

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

Nicolas Nisse, Alexandre Salch, Valentin Weber

► **To cite this version:**

Nicolas Nisse, Alexandre Salch, Valentin Weber. Recovery of disrupted airline operations using k-Maximum Matching in Graphs. [Research Report] RR-8679, Inria Sophia Antipolis; INRIA. 2015. <hal-01116487>

HAL Id: hal-01116487

<https://hal.inria.fr/hal-01116487>

Submitted on 17 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



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

Nicolas Nisse, Alexandre Salch, Valentin Weber

**RESEARCH
REPORT**

N° 8679

February 2015

Project-Teams COATI



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

Nicolas Nisse^{*†}, Alexandre Salch[‡], Valentin Weber[‡]

Project-Teams COATI

Research Report n° 8679 — February 2015 — 12 pages

Abstract: When an aircraft is approaching an airport, it gets a short time interval (called *slot*) that it can use to land. If the landing of the aircraft is delayed (because of bad weather, or if it arrives late, or if other aircrafts have to land first), it loses its slot and Air traffic controllers have to assign it a new slot. However, slots for landing are a scarce resource of the airports and, to avoid that an aircraft waits too much time, Air traffic controllers have to regularly modify the assignment of the slots of the aircrafts. Unfortunately, for legal and economical reasons, Air traffic controllers can modify the slot-assignment only using two kind of operations: either assign to aircraft *A* a slot that was free, or give to *A* the slot of another aircraft *B* and assign to *B* a free slot. The problem is then the following.

Let $k \geq 1$ be an odd integer and let G be a graph and M be a matching (set of pairwise disjoint edges) of G . What is the maximum size of a matching that can be obtained from M by using only augmenting paths of length at most k ? Moreover, how to compute such a maximum matching? This problem has already been studied in the context of wireless networks, mainly because it provides a simple approximation for the classical matching problem. We prove that this problem can be solved in polynomial-time when $k \leq 3$. Then, we show that, for any odd integer $k \geq 5$, the problem is NP-complete in planar bipartite graphs with maximum degree at most 3.

Key-words: Matching, bipartite graphs, planar graphs

* Inria, France

† Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France

‡ Innovation & Research, Amadeus IT Group SA

**RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE**

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Réaffectation de créneaux d'atterrissage aux aéroports

Résumé : Lorsqu'un avion approche d'un aéroport, il dispose d'un intervalle de temps (*slot*) très limité (une vingtaine de minutes) pour atterrir. Si l'avion a du retard (à cause des conditions météorologiques, à cause du retard d'autres avions, ou si lui-même a eu du retard au décollage), il perd son slot et il faut qu'un nouveau slot lui soit attribué par les contrôleurs aériens. Cependant, les slots d'atterrissage sont une denrée rare et, pour éviter qu'un avion A n'attende trop longtemps, les contrôleurs doivent régulièrement modifier l'attribution des slots d'autres avions afin d'en affecter un à l'avion A . Ce problème peut aisément être modélisé comme un problème de couplage dans un graphe biparti. Malheureusement, pour des raisons légales et économiques, les contrôleurs aériens ne peuvent effectuer leurs modifications qu'en effectuant deux types d'opérations: soit attribuer à l'avion A un slot libre, soit donner à l'avion A le slot d'un avion B et attribuer un slot libre à ce dernier. Le problème devient donc le suivant.

Soit G un graphe et M un couplage (ensemble d'arêtes deux-à-deux disjointes) de G . Comment calculer un couplage maximum pouvant être obtenu à partir de M en utilisant uniquement des chemins augmentants de longueurs au plus k ? Ce problème a déjà été étudié dans le cadre des réseaux sans-fil car il fournit une approximation simple au problème de couplage maximum. Nous prouvons que, pour $k = 3$, ce problème peut être résolu en temps polynomial, fournissant du même coup un algorithme efficace pour les contrôleurs aériens. Nous prouvons ensuite que, pour tout entier impair $k \geq 5$, le problème est NP-complet dans les graphes bipartis planaires de degré au plus 3.

Mots-clés : Couplage, graphes bipartis, graphes planaires

1 Introduction

For obvious safety reasons, the airport authorities should ensure that the number of landing aircrafts in a given time slot is less than the capacity of the airport. Thus, airports have time intervals (called *slots*) that are initially assigned to aircrafts according to their arrival schedule. Of course, an aircraft can only be assigned a slot that is compatible with its arrival time. However, if a flight is delayed A (“for reasons beyond our control”) the operations controllers must assign a new slot via the information system managing these exchanges. The strict regulations that respects this system are that only two operations are possible. Either an available slot is assigned to the aircraft A , or a slot S which is already assigned to another aircraft B is reassigned to A while B is assigned an available slot S' . In both cases, the slot S should be compatible with the schedule of A and in the second case, S and S' must be compatible with the schedule of B . If several planes fall behind and lose their slots, the resolution of these problems is difficult for operations controllers that do not have the tools to make these changes and must ensure “by hand” that all aircrafts are going to get a slot.

