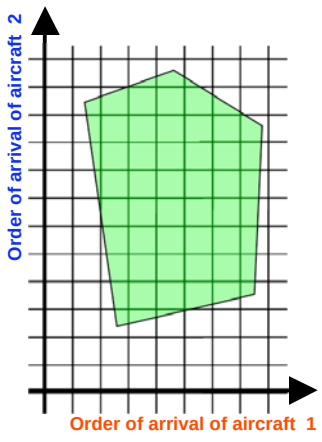
Lecture 7: Using MILP to solve decision problems

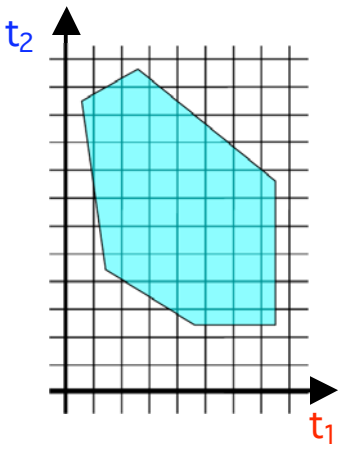
- Decision problems
- Absolute values
- Transformation of a logical “or” into a logical “and”
- A formal definition of MILP
- Example: holding patterns
- Posing the number of holding patterns as a MILP
- A real example from Air Traffic Control

Graphical interpretation of decision problems

Some variables are on a grid, some are not:



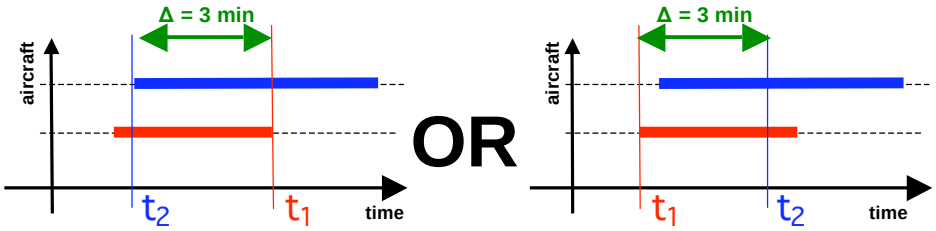
The solution for these two variables has to be on the grid.



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

Discrete order of arrival: deciding the order

First kind of decision variables: order of arrival



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

Another way to express this:

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

Now, every time you see an absolute value

Absolute value: not linear, not affine → difficult

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

Absolute value can be expressed as a logical disjunction (this is just a fancy way to say « or »).

$$t_2 - t_1 \geq \Delta \quad \textbf{OR} \quad t_1 - t_2 \geq \Delta$$

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

« and » is easy, « or » is difficult

Reminder: you have already used « and » many times

$$\begin{aligned}
&\textbf{Minimize:} && c(x_1, x_2) = a_1 x_1 + a_2 x_2 \\
&\textbf{Subject to:} && a_1 x_1 \leq c_{\max} \\
& && a_2 x_2 \geq a_{\min} \\
& && a_1 x_1 + a_2 x_2 \geq a_{\min} \\
& && a_2 x_2 \geq 2 a_1 x_1 \\
& && a_2 x_2 \leq 2 a_1 x_1
\end{aligned}$$

All these are logical “and”

Tranformation of an « or » into an « and »

Let us pick a very large number M
Let us pick a decision variable $d \in \{0, 1\}$

The two following statements are equivalent:

or
$$\begin{cases} t_1 - t_2 \geq \Delta \\ t_2 - t_1 \geq \Delta \end{cases}$$

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

Logical explanation

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

Logical explanation

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

$t_1 - t_2 \leq -\Delta + M(1-d)$ **becomes** $t_1 - t_2 \leq -\Delta + M$

Logical explanation

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

$t_1 - t_2 \leq -\Delta + M(1-d)$ **becomes** $t_1 - t_2 \leq -\Delta + M$

i.e. $t_1 - t_2 \leq -\Delta + 1000000$

Logical explanation

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

~~$t_1 - t_2 \leq -\Delta + M(1-d)$ **becomes** $t_1 - t_2 \leq -\Delta + M$~~

~~**i.e.** $t_1 - t_2 \leq -\Delta + 1000000$~~

This is always true (just take M large enough).

Therefore, the second condition can be discarded.

