

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 j . 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: interval of safe earliest time for flight can traverse node $v \in V$

```

{create the hypergraph}
 $G' = (V, A, A') \leftarrow (V, A, \emptyset)$ 
for all  $i \in F$  do
  for all  $j \in F$  do
    if  $P_i \cap P_j \neq \emptyset$  then
      for all  $x \in P_i \cap P_j$  do
        if  $x_{i,j} = 0$  then
           $a \leftarrow \text{prec}_i(x), x$ 
           $a.\text{flight} \leftarrow i$ 
           $A' \leftarrow A' \cup a$ 
        end if
      end for
    end if
  end for
end for
{define t}
for all  $i \in F$  do
  for all  $x \in V : x \notin s(P_i)$  do
     $t(x, i) = -1$ 
  end for
end for
for all  $i \in F$  do
  for  $(x, y) \in P_i$  do
    if  $\exists z : (z, y) \in A'$  then
      BackPropagation( $G', z, (z, y).\text{flight}, \emptyset$ )
    else
      propagate time
    end if
  end for
end for

```

Require: hyper-graph $G' = (V, A, A')$, node x to back-propagate, flight f , set of considered flight Cf

Ensure: interval of safe earliest of x

```

if  $z$  is a starting point then
    return propagated time
end if
if  $\exists z : (z, x) \in A'$  then
     $CF \leftarrow CF \cup (z, x).flight$ 
     $T \leftarrow \text{BackPropagation}(G', x, x.flight, Cf)$ 
else
     $T \leftarrow \text{BackPropagation}(G', z, f, \emptyset)$ 
end if
return propagate time using  $T$ 

```

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.