Recovery of disrupted airline operations using k-Maximum Matching in graphs

Julien Bensmail[1] Valentin Garnero[1] Nicolas Nisse[1]
Alexandre Salch[2] Valentin Weber[3]

[1] Université Côte d'Azur, Inria, CNRS, I3S, France

[2] Innovation & Research, Amadeus IT Group SA

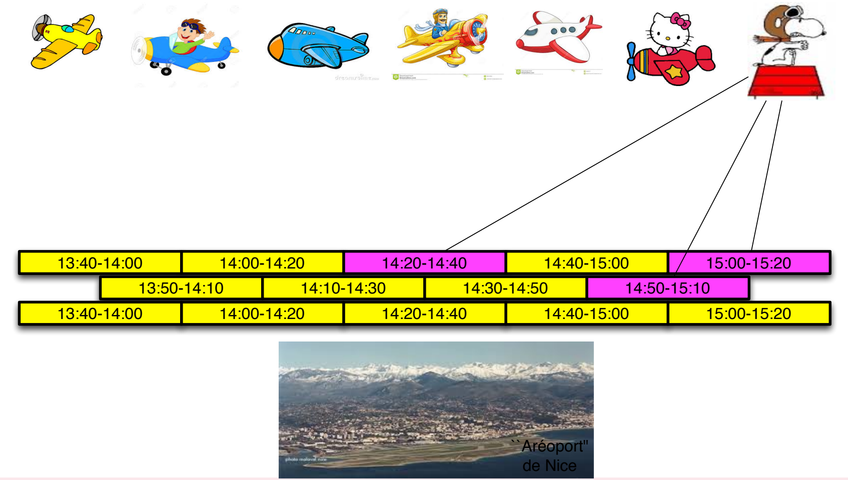
[3] Innovation & Research, Amadeus IT Pacific