Logical explanation

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

Case 2: d=1

$t_1 - t_2 \leq -\Delta + M(1-d)$ **becomes** $t_1 - t_2 \leq -\Delta$

Logical explanation

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

Case 2: d=1

$t_1 - t_2 \leq -\Delta + M(1-d)$ **becomes** $t_1 - t_2 \leq -\Delta$

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta - M$

i.e. $t_1 - t_2 \geq \Delta - 1000000$

This is always true (just take M large enough).
Therefore, the second condition can be discarded.

Logical explanation

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

Case 2: d=1

$t_1 - t_2 \leq -\Delta + M(1-d)$ **becomes** $t_1 - t_2 \leq -\Delta$

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta - M$

i.e. $t_1 - t_2 \geq \Delta - 1000000$

This is always true (just take M large enough).
Therefore, the second condition can be discarded.

Summary

Two cases to investigate:

Case 1: d=0

$t_1 - t_2 \geq \Delta - Md$ **becomes** $t_1 - t_2 \geq \Delta$

Case 2: d=1

$t_1 - t_2 \leq -\Delta + M(1-d)$ **becomes** $t_1 - t_2 \leq -\Delta$

Summary

Depending on the value of d:

Case 1: d=0 $t_1 - t_2 \geq \Delta$

Case 2: d=1 $t_2 - t_1 \geq \Delta$

In other words $t_1 - t_2 \geq \Delta$ **or** $t_2 - t_1 \geq \Delta$

Tranformation of an « or » into an « and »

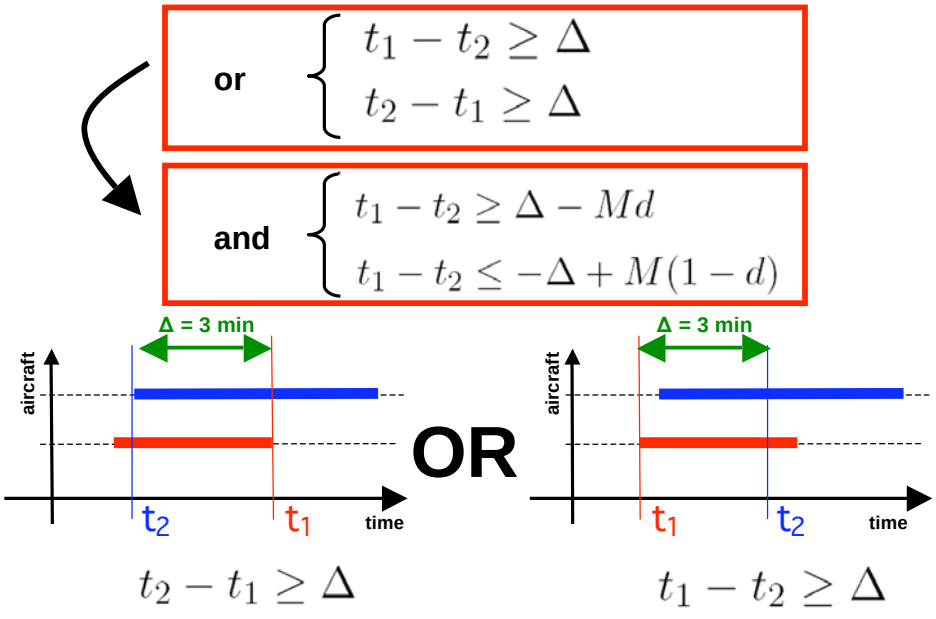
Let us pick a very large number M
Let us pick a decision variable $d \in \{0, 1\}$

The two following statements are equivalent:

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

or
$$\begin{cases} t_1 - t_2 \geq \Delta \\ t_2 - t_1 \geq \Delta \end{cases}$$

Tranformation of an « or » into an « and »



Why is it useful?

Now you can pose the problem of earliest arrival time of the last aircraft with decision enabled for order of arrival: you can deal with continuous and discrete variables.

t1 if d=1, t2 if d=0

$$
\begin{aligned}
\textbf{min:} \quad & \overbrace{dt_1 + (1-d)t_2} \\
\textbf{Subject to:} \quad & t_1 - t_2 \geq \Delta - Md \\
& t_1 - t_2 \leq -\Delta + M(1-d) \\
& t_1 \leq b_1 \\
& t_1 \geq a_1 \\
& t_2 \leq b_2 \\
& t_2 \leq a_2
\end{aligned}
$$

Why is it useful?

