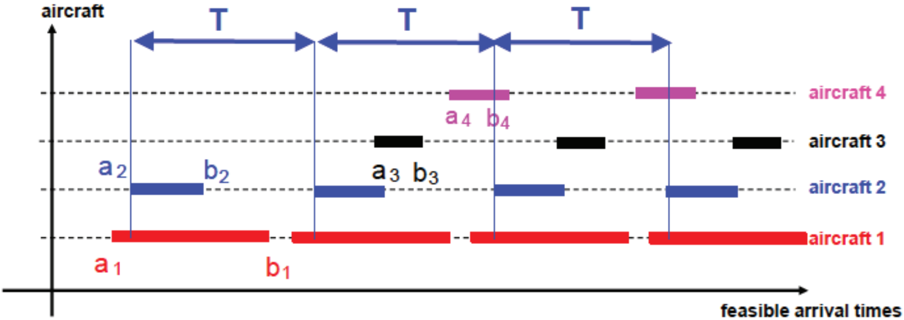


A holding pattern can be expressed as a MILP

This is can be expressed in terms of two linear constraints involving integer and continuous variables, more generally, for any admissible interval for aircraft 2:

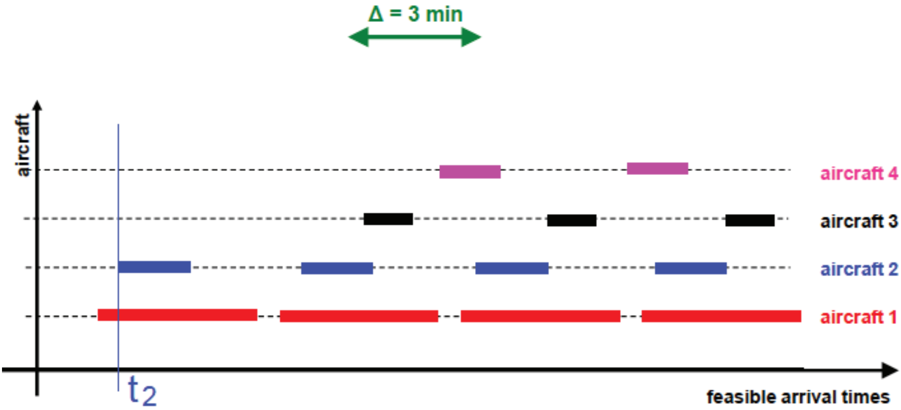
$$t_2 \geq a_2 + nT$$
$$t_2 \leq b_2 + nT$$

For n holding patterns



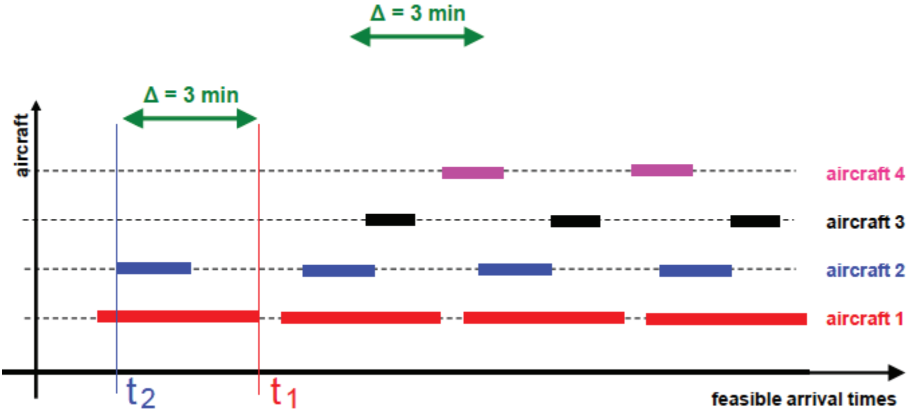
A real-world example: Air Traffic Control

Problem: separating aircraft by Δ = 3 min: how to schedule the aircraft so the last aircraft comes as early as possible.



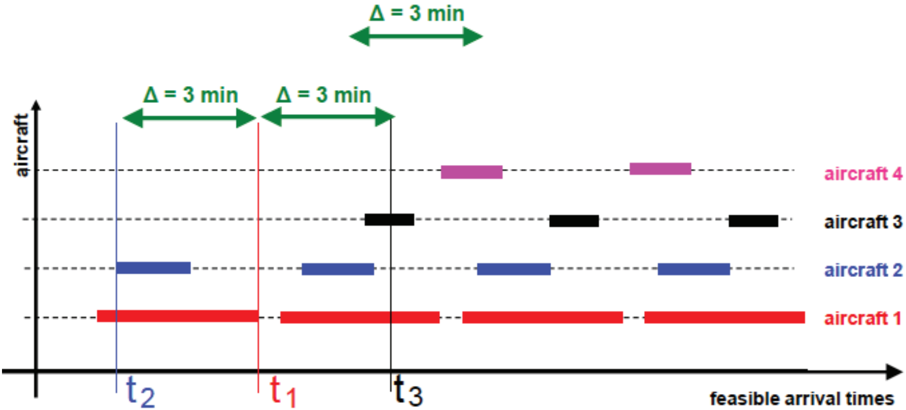
A real-world example: Air Traffic Control

Problem: separating aircraft by Δ = 3 min: how to schedule the aircraft so the last aircraft comes as early as possible.