LAGOS 2017, Marseille, 11st September 2017

Assignment of slots for landing to aircrafts

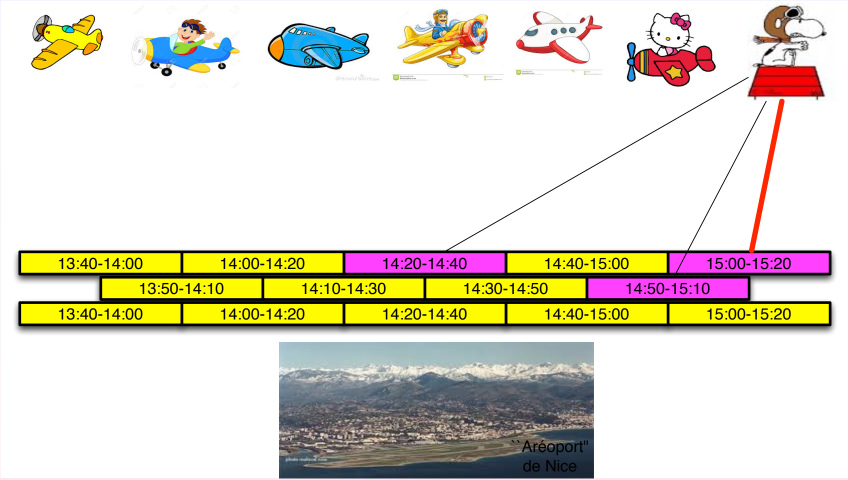
Aircrafts arriving at some airport

Assignment of slots for landing to aircrafts



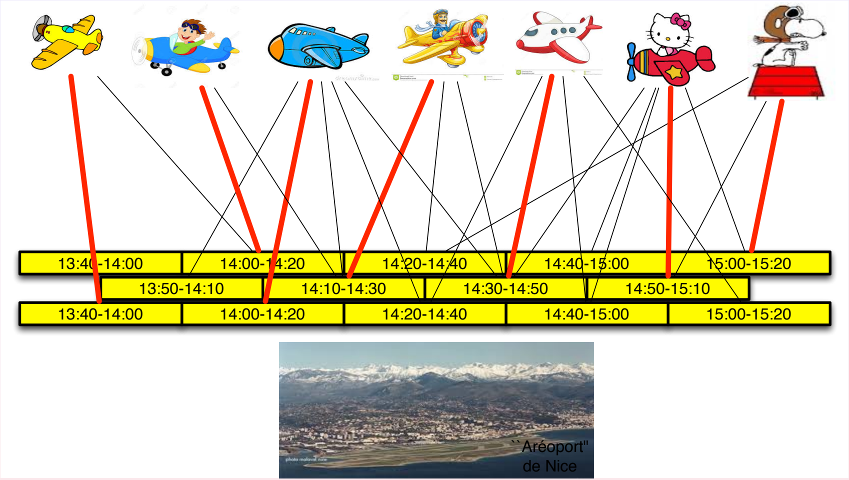
Each aircraft has a set of available and compatible slots depending on the tracks, the schedules, the companies...

Assignment of slots for landing to aircrafts



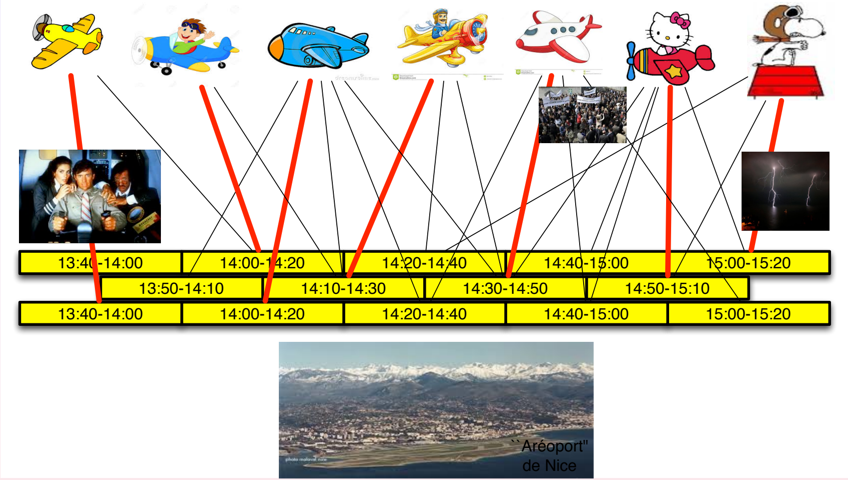
Initially, one compatible slot is assigned to each aircraft.

Assignment of slots for landing to aircrafts



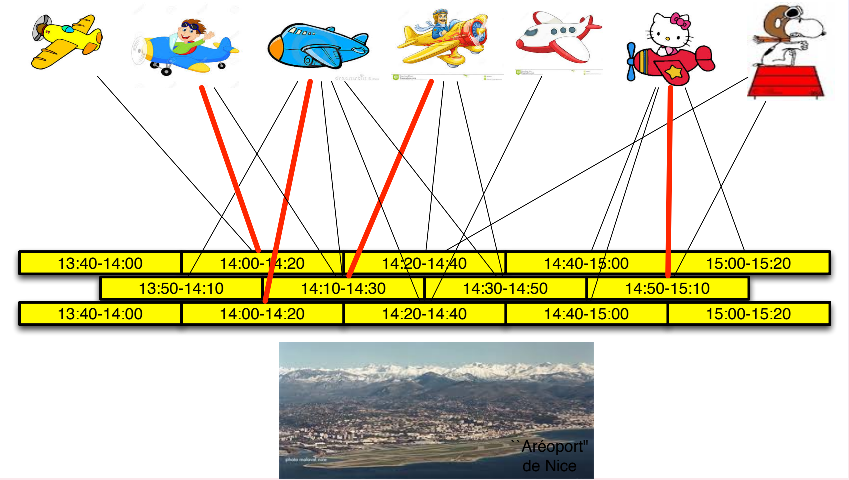
Initially, one compatible slot is assigned to each aircraft.