Now you can pose the problem of earliest arrival time of the last aircraft with decision enabled for order of arrival: you can deal with continuous and discrete variables.

Aircraft 1 and 2 are separated by more than Δ (previous slides)

$$
\begin{aligned}
\textbf{min:} \quad & dt_1 + (1-d)t_2 \\
\textbf{Subject to:} \quad & t_1 - t_2 \geq \Delta - Md \\
& t_1 - t_2 \leq -\Delta + M(1-d) \qquad \} \; |t_1 - t_2| \geq \Delta \\
& t_1 \leq b_1 \\
& t_1 \geq a_1 \\
& t_2 \leq b_2 \\
& t_2 \leq a_2
\end{aligned}
$$

Why is it useful?

Now you can pose the problem of earliest arrival time of the last aircraft with decision enabled for order of arrival: you can deal with continuous and discrete variables.

$$
\begin{aligned}
\textbf{min:} \quad & dt_1 + (1-d)t_2 \\
\textbf{Subject to:} \quad & t_1 - t_2 \geq \Delta - Md \\
& t_1 - t_2 \leq -\Delta + M(1-d) \qquad |t_1 - t_2| \geq \Delta \\
& t_1 \leq b_1 \\
& t_1 \geq a_1 \qquad \} \text{ Aircraft 1 arrives between a1 and b1} \\
& t_2 \leq b_2 \\
& t_2 \leq a_2 \qquad \} \text{ Aircraft 2 arrives between a2 and b2}
\end{aligned}
$$

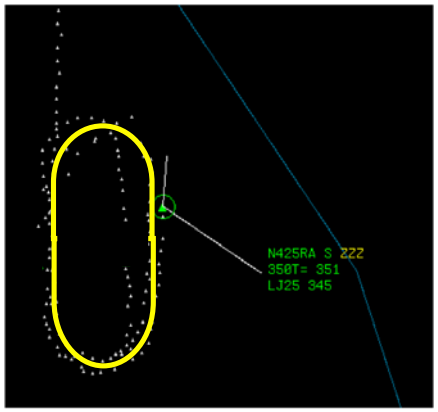
A formal definition of a MILP

A Mixed Integer Linear Program is a Linear Program in which some of the variables are continuous, and some are integer.

$$
\begin{aligned}
\textbf{min:} \quad & dt_1 + (1-d)t_2 \\
\textbf{Subject to:} \quad & t_1 - t_2 \geq \Delta - Md \\
& t_1 - t_2 \leq -\Delta + M(1-d) \\
& t_1 \leq b_1 \\
& t_1 \geq a_1 \\
& t_2 \leq b_2 \\
& t_2 \leq a_2
\end{aligned}
$$

Holding patterns: how many should an aircraft fly?

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

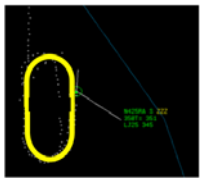


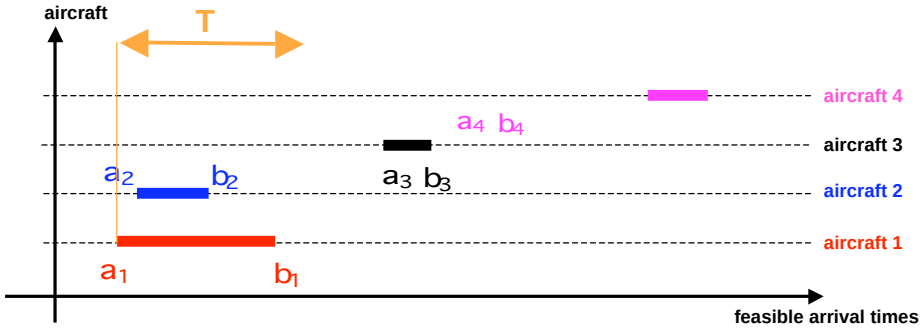
CTAS tracks courtesy of NASA Ames

Question for ATC:

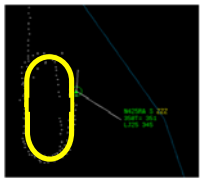
how many holding patterns should one aircraft do before it is allowed to land?