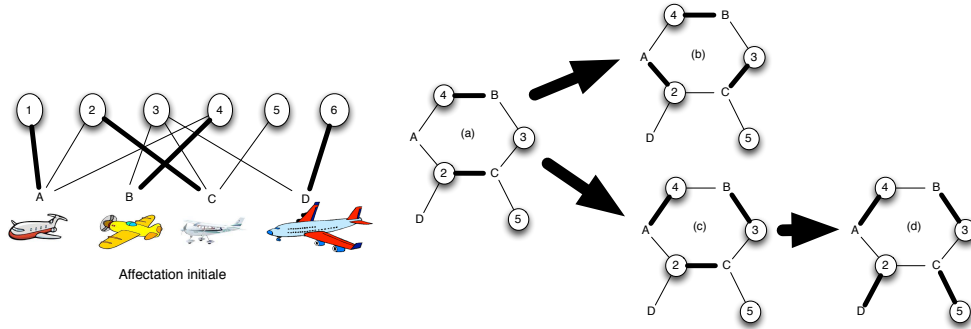


Figure 1: Example of a simple scenario with 4 aircrafts and 6 slots.

An example of a simple scenario is shown in Figure 1. There are 4 planes denoted by A, B, C, D and initially, there are 6 slots denoted by $1, \dots, 6$. On the left, each aircraft is linked to the slots that are compatible with it. Initially, slots 1, 2, 4 and 6 are assigned to aircrafts A, C, B and D , respectively. Edges of the matching are depicted in bold. After aircrafts A and D have been delayed, they are not compatible with slots 1 and 6 anymore and, so, the configuration becomes the one depicted in Figure 1(a). From Configuration (a), if Slot 2 is reassigned to plane A and Slot 3 is assigned to Plane C (i.e., the path $(A2C3)$ is augmented), then we reach Configuration (b) where no allowed modification is possible anymore (there exists no augmenting path of length at most 3). Another solution would be to first augment the path $(A4B3)$ (reaching Configuration (c)) and then the path $(D2C5)$, reaching the Configuration of Figure 1(d) where all aircrafts are assigned a slot.

The problem of *reassignment of slots* can of course be modeled as a problem of matching in graphs. Let $G = (V, E)$ be a graph. A *matching* $M \subseteq E$ of G is a set of edges pairwise disjoint. A vertex $v \in V$ is *covered* by M if there is $e \in M$ such that $v \in e$. Otherwise, v is said *exposed*. The size of a maximum matching in G is denoted by $\mu(G)$. The problem of computing a maximum matching has been widely studied and it is well known that it can be solved in polynomial-time [Edm65]. A key ingredient in most of the work on matching is the notion of *augmenting path*. A path $P = (v_0, \dots, v_k)$ of G is a sequence of pairwise distinct vertices such that $e_i = \{v_i, v_{i+1}\} \in E$ for each $0 \leq i < k$. The path P is said M -*augmenting* if, v_0 and v_k are

exposed and, for any $0 \leq i < k$, $e_i = \{v_i, v_{i+1}\} \in M$ if and only if i is odd. In particular, an augmenting path has always an odd length. A well known theorem of Claude Berge states that a matching M is maximum if and only if there are no M -augmenting paths. In particular, it is easy to check that, if P is M -augmenting, then $M \Delta E(P)$ is a matching of G with size $|M| + 1$, where $E(P)$ is the set of edges of P and Δ is the symmetric difference. When passing from a matching M to the matching $M \Delta E(P)$, we say that the M -augmenting path P is *augmented*.

The problem of *reassignment of slots* described above can be modeled as follows. Let $G = (X \cup Y, E)$ be a bipartite graph. The part X represents the set of aircrafts and the part Y represents the set of available slots. There is an edge between $a \in X$ and $s \in Y$ if the slot s is compatible with the schedules of the aircraft a . Let M be a matching of G that corresponds to a pre-established assignment of some slots to some aircrafts (see Fig 1 (a)). The problem of reassignment of slots is equivalent to compute a matching maximum that can be obtained from M by augmenting only paths of length at most 3.

1.1 Our results

We consider the problem that takes a graph $G = (V, E)$, a matching $M \subseteq E$ and an odd integer $k \geq 1$ as inputs. Let $\mu_k(G, M)$ denote the size of a maximum matching that can be obtained from M in G by augmenting paths of length at most k . The problem consists in computing a sequence of augmenting-paths of length $\leq k$ that allow to obtain, from M , a matching of size $\mu_k(G, M)$. Let us emphasize that only odd-length paths can be augmented.

In the cases $k \in \{1, 3\}$, we prove that the computational complexity of the problem is equivalent to the one of the classical maximum matching problem (without any constraint on the length of paths). Hence, for $k \in \{1, 3\}$, it can be solved in polynomial time. Then, we show that, for any odd integer $k \geq 5$, the problem is NP-complete in planar bipartite graphs with maximum degree at most 3.