Assignment of slots for landing to aircrafts



Imponderable problems may happen (no refund...)

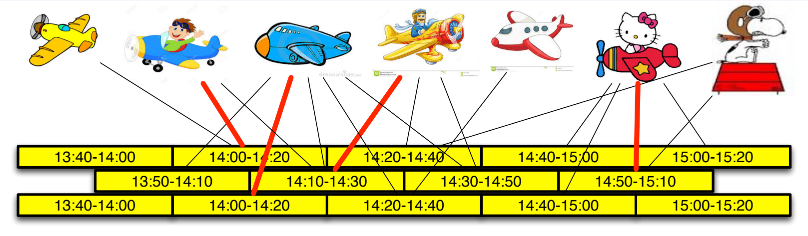
Assignment of slots for landing to aircrafts



How to return to a normal situation?

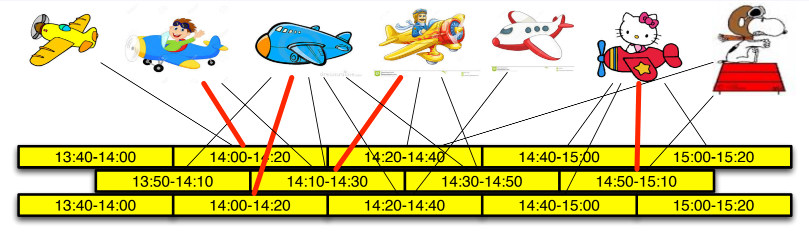
i.e., maximize # of aircrafts having a slot for landing!!

A simple matching problem in bipartite graphs?



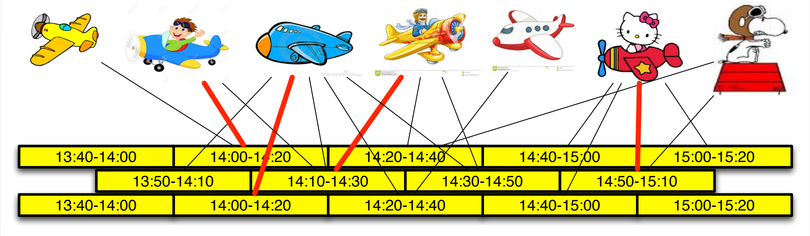
Restart from scratch and compute a maximum matching?

A simple matching problem in bipartite graphs?

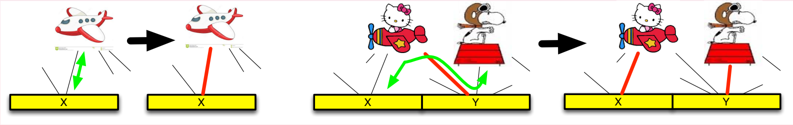


Restart from scratch and compute a maximum matching?
No!! Constraints due to the system/to the companies' policies...

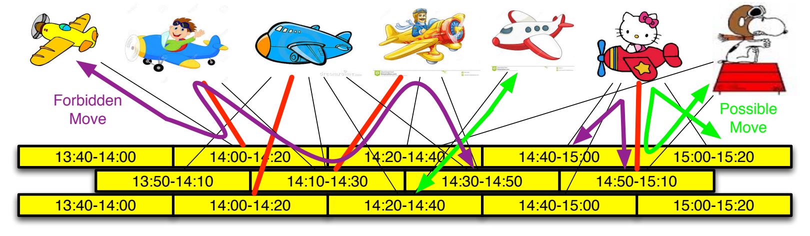
A simple matching problem in bipartite graphs?