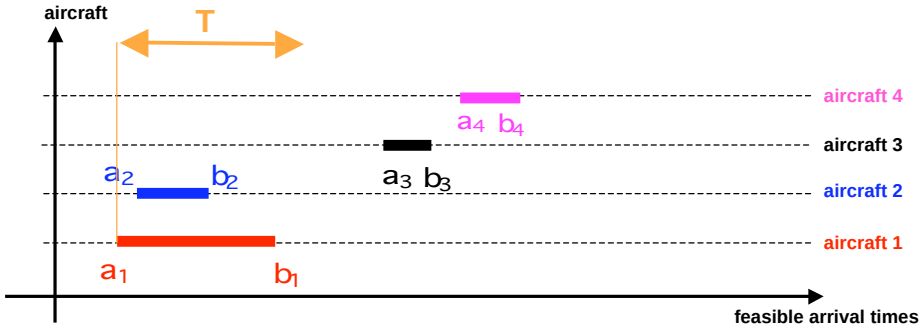
A holding pattern is a shift by T



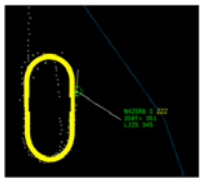


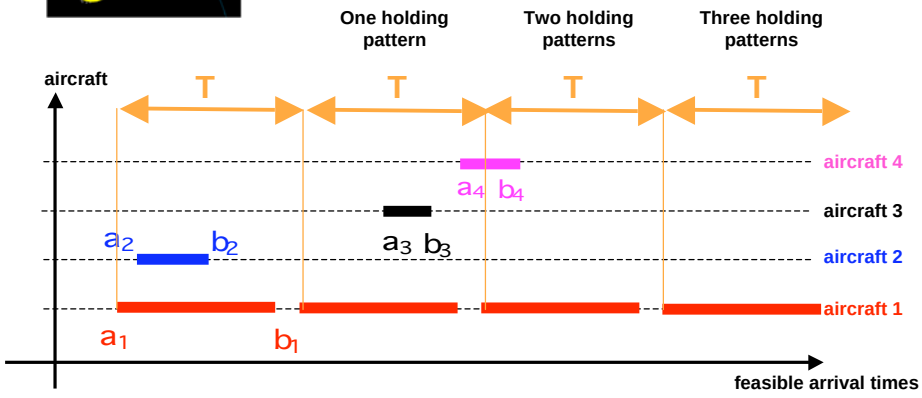
A holding pattern is a shift by T



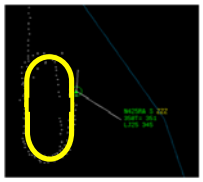


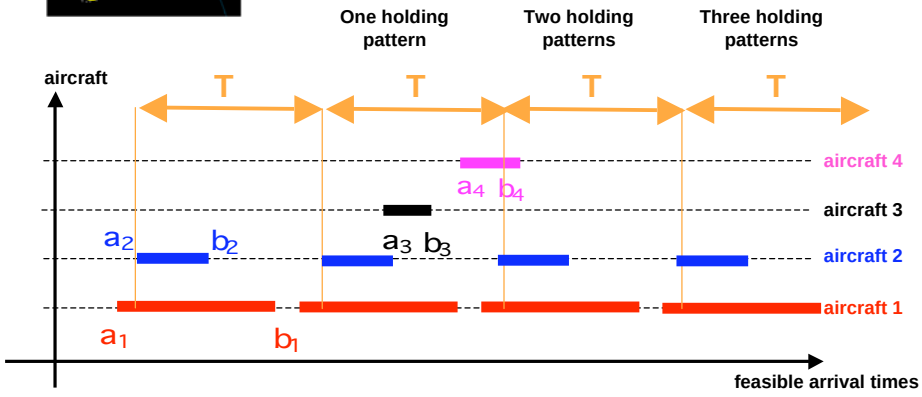
A holding pattern is a shift by T



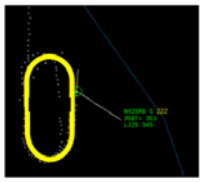


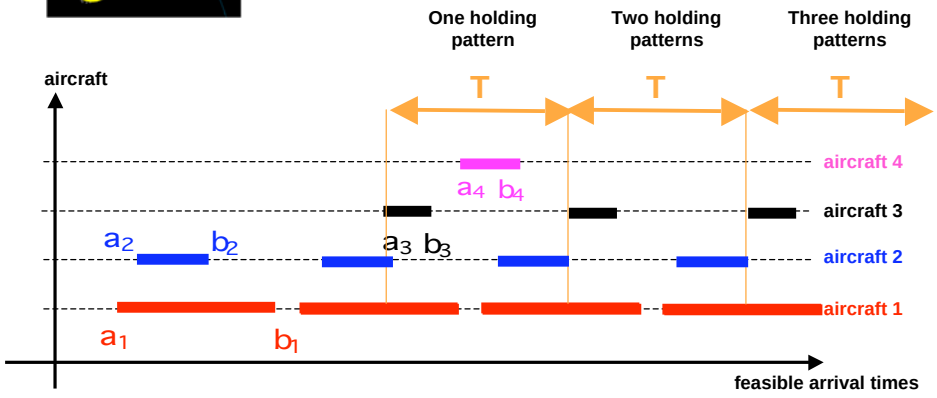
A holding pattern is a shift by T