1.2 Related work

the first algorithm for solving the maximum matching problem in polynomial-time in general graphs is due to Edmonds [Edm65]. Then, many work has been dedicated to design more efficient algorithms [HK73, MV80, DP14]. In particular, the algorithms in [HK73, MV80] are based on augmenting paths in the non-decreasing order of their lengths. Such a method gives a good approximation since augmenting only the paths of length at most $2k - 3$ provides a $(1 - 1/k)$ -approximation of the maximum matching [HK73].

The problem of matching with bounded-length paths has also been studied in the context of wireless networks. In particular, it provides simple distributed algorithms to compute the scheduling of transmissions with interference [WS05, BSS09].

2 Maximum matching with bounded-length augmenting paths: polynomial cases

As a warm-up, let us start with trivial cases.

Claim 1 *For any odd $k \geq 1$, the problem of computing $\mu_k(G, \emptyset)$ is equivalent to the classical maximum matching problem.*

Proof. It is sufficient to compute a (classical) maximum matching M of G and then to augment all edges of M one by one. ■

Claim 2 For any initial matching M , the problem of computing $\mu_1(G, M)$ is equivalent to the classical maximum matching problem.

Proof. For $k = 1$, the single edges that can be augmented are the edges between two nodes initially exposed by M . In particular, all edges of M have to be kept.

Let H be the graph obtained from G by removing all vertices of the edges of M . Let M' be a (classical) maximum matching of H . Then, $M \cup M'$ is a matching of G with size $\mu_1(G, M)$ and that can be obtained from M by augmenting all edges of M' one by one. ■

When $k \geq 3$, the difficulty arises from the fact that the order in which the paths are augmented is important. This fact is illustrated in Figure 1 where augmenting first the path (A2C3) leads to non-optimal dead end configuration. In addition, the order in which the paths are augmented has an impact on the creation or non-creation of new augmenting paths of length at most k . For instance, for $k = 5$, let us consider the graph that consists of a path (v_1, \dots, v_7) plus three edges $\{v_5, v_8\}, \{v_8, v_9\}$ et $\{v_9, v_{10}\}$. The initial matching is $\{\{v_2, v_3\}, \{v_4, v_5\}, \{v_8, v_9\}\}$. Initially, there are 2 augmenting paths of length at most 5: $P_1 = (v_6, v_7)$ et $P_2 = (v_1, \dots, v_6)$. If P_1 is augmented first, then it remains no augmenting paths of length at most 5 anymore. However, augmenting P_2 first "destroys" the path P_1 but creates a new path $(v_{10}, v_9, v_8, v_5, v_6, v_7)$ that can be augmented.

The main difference between the cases $k = 3$ and $k \geq 5$ is that, in the former case, it is possible to ignore the augmenting paths that were not present in the initial matching M . In other words, when $k = 3$, if a new augmenting path P of length at most k is created after augmenting some path, then it is not necessary to augment P in order to achieve a matching of size $\mu_3(G, M)$.

Let G be a graph and $M \subseteq E(G)$ be a matching of G , let $\mathcal{P}_3(G, M)$ denote the set of M -augmenting paths of length at most 3 in G .

Lemma 1 Let G be a graph and M be a matching in G . Then, there exists $\mathcal{P} \subseteq \mathcal{P}_3(G, M)$ such that a matching of size $\mu_3(G, M)$ can be obtained from M by augmenting the paths in \mathcal{P} in any order.

Proof. Let $M = M_0$. Let (P_0, \dots, P_r) be a sequence of paths of length at most 3 such that, for any $0 \leq i \leq r$, $M_{i+1} = M_i \Delta E(P_i)$ and $P_i \in \mathcal{P}_3(G, M_i)$ and such that $|M_{r+1}| = \mu_3(G, M)$.

Let $h \leq r$ be the maximum integer such that $P_h \notin \mathcal{P}_3(G, M)$. If such a path exists, the following process returns a new sequence that either has strictly less paths not in $\mathcal{P}_3(G, M)$, or that strictly decrease the largest index of such a path. Hence, iterating this process eventually achieves a sequence of M -augmenting paths that result in the same matching that the initial sequence.

First, we prove that P_h has length 3. Indeed, otherwise, $P_h = (u, v)$ where u and v are exposed by the matching M_h . Since an non-exposed vertex cannot become exposed, then u and v are exposed by M_0 and then $P_h \in \mathcal{P}_3(G, M_0)$, a contradiction.

Hence, $P_h = (u, x, y, v)$, with $\{x, y\} \in M_h \setminus M_{h-1}$ and u, v are exposed in M_h and then u, v are exposed in M . In particular, $\{x, y\} \notin M_0$ since otherwise $P_h \in \mathcal{P}_3(G, M)$. Let $0 < m \leq h$ be the maximum integer such that $\{x, y\} \in M_m$. Therefore, there is a M_{m-1} -augmenting path P_t , $t < h$, that is augmented to include $\{x, y\}$ in the matching. There are two cases to be considered.