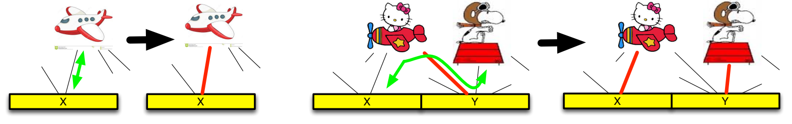
Restart from scratch and compute a maximum matching?
No!! Constraints due to the system/to the companies' policies...
ONLY 2 possible "moves" to satisfy all demands



A simple matching problem in bipartite graphs?

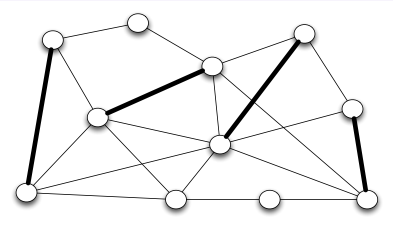


Restart from scratch and compute a maximum matching?
No!! Constraints due to the system/to the companies' policies...
ONLY 2 possible "moves" to satisfy all demands



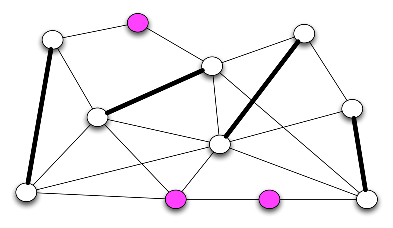
Reminder on Matchings in Graphs

Let $G = (V, E)$ be a graph.
A matching $M \subseteq E$ is a set of pairwise disjoint edges



Reminder on Matchings in Graphs

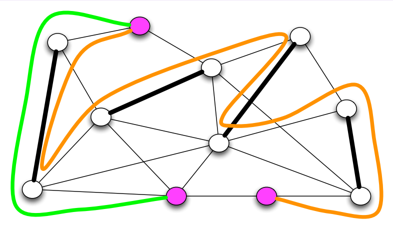
Let $G = (V, E)$ be a graph.
A matching $M \subseteq E$ is a set of pairwise disjoint edges



Exposed vertex: do not belong to the matching

Reminder on Matchings in Graphs

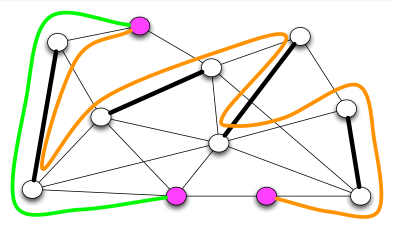
Let $G = (V, E)$ be a graph.
A matching $M \subseteq E$ is a set of pairwise disjoint edges



Exposed vertex: do not belong to the matching
M-augmenting path : "alternating" with both ends exposed

Reminder on Matchings in Graphs

Let $G = (V, E)$ be a graph.
A matching $M \subseteq E$ is a set of pairwise disjoint edges



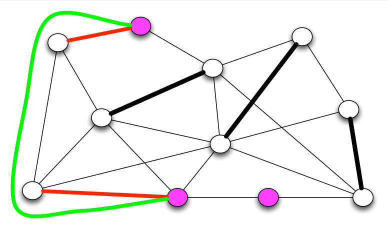
Exposed vertex: do not belong to the matching
M-augmenting path : "alternating" with both ends exposed

[Berge 1957] : Let G be a graph

M maximum matching ($|M| = \mu(G)$) iff no M-augmenting path.

Reminder on Matchings in Graphs

Let $G = (V, E)$ be a graph.
A matching $M \subseteq E$ is a set of pairwise disjoint edges



Exposed vertex: do not belong to the matching
M-augmenting path : "alternating" with both ends exposed

[Berge 1957] : Let G be a graph

M maximum matching ($|M| = \mu(G)$) iff no M-augmenting path.

