We have pre-calculated the order in which the flight arrive in each node $x \in V$, and we say that, if there is a conflict, then $x_{i,j} = 0$ means that $x_{j,i} = 1$ and then that the flight i pass before flight i, if there are no conflict $x_{i,j} = x_{j,i} = 1$

Require: graph $G = (V, E)$, set of flight F, path of each flight $P_i \forall i \in F$, conflict variable $x_{i,j}$ (see above), $t(s,i) \forall i \in F, \forall s(P_i)$ starting time for the initial node of the path
Ensure: earliest time an arc can come to a node $v \in V$

```
{calculate the number number of conflict between two flights}
Mc ← 0
for all i ∈ F do
  for all j ∈ F do
    tmp ← |P_i ∩ P_j|
    if tmp > Mc then
      Mc ← tmp
    end if
  end for
end for
{define t}
for all i ∈ F do
  for all x ∈ V : x ∉ s(P_i) do
    t(x, i) = −1
  end for
end for
{propagation of time}
for |F|^Mc do
  for all i ∈ F do
    for (x, y) ∈ P_i do
      if ∃j ∈ F : y_{i,j} = 0 then
        if t(y, j) = −1 then
          break
        else
          propagate time s.t. it's far enough from the conflict {}
        end if
      else
        propagate time {t(y, i) ← t(x, i) + d/v_min}
      end if
    end for
  end for
end for
```

TODO: check if the next function it's working, with d the distance, s the maximum percentage change, $v_{i,j,z}$ the speed of the arc $(i, j) \in A$ and flight z, D the safety distance ($\underline{t}, \overline{t}$ are the min and max safaty time)

$$\forall i \in F, \forall (x, y) \in A$$

$$\overline{t}_{i,y} = \begin{cases} \overline{t}_{i,x} + \dots \frac{d}{\max\{\underline{v}_{i,y},(1-s)*v_{x,y,i}\}} & \text{if } \nexists j \in F : y_{i,j} \neq 1 \\ \underline{t}_{j,y} + \dots & \text{if } \exists j \in F : y_{i,j} = 1 \end{cases}$$

$$\underline{\mathrm{t}}_{i,y} = \begin{cases} \underline{t}_{i,x} + \frac{d}{\min\{\overline{v}_{i,y},(1+s)*v_{x,y,i}\}} & \text{if } \nexists j \in F : y_{i,j} \neq 1 \\ \overline{t}_{j,y} + \dots & \text{if } \exists j \in F : y_{i,j} = 1 \end{cases}$$

Possibili migliorie da inserire:
Per ogni volo, mettere in un buffer l'arco/l'ultimo nodo definito
Idee propagazione:
usare range permesso, quindi impostare nei nodi di conflitto le velocità e trattare negli altri posti tramite propagazione raggiungere gli altri, per i successivi non è un problema dato che dobbiamo attendere quelli che vengono prima.

For latest time, just use inverse the order of the path and some other small differences.