- Let us first assume that $P_t = (x, y)$. This implies that x and y were exposed. In that case, we replace the sequence of augmenting paths by

$$(P_1, \dots, P_{t-1}, P_{t+1}, \dots, P_{h-1}, (u, x), (y, v), P_{h+1}, \dots, P_r).$$

That is, the paths P_t and P_h are removed (they are not augmented anymore) and, instead, we augment the two paths (u, x) and (y, v) that are in $\mathcal{P}_3(G, M)$. It can be checked that this is a valid sequence of augmenting paths and it results in the same matching as the initial sequence. Hence, we have decrease the number of paths that are not in $\mathcal{P}_3(G, M)$.

- $P_t = (a, b, x, y)$ a M_{m-1} -augmenting path, i.e., such that a and y are exposed in M_{m-1} and so in M_0 and $\{b, x\} \in M_{m-1}$. Moreover, a and b are not exposed in M_m (and so in M_h by Claim) and then $\{a, b\} \cap \{u, v\} = \emptyset$. In that case, we replace the sequence of augmenting paths by

$$(P_1, \dots, P_{t-1}, (a, b, x, u), P_{t+1}, \dots, P_{h-1}, (y, v), P_{h+1}, \dots, P_r).$$

That is, instead of augmenting P_t , we use the path (a, b, x, u) and instead of augmenting P_h , we use the path $(y, v) \in \mathcal{P}_3(G, M)$. It can be checked that this is a valid sequence of augmenting paths and it results in the same matching as the initial sequence. Hence, we have decrease the largest index of a path that is not in $\mathcal{P}_3(G, M)$.

Hence, there exists $\mathcal{P} \subseteq \mathcal{P}_3(G, M)$ such that a matching of size $\mu_3(G, M)$ can be obtained from M by augmenting the paths in \mathcal{P} . Moreover, since all these paths are initially present and can all be augmented, they do not interfere one on each other. Hence, they can be augmented in any order. \blacksquare

Let G be a graph and $M \subseteq E(G)$ be a matching of G . An edge $e \in E(G) \setminus M$ is said *useless* if e is incident to two edges of M , i.e., if each endpoint of e is incident to one edge of M . An edge $e \in M$ is said *forced* if at least one of its endpoints has degree one in G .

Finally, let (\hat{G}, \hat{M}) be the graph and matching obtained from M by first removing all useless edges (but keeping their endpoints), and then by recursively removing the forced edges and their endpoints. Note that (\hat{G}, \hat{M}) has no useless edges nor forced edges. Let f_M denote the number of forced edges that have been removed to obtain (\hat{G}, \hat{M}) .

Lemma 2 *Let G be a graph and M a matching. Any sequence of augmenting paths of length at most 3 in one of G_M or (\hat{G}, \hat{M}) is a sequence of augmenting paths in the other graph, and results in the same matching in both graph. Therefore, $\mu_3(G, M) = \mu_3(\hat{G}, \hat{M}) + f_M$.*

Proof. First, no useless edge of (G, M) belongs to an augmenting path of length at most 3. Similarly, none of the forced edges that are removed to obtain (\hat{G}, \hat{M}) belongs to an augmenting path of length at most 3. Hence, $\mathcal{P}_3(G, M) \subseteq \mathcal{P}_3(\hat{G}, \hat{M})$. Reciprocally, removing useless or forced edges cannot create new augmenting path of length at most 3. Hence, $\mathcal{P}_3(G, M) = \mathcal{P}_3(\hat{G}, \hat{M})$.

Then, the result follows Lemma 1. \blacksquare

Lemma 3 *Let G be a graph and M a matching such that any edge in $E(G) \setminus M$ has at least one exposed end, and any matched vertex has degree at least two. Then $\mu_3(G, M)$ equals the size $\mu(G)$ of a maximum matching in G .*

Proof. Let M^* be a maximum matching in G . We describe a sequence of augmenting paths of length at most 3 of G_M and achieving a matching of size $\mu(G) = |M^*|$.

For any $e \in M$, let H_e be the set of edges that consists of e and any edge incident to e , i.e., $H_e = \{f \mid e \cap f \neq \emptyset\}$. Let $H = \bigcup_{e \in M} \{H_e\}$. The key point is that, because there are no useless edges, the sets H_e are pairwise disjoint. Moreover, any path in $\mathcal{P}_3(G, M)$ is either one edge of $E(G) \setminus H$ (type 1) or it contains $e \in M$ and its 3 edges are in H_e (type 2).

First, let us slightly modify the maximum matching M^* . Note that, for any $e \in M$, $1 \leq |H_e \cap M^*| \leq 2$. Let $J_i = \{e \in M \mid |H_e \cap M^*| = i\}$ and $I_i = \{H_e \cap M^* \mid e \in J_i\}$.

For any $e \in M$ such that $|H_e \cap M^*| = 1$, we replace $H_e \cap M^*$ by e in M^* . More formally, let $M' = (M^* \setminus I_1) \cup \bigcup_{e \in J_1} \{e\}$. Then, M' is a matching of same size as M^* , i.e., M' is also a maximum matching of G .