$\Rightarrow$ the order in which the augmenting paths are augmented is not important

Computing a maximum matching

Maximum Matching in Bipartite Graphs (flow problem)

"easy" [Hungarian method, Kuhn 1955]

Maximum Matching $\mu(G)$

Polynomial [Edmonds 1965]

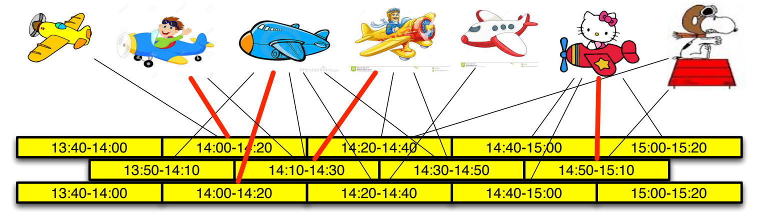
finding an augmenting path in polynomial time + Berge's theorem

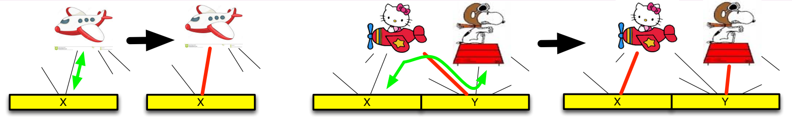
Augment "greedily" paths of length $\leq 2k-3$

$(1-\frac{1}{k})$-Approximation for $\mu(G)$ [Hopcroft,Kraft 1973]

Applications in wireless networks

Let's go back to slots assignment





Let G be a graph, M be a (partial) matching and $k \in \mathbb{N}$ odd

Let $\mu_k(G, M)$ be the maximum size of a matching that can be obtained from M by augmenting only paths of lenghts $\leq k$. here $k = 3$

Our problem

Given a graph G, a matching M and $k \in \mathbb{N}$ odd

Compute a matching of size $\mu_k(G, M)$ that can be obtained from M by augmenting only paths of lenghts $\leq k$.

Goal: algorithm that computes a sequence $(P_1, \cdots, P_r)$ such that:

- $\forall i \leq r$, P_i a path of length $\leq k$ in G
- $\forall i \leq r$, after augmenting $P_1, \cdots, P_{i-1}$ starting from M
  P_i is augmenting
- and r is maximum (w.r.t. these constraints)
  $(r_{max} + |M| = \mu_k(G, M))$

Easy Results

Problem: Given a graph G, a matching M and $k \in \mathbb{N}$ odd

Compute a matching of size $\mu_k(G,M)$ that can be obtained from M by augmenting only paths of lenghts $\leq k$.

Goal: algorithm that computes a sequence $(P_1, \cdots, P_r)$ such that:

- $\forall i \leq r$, P_i a path of length $\leq k$ in G
- $\forall i \leq r$, after augmenting $P_1, \cdots, P_{i-1}$ starting from M P_i is augmenting
- and r is maximum (w.r.t. these constraints)

Case $M = \emptyset$. For any odd $k \geq 0$, $\mu_k(G,\emptyset) = \mu(G)$

Compute a maximum matching, augment its edges one by one.

Case $k = 1$. For any matching M, $\mu_1(G,M) = \mu(G \setminus V(M)) + |M|$.

The edges of M cannot be "modified"
Compute a max. matching in $G \setminus V(M)$, augment these edges 1 by 1

Our contributions

$k = 3$

Let G be a graph and M be a matching

Computing a matching of size $\mu_3(G, M)$, obtained from M by augmenting paths of length ≤ 3, is in P.

$k \geq 5$

Let G be a planar bipartite graph of max. degree 3 and M be a matching

Computing a matching of size $\mu_k(G, M)$, obtained from M by augmenting paths of length at most $k \geq 5$, is NP-complete.

$G = T$ is a tree.

Computing $\mu_k(T, M)$ (k odd) can be done in polynomial-time in trees T

- with bounded max. degree Δ (dynamic prog., FPT in $k + \Delta$)
- or with vertices of degree ≥ 3 pairwise at distance $> k$.

Our contributions

$k = 3$

Let G be a graph and M be a matching

Computing a matching of size $\mu_3(G, M)$, obtained from M by augmenting paths of length ≤ 3, is in P.