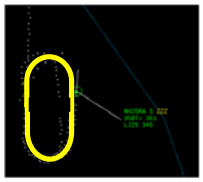


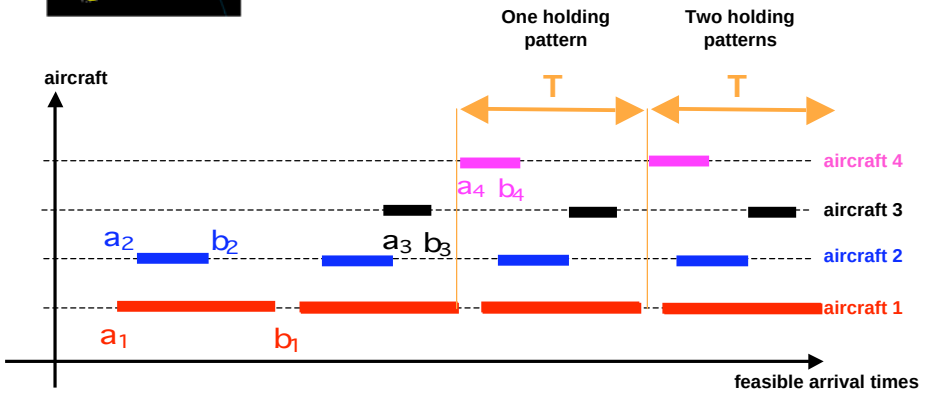
A holding pattern is a shift by T



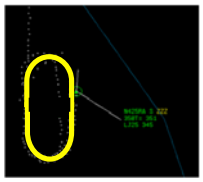


A holding pattern is a shift by T

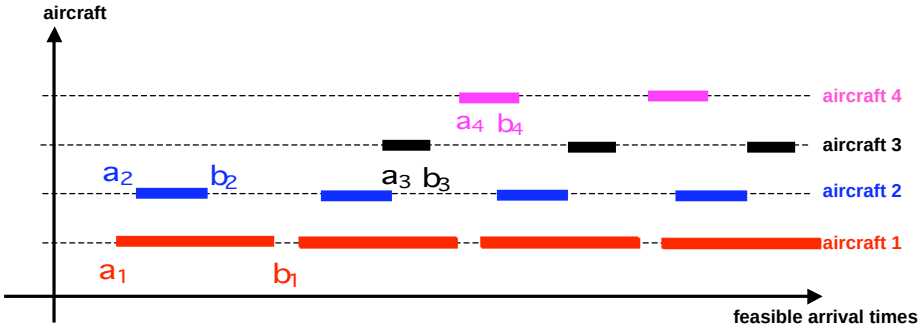




A holding pattern is a shift by T

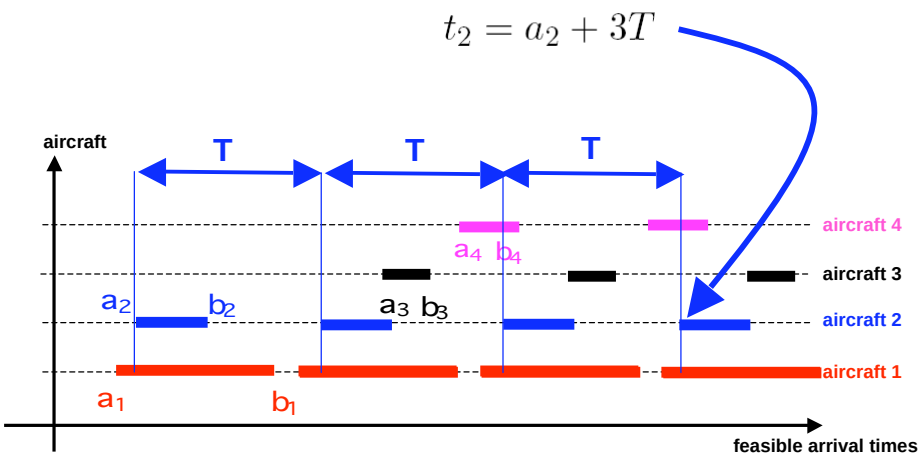


This is the set in which we want to schedule aircraft. We seek one arrival time for each aircraft in each of the colored sets.



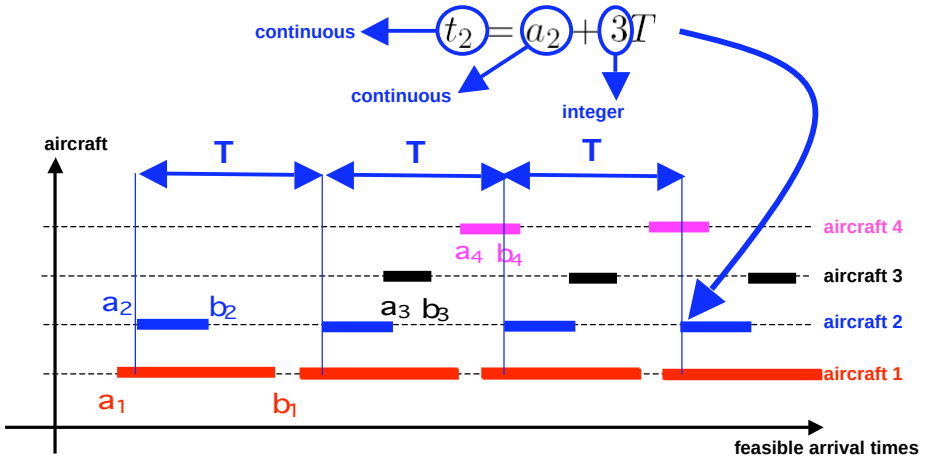