We now show that M' can be obtained by augmenting paths in $\mathcal{P}_3(G, M)$. Indeed, for any $e \in J_2$, augment the path with edges e and $H_e \cap M^*$. Then, for any $e \in M^* \setminus H$, augment the path that only consists of e . The resulting matching is precisely M' . ■

From Lemmas 2 and 3, we get:

Theorem 1 $\mu_3(G, M)$ and a corresponding matching can be computed in polynomial time for any graph G and matching M .

Proof. Compute (\hat{G}, \hat{M}) and apply your favorite classical matching algorithm on it. ■

3 NP-completeness when $k \geq 5$

Let $\mu_k(G, M)$ be the maximum size of a matching that can be obtained from M by augmenting paths of length at most k in G .

Theorem 2 Let $k \geq 5$ be an odd integer. Computing $\mu_k(G, M)$ is NP-hard in the class of bipartite planar graphs with maximum degree 3.

Proof. The proof is a reduction from 3-SAT.

Let us consider an instance of 3-SAT which is a formula Φ with variables v_1, \dots, v_n and clauses C_1, \dots, C_m . For any $i \leq n$, let o_i denote the number of clauses where variable v_i appears positively or negatively.

We aim at building a graph $G(\Phi)$ and an initial matching $M(\Phi)$ such that $\mu_k(G_M(\Phi)) = |M(\Phi)| + m + \sum_{i \leq n} o_i$ if and only if Φ is satisfiable. In other words, starting from $M(\Phi)$, it is possible to augment $m + \sum_{i \leq n} o_i$ paths of length at most k if Φ is satisfiable, otherwise only strictly less paths can be augmented.

Gadget-graph G_{ij} . Let $i \leq n$ and let $j \leq m$ such that the variable v_i appears in Clause C_j (positively or negatively). Let us build the gadget-graph G_{ij} as follows. The graph G_{ij} is a tree with $3k - 1$ vertices $\{u_1^{ij}, \dots, u_{3k-1}^{ij}\}$. It is obtained from the disjoint union of a path $(u_1^{ij}, \dots, u_{2k+1}^{ij})$ and a path $(u_{2k+2}^{ij}, \dots, u_{3k-1}^{ij})$ by adding an edge between u_k^{ij} and u_{2k+2}^{ij} if v_i appears positively in C_j , or by adding the edge $\{u_{k+2}^{ij}, u_{2k+2}^{ij}\}$ if v_i appears negatively in C_j . The initial matching $M(\Phi)$ restricted to the subgraph G_{ij} is denoted by $M_{ij} = M(\Phi) \cap E(G_{ij})$ and consists of the edges $\{u_2^{ij}, u_3^{ij}\}, \{u_4^{ij}, u_5^{ij}\}, \dots, \{u_{k-1}^{ij}, u_k^{ij}\}, \{u_{k+2}^{ij}, u_{k+3}^{ij}\}, \{u_{k+4}^{ij}, u_{k+5}^{ij}\}, \dots, \{u_{2k-1}^{ij}, u_{2k}^{ij}\}, \{u_{2k+2}^{ij}, u_{2k+3}^{ij}\}, \{u_{2k+4}^{ij}, u_{2k+5}^{ij}\}, \dots, \{u_{3k-3}^{ij}, u_{3k-2}^{ij}\}$.

Finally, let us denote u_1^{ij} by t_{ij} , u_{2k+1}^{ij} by f_{ij} , u_{k+1}^{ij} by c_{ij} , u_{3k-1}^{ij} by v_{ij} , u_k^{ij} by p_{ij} and u_{k+2}^{ij} by n_{ij} .

Note that in G_{ij} , there are exactly two M_{ij} -augmenting paths of length at most k , the path T_{ij} from c_{ij} to t_{ij} and the path F_{ij} from c_{ij} to f_{ij} . Intuitively, augmenting the path P_{ij} will correspond to assign v_i to *True* while augmenting the path F_{ij} will correspond to assign v_i to *False*.

The gadget-graph G_{ij} and the initial matching M_{ij} are depicted in Figure 2.

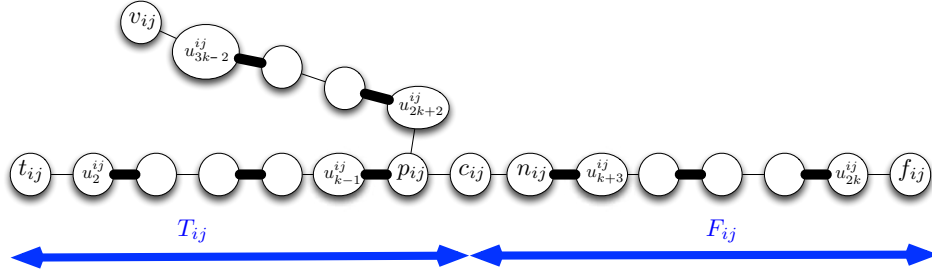


Figure 2: Gadget-graph G_{ij} for Variable v_i appearing positively in Clause C_j . Edges of the initial matching M_{ij} are depicted in bold. If Variable v_i appears negatively in Clause C_j , then the edge $\{p_{ij}, u_{2k+2}^{ij}\}$ is replaced by the edge $\{n_{ij}, u_{2k+2}^{ij}\}$.