$k \geq 5$

Let G be a planar bipartite graph of max. degree 3 and M be a matching

Computing a matching of size $\mu_k(G, M)$, obtained from M by augmenting paths of length at most $k \geq 5$, is NP-complete.

$G = T$ is a tree.

Computing $\mu_k(T, M)$ (k odd) can be done in polynomial-time in trees T

- with bounded max. degree Δ (dynamic prog., FPT in $k + \Delta$)
- or with vertices of degree ≥ 3 pairwise at distance $> k$.

Our contributions

$k = 3$

Let G be a graph and M be a matching

Computing a matching of size $\mu_3(G, M)$, obtained from M by augmenting paths of length ≤ 3, is in P.

$k \geq 5$

Let G be a planar bipartite graph of max. degree 3 and M be a matching

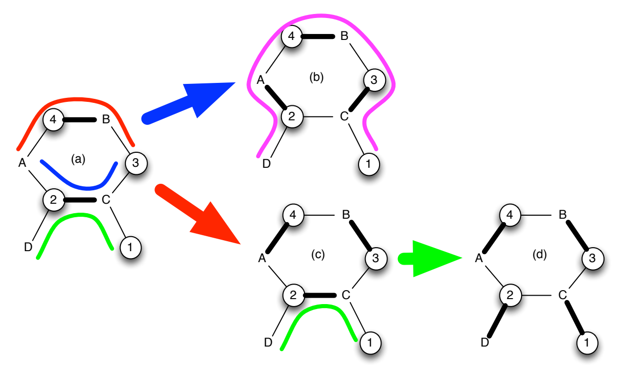
Computing a matching of size $\mu_k(G, M)$, obtained from M by augmenting paths of length at most $k \geq 5$, is NP-complete.

$G = T$ is a tree.

Computing $\mu_k(T, M)$ (k odd) can be done in polynomial-time in trees T

- with bounded max. degree Δ (dynamic prog., FPT in $k + \Delta$)
- or with vertices of degree ≥ 3 pairwise at distance $> k$.

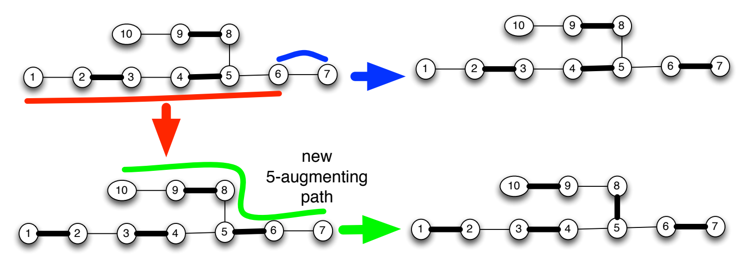
Why bounding the length of the augmenting paths makes the problem harder? ex: $k = 3$



$(a) \to (b)$ not optimal, but $(a) \to (c) \to (d)$ optimal

$\Rightarrow$ The result is impacted by the order in which paths are augmented.

Why bounding the length of the augmenting paths makes the problem harder? ex: $k = 5$



$\Rightarrow$ The order in which paths are augmented impacts the creation of new augmenting paths that are necessary to reach the optimum

Our contributions: ideas of the proofs

$k = 3$

$\mu_3(G, M)$ can be computed in polynomial time

- possible to focus only on 3-augmenting paths initially present;
- after "reducing" the graph, augmentations may be in any order.

$k \geq 5$

$\mu_5(G, M)$ is NP-hard even if G planar bipartite of max. degree 3.

Reduction from 3-SAT where "creating" new augmenting paths corresponds to a choice in the assignment.

Our contributions: ideas of the proofs

$k = 3$

$\mu_3(G, M)$ can be computed in polynomial time

- possible to focus only on 3-augmenting paths initially present;
- after "reducing" the graph, augmentations may be in any order.