A holding pattern can be expressed as a MILP

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



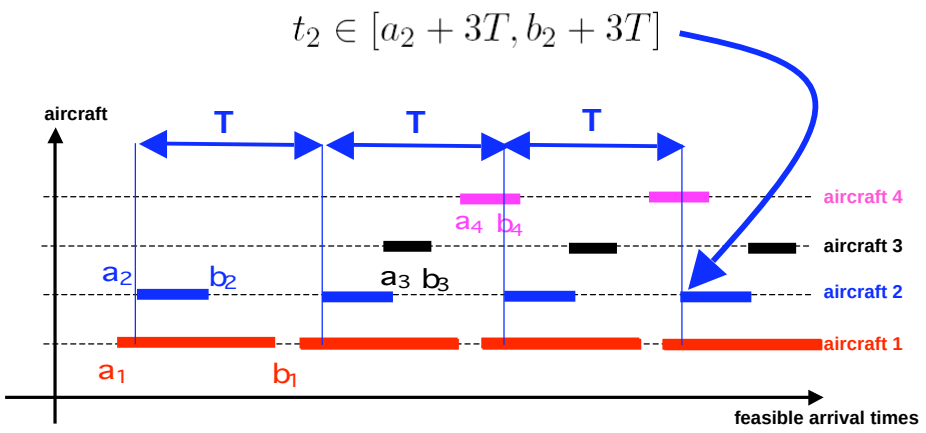
A holding pattern can be expressed as a MILP

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

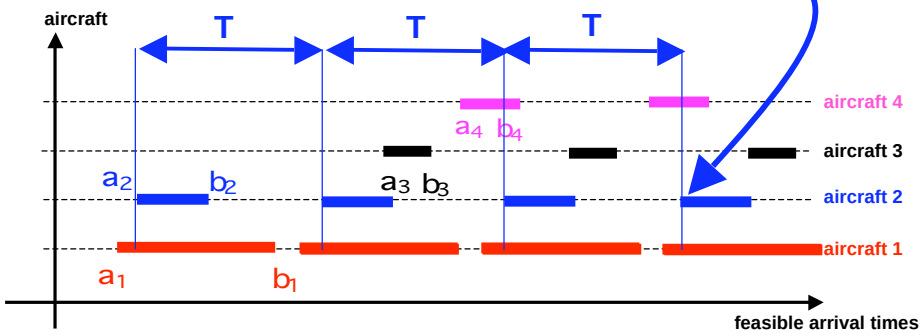
Actually, the human air traffic controller has the possibility to schedule aircraft 2 anywhere in the fourth time interval (i.e. with three holding patterns).