Variable-graph G_i . Let $i \leq n$, we build a Variable-gadget graph that corresponds to Variable v_i . Let $J_i = \{j_1^i, \dots, j_{o_i}^i\}$ be the set of integers j such that Variable v_i appears (positively or negatively) in Clause C_j . The Variable-graph G_i is obtained from the disjoint union of the gadget-graphs $G_{i,j_1^i}, \dots, G_{i,j_{o_i}^i}$ by combining them into a “cycle”, i.e., the paths between t_{i,j_k^i} and f_{i,j_k^i} in G_{i,j_k^i} are “glued together” for $\ell = 1$ to $\ell = o_i$. More precisely, let us identify t_{i,j_1^i} and $f_{i,j_{o_i}^i}$ and, for $1 \leq \ell < o_i$, let us identify $t_{i,j_{\ell+1}^i}$ and f_{i,j_ℓ^i} .

The initial matching M_i in G_i is simply the union of the initial matchings M_{ij} for $j \in J_i$, i.e., $M_i = \bigcup_{j \in J_i} M_{ij}$. It is important to note that the augmenting paths in G_i are the augmenting paths in the G_{ij} 's, for $j \in J_i$, i.e., no new augmenting paths are created by combining the G_{ij} 's. This can be proved because the G_{ij} 's are “joined” by identifying exposed vertices, hence each augmenting path is fully included in some G_{ij} . An example is depicted in Figure 3.

Note that the order in which the graphs G_{ij} are combined will be relevant only for ensuring the planarity of the graph. However, keeping the same “orientation” for each of them (the end t of one such graph being identified to the end f of the next one) is crucial. The key point of combining the graphs G_{ij} in such a way is that the only way to augment one of the paths T_{ij} or F_{ij} for all $j \in J_i$ is to do the same choice for each $j \in J_i$, i.e., either all paths T_{ij} , $j \in J_i$, are augmented, or all paths F_{ij} , $j \in J_i$, are augmented. In other words, if there is $x, y \in J_i$ such that T_{ix} and F_{iy} are augmented, then there exists $z \in J_i$ such that neither T_{iz} nor F_{iz} can be augmented.

Clause-graph H_j and main graph $G_M(\Phi)$. For any $j \leq m$, the Clause-graph H_j simply consists of one node γ_j . The main graph $G(\Phi)$ is obtained from the disjoint union of the graphs G_i , $i \leq n$, and H_j , $j \leq m$, by adding an edge $\{\gamma_j, c_{ij}\}$ for any Variable v_i appearing (positively or negatively) in Clause C_j . Finally, the initial matching $M(\Phi)$ equals $\bigcup_{i \leq n} M_i$, i.e., the union of the initial matchings in the Variable-gadgets.

An example with formula $\Phi = (b \vee c \vee d) \wedge (\bar{b} \vee c \vee \bar{e}) \wedge (\bar{c} \vee d \vee e) \wedge (\bar{b} \vee \bar{d} \vee \bar{e})$ is depicted in Figure 4.

For any $i \leq n$, $j \leq m$ such that Variable v_i appears in Clause C_j , let R_{ij} be the shortest path in G between γ_j and v_{ij} and let S_{ij} be the shortest path in $G(\Phi)$ between γ_j and c_{ij} . Note that S_{ij} is initially an augmenting path of length 1.

Before going on, let us consider the behavior of the combination of a gadget-graph G_{ij} with Clause-graph H_j where Variable v_i appears in Clause C_j . In the subgraph induced by G_{ij} and

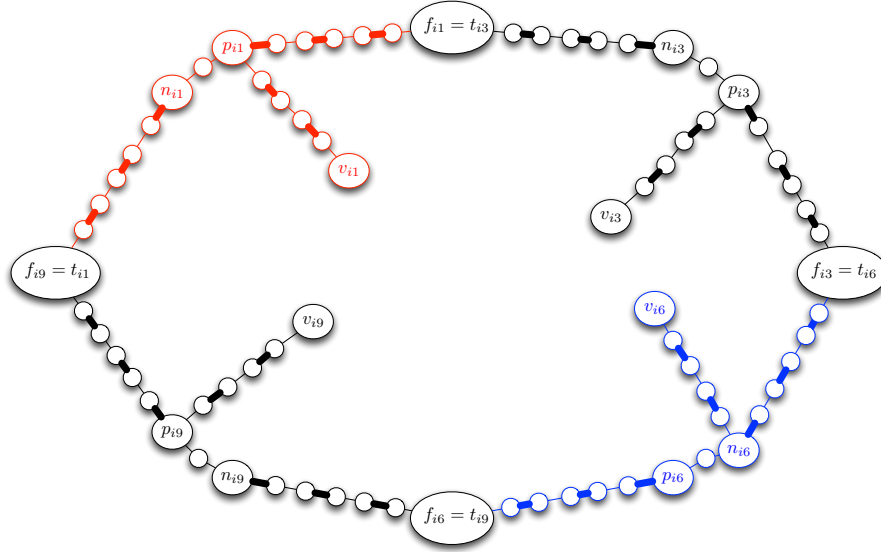


Figure 3: Variable-gadget G_i for $k = 7$ and Variable v_i appearing positively in Clauses C_1, C_3, C_9 and negatively in Clause C_6 . The edges of the initial matching appear in bold. Colors are used only to better distinguish the four gadget-graphs G_{i1} (red), G_{i3} (black), G_{i6} (blue) and G_{i9} (black).

H_j , there are initially three augmenting paths of length at most k , T_{ij} , F_{ij} and S_{ij} . See Figure 5.

Augmenting the path S_{ij} is depicted in Figure 6. It does not create new augmenting path of length at most k and remove T_{ij} , F_{ij} and S_{ij} , for any Variable $v_{i'}$ in Clause C_j , from the possible augmenting paths.

Augmenting the path F_{ij} is depicted in Figure 7. It does not create new augmenting path of length at most k and remove T_{ij} and S_{ij} from the possible augmenting paths. It also remove $T_{ij'}$ from the possible augmenting paths of length at most k where j' is the predecessor of j in J_i .

Augmenting the path T_{ij} is depicted in Figure 8. It does create the new augmenting path R_{ij} of length at most k and remove F_{ij} and S_{ij} from the possible augmenting paths. It also remove $F_{ij'}$ from the possible augmenting paths of length at most k where j' is the successor of j in J_i .

Claim 3 *If Φ is satisfiable, then $\mu_k(G_M(\Phi)) \geq |M(\Phi)| + m + \sum_{i \leq n} o_i$*

Proof. Consider a truth assignment for Φ and the graph $G_M(\Phi)$ built as previously.

For any $i \leq n$, if v_i is assigned to *True*, then augment each path T_{ij} for any $j \in J_i$. If v_i is assigned to *False*, then augment each path F_{ij} for any $j \in J_i$.

Finally, for any $j \leq m$, let i_j be one variable such that v_{i_j} validates C_j . Then augment the path $R_{i_j, j}$. ■

Claim 4 *If $\mu_k(G_M(\Phi)) \geq |M(\Phi)| + m + \sum_{i \leq n} o_i$, then Φ is satisfiable.*

Proof. The intuition is that to augment M as required, the only way to augment the paths as in the previous claim. Then, it would provide a truth assignment. ■

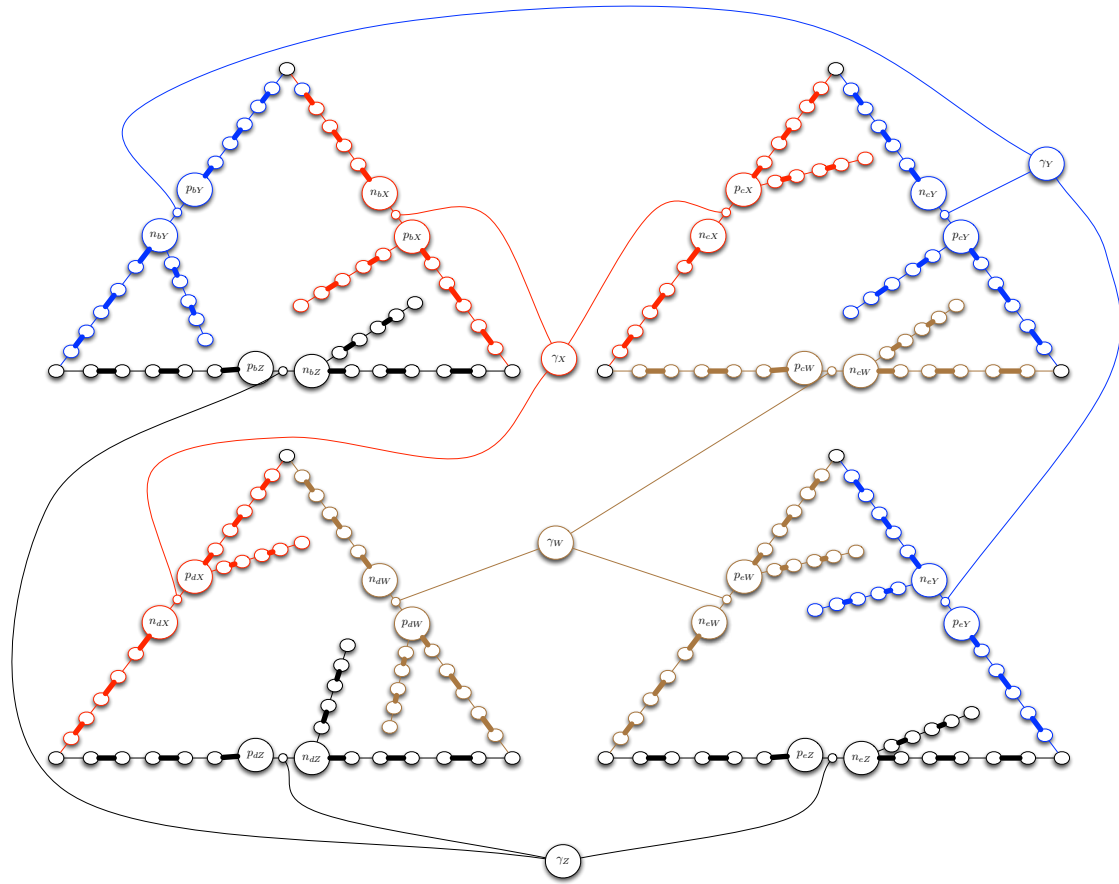


Figure 4: Graph $G_M(\Phi)$ built for $k = 7$ and from the formula $\Phi = (b \vee c \vee d) \wedge (\bar{b} \vee c \vee \bar{e}) \wedge (\bar{c} \vee d \vee e) \wedge (\bar{b} \vee \bar{d} \vee \bar{e}) = X \wedge Y \wedge Z \wedge W$ with Variables b, c, d, e and Clauses $X = b \vee c \vee d$ (red), $Y = \bar{b} \vee c \vee \bar{e}$ (blue), $W = \bar{c} \vee d \vee e$ (brown) and $Z = \bar{b} \vee \bar{d} \vee \bar{e}$ (black). The edges of the initial matching appear in bold.

■

References

- [BSS09] L. Bui, S. Sanghavi, and R. Srikant. Distributed link scheduling with constant overhead. *IEEE/ACM Trans. Netw.*, 17(5):1467–1480, 2009.
- [DP14] R. Duan and S. Pettie. Linear-time approximation for maximum weight matching. *J. ACM*, 61(1), 2014.
- [Edm65] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

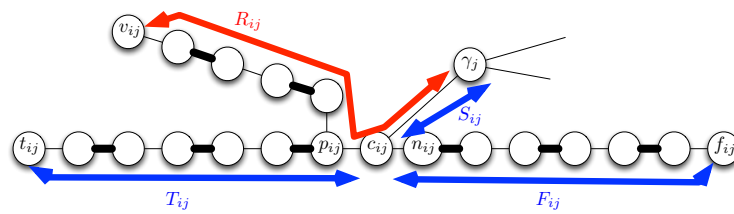


Figure 5: Initial augmenting paths T_{ij} , F_{ij} and S_{ij} in the subgraph induced by G_{ij} and H_j (case when Variable v_i appears positively in Clause C_j).

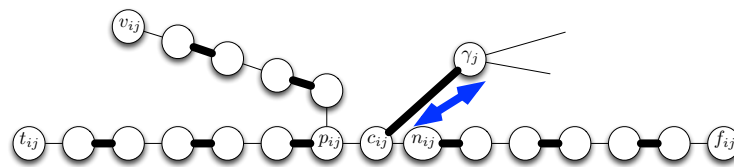


Figure 6: Augmenting the path S_{ij}

- [HK73] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.
- [MB10] D. Michalek and H. Balakrishnan. Dynamic reconfiguration of terminal airspace during convective weather. In *49th Conference on Decision and Control*, pages 4875–4881. IEEE, 2010.
- [MV80] S. Micali and V. V. Vazirani. An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In *21st Symp. on Foundations of Comp. Sc. (FOCS)*, pages 17–27. IEEE, 1980.
- [NSW15] N. Nisse, A. Salch, and V. Weber. Recovery of disrupted airline operations, 2015. <http://hal.inria.fr/Something>.
- [WS05] X. Wu and R. Srikant. Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with nodeexclusive spectrum sharing. In *IEEE Conf. on Decision and Control*, 2005.

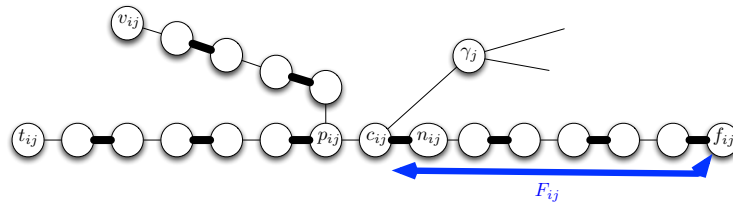


Figure 7: Augmenting the path F_{ij}

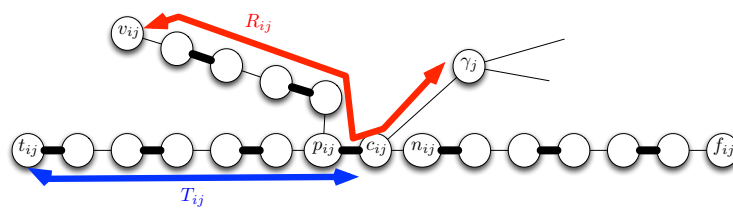


Figure 8: Augmenting the path T_{ij}



**RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE**

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399