$k \geq 5$

$\mu_k(G, M)$ is NP-hard even if G planar bipartite of max. degree 3.

Reduction from 3-SAT where "creating" new augmenting paths corresponds to a choice in the assignment.

Question: Complexity of $\mu_k(T, M)$ in trees T?

What we know:

Computing $\mu_k(T, M)$ (k odd) can be done in polynomial-time in trees T

- with bounded max. degree Δ (dynamic prog., FPT in $k + \Delta$)
- or with vertices of degree ≥ 3 pairwise at distance $> k$.

Going further: $\Rightarrow$ a new problem

New Problem: Given a graph G, a matching M and $k \in \mathbb{N}$ odd

Compute a matching of size $\mu_{=k}(G, M)$ that can be obtained from M by augmenting only paths of lenghts $= k$.

Our results

- $\forall G$ and matching M, deciding if $\mu_{=3}(G, M) \leq q$ is NP-complete
- given a tree T, M a matching and $q, k \in \mathbb{N}$ as inputs, deciding if $\mu_{=k}(T, M) \leq q$ is NP-complete

Question: Complexity of $\mu_k(T, M)$ in trees T?

What we know:

Computing $\mu_k(T, M)$ (k odd) can be done in polynomial-time in trees T

- with bounded max. degree Δ (dynamic prog., FPT in $k + \Delta$)
- or with vertices of degree ≥ 3 pairwise at distance $> k$.

Going further: $\Rightarrow$ a new problem

New Problem: Given a graph G, a matching M and $k \in \mathbb{N}$ odd

Compute a matching of size $\mu_{=k}(G, M)$ that can be obtained from M by augmenting only paths of lenghts $= k$.

Our results

- $\forall G$ and matching M, deciding if $\mu_{=3}(G, M) \leq q$ is NP-complete
- given a tree T, M a matching and $q, k \in \mathbb{N}$ as inputs, deciding if $\mu_{=k}(T, M) \leq q$ is NP-complete

Question: Complexity of $\mu_k(T, M)$ in trees T?

What we know:

Computing $\mu_k(T, M)$ (k odd) can be done in polynomial-time in trees T

- with bounded max. degree Δ (dynamic prog., FPT in $k + \Delta$)
- or with vertices of degree ≥ 3 pairwise at distance $> k$.

Going further: $\Rightarrow$ a new problem

New Problem: Given a graph G, a matching M and $k \in \mathbb{N}$ odd

Compute a matching of size $\mu_{=k}(G, M)$ that can be obtained from M by augmenting only paths of lenghts $= k$.

Our results

- $\forall G$ and matching M, deciding if $\mu_{=3}(G, M) \leq q$ is NP-complete
- given a tree T, M a matching and $q, k \in \mathbb{N}$ as inputs, deciding if $\mu_{=k}(T, M) \leq q$ is NP-complete

Conclusion

Firstly: Learn how to communicate with companies

After many months of discussion, hypotheses are changing...

Actual (?) constraints of the system/of Airline Operation Controllers

- Switch alternating cycles of size 4 and augment paths of length 1
- two classes of slots : the ones of the company and the others

Conclusion

Firstly: Learn how to communicate with companies

After many months of discussion, hypotheses are changing...

Actual (?) constraints of the system/of Airline Operation Controllers

- Switch alternating cycles of size 4 and augment paths of length 1
- two classes of slots : the ones of the company and the others

Further theoretical work

- complexity of $\mu_k(G, M)$ in trees?
- in other graph classes?

Conclusion

Firstly: Learn how to communicate with companies

After many months of discussion, hypotheses are changing...

Actual (?) constraints of the system/of Airline Operation Controllers

- Switch alternating cycles of size 4 and augment paths of length 1
- two classes of slots : the ones of the company and the others

Further theoretical work

- complexity of $\mu_k(G, M)$ in trees?
- in other graph classes?

Muchas Gracias / Muito Obrigado / Gramàci ... !