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

$$t_2 \geq a_2 + 3T$$
$$t_2 \leq b_2 + 3T$$

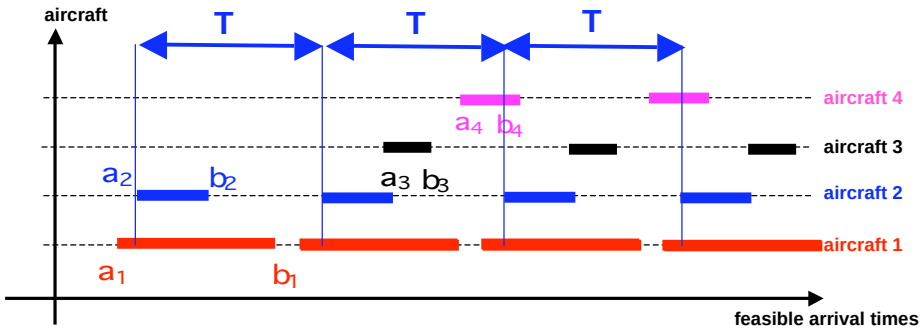


A holding pattern can be expressed as a MILP

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:

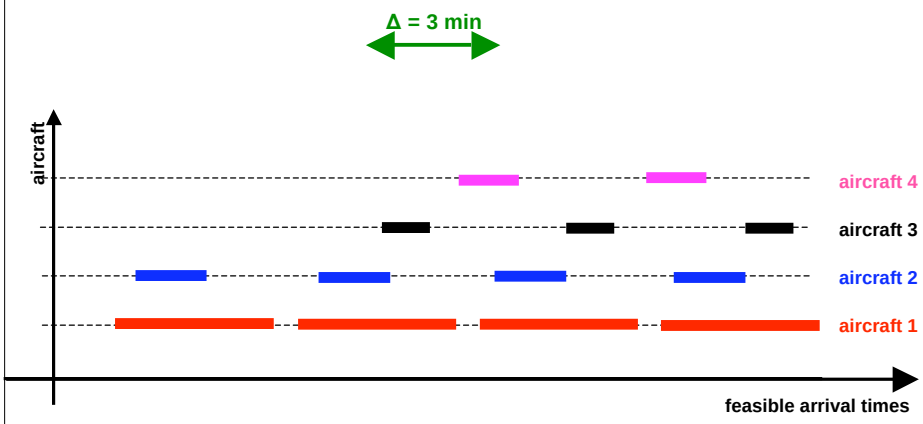
$$t_2 \geq a_2 + nT$$
$$t_2 \leq b_2 + nT$$

For n holding patterns



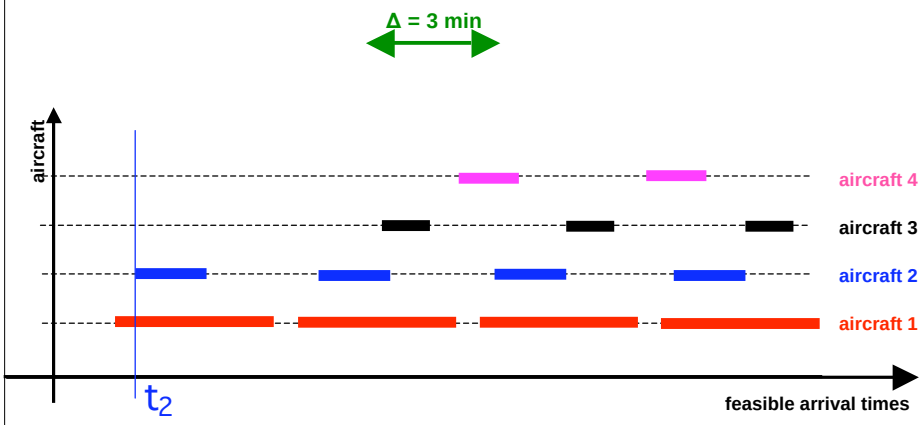
An real example from transportation (air traffic control)

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



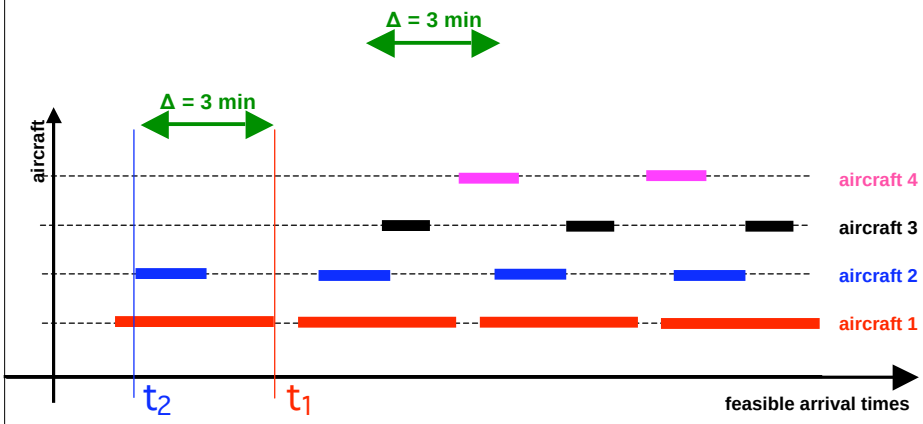
An real example from transportation (air traffic control)

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



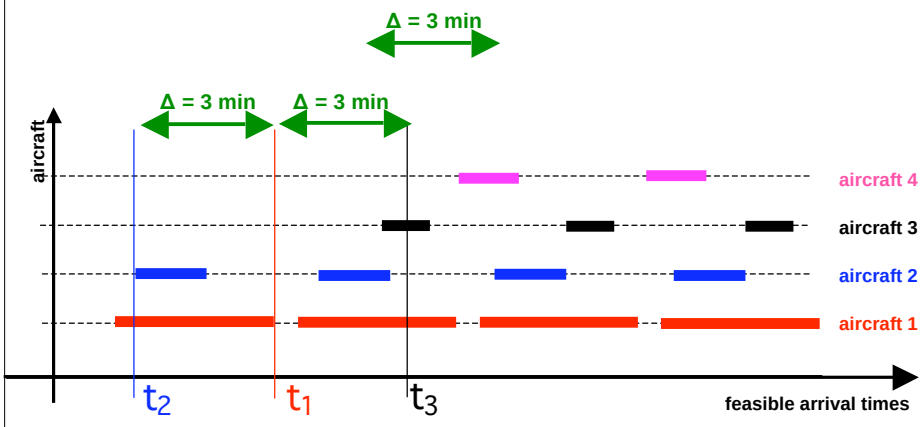
An real example from transportation (air traffic control)

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



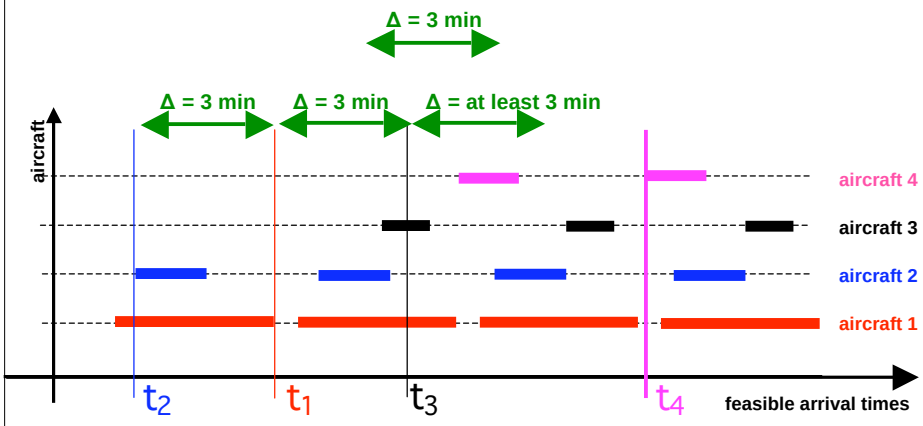
An real example from transportation (air traffic control)

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



An real example from transportation (air traffic control)

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



An real example from transportation (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.