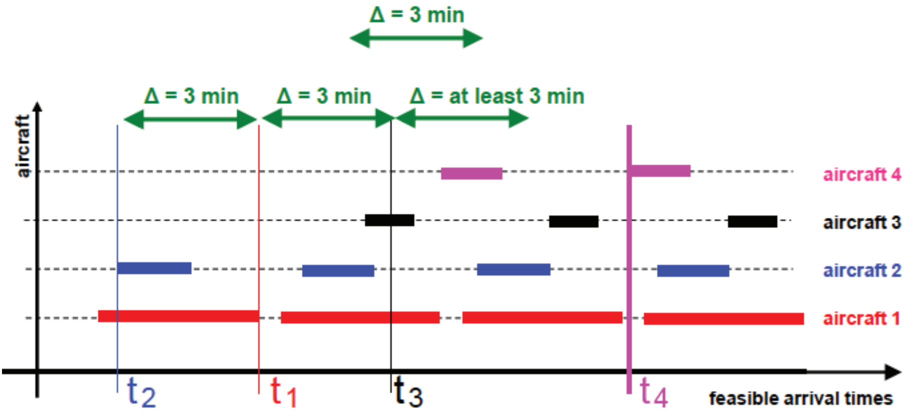
A real-world example: Air Traffic Control

Problem: separating aircraft by $\Delta = 3$ min: how to schedule the aircraft so the last aircraft comes as early as possible.



A real-world example: Air Traffic Control

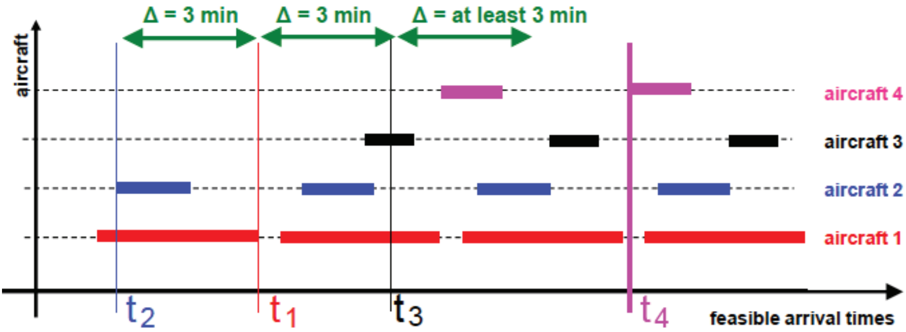
Problem: separating aircraft by Δ = 3 min: how to schedule the aircraft so the last aircraft comes as early as possible.



A real-world example: Air Traffic Control

This is a Mixed Integer Linear Program (MILP):

- Some variables are integer (the order of arrival): 3, 1, 2, 4…
- Some variables are continuous (the times of arrival)
- The problem can be posed as a linear program involving both integer and continuous variables.



MIP Formulation

Notation

m	Number of aircraft
$[a_i, b_i]$	Feasible arrival time interval for airplane i
T	Length of holding pattern
t_i	Arrival time of aircraft i
Δ	Minimum time separation between landings
n_i	No. of holding patterns for aircraft i
M	Large number
d_{ij}	Encoding of relative order for aircrafts i and j

Constraints

Non-negativity, integer : $t_i \geq 0, \;\; n_i \in \mathbb{Z} \quad \forall i = 1, \cdots, m$

Feasible arrival times : $a_i + n_i T \leq t_i \leq b_i + n_i T, \quad \forall i = 1, \cdots, m$

Minimum time separation : $|t_i - t_j| \geq \Delta, \quad \forall\, i, j = 1, \cdots, m \quad i \neq j$

MIP Formulation

Notation

m	Number of aircraft
$[a_i, b_i]$	Feasible arrival time interval for airplane i
T	Length of holding pattern
t_i	Arrival time of aircraft i
Δ	Minimum time separation between landings
n_i	No. of holding patterns for aircraft i
M	Large number
d_{ij}	Encoding of relative order for aircrafts i and j

Constraints

Non-negativity, integer : $t_i \geq 0, \;\; n_i \in \mathbb{Z} \quad \forall i = 1, \cdots, m$

Feasible arrival times : $a_i + n_i T \leq t_i \leq b_i + n_i T, \quad \forall i = 1, \cdots, m$

Minimum time separation :

$$\begin{aligned} t_i - t_j &\geq \Delta - Md \\ t_i - t_j &\leq -\Delta + M(1-d), \\ d_{ij} &\in \{0, 1\} \qquad \forall\, i, j = 1, \cdots, m \quad i \neq j \end{aligned}$$

Additional Reading

Revelle Section 7